Research Article

The Splitting Crank–Nicolson Scheme with Intrinsic Parallelism for Solving Parabolic Equations

Guanyu Xue,1 Yunjie Gong,1 and Hui Feng2

1School of Mathematics and Information Sciences, Yantai University, Yantai 264005, China
2School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Guanyu Xue; xueguanyu2011@163.com

Received 10 February 2020; Accepted 28 February 2020; Published 30 March 2020

Academic Editor: Liguang Wang

Copyright © 2020 Guanyu Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a splitting Crank–Nicolson (SC-N) scheme with intrinsic parallelism is proposed for parabolic equations. The new algorithm splits the Crank–Nicolson scheme into two domain decomposition methods, each one is applied to compute the values at \((n+1)^{\text{th}}\) time level by use of known numerical solutions at \(n^{\text{th}}\) time level, respectively. Then, the average of the above two values is chosen to be the numerical solutions at \((n+1)^{\text{th}}\) time level. The new algorithm obtains accuracy of the Crank–Nicolson scheme while maintaining parallelism and unconditional stability. This algorithm can be extended to solve two-dimensional parabolic equations by alternating direction implicit (ADI) technique. Numerical experiments illustrate the accuracy and efficiency of the new algorithm.

1. Introduction

The parallel numerical schemes for parabolic equations have been studied rapidly. In 1985, Evans [1] proposed the Alternating Group Explicit (AGE) scheme for the diffusion equation by Saul’yev asymmetric schemes [2]. The Alternating Segment Crank–Nicolson (ASC-N) scheme was designed in [3]. Afterwards, the alternating segment algorithms (AGE scheme and ASC-N scheme) above became very effective methods for parabolic equations, such as convection-diffusion equation [4], dispersive equation [5–8], and fourth-order parabolic equation [9, 10]. Meanwhile, domain decomposition methods (DDMs) for the partial differential equations have been developed. Since the advantages of the low computation and communication costs at each time step, the nonoverlapping DDMs have been studied extensively [11–20]. The concept called “intrinsic parallelism” was presented in [21–23]. In 1999, the alternating difference schemes were presented, and the unconditional stability analysis was given in [24]. The unconditionally stable domain decomposition method was obtained by the alternating technique in [25–30]. Recently, the numerical methods for parabolic equations have attracted great attention of scholars [31–42]. Meanwhile, the finite volume element methods for elliptic problems are studied by Bi et al. [43, 44], and the finite volume element method for second-order hyperbolic equations is proposed by Chen et al. [45]. Obviously, the Crank–Nicolson scheme is one of the most classical methods for PDEs [46, 47]. But, there is litter work on parallel algorithms that satisfy the accuracy and stability of the Crank–Nicolson scheme.

On this basis, a new parallel algorithm called splitting Crank–Nicolson scheme for parabolic equations will be presented in this paper. The idea of the new algorithm is to divide the classical Crank–Nicolson scheme into two parts, which are DDMs, each of which is used in computations at \((n+1)^{\text{th}}\) time level utilizing the numerical solution at time level \(n\). Then, the average of the above two values is chosen to be the numerical solutions at \((n+1)^{\text{th}}\) time level. It can be described as follows:

1. Split the Crank–Nicolson scheme into DDM I and DDM II by Saul’yev asymmetric schemes.
2. DDM I is applied to compute the values at \((n+1)^{\text{th}}\) time level noted as \(V^{n+1}\) by use of known value \(U^n\) at
n-th time level. DDM II is also used to compute the values at (n + 1)th time level noted as $W^{n+1}$ by use of $U^n$ at n-th time level.

(3) The value $U^{n+1} = (V^{n+1} + W^{n+1})/2$ are set as numerical solutions at (n + 1)th time level.

The advantage of the SC-N scheme is splitting the C-N scheme into two parallel algorithms, which can be computed by parallel computers. Then, the average value obtained is restored to C-N scheme approximately. This paper is organized as follows: in Section 2, we introduce a SC-N scheme for parabolic equations. For simplicity of presentation, we focus on a model problem, namely, one-dimensional parabolic equations. The new algorithm and detailed presentations are given. Then, we extend the new algorithm to solve two-dimensional parabolic equations by ADI technique. Finally, numerical experiments illustrated the accuracy of SC-N scheme is approximate to the C-N scheme and the new algorithm is efficient.

2. Algorithm Presentation

Considering the model problem of one-dimensional parabolic equations,

$$\frac{\partial u}{\partial t} - a \frac{\partial^2 u}{\partial x^2} = f(x,t), \quad x \in (0, l), \quad t \in (0, T), \quad (1)$$

with the initial and boundary conditions

$$u(x,0) = u_0(x), \quad x \in [0, l],$$
$$u(0,t) = g_0(t), \quad u(l,t) = g_1(t), \quad t \in (0, T), \quad (2)$$

where $a \geq 0$ is a constant.

Let $h$ and $\tau$ be the spatial and temporal step sizes, respectively. Denote $x_j = jh, j = 0, 1, \ldots, m$ and $t_n = n\tau, n = 0, 1, \ldots, N$. Let $u^n_j$ be the approximate solution at $(x_j, t_n)$. $u(x, t)$ represents the exact solution of (1).

The well-known Crank–Nicolson scheme can be written as

$$\frac{1}{\tau} u^n_j + (1 + \mu) u^n_j - \frac{1}{h^2} u^n_{j-1} = \frac{\mu}{2} f^n_j, \quad (3)$$

where $\mu = \frac{\tau}{h^2}$.

Let $\mu = ar$, and (3) can be written as the matrix form

$$AU^{n+1} = BU^n + F^n, \quad (4)$$

where $U^n = (u^n_0, u^n_1, \ldots, u^n_m)^T$ and $F^n = ((\mu/2)(u^n_0 + u^n_m) + t f^n_0, \ldots, (\mu/2)(u^n_m + u^n_{m-1}) + t f^n_m)^T$.

The matrices $A$ and $B$ are as follows:

$$A = \begin{bmatrix} 1 + \mu & -\frac{\mu}{2} \\ \frac{\mu}{2} & 1 + \mu \end{bmatrix}, \quad B = \begin{bmatrix} \frac{\mu}{2} & 1 - \mu \\ 1 - \mu & \frac{\mu}{2} \end{bmatrix} \quad (5)$$

Splitting Crank–Nicolson scheme (4), we obtain

$$(A_1 + A_2)U^{n+1} = (B_1 + B_2)U^n + 2F^n, \quad (6)$$

where $A_1$ and $A_2$ are block diagonal matrices, respectively.

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \quad (7)$$

$V^{n+1}$ and $W^{n+1}$ can be defined as follows:

$$A_1 V^{n+1} = B_1 U^n + F^n, \quad (8)$$
$$A_2 W^{n+1} = B_2 U^n + F^n, \quad (9)$$

where $V^{n+1}$ and $W^{n+1}$ approximate to $U^{n+1}$, respectively. Then,

$$U^{n+1} = \frac{V^{n+1} + W^{n+1}}{2}. \quad (10)$$

Scheme (8) is DDM I, and scheme (9) is called DDM II. The two DDMs aforementioned are suitable for parallel computing.

In order to construct two domain decomposition methods, DDM I and DDM II, at (n + 1)th time level, we need to consider four forms of Saul’yev asymmetric difference schemes corresponding to (3) as follows:
\begin{align}
\frac{ar}{2}u_{j+1} + \left(1 + \frac{3ar}{2}\right)u_j - aru_{j-1} &= \frac{ar}{2}u_{j-1} + \left(1 - \frac{ar}{2}\right)u_j + \tau f^n_j,
\end{align}
\tag{11}

\begin{align}
-aru_{j-1} + \left(1 + \frac{3ar}{2}\right)u_j - \frac{ar}{2}u_{j+1} &= \left(1 - \frac{ar}{2}\right)u_j + \frac{ar}{2}u_{j+1} + \tau f^n_j,
\end{align}
\tag{12}

\begin{align}
\left(1 + \frac{ar}{2}\right)u_j - \frac{ar}{2}u_{j+1} &= aru_{j-1} + \left(1 - \frac{3ar}{2}\right)u_j + \frac{ar}{2}u_{j+1} + \tau f^n_j,
\end{align}
\tag{13}

\begin{align}
\frac{ar}{2}u_{0} + (1 + ar)u_1 - \frac{ar}{2}u_2 &= \frac{ar}{2}u_0 + (1 - ar)u_1 + \frac{ar}{2}u_2 + \tau f^n_1,
\end{align}
\tag{15}

\begin{align}
\frac{ar}{2}u_1 + (1 + ar)u_2 - \frac{ar}{2}u_3 &= \frac{ar}{2}u_1 + (1 - ar)u_2 + \frac{ar}{2}u_3 + \tau f^n_2,
\end{align}
\tag{15}

\begin{align}
\frac{ar}{2}u_2 + (1 + ar)u_3 - \frac{ar}{2}u_4 &= \frac{ar}{2}u_2 + (1 - ar)u_3 + \frac{ar}{2}u_4 + \tau f^n_3.
\end{align}
\tag{15}

The flow chart of schemes (11)–(14) are displayed in Figure 1. Assume \( m - 1 = 6K \), where \( K \) is a positive integer.

2.1. DDM I. For the values \( u^n_1, u^n_2, \) and \( u^n_3 \), we use the formulas as follows:

\begin{align}
\left(1 + \frac{ar}{2}\right)u^n_{0k-2} - \frac{ar}{2}u^n_{0k-1} = aru^n_{0k-3} + \left(1 - \frac{3ar}{2}\right)u^n_{0k-2} + \frac{ar}{2}u^n_{0k-1} + \tau f^n_{0k-2},
\end{align}
\tag{16}

\begin{align}
\frac{ar}{2}u^n_{0k-2} + (1 + ar)u^n_{0k-1} - \frac{ar}{2}u^n_{0k} &= \frac{ar}{2}u^n_{0k-2} + (1 - ar)u^n_{0k-1} + \frac{ar}{2}u^n_{0k} + \tau f^n_{0k-1},
\end{align}
\tag{16}

\begin{align}
\frac{ar}{2}u^n_{0k} + \left(1 + \frac{3ar}{2}\right)u^n_{0k+1} - aru^n_{0k+1} &= \frac{ar}{2}u^n_{0k} + \left(1 - \frac{ar}{2}\right)u^n_{0k+1} + \tau f^n_{0k},
\end{align}
\tag{16}

\begin{align}
-aru^n_{0k} + \left(1 + \frac{3ar}{2}\right)u^n_{0k+1} - \frac{ar}{2}u^n_{0k+2} &= \left(1 - \frac{ar}{2}\right)u^n_{0k+1} + \frac{ar}{2}u^n_{0k+2} + \tau f^n_{0k+1},
\end{align}
\tag{16}

\begin{align}
\frac{ar}{2}u^n_{0k+1} + (1 + ar)u^n_{0k+2} - \frac{ar}{2}u^n_{0k+3} &= \frac{ar}{2}u^n_{0k+1} + (1 - ar)u^n_{0k+2} + \frac{ar}{2}u^n_{0k+3} + \tau f^n_{0k+2},
\end{align}
\tag{16}

\begin{align}
-aru^n_{0k+1} + \left(1 + \frac{3ar}{2}\right)u^n_{0k+2} - \frac{ar}{2}u^n_{0k+3} &= \frac{ar}{2}u^n_{0k+1} + \left(1 - \frac{ar}{2}\right)u^n_{0k+2} + \frac{ar}{2}u^n_{0k+3} + \tau f^n_{0k+3}.
\end{align}
\tag{16}

Find the values \([u^n_{0k-2}, u^n_{0k-1}, u^n_{0k}, u^n_{0k+1}, u^n_{0k+2}, u^n_{0k+3}]\) by using the following formulas (\( k = 1, 2, \ldots, K - 1 \)):

\begin{align}
\left(1 + \frac{ar}{2}\right)u^n_{m-3} - \frac{ar}{2}u^n_{m-2} &= aru^n_{m-4} + \left(1 - \frac{3ar}{2}\right)u^n_{m-3} + \frac{ar}{2}u^n_{m-2} + \tau f^n_{m-3},
\end{align}
\tag{17}

\begin{align}
-aru^n_{m-3} + \left(1 + ar\right)u^n_{m-2} - \frac{ar}{2}u^n_{m-1} &= aru^n_{m-3} + \frac{ar}{2}u^n_{m-2} + \left(1 - ar\right)u^n_{m-1} + \tau f^n_{m-2},
\end{align}
\tag{17}

\begin{align}
\frac{ar}{2}u^n_{m-2} + (1 + ar)u^n_{m-1} - \frac{ar}{2}u^n_{m} &= \frac{ar}{2}u^n_{m-2} + (1 - ar)u^n_{m-1} + \frac{ar}{2}u^n_{m} + \tau f^n_{m-1}.
\end{align}
\tag{17}

Obviously, each subdomain contains 6 nodes which can be computed by (16) independently.

For the values \([u^n_{m-3}, u^n_{m-2}, u^n_{m-1}]\), by using the formulas as follows:
DDM I can be written as the matrix form

\[ A_1 U^{n+1} = B_1 U^n + F^n, \]

where \( U^n = (u_{11}^n, u_{12}^n, \ldots, u_{m-1}^n)^T \), \( A_1 = (I + \mu G_1) \), and \( B_1 = (I - \mu G_2) \).

The matrices \( G_1 \) and \( G_2 \) are block diagonal matrices which are as follows:

\[ G_1 = \begin{bmatrix} P_1 & Q & \cdots & Q \\ Q & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & Q \\ Q & \cdots & \ddots & P_2 \end{bmatrix}, \]

\[ G_2 = \begin{bmatrix} Q_1 & \cdots & \cdots & Q_2 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ Q_2 & \cdots & \cdots & Q_1 \end{bmatrix}, \]

where

\[ P_1 = \begin{bmatrix} 1 & 1/2 & 0 \\ 1/2 & 1 & 1/2 \\ 0 & 1/2 & 1/2 \end{bmatrix}, \]

\[ P_2 = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1 & 1/2 \\ 0 & 1/2 & 1 \end{bmatrix}, \]

\[ Q = \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \\ -1 & 3/2 \end{bmatrix}, \]

\[ Q_1 = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \\ 0 & 1/2 \\ -1 & 3/2 \end{bmatrix}, \]

\[ Q_2 = \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \\ -1 & 3/2 \end{bmatrix}. \]

Each block matrix system (i.e., each subdomain) can be solved independently. It is evident that DDM I (18) has intrinsic parallelism.

\[ 2.2. \text{DDM II.} \text{ For the values } u_{11}^{n+1}, u_{22}^{n+1}, \ldots, u_{m-1}^{n+1} \text{ by using the following formulas:} \]

![Figure 1: The diagram of Saul'yev asymmetric difference schemes. (a) Scheme (11). (b) Scheme (12). (c) Scheme (13). (d) Scheme (14).](image-url)
\[
\begin{align*}
-\frac{ar}{2}u_{n+1} + (1 + ar)u_1 - \frac{ar}{2}u_2 &= \frac{ar}{2}u_0 + (1 - ar)u_1 + \frac{ar}{2}u_2 + \tau f_1^n, \\
-\frac{ar}{2}u_1 + (1 + ar)u_2 - \frac{ar}{2}u_3 &= \frac{ar}{2}u_1 + (1 - ar)u_2 + \frac{ar}{2}u_3 + \tau f_2^n, \\
-\frac{ar}{2}u_2 + (1 + 3ar)u_3 - aru_4 &= \frac{ar}{2}u_2 + (1 - ar)u_3 + \frac{ar}{2}u_4 + \tau f_3^n, \\
-aru_4 + (1 + 3ar)u_5 - \frac{ar}{2}u_6 &= \frac{ar}{2}u_4 + (1 - ar)u_5 + \frac{ar}{2}u_6 + \tau f_4^n, \\
-\frac{ar}{2}u_4 + (1 + ar)u_5 - \frac{ar}{2}u_6 &= \frac{ar}{2}u_4 + (1 - ar)u_5 + \frac{ar}{2}u_6 + \tau f_5^n, \\
-\frac{ar}{2}u_5 + (1 + ar)u_6 &= \frac{ar}{2}u_5 + (1 - 3ar)u_6 + \tau f_6^n. 
\end{align*}
\]

Find the values \([u_{n+1}^{(1)}, u_{n+1}^{(2)}, u_{n+1}^{(3)}, u_{n+1}^{(4)}, u_{n+1}^{(5)}, u_{n+1}^{(6)}]\) by using the formulas as follows (\(k = 1, 2, \ldots, K - 2\)):

\[
\begin{align*}
\left(1 + \frac{ar}{2}\right)u_{n+1}^{(k+1)} - \frac{ar}{2}u_{n+1}^{(k+2)} &= aru_n^{(k)} + \left(1 - \frac{3ar}{2}\right)u_{n+1}^{(k+1)} + \frac{ar}{2}u_{n+1}^{(k+2)} + \tau f_{n+1}^{(k+1)}, \\
\frac{ar}{2}u_{n+1}^{(k+1)} + (1 + ar)u_{n+1}^{(k+2)} - \frac{ar}{2}u_{n+1}^{(k+3)} &= \frac{ar}{2}u_{n+1}^{(k+1)} + (1 - ar)u_{n+1}^{(k+2)} + \frac{ar}{2}u_{n+1}^{(k+3)} + \tau f_{n+1}^{(k+2)}, \\
\frac{ar}{2}u_{n+1}^{(k+1)} + (1 + 3ar)u_{n+1}^{(k+2)} - aru_{n+1}^{(k+3)} &= \frac{ar}{2}u_{n+1}^{(k+1)} + (1 - ar)u_{n+1}^{(k+2)} + \frac{ar}{2}u_{n+1}^{(k+3)} + \tau f_{n+1}^{(k+3)}, \\
-aru_{n+1}^{(k+3)} + (1 + 3ar)u_{n+1}^{(k+4)} - \frac{ar}{2}u_{n+1}^{(k+5)} &= \frac{ar}{2}u_{n+1}^{(k+3)} + (1 - ar)u_{n+1}^{(k+4)} + \frac{ar}{2}u_{n+1}^{(k+5)} + \tau f_{n+1}^{(k+4)}, \\
-\frac{ar}{2}u_{n+1}^{(k+4)} + (1 + ar)u_{n+1}^{(k+5)} - \frac{ar}{2}u_{n+1}^{(k+6)} &= \frac{ar}{2}u_{n+1}^{(k+4)} + (1 - ar)u_{n+1}^{(k+5)} + \frac{ar}{2}u_{n+1}^{(k+6)} + \tau f_{n+1}^{(k+5)}, \\
-\frac{ar}{2}u_{n+1}^{(k+5)} + (1 + ar)u_{n+1}^{(k+6)} &= \frac{ar}{2}u_{n+1}^{(k+5)} + (1 - 3ar)u_{n+1}^{(k+6)} + \tau f_{n+1}^{(k+6)}. 
\end{align*}
\]

Obviously, each subdomain contains 6 nodes which can be computed by (22) independently.

Find the values \([\nu_{m-1}^{(1)}, \nu_{m-1}^{(2)}, \ldots, \nu_{m-1}^{(6)}]\) by using the formulas as follows:
DDM II can be written as the matrix form

\[ A_1 U^{n+1} = B_2 U^n + F^n, \]

where \( A_1 = (I + \mu G_2) \) and \( B_2 = (I - \mu G_1) \).

The matrices \( G_1 \) and \( G_2 \) are block diagonal matrices which are as follows:

\[
G_1 = \begin{bmatrix}
P_1 & Q & \cdots & Q \\
\end{bmatrix},
\]

\[
G_2 = \begin{bmatrix}
P_1 & Q & \cdots & Q \\
\end{bmatrix}.
\]

Each block matrix system (i.e., each subdomain) can be solved independently. It is evident that DDM II (24) has intrinsic parallelism.

Schemes (15)–(17) and schemes (21)–(23) construct two domain decomposition methods (18) and (24), respectively. The corresponding algorithm can be described in Algorithm 1. The matrix form of Algorithm 1 can be written as follows:

\[
\begin{align*}
A_1 V^{n+1} &= B_1 U^n + F^n, \\
A_2 W^{n+1} &= B_2 U^n + F^n, \\
U^{n+1} &= \frac{1}{2} (V^{n+1} + W^{n+1}), \\
n &= 0, 1, 2, \ldots,
\end{align*}
\]

where \( U^n = (u^n_1, u^n_2, \ldots, u^n_{m-1})^T \), \( V^n = (v^n_1, v^n_2, \ldots, v^n_{m-1})^T \), and \( W^n = (w^n_1, w^n_2, \ldots, w^n_{m-1})^T \).

**Remark 1.** Obviously, C-N scheme (4) is equal to \([(A_1 + A_2)/2)U^{n+1} = ((B_1 + B_2)/2)U^n + F^n]\).

**Remark 2.** Because the SC-N scheme is derived from the C-N scheme, it also has the properties of the C-N scheme, i.e., the SC-N scheme is unconditionally stable and maintains the second-order numerical accuracy \( O(\tau^2 + h^2) \).

### 3. Extension to Two-Dimensional Parabolic Equations

In this section, we will extend Algorithm 1 to solve two-dimensional parabolic equations:

\[
\frac{\partial u}{\partial t} - \left( \frac{1}{2} \frac{\partial^2 u}{\partial x^2} - \frac{1}{2} \frac{\partial^2 u}{\partial y^2} \right) \frac{\partial u}{\partial y} \frac{\partial u}{\partial y} = f(x, y, t), \quad (x, y, t) \in \Omega, \quad t \in (0, T],
\]

\[
u(x, y, 0) = u_0(x, y), \quad (x, y) \in \Omega,
\]

\[
u(x, y, t) = 0, \quad (x, y) \in \partial \Omega, \quad t \in (0, T],
\]

where domain \( \Omega \in (0, L_x) \times (0, L_y) \) and \( a > 0 \) and \( b > 0 \) are diffusion coefficients.

Let \( u^n_{i,j} \) be the approximate solution at \((x_i, y_j, t_n)\), and \( u(x, y, t) \) represents the exact solution of (27). With the same time and space discretization of Algorithm 1, we obtain its extended algorithm by alternating direction implicit (ADI) technique [48] for equations (27)–(29).

#### 3.1. x-Direction

Let \( r_1 = \alpha \tau/(2h^2) \) and \( r_2 = \beta \tau/(2h^2) \), and the matrix form of the SC-N scheme in x-direction can be written as follows:
The matrix form of the SC-N scheme in y-direction can be written as follows:

\[
\begin{equation}
\begin{cases}
(I + r_2 G_1) V_{m}^{n+1/2} = (I - r_2 G_2) U_{m}^{n+1/2} + b_2^{n+1/2}, \\
(I + r_2 G_2) W_{m}^{n+1/2} = (I - r_2 G_1) U_{m}^{n+1/2} + b_2^{n+1/2}, \\
U_{m}^{n+1} = \frac{1}{2} (V_{m}^{n+1/2} + W_{m}^{n+1/2}),
\end{cases}
\end{equation}
\]

where

\[
b_1^n = \begin{bmatrix}
\left( r_2 u_{1,0,j}^n + r_1 u_{1,0,j}^{n+1/2} \right) / 2 \\
\vdots \\
\left( r_2 u_{m-1,j}^n + r_1 u_{m-1,j}^{n+1/2} \right) / 2 \\
\end{bmatrix} + r_2 \left( u_{1,j}^n - 2u_{1,j}^{n+1/2} + u_{1,j+1}^n \right) + \left( \tau f_{1,j}^n / 2 \right)
\]

\[
b_2^n = \begin{bmatrix}
\left( r_2 u_{1,0,j}^n + r_1 u_{1,0,j}^{n+1/2} \right) / 2 \\
\vdots \\
\left( r_2 u_{m-1,j}^n + r_1 u_{m-1,j}^{n+1/2} \right) / 2 \\
\end{bmatrix} + r_2 \left( u_{m-1,j}^n - 2u_{m-1,j}^{n+1/2} + u_{m-1,j+1}^n \right) + \left( \tau f_{m-1,j}^n / 2 \right)
\]

\[
U_{m}^{n+1} = \begin{bmatrix}
u_{1,j}^{n+1/2}, u_{2,j}^{n+1/2}, \ldots, u_{m,j}^{n+1/2}
\end{bmatrix}^T,
\]

\[
V_{m}^{n+1/2} = \begin{bmatrix}
u_{1,j}^{n+1/2}, v_{2,j}^{n+1/2}, \ldots, v_{m,j}^{n+1/2}
\end{bmatrix}^T,
\]

\[
W_{m}^{n+1/2} = \begin{bmatrix}w_{1,j}^{n+1/2}, w_{2,j}^{n+1/2}, \ldots, w_{m,j}^{n+1/2}\end{bmatrix}^T,
\]

\[
j = 1, 2, \ldots, m - 1.
\]
Algorithm 2. We divide the mesh point into many segments, such as convergence in the space is time step and then solve the values along Remark 4. In Algorithm 2, the domain is divided into many first solve the values along direction, and (32) at the half-
where
\[
\begin{align*}
 b_2^{r(1/2)} = \left[ (r_2 u_{j,0}^{r(1/2)} + r_1 u_{j,1}^{r(1/2)})/2 \right] + r_1 \left( u_{j-1,1}^{r(1/2)} - 2u_{j,1}^{r(1/2)} + u_{j+1,1}^{r(1/2)} \right) + (r_1 f_{j,1}^{r(1/2)})/2 \\
- r_1 \left( u_{j-1,2}^{r(1/2)} - 2u_{j,2}^{r(1/2)} + u_{j+1,2}^{r(1/2)} \right) + (r_1 f_{j,2}^{r(1/2)})/2 \\
- r_1 \left( u_{j-1,m-2}^{r(1/2)} - 2u_{j,m-2}^{r(1/2)} + u_{j+1,m-2}^{r(1/2)} \right) + (r_1 f_{j,m-2}^{r(1/2)})/2 \\
\left[ (r_2 u_{j,m}^{r(1/2)} + r_1 u_{j,m-1}^{r(1/2)})/2 \right] + r_1 \left( u_{j-1,m-1}^{r(1/2)} - 2u_{j,m-1}^{r(1/2)} + u_{j+1,m-1}^{r(1/2)} \right) + (r_1 f_{j,m-1}^{r(1/2)})/2 \\
\end{align*}
\]

(33)

The corresponding algorithm can be described in Algorithm 2.

Similar to Algorithm 1, it is obvious that Algorithm 2 has unconditional stability and parallelism.

Remark 3. In fact, \(((I + r_1 G_1) + (I + r_2 G_2))/2) U^{r+1(1/2)} = ((I - r_1 G_1) + (I - r_2 G_2))/2) U^r + b_1^T \) is the C-N scheme in x-direction, and \(((I + r_1 G_1) + (I + r_2 G_2))/2) U^{r+1} = ((I - r_1 G_1) + (I - r_2 G_2))/2) U^{r+1(1/2)} + b_2^{r+1(1/2)} \) is the C-N scheme in y-direction.

Remark 4. In Algorithm 2, the domain is divided into many subdomains by using two DDMs. In each time interval, we first solve the values along x-direction by (30) at the half-time step and then solve the values along y-direction by (32) at the next half-time step. Schemes (30) and (32) lead to block diagonal algebraic systems that can be solved independently so, Algorithm 2 not only is suitable for parallel computation but also maintains the accuracy. Based on the advantage of ADI technique, Algorithm 2 reduces computational complexities. Though it is developed for two-dimensional problems, Algorithm 2 can be easily extended to solve high-dimensional parabolic equations.

4. Numerical Experiments

To illustrate the efficiency of the SC-N scheme for parabolic equations, we will compare the accuracy of the new algorithm with the existing method.

**Example 1**

\[
\begin{align*}
\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} &= (1 + \pi^2) e^{3x} \sin(\pi x), \quad x \in (0, 1), t \in (0, 1], \\
u(x, 0) &= \sin(\pi x), \quad x \in [0, 1], \\
u(0, t) &= u(1, t) = 0, \quad t \in (0, 1].
\end{align*}
\]

(34)

The exact solution of Example 1 is
\[
u(x, t) = e^x \sin(\pi x).
\]

(35)

Firstly, we examine the convergence rate of Algorithm 1. We divide the mesh point into many segments, such as K = 3, K = 4, K = 5, and K = 6. We calculate errors \( L^\infty = \|u(x, t) - u^h \|_\infty \) taking \( r = 0.001 \), and the following rate of convergence in the space is

\[
\text{rate} \approx \frac{\log(L^\infty(h_1)/L^\infty(h_2))}{\log(h_1/h_2)}
\]

(36)

Clearly, the errors appear to be of order \( O(h^2) \) in Table 1. Next, we present the error results of the SC-N scheme in terms of the absolute errors and the relative errors, where the absolute error (A. E.) is defined by

\[
\epsilon_j^n = |u_j^n - u(x_j, t_n)|,
\]

(37)

and the relative error (R. E.) is defined by

\[
E_j^n = \left( \frac{\epsilon_j^n}{u(x_j, t_n)} \right) \times 100%.
\]

(38)
Table 1: Convergence rate of Algorithm 1 for \( h = 0.4 \).

| \( h \)  | \( L^\infty \) error | Rate |
|--------|------------------|------|
| 1/19 (K = 3) | 4.998746e - 4 | – |
| 1/25 (K = 4) | 3.085232e - 4 | 2.0167 |
| 1/31 (K = 5) | 2.002844e - 4 | 1.9955 |
| 1/37 (K = 6) | 1.412298e - 4 | 1.9903 |

Table 2: The absolute errors and relative errors of numerical solutions to Example 1 for \( h = 1/19 \) (i.e., \( K = 3 \)).

| \( x \) | \( y \) | \( t = 0.2 \) | \( t = 0.4 \) | \( t = 0.8 \) |
|--------|--------|-------------|-------------|-------------|
| 0.11   | 0.3731e - 3 | 1.7559e - 2 | 1.9312e - 2 | 1.9534e - 2 |
| 0.21   | 0.6184e - 3 | 1.5402e - 2 | 1.7176e - 2 | 1.7399e - 2 |
| 0.31   | 0.8750e - 3 | 1.5999e - 2 | 1.7754e - 2 | 1.7976e - 2 |
| 0.42   | 0.9753e - 3 | 1.5400e - 2 | 1.7165e - 2 | 1.7388e - 2 |
| 0.52   | 1.0066e - 3 | 1.5496e - 2 | 1.7225e - 2 | 1.7447e - 2 |
| 0.63   | 0.9135e - 3 | 1.5280e - 2 | 1.7037e - 2 | 1.7259e - 2 |
| 0.73   | 0.7582e - 3 | 1.5778e - 2 | 1.7534e - 2 | 1.7757e - 2 |
| 0.84   | 0.5469e - 3 | 1.7563e - 2 | 1.9316e - 2 | 1.9538e - 2 |
| 0.95   | 0.1891e - 3 | 1.7556e - 2 | 1.9309e - 2 | 1.9532e - 2 |

Tables 2 and 3 display the absolute errors and the relative errors obtained by presented Algorithm 1 for \( h = 1/19 \) (i.e., \( K = 3 \)) and \( h = 1/25 \) (i.e., \( K = 4 \)) at \( t = 0.2 \), \( t = 0.4 \), and \( t = 0.8 \) when taking \( r = 1.5 \) (\( r = t/h^2 \)). From Tables 2 and 3, it is obvious that our algorithm has high accuracy.

We compare Algorithm 1 with the ASC-N scheme in [3] and C-N scheme (3) by the maximum errors for \( h = 1/19 \) (i.e., \( K = 3 \)) and \( h = 1/25 \) (i.e., \( K = 4 \)) at different times \( t = 0.2 \), \( t = 0.4 \), \( t = 0.6 \), and \( t = 0.8 \). With the increasing of computation time, the errors of the ASC-N scheme in [3] increase more than those of Algorithm 1 for different \( r (r = t/h^2) \) in Tables 4 and 5. Moreover, the results show that Algorithm 1 can achieve the same accuracy as the classic C-N scheme while maintaining parallelism. We consider an example for \( h = 1/121 \) (i.e., \( K = 20 \)) with large grid ratio \( r = 15 \) (\( r = t/h^2 \)). Table 6 shows that Algorithm 1 has a better accuracy than two others, and it is indicated that Algorithm 1 is stable.

Example 2

\[
\begin{align*}
\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= f(x, y, t), (x, y) \in \Omega, \quad t \in (0, 1], \\
u(x, y, 0) &= \sin(\pi x)\sin(\pi y), \quad (x, y) \in \Omega, \\
u(u, 0, t) &= u(1, 1, t) = 0, \quad y \in [0, 1], t \in (0, 1], \\
u(x, 0, t) &= u(x, 1, t) = 0, \quad x \in [0, 1], t \in (0, 1],
\end{align*}
\]

where the domain is \( \Omega = (0, 1) \times (0, 1) \), and the right side function is

\[
f(x, y, t) = (1 + 2\pi^2)e^{t}\sin(\pi x)\sin(\pi y).
\]
(41)

Let $h = 1/19$ (i.e., $K = 3$), $h = 1/25$ (i.e., $K = 4$), $h = 1/31$ (i.e., $K = 5$), and $h = 1/37$ (i.e., $K = 6$), and the maximum errors of Algorithm 2 for $\tau = 0.0004$ at different times are displayed in Table 7. It is obvious that the accuracy of Algorithm 2 is satisfied. Table 8 shows the comparison of the CPU calculation time among Algorithm 2, the classical C-N scheme, and the ASC-N scheme. Algorithm 2 not only has high accuracy but also has good parallel efficiency.
Table 7: The maximum errors of Algorithm 2 to Example 2 for $\tau = 0.0004$.

| $h$     | $t = 0.2$  | $t = 0.4$  | $t = 0.5$  | $t = 0.6$  | $t = 0.8$  |
|---------|------------|------------|------------|------------|------------|
| $h = 1/19$ | 6.0275$e-4$ | 1.5451$e-4$ | 8.0677$e-5$ | 4.9096$e-5$ |
| $h = 1/25$ | 4.9767$e-4$ | 1.9272$e-4$ | 1.0052$e-4$ | 6.1127$e-5$ |
| $h = 1/31$ | 5.1593$e-4$ | 2.1305$e-4$ | 1.1113$e-4$ | 6.7574$e-5$ |
| $h = 1/37$ | 5.7284$e-4$ | 2.3546$e-4$ | 1.2282$e-4$ | 7.4684$e-5$ |
| $h = 1/37$ | 7.0014$e-4$ | 2.8760$e-4$ | 1.5001$e-4$ | 9.1219$e-5$ |

Table 8: Comparison of three schemes’ calculation time for $\tau = 1.2$ at $t = 0.5$.

| $h$     | Algorithm 2 | C-N scheme | ASC-N scheme |
|---------|-------------|------------|--------------|
| $1/37 (K = 6)$ | 6.4121s  | 20.2625s  | 16.7089s  |
| $1/49 (K = 8)$ | 21.4226s  | 49.9863s  | 40.0513s  |
| $1/61 (K = 10)$ | 53.2623s  | 175.2632s | 146.0665s |
| $1/91 (K = 15)$ | 341.5628s | 1167.2047s | 966.9273s |

5. Conclusion

We have proposed and analyzed the SC-N scheme for parabolic equations. This algorithm consists of two DDMs, which split the Crank–Nicolson scheme; each one is used to solve the values at the same time level. Then, the average of two values is calculated. The SC-N scheme maintains the properties of the C-N scheme, i.e., unconditional stability and second-order numerical accuracy. Then, we extend the new algorithm to two-dimensional parabolic equations by ADI technique, which means that high-dimensional parabolic equations can be solved by the proposed algorithm in this paper. Numerical experiments illustrate the good performance of the new algorithm.

Data Availability

All data generated or analyzed during this study are included in this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank the anonymous referees for helpful comments and suggestions. This research was supported by the PhD research startup foundation of Yantai University (no. 2219002).

References

[1] D. J. Evans, "Alternating group explicit method for the diffusion equation," Applied Mathematical Modelling, vol. 9, no. 3, pp. 201–206, 1985.
[2] V. K. Saul’yev, Integration of Equations of Parabolic Type by Method of Nets, Elsevier, Amsterdam, Netherlands, 1964.
[3] B. Zhang and W. Li, "On alternating segment Crank-Nicolson scheme," Parallel Computing, vol. 20, no. 6, pp. 897–902, 1994.
[4] W. Wang, "A class of alternating segment Crank-Nicolson methods for solving convection-diffusion equations," Computing, vol. 73, pp. 41–55, 2004.
[5] W. Wang and S. Fu, "An unconditionally stable alternating segment difference scheme of eight points for the dispersive equation," International Journal for Numerical Methods in Engineering, vol. 67, no. 3, pp. 435–447, 2006.
[6] Q. Zhang and W. Wang, "A four-order alternating segment Crank-Nicolson scheme for the dispersive equation," Computers & Mathematics with Applications, vol. 57, no. 2, pp. 283–289, 2009.
[7] W.-Q. Wang and Q. Zhang, "A highly accurate alternating 6-point group method for the dispersive equation," International Journal of Computer Mathematics, vol. 87, no. 7, pp. 1512–1521, 2010.
[8] G. Guo, S. Lü, and B. Liu, "Unconditional stability of alternating difference schemes with variable time steplengths for dispersive equation," Applied Mathematics and Computation, vol. 262, pp. 249–259, 2015.
[9] G. Guo and B. Liu, "Unconditional stability of alternating difference schemes with intrinsic parallelism for the fourth-order parabolic equation," Applied Mathematics and Computation, vol. 219, no. 14, pp. 7319–7328, 2013.
[10] G. Guo and S. Lü, "Unconditional stability of alternating difference schemes with intrinsic parallelism for two-dimensional fourth-order diffusion equation," Computers & Mathematics with Applications, vol. 71, no. 10, pp. 1944–1959, 2016.
[11] Y. Bonbendir, X. Antonie, and C. Geuzaine, "A quasi-optimal non-overlapping domain decomposition algorithm for Helmholtz equations," Journal of Computational Physics, vol. 231, pp. 262–280, 2012.
[12] S. Zhu, "Conservative domain decomposition procedure with unconditional stability and second-order accuracy," Applied Mathematics and Computation, vol. 216, no. 11, pp. 3275–3282, 2010.
[13] W. Hao and S. Zhu, "Domain decomposition schemes with high-order accuracy and unconditional stability," Applied Mathematics and Computation, vol. 219, no. 11, pp. 6170–6181, 2013.
[14] Q. Du, M. Mu, and Z. Wu, "Efficient parallel algorithms for parabolic problems," SIAM Journal on Numerical Analysis, vol. 39, no. 5, pp. 1469–1487, 2001.
[15] H. S. Shi and H.-L. Liao, "Unconditional stability of corrected explicit-implicit domain decomposition algorithms for
parallel approximation of heat equations,” SIAM Journal on Numerical Analysis, vol. 44, no. 4, pp. 1584–1611, 2006.

[16] M. Dryja and X. Tu, “A domain decomposition discretization of parabolic problems,” Numerische Mathematik, vol. 107, no. 4, pp. 625–640, 2007.

[17] C. Li and Y. Yuan, “A modified upwind difference domain decomposition method for convection-diffusion equations,” Applied Numerical Mathematics, vol. 59, no. 7, pp. 1584–1598, 2009.

[18] D. Liang and C. Du, “The efficient S-DDM scheme and its analysis for solving parabolic equations,” Journal of Computational Physics, vol. 272, pp. 46–69, 2014.

[19] S. Kumar and M. Kumar, “An analysis of overlapping domain decomposition methods for singularly perturbed reaction-diffusion problems,” Journal of Computational and Applied Mathematics, vol. 281, pp. 250–262, 2015.

[20] Z. Zhou and D. Liang, “A time second-order mass-conserved implicit-explicit domain decomposition scheme for solving the diffusion equations,” Advances in Applied Mathematics and Mechanics, vol. 9, no. 4, pp. 795–817, 2017.

[21] Y. Zhou, “Finite difference method with intrinsic parallelism for quasilinear parabolic systems,” Beijing Math., vol. 2, pp. 1–19, 1996.

[22] Y. Zhou, “General finite difference schemes with intrinsic parallelism for nonlinear parabolic system,” Beijing Math., vol. 2, pp. 20–36, 1996.

[23] Y. Zhou, L. Shen, and G. Yuan, “Some practical difference schemes with intrinsic parallelism for nonlinear parabolic systems,” Chinese Journal of Numerical Mathematics and Applications, vol. 19, pp. 46–57, 1997.

[24] G.-W. Yuan, L.-J. Shen, and Y.-L. Zhou, “Unconditional stability of alternating difference schemes with intrinsic parallelism for two-dimensional parabolic systems,” Numerical Methods for Partial Differential Equations, vol. 15, no. 6, pp. 625–636, 1999.

[25] Z. Sheng, G. Yuan, and X. Hang, “Unconditional stability of parallel difference schemes with second order accuracy for parabolic equation,” Applied Mathematics and Computation, vol. 184, no. 2, pp. 1015–1031, 2007.

[26] G. Yuan, Z. Sheng, and X. Hang, “The unconditional stability of parallel difference schemes with scheme order convergence for nonlinear parabolic system,” Numerical Methods for Partial Differential Equations, vol. 20, no. 1, pp. 45–64, 2007.

[27] Y. Zhuang, “An alternating explicit-implicit domain decomposition method for the parallel solution of parabolic equations,” Journal of Computational and Applied Mathematics, vol. 206, no. 1, pp. 549–566, 2007.

[28] G. Xue and H. Feng, “A new parallel algorithm for solving parabolic equations,” Advances in Difference Equations, vol. 174, pp. 1–16, 2018.

[29] G. Xue and H. Feng, “New parallel algorithm for convection-dominated diffusion equation,” East Asian Journal on Applied Mathematics, vol. 8, no. 2, pp. 261–279, 2018.

[30] G. Xue and H. Feng, “An alternating segment explicit-implicit scheme with intrinsic parallelism for Burgers’ equation,” Journal of Computational and Theoretical Transport, vol. 49, no. 1, pp. 16–30, 2020.

[31] C. Chen and H. Liu, “A two-grid finite volume element method for a nonlinear parabolic problem,” International Journal of Numerical Analysis and Modeling, vol. 12, pp. 197–210, 2015.

[32] C. Chen and X. Zhao, “A posteriori error estimate for finite volume element method of the parabolic equations,” Numerical Methods for Partial Differential Equations, vol. 33, no. 1, pp. 259–275, 2017.

[33] C. Chen, X. Zhang, G. Zhang, and Y. Zhang, “A two-grid finite element method for nonlinear parabolic integro-differential equations,” International Journal of Computer Mathematics, vol. 96, no. 10, pp. 2010–2023, 2019.

[34] C. Chen, W. Liu, and C. Bi, “A two-grid characteristic finite volume element method for semilinear advection-dominated diffusion equations,” Numerical Methods for Partial Differential Equations, vol. 29, no. 5, pp. 1543–1562, 2013.

[35] M. Yang, “Higher-order finite volume element methods based on Barlow points for one-dimensional elliptic and parabolic problems,” Numerical Methods for Partial Differential Equations, vol. 31, no. 4, pp. 977–994, 2015.

[36] W. Gong, M. Hinze, and Z. Zhou, “Finite element method and a priori error estimates for dirichlet boundary control problems governed by parabolic pdes,” Journal of Scientific Computing, vol. 66, no. 3, pp. 941–967, 2016.

[37] N. An, X. Yu, and C. Huang, “Local discontinuous Galerkin methods for parabolic interface problems with homogeneous and non-homogeneous jump conditions,” Computers & Mathematics with Applications, vol. 74, no. 10, pp. 2572–2598, 2017.

[38] X. Wang and Y. Tao, “A new Newton method with memory for solving nonlinear equations,” Mathematics, vol. 8, no. 1, p. 108, 2020.

[39] X. Li, Q. Xu, and A. Zhu, “Weak galerkin mixed finite element methods for parabolic equations with memory,” Discrete & Continuous Dynamical Systems-S, vol. 12, no. 3, pp. 513–531, 2019.

[40] H. Cheng and R. Yuan, “Discontinuous mixed covolume methods for parabolic problems,” The Scientific World Journal, vol. 2014, Article ID 867863, 8 pages, 2017.

[41] W. Gong, M. Hinze, and Z. J. Zhou, “Space-time finite element approximation of parabolic optimal control problems,” Journal of Numerical Mathematics, vol. 20, no. 2, pp. 111–145, 2012.

[42] Q. Yang and X. Zhang, “Discontinuous Galerkin immersed finite element methods for parabolic interface problems,” Journal of Computational and Applied Mathematics, vol. 299, pp. 127–139, 2016.

[43] C. Bi, Y. Lin, and M. Yang, “Finite volume element method for monotone nonlinear elliptic problems,” Numerical Methods for Partial Differential Equations, vol. 29, no. 4, pp. 1097–1120, 2013.

[44] C. Bi and M. Liu, “A discontinuous finite volume element method for second-order elliptic problems,” Numerical Methods for Partial Differential Equations, vol. 28, no. 2, pp. 425–440, 2012.

[45] C. Chen, X. Zhao, and Y. Zhang, “A posteriori error estimate for finite volume element method of the second-order hyperbolic equations,” Mathematical Problems in Engineering, vol. 2015, Article ID 510241, 11 pages, 2015.

[46] X. Shen and A. Zhu, “A Crank-Nicolson linear difference scheme for a BBM equation with a time fractional nonlocal viscous term,” Advances in Difference Equations, vol. 351, pp. 1–12, 2018.

[47] C. Chen, K. Li, Y. Chen, and Y. Huang, “Two-grid finite element methods combined with Crank-Nicolson scheme for nonlinear Sobolev equations,” Advances in Computational Mathematics, vol. 45, no. 2, pp. 611–630, 2019.

[48] F. Zeng, Z. Zhang, and G. E. Karniadakis, “Fast difference schemes for solving high-dimensional time-fractional sub-diffusion equations,” Journal of Computational Physics, vol. 307, pp. 15–33, 2016.