Domain decomposition schemes with high-order accuracy and unconditional stability

Wenrui Hao* Shaohong Zhu†

March 7, 2012

Abstract

Parallel finite difference schemes with high-order accuracy and unconditional stability for solving parabolic equations are presented. The schemes are based on domain decomposition method, i.e., interface values between subdomains are computed by the explicit scheme; interior values are computed by the implicit scheme. The numerical stability and error are derived in the H^1 norm in one dimensional case. Numerical results of both one and two dimensions examining the stability, accuracy, and parallelism of the procedure are also presented.

Keywords: Domain decomposition, Finite difference, Parabolic equation, High-order accuracy, Unconditional stability.

1 Introduction

Domain decomposition is a powerful tool for devising parallel methods to solve time-dependent partial differential equations. There is rich literature on domain decomposition finite difference methods for solving parabolic equations on parallel computers. For the non-overlapping domain decomposition methods, the explicit nature of the calculation at the interface of sub-domain leads some domain decomposition schemes to be conditionally stable, which implies that they have to suffer from temporal step-size restrictions (see [1]-[5]). Schemes with unconditional stability as well as high-order accuracy being desired in the applications, many investigators have turned to improve the stability of the domain decomposition method. For example, the corrected explicit-implicit domain decomposition algorithms were presented in [6] and [7]. By adding the correction step to explicit-implicit domain decomposition methods, updating the interface solutions at each time level, the corrected methods were proved to

*Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556 USA (whao@nd.edu).

†School of Mathematics Science and LPMC, Nankai University, Tianjin, 300071, China (shhzhu@nankai.edu.cn).

have the unconditional stability. While the needless corrected domain decomposition schemes with unconditional stability were presented in [8] and [9]. All of these methods with unconditional stability reach the second order accuracy at most.

The purpose of this paper is to present the domain decomposition finite difference procedure with third-order accuracy and unconditional stability. We first consider the following Dirichlet boundary problem

$$\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2} = 0, x \in (0,1), t \in (0,T], \tag{1.1}$$

$$U(0,t) = U(1,t) = 0, t \in (0,T], \tag{1.2}$$

$$U(x,0) = U_0(x), 0 \leq x \leq 1, \tag{1.3}$$

where the initial function $U_0(x)$ satisfies the boundary condition, i.e., $U_0(0) = U_0(1) = 0$. Then we extend the method to the problem of two dimensional space.

We will introduce two new finite difference schemes for solving (1.1)-(1.3) in Section 2, and sketch the domain decomposition procedure, the numerical stability and convergence in Section 3. In section 4, the proof of the unconditional stability and the error estimate will be given. Numerical examples and examination of the algorithm will be provided in Section 5. In Sections 6 and 7, we extend the method to the problem of two dimensional space and test some examples.

2 Two new finite difference schemes

Taking the usual h, τ mesh in x and t, and denoting the approximate value of $U(x_j, t^n) \equiv U_j^n$ by u_j^n, where $x_j = jh$ and $t^n = n\tau$, we define the following operators

$$\Delta_+ u_j^n = u_{j+1}^n - u_j^n,\ \Delta_- u_j^n = u_j^n - u_{j-1}^n,\ \Delta_\tau u_j^n = \frac{1}{\tau}(u_j^n - u_j^{n-1}).$$

It is well known that the following Taylor expansion resulting in the fully implicit finite difference scheme is valid.

$$\Delta_\tau U_j^{n+1} - \frac{\Delta_+\Delta_- U_j^{n+1}}{h^2} = -\left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{\partial^2 U}{\partial t^2}(x_j, t^{n+1}) + O(\tau^2 + h^4). \tag{2.4}$$

Noticing that

$$\frac{\partial^2 U}{\partial t^2}(x_j, t^{n+1}) = \frac{U_j^{n+1} - 2U_j^n + U_j^{n-1}}{\tau^2} + O(\tau), \tag{2.5}$$

and substituting $\frac{\partial^2 U}{\partial t^2}(x_j, t^{n+1})$ into (2.4), we obtain

$$\Delta_\tau U_j^{n+1} - \frac{\Delta_+\Delta_- U_j^{n+1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{U_j^{n+1} - 2U_j^n + U_j^{n-1}}{\tau^2} = O(\tau^2 + h^4 + \tau h^2). \tag{2.6}$$

Replacing j by $j+1$ and $j-1$ in (2.5), we have

$$U_{j+1}^{n+1} = 2U_{j+1}^{n} - U_{j+1}^{n-1} + \tau^2 \frac{\partial^2 U}{\partial t^2}(x_{j+1}, t^{n+1}) + O(\tau^3)$$

and

$$U_{j-1}^{n+1} = 2U_{j-1}^{n} - U_{j-1}^{n-1} + \tau^2 \frac{\partial^2 U}{\partial t^2}(x_{j-1}, t^{n+1}) + O(\tau^3).$$

Substituting $U_{j\pm1}^{n+1}$into (2.6), we can get

$$\begin{aligned} &\Delta_\tau U_j^{n+1} - \frac{2U_{j+1}^{n} - U_{j+1}^{n-1} - 2U_j^{n+1} + 2U_{j-1}^{n} - U_{j-1}^{n-1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{U_j^{n+1} - 2U_j^n + U_j^{n-1}}{\tau^2} \\ &= \frac{\tau^2}{h^2}\left(\frac{\partial^2 U}{\partial t^2}(x_{j+1}, t^{n+1}) + \frac{\partial^2 U}{\partial t^2}(x_{j-1}, t^{n+1})\right) + O(\tau^2 + h^4 + \tau h^2 + \frac{\tau^3}{h^2}). \end{aligned} \tag{2.7}$$

By omitting the high order term, (2.6) and (2.7) yield two new finite difference schemes for (1.1) :

$$\Delta_\tau u_j^{n+1} - \frac{\Delta_+\Delta_- u_j^{n+1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{\tau^2} = 0, \tag{2.8}$$

$$\begin{aligned} &\Delta_\tau u_j^{n+1} - \frac{\Delta_+\Delta_- u_j^{n+1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{\tau^2} \\ &-r(\Delta_\tau u_{j+1}^n - \Delta_\tau u_{j+1}^{n+1}) - r(\Delta_\tau u_{j-1}^n - \Delta_\tau u_{j-1}^{n+1}) = 0, \end{aligned} \tag{2.9}$$

where $r = \dfrac{\tau}{h^2}$. From the derivation of (2.8) and (2.9), we know that the truncation errors of (2.8) and (2.9) are $O(\tau^2+h^4+\tau h^2)$ and $O\left(\dfrac{\tau^2}{h^2}+\tau^2+h^4+\tau h^2+\dfrac{\tau^3}{h^2}\right)$ respectively. If r is any positive real number, the truncation errors become $O(h^4)$ and $O(h^2)$.

3 Domain decomposition procedure and main results

Suppose $Jh = 1$, $N\tau = T$. For simplicity, we will consider a domain decomposition which involves in decomposing $(0,1)$ into only two subdomains, $(0,\overline{x})$ and $(\overline{x},1)$, where $\overline{x} = x_k$ for some integer k $(1 < k < J-1)$. We use the explicit scheme (2.9) to compute the solution value u_k^{n+1} and the implicit scheme (2.8) to compute other solution values $u_j^{n+1}(j \neq k)$ respectively. The system can be written as

$$\begin{aligned} &L(u_j^{n+1}) = 0,\ 1 \le n \le N-1, 0 < j < J, \\ &u_j^{n+1} = 0,\ j = 0, J, \end{aligned} \tag{3.10}$$

where the linear operator L is

$$L(u_j^{n+1}) = \begin{cases} \Delta_\tau u_j^{n+1} - \dfrac{\Delta_+\Delta_- u_j^{n+1}}{h^2} + \left(\dfrac{\tau}{2} + \dfrac{h^2}{12}\right)\dfrac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{\tau^2}, j \neq k, \\ \Delta_\tau u_j^{n+1} - \dfrac{\Delta_+\Delta_- u_j^{n+1}}{h^2} + \left(\dfrac{\tau}{2} + \dfrac{h^2}{12}\right)\dfrac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{\tau^2} \\ -r(\Delta_\tau u_{j+1}^n - \Delta_\tau u_{j+1}^{n+1}) - r(\Delta_\tau u_{j-1}^n - \Delta_\tau u_{j-1}^{n+1}), j = k. \end{cases} \quad (3.11)$$

The resulting system of equations decouples into two disjoint sets of equations corresponding to the subdomains. These systems can be solved in parallel.

We are now in position to state two main theorems of this paper, which will be proved in the forthcoming sections. For the discrete function $u^n = \{u_j^n | j = 0, 1, ..., J,\ u_0^n = u_J^n = 0\}$, define

$$\|u^n\|^2 = \sum_{j=1}^{J-1} |u_j^n|^2 h, \quad \|\Delta_+ u^n\|^2 = \sum_{j=0}^{J-1} |\Delta_+ u_j^n|^2 h.$$

We have the following theorems.

Theorem 3.1 (Stability) *For any given $r > 0$, the finite difference solutions of the parallel scheme (3.10)-(3.11) satisfy*

$$\max_n \|\Delta_+ u^n\| \leq \sqrt{\frac{1}{2r} + \frac{1}{12r^2} + 1}\,(\|\Delta_+ u^1\| + \|u^1 - u^0\|).$$

Theorem 3.2 (Convergence) *Let $e_j^n = U(x_j, t^n) - u_j^n$. For any given $r > 0$, the finite difference solutions of the parallel scheme (3.10)-(3.11) satisfy*

$$\max_n \|\Delta_+ e^n\| \leq C(\|\Delta_+ e^1\| + \|e^1 - e^0\| + h^4),$$

where C is a positive constant independent of h and τ.

Since the finite difference scheme (3.10) has three time levels, besides taking $u_j^0 = U_0(jh)$, thus $e_j^0 = 0$, we need to find other methods to solve $u_j^1, \forall j$. In order to match high order accuracy, we can use either fourth order explicit schemes such as the impact scheme or the high-order parallel iterative method [10] to compute u_j^1, i.e., let $|e_j^1| \leq O(h^4)$. Then according to Theorem 3.2, scheme (3.10) will reach third-order accuracy.

4 Proof of stability and convergence

We first state three auxiliary lemmas. The stability and convergence results are then derived.

Lemma 4.1 (Discrete Poincare Inequality) *For the discrete function* $u^n = \{u_j^n | j = 0, 1, ..., J,\ u_0^n = u_J^n = 0\}$*, there exists*

$$\|u^n\| \le \left\| \frac{\Delta_+ u^n}{h} \right\|.$$

Lemma 4.2 (Discrete Green Theorem) *If* u_j *and* v_j *are discrete functions on the set* $\{x_j | j = 0, 1, \cdots, J\}$*, then we have*

$$\sum_{j=1}^{J-1} u_j \Delta_+ v_j = -\sum_{j=1}^{J} v_j \Delta_- u_j - u_0 v_1 + u_J v_J.$$

Lemma 4.3 *For any given* $r > 0$*,* $f \in L_\infty(0, T; L_2(0,1))$*, the finite difference solutions of system* $L(u_j^{n+1}) = f_j^{n+1}$ *with the Dirichlet boundary condition satisfy*

$$\max_n \|\Delta_+ u^n\|^2 \le \|\Delta_+ u^1\|^2 + \left(\frac{1}{2r} + \frac{1}{12r^2} + 1 \right) \|u^1 - u^0\|^2 + \frac{Th^2}{r} \max_n \|f^n\|^2.$$

Lemma 4.1 and Lemma 4.2 are proved in [11]. Before proving the stability and convergence, we will give the proof of Lemma 4.3.

4.1 Proof of Lemma 4.3

Denoting $w_j^{n+1} = u_j^{n+1} - u_j^n$, we can rewrite $L(u_j^{n+1}) = f_j^{n+1}$ with the Dirichlet boundary condition as

$$\begin{cases} w_j^{n+1} - r\Delta_+\Delta_- u_j^{n+1} + \left(\frac{1}{2} + \frac{1}{12r} \right)(w_j^{n+1} - w_j^n) = f_j^{n+1}\tau, j \ne k, \\ w_j^{n+1} - r\Delta_+\Delta_- u_j^{n+1} + \left(\frac{1}{2} + \frac{1}{12r} \right)(w_j^{n+1} - w_j^n) \\ -r(w_{j+1}^n - w_{j+1}^{n+1}) - r(w_{j-1}^n - w_{j-1}^{n+1}) = f_j^{n+1}\tau, j = k, \\ u_j^{n+1} = 0,\ j = 0, J. \end{cases}$$

Multiplying the above equations by $w_j^{n+1} h, j = 1, 2, \cdots, J-1$ and summing them up respect to j, we have

$$\sum_{j=1}^{J-1} (w_j^{n+1})^2 h - r \sum_{j=1}^{J-1} w_j^{n+1} \Delta_+\Delta_- u_j^{n+1} h + \left(\frac{1}{2} + \frac{1}{12r} \right) \sum_{j=1}^{J-1} (w_j^{n+1} - w_j^n) w_j^{n+1} h$$
$$-r w_k^{n+1} (w_{k+1}^n - w_{k+1}^{n+1}) h - r w_k^{n+1} (w_{k-1}^n - w_{k-1}^{n+1}) h = \sum_{j=1}^{J-1} f_j^{n+1} w_j^{n+1} \tau h. \quad (4.12)$$

From Lemma 4.2, we have

$$\sum_{j=1}^{J-1} w_j^{n+1}\Delta_+\Delta_- u_j^{n+1} = -\sum_{j=1}^{J}\Delta_- u_j^{n+1}\Delta_- w_j^{n+1}$$

$$= \quad -\frac{1}{2}\sum_{j=0}^{J-1}(\Delta_+ u_j^{n+1})^2 + \frac{1}{2}\sum_{j=0}^{J-1}(\Delta_+ u_j^{n})^2 - \frac{1}{2}\sum_{j=0}^{J-1}(\Delta_+ u_j^{n+1} - \Delta_+ u_j^{n})^2.$$

Then (4.12) becomes

$$\begin{aligned}
&\|w^{n+1}\|^2 + \frac{r}{2}\Big[\sum_{j=0}^{J-1}(\Delta_+ u_j^{n+1})^2 h - \sum_{j=0}^{J-1}(\Delta_+ u_j^{n})^2 h\Big] + \frac{r}{2}\sum_{j=0}^{J-1}(\Delta_+ u_j^{n+1} - \Delta_+ u_j^{n})^2 h \\
&+\Big(\frac{1}{2} + \frac{1}{12r}\Big)\Big[\frac{1}{2}\sum_{j=1}^{J-1}(w_j^{n+1})^2 h - \frac{1}{2}\sum_{j=1}^{J-1}(w_j^{n})^2 h + \frac{1}{2}\sum_{j=1}^{J-1}(w_j^{n+1} - w_j^{n})^2 h\Big] \\
&+rhw_k^{n+1}(w_{k+1}^{n+1} + w_{k-1}^{n+1}) \\
= \quad & rhw_k^{n+1}(w_{k+1}^{n} + w_{k-1}^{n}) + \sum_{j=1}^{J-1} w_j^{n+1} f_j^{n+1}\tau h \\
\leq \quad & rh\Big(\frac{(w_k^{n+1})^2 + (w_{k+1}^{n})^2}{2} + \frac{(w_k^{n+1})^2 + (w_{k-1}^{n})^2}{2}\Big) + \sum_{j=1}^{J-1}\frac{(w_j^{n+1})^2 + (f_j^{n+1}\tau)^2}{2} h \\
\leq \quad & rh(w_k^{n+1})^2 + rh\frac{(w_{k+1}^{n})^2 + (w_{k-1}^{n})^2}{2} + \frac{1}{2}\|w^{n+1}\|^2 + \frac{\tau^2}{2}\|f^{n+1}\|^2. \qquad (4.13)
\end{aligned}$$

Noting that

$$\begin{aligned}
&\frac{r}{2}\sum_{j=0}^{J-1}(\Delta_+ u_j^{n+1} - \Delta_+ u_j^{n})^2 h + rhw_k^{n+1}(w_{k+1}^{n+1} + w_{k-1}^{n+1}) \\
= \quad & \frac{r}{2}\sum_{j=0, j\neq k, k-1}^{J-1}(\Delta_+ w_j^{n+1})^2 h + rh\Big[\frac{(w_{k+1}^{n+1})^2 + (w_{k-1}^{n+1})^2}{2}\Big] + rh(w_k^{n+1})^2,
\end{aligned}$$

we can simplify (4.13) as follows

$$\begin{aligned}
&\frac{1}{2}\|w^{n+1}\|^2 + \frac{r}{2}\Big[\|\Delta_+ u^{n+1}\|^2 - \|\Delta_+ u^{n}\|^2\Big] + \frac{r}{2}\sum_{j=0, j\neq k, k-1}^{J-1}(\Delta_+ w_j^{n+1})^2 h \\
&+\Big(\frac{1}{4} + \frac{1}{24r}\Big)(\|w^{n+1}\|^2 - \|w^{n}\|^2 + \|w^{n+1} - w^{n}\|^2) \\
&+rh\Big[\frac{(w_{k+1}^{n+1})^2 + (w_{k-1}^{n+1})^2}{2} - \frac{(w_{k+1}^{n})^2 + (w_{k-1}^{n})^2}{2}\Big] \leq \frac{\tau^2}{2}\|f^{n+1}\|^2
\end{aligned}$$

Since $\|w^{n+1}\|^2 \geq 0$, $\sum_{j=0, j\neq k,k-1}^{J-1} (\Delta_+ w_j^{n+1})^2 h \geq 0$, and $\|w^{n+1} - w^n\|^2 \geq 0$, then we have

$$\frac{r}{2}\Big[\|\Delta_+ u^{n+1}\|^2 - \|\Delta_+ u^n\|^2\Big] + \left(\frac{1}{4} + \frac{1}{24r}\right)(\|w^{n+1}\|^2 - \|w^n\|^2)$$
$$+rh\Big[\frac{(w_{k+1}^{n+1})^2 + (w_{k-1}^{n+1})^2}{2} - \frac{(w_{k+1}^{n})^2 + (w_{k-1}^{n})^2}{2}\Big] \leq \frac{\tau^2}{2}\|f^{n+1}\|^2$$

Summing up respect to n, we get

$$\frac{r}{2}\|\Delta_+ u^{n+1}\|^2 + \left(\frac{1}{4} + \frac{1}{24r}\right)\|w^{n+1}\|^2 + rh\frac{(w_{k+1}^{n+1})^2 + (w_{k-1}^{n+1})^2}{2}$$
$$\leq \frac{r}{2}\|\Delta_+ u^1\|^2 + \left(\frac{1}{4} + \frac{1}{24r}\right)\|w^1\|^2 + rh\frac{(w_{k+1}^{1})^2 + (w_{k-1}^{1})^2}{2} + \frac{T\tau}{2}\max_n \|f^n\|^2.$$

Thus

$$\|\Delta_+ u^{n+1}\|^2 \leq \|\Delta_+ u^1\|^2 + \left(\frac{1}{2r} + \frac{1}{12r^2} + 1\right)\|u^1 - u^0\|^2 + \frac{Th^2}{r}\max_n \|f^n\|^2.$$

Lemma 4.3 is proved. □

4.2 Proof of stability

Let $f = 0$ in Lemma 4.3, thus Theorem 3.1 is proved. □

4.3 Proof of convergence

According to the parallel scheme (3.10)-(3.11), the errors e_j^{n+1} $(1 \leq n < N)$ follows the below equations:

$$\Delta_\tau e_j^{n+1} - \frac{\Delta_+\Delta_- e_j^{n+1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{e_j^{n+1} - 2e_j^n + e_j^{n-1}}{\tau^2} = G_j^{n+1}, j \neq k,$$
$$\Delta_\tau e_j^{n+1} - \frac{\Delta_+\Delta_- e_j^{n+1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{e_j^{n+1} - 2e_j^n + e_j^{n-1}}{\tau^2}$$
$$-r(\Delta_\tau e_{j+1}^n - \Delta_\tau e_{j+1}^{n+1}) - r(\Delta_\tau e_{j-1}^n - \Delta_\tau e_{j-1}^{n+1}) = \Phi^{n+1}h^2 + G_j^{n+1}, j = k,$$
$$e_j^{n+1} = 0,\ j = 0, J,$$

where $\Phi^{n+1} = 2r^2\frac{\partial^2 U}{\partial t^2}(x_k, t^{n+1}), |G_j^{n+1}| \leq O(h^4)$. In order to get the error estimates for e_j^{n+1}, we assume that $e_j^{n+1} = p_j^{n+1} + q_j^{n+1}$, where p_j^{n+1} and q_j^{n+1} are the solutions of the following problems respectively.

- problem I

$$\begin{aligned}
&\Delta_\tau p_j^{n+1} - \frac{\Delta_+\Delta_- p_j^{n+1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{p_j^{n+1} - 2p_j^n + p_j^{n-1}}{\tau^2} = G_j^{n+1}, j \neq k,\\
&\Delta_\tau p_j^{n+1} - \frac{\Delta_+\Delta_- p_j^{n+1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{p_j^{n+1} - 2p_j^n + p_j^{n-1}}{\tau^2}\\
&-r(\Delta_\tau p_{j+1}^n - \Delta_\tau p_{j+1}^{n+1}) - r(\Delta_\tau p_{j-1}^n - \Delta_\tau p_{j-1}^{n+1}) = G_j^{n+1}, j = k, \qquad (4.14)\\
&p_j^{n+1} = 0,\ j = 0, J,\\
&p_j^0 = e_j^0,\ p_j^1 = e_j^1, \forall j.
\end{aligned}$$

- problem II

$$\begin{aligned}
&\Delta_\tau q_j^{n+1} - \frac{\Delta_+\Delta_- q_j^{n+1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{q_j^{n+1} - 2q_j^n + q_j^{n-1}}{\tau^2} = 0, j \neq k,\\
&\Delta_\tau q_j^{n+1} - \frac{\Delta_+\Delta_- q_j^{n+1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{q_j^{n+1} - 2q_j^n + q_j^{n-1}}{\tau^2}\\
&-r(\Delta_\tau q_{j+1}^n - \Delta_\tau q_{j+1}^{n+1}) - r(\Delta_\tau q_{j-1}^n - \Delta_\tau q_{j-1}^{n+1}) = \Phi^{n+1}h^2, j = k,\\
&q_j^{n+1} = 0,\ j = 0, J,\\
&q_j^0 = q_j^1 = 0, \forall j.
\end{aligned}$$

From lemma 4.3, we can obtain the estimate for p_j^{n+1}, i.e.,

$$\begin{aligned}
\|\Delta_+ p^n\|^2 &\le \|\Delta_+ p^1\|^2 + \left(\frac{1}{2r} + \frac{1}{12r^2} + 1\right)\|p^1 - p^0\|^2 + \frac{Th^2}{r}\max_n \|G^n\|^2\\
&\le \left(\frac{1}{2r} + \frac{1}{12r^2} + 1\right)(\|\Delta_+ e^1\| + \|e^1 - e^0\|)^2 + O(h^{10}). \qquad (4.15)
\end{aligned}$$

For estimating q_j^{n+1}, we first consider $\overline{q}_j^{n+1}$ $(-1 \le n \le N-1)$which satisfies the following equations

$$\begin{aligned}
&-\frac{\Delta_+\Delta_- \overline{q}_j^{n+1}}{h^2} = 0, j \neq k,\\
&-\frac{\Delta_+\Delta_- \overline{q}_k^{n+1}}{h^2} = \Phi^{n+1}h^2,\\
&\overline{q}_0^{n+1} = \overline{q}_J^{n+1} = 0.
\end{aligned}$$

Then the formula of $\overline{q}_j^{n+1}$ is

$$\overline{q}_j^{n+1} = \begin{cases} 0, j = 0, J,\\ \dfrac{j}{J}\displaystyle\sum_{k+1}^{J} \Phi^{n+1}h^4, 1 \le j \le k,\\ \dfrac{J-j}{J}\displaystyle\sum_{i=1}^{k-1} \Phi^{n+1}h^4, k < j < J. \end{cases}$$

Hence

$$\|\Delta_+\overline{q}^{n+1}\| \le O(h^4) \text{ and } \|\overline{q}^{n+1} - \overline{q}^n\| \le O(\tau h^3).$$

Define $\widetilde{q}_j^{n+1} = q_j^{n+1} - \overline{q}_j^{n+1}$, then we have

$$\begin{aligned}
&\Delta_\tau \widetilde{q}_j^{n+1} - \frac{\Delta_+\Delta_-\widetilde{q}_j^{n+1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{\widetilde{q}_j^{n+1} - 2\widetilde{q}_j^n + \widetilde{q}_j^{n-1}}{\tau^2} = -R_j^{n+1}, j \ne k,\\
&\Delta_\tau \widetilde{q}_j^{n+1} - \frac{\Delta_+\Delta_-\widetilde{q}_j^{n+1}}{h^2} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{\widetilde{q}_j^{n+1} - 2\widetilde{q}_j^n + \widetilde{q}_j^{n-1}}{\tau^2}\\
&-r(\Delta_\tau \widetilde{q}_{j+1}^n - \Delta_\tau \widetilde{q}_{j+1}^{n+1}) - r(\Delta_\tau \widetilde{q}_{j-1}^n - \Delta_\tau \widetilde{q}_{j-1}^{n+1}) = -R_j^{n+1}, j = k,\\
&\widetilde{q}_j^{n+1} = 0,\ j = 0, J,\\
&\widetilde{q}_j^0 = -\overline{q}_j^0,\ \widetilde{q}_j^1 = -\overline{q}_j^1, \forall j,
\end{aligned}$$

where

$$R_j^{n+1} = \begin{cases}
\Delta_\tau \overline{q}_j^{n+1} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{\overline{q}_j^{n+1} - 2\overline{q}_j^n + \overline{q}_j^{n-1}}{\tau^2}, j \ne k,\\
\Delta_\tau \overline{q}_j^{n+1} + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{\overline{q}_j^{n+1} - 2\overline{q}_j^n + \overline{q}_j^{n-1}}{\tau^2} - r(\Delta_\tau \overline{q}_{j+1}^n - \Delta_\tau \overline{q}_{j+1}^{n+1})\\
-r(\Delta_\tau \overline{q}_{j-1}^n - \Delta_\tau \overline{q}_{j-1}^{n+1}), j = k,
\end{cases}$$

and $\|R^{n+1}\| \le O(h^3)$. From Lemma 4.3, we can obtain the estimate of $\|\Delta_+\widetilde{q}^{n+1}\|$.

$$\|\Delta_+\widetilde{q}^{n+1}\|^2 \quad \le \quad \|\Delta_+\widetilde{q}^1\|^2 + \left(\frac{1}{2r} + \frac{1}{12r^2} + 1\right)\|\widetilde{q}^1 - \widetilde{q}^0\|^2 + \frac{Th^2}{r}\max_n \|R^n\|^2 \le O(h^8).$$

Thus,

$$\|\Delta_+ q^{n+1}\| \le \|\Delta_+\widetilde{q}^{n+1}\| + \|\Delta_+\overline{q}^{n+1}\| \le O(h^4).$$

Combining with (4.15), we get

$$\|\Delta_+ e^{n+1}\| \le \|\Delta_+ p^{n+1}\| + \|\Delta_+ q^{n+1}\| \le C(\|\Delta_+ e^1\| + \|e^1 - e^0\| + h^4),$$

where C is a positive constant independent of h and τ. Theorem 3.2 is proved. □

5 Numerical experiments

In this section, some numerical results are presented to show the stability, accuracy, and parallelism of the scheme described above, and the computational costs are also presented. All the experiments are run on a cluster consisting of a manager that uses one core of a Xeon 5410 processor and up to 32 computing nodes, each containing two Xeon 5410 processors running 64-bit Linux, i.e., each node consists of 8 processing cores.

We consider the problem defined in equations (1.1)-(1.3) with $U_0(x) = \sin(\pi x)$. Obviously the exact solution of the equations is $U(x,t) = e^{-\pi^2 t}\sin(\pi x)$.

First, we verify the stability of the scheme by taking the step size $h = 10^{-3}$, $r = 1, 10, 100, 1000$ with two subdomains. FIG. 1 clearly shows that the norm of u^n doesn't occur blowing up even if r is large enough. This explains the unconditional stability of the scheme.

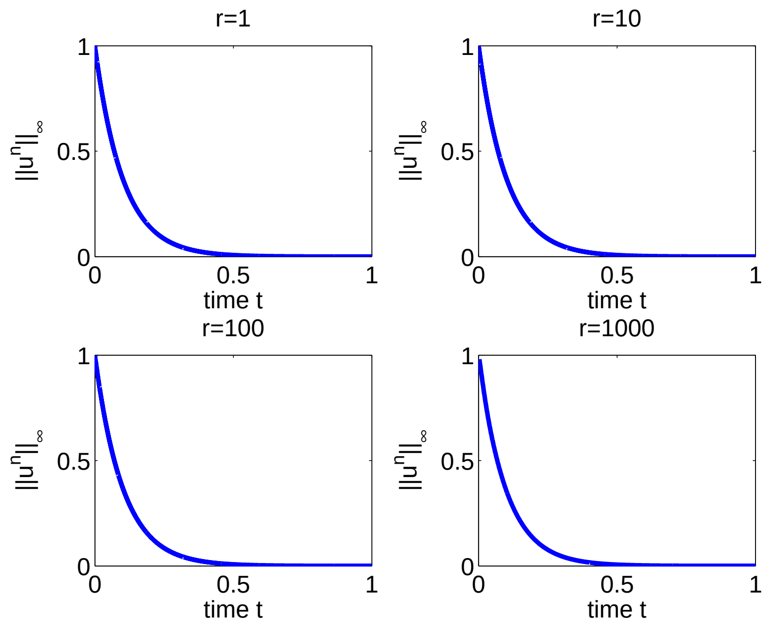

Figure 1: The infinity norm of u^n v.s. t

Second, we examine the numerical errors in the solutions. Table 1 shows that the errors for each case are roughly of the same order of magnitude, and the errors appear to be $O(h^3)$ in each case.

Third, we test the speed-up for the scheme. Here we take $h = 10^{-5}$, $\tau = 10^{-6}$, i.e., $r = 10^4$, $T = 0.5$ and list the time for computing and the speed-up in Table 5, which shows that the scheme has a coemptive parallelism.

6 Extension to two dimensional case

In this section, $U(x, y, t)$ will be a solution of the following Dirichlet boundary problem on $\Omega = (0, 1) \times (0, 1)$,

Table 1: Numerical errors for different grid points

$r = 10, T = 1$					
2 processors		4 processors		8 processors	
J	$\|u^n - U^n\|/h^3$	J	$\|u^n - U^n\|/h^3$	J	$\|u^n - U^n\|/h^3$
1000	2.06821	1000	4.05448	1000	8.07385
2000	2.06715	2000	4.05336	2000	8.07276
4000	2.06316	4000	4.04938	4000	8.06881
8000	1.94919	8000	3.93590	8000	7.95632
$r = 100, T = 1$					
2 processors		4 processors		8 processors	
J	$\|u^n - U^n\|/h^3$	J	$\|u^n - U^n\|/h^3$	J	$\|u^n - U^n\|/h^3$
1000	2.06703e+02	1000	4.04766e+02	1000	8.04137e+02
2000	2.06729e+02	2000	4.05299e+02	2000	8.06955e+02
4000	2.06690e+02	4000	4.05307e+02	4000	8.07216e+02
8000	2.06820e+02	8000	4.05434e+02	8000	8.07356e+02
$r = 1000, T = 1$					
2 processors		4 processors		8 processors	
J	$\|u^n - U^n\|/h^3$	J	$\|u^n - U^n\|/h^3$	J	$\|u^n - U^n\|/h^3$
1000	2.01883e+04	1000	4.04424e+04	1000	8.03459e+04
2000	2.03740e+04	2000	3.93818e+04	2000	8.03843e+04
4000	2.06360e+04	4000	4.03957e+04	4000	8.01579e+04
8000	2.06634e+04	8000	4.05137e+04	8000	8.06558e+04

$$\frac{\partial U}{\partial t} - \Delta U = 0, (x, y) \in \Omega, t \in (0, T], \tag{6.16}$$

$$U(x, y, t) = 0, (x, y) \in \partial\Omega, t \in (0, T], \tag{6.17}$$

$$U(x, y, 0) = U_0(x, y), (x, y) \in \Omega, \tag{6.18}$$

where the initial function $U_0(x, y)$ satisfies the boundary condition, and $\Delta U = \dfrac{\partial^2 U}{\partial x^2} + \dfrac{\partial^2 U}{\partial y^2}$. We extend the domain decomposition method stated in Section 3 to the above problem. Take

$$\Omega_1 = \{(x, y) \in \Omega : x < \overline{x}\},$$
$$\Omega_2 = \{(x, y) \in \Omega : x > \overline{x}\}.$$

Let $x_i = ih$ same as in Section 3, and let $y_j = jh$. Suppose that there exists an integer k such that $\overline{x} = x_k$ (see Fig. 2). In analogy with Section 3 we call points (x_i, y_j, t^n) as boundary points if $(x_i, y_j) \in \partial\Omega$, and interface points if $i = k$. Otherwise, we call them interior points. The values $u^n_{i,j}$ will approximate $U(x_i, y_j, t^n) \equiv U^n_{i,j}$.

We denote

$$\Delta_+\Delta_- u_{i,j} = u_{i+1,j} - 2u_{i,j} + u_{i-1,j}, \quad \delta_+\delta_- u_{i,j} = u_{i,j+1} - 2u_{i,j} + u_{i,j-1},$$

Table 2: Comparison of time for different processors

Processors n	Time t_n (seconds)	Speed-up t_1/t_n
1	4862.66	-
2	2450.82	1.98
4	1250.29	3.89
8	621.74	7.82
16	325.05	14.96
32	165.85	29.32
64	100.45	48.40

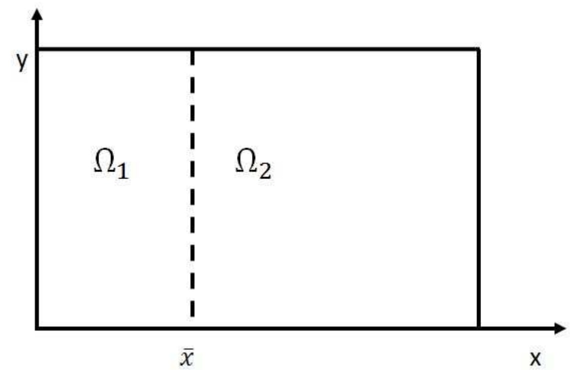

Figure 2: domain decomposition for two dimensional case

and

$$\Delta_\tau u_{i,j}^n = \frac{u_{i,j}^n - u_{i,j}^{n-1}}{\tau}.$$

Then

$$\begin{aligned}
\frac{\Delta_+\Delta_- U_{i,j}}{h^2} &= \frac{\partial^2 U}{\partial x^2}(x_i,y_j,t) + \frac{h^2}{12}\frac{\partial^4 U}{\partial x^4}(x_i,y_j,t) + O(h^4),\\
\frac{\delta_+\delta_- U_{i,j}}{h^2} &= \frac{\partial^2 U}{\partial y^2}(x_i,y_j,t) + \frac{h^2}{12}\frac{\partial^4 U}{\partial y^4}(x_i,y_j,t) + O(h^4),\\
\frac{\delta_+\delta_-\Delta_+\Delta_- U_{i,j}}{h^4} &= \frac{\partial^4 U}{\partial x^2 \partial y^2}(x_i,y_j,t) + O(h^2).
\end{aligned}$$

Thus we have

$$\begin{aligned}
&\frac{\Delta_+\Delta_- U_{i,j} \;+\; \delta_+\delta_- U_{i,j}}{h^2} + \frac{1}{6h^2}\delta_+\delta_-\Delta_+\Delta_- U_{i,j}\\
=\;& \Delta U(x_i,y_j,t) + \frac{h^2}{12}\left(\frac{\partial^4 U}{\partial x^4}(x_i,y_j,t) + 2\frac{\partial^4 U}{\partial x^2\partial y^2}(x_i,y_j,t) + \frac{\partial^4 U}{\partial y^4}(x_i,y_j,t)\right) + O(h^4)\\
=\;& \Delta U(x_i,y_j,t) + \frac{h^2}{12}\frac{\partial^2 U}{\partial t^2}(x_i,y_j,t) + O(h^4). \qquad (6.19)
\end{aligned}$$

Moreover,

$$\Delta_\tau U_{i,j}^{n+1} = \frac{\partial U}{\partial t}(x_i, y_j, t^{n+1}) - \frac{\tau}{2}\frac{\partial^2 U}{\partial t^2}(x_i, y_j, t^{n+1}) + O(\tau^2). \tag{6.20}$$

Subtract (6.20) with (6.19), we obtain

$$\begin{aligned} & \Delta_\tau U_{i,j}^{n+1} - \frac{\Delta_+\Delta_- U_{i,j}^{n+1} + \delta_+\delta_- U_{i,j}^{n+1}}{h^2} - \frac{1}{6h^2}\delta_+\delta_-\Delta_+\Delta_- U_{i,j}^{n+1} \\ = \quad & -\left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{\partial^2 U}{\partial t^2}(x_i, y_j, t^{n+1}) + O(\tau^2 + h^4). \end{aligned}$$

Therefore, the fourth order finite difference scheme for (6.16) is

$$\begin{aligned} \Delta_\tau u_{i,j}^{n+1} - \frac{\Delta_+\Delta_- u_{i,j}^{n+1} + \delta_+\delta_- u_{i,j}^{n+1}}{h^2} - \frac{1}{6h^2}\delta_+\delta_-\Delta_+\Delta_- u_{i,j}^{n+1} \\ + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{u_{i,j}^{n+1} - 2u_{i,j}^{n} + u_{i,j}^{n-1}}{\tau^2} = 0, \end{aligned} \tag{6.21}$$

which is a nine-point finite difference scheme with three levels (see Fig. 3).

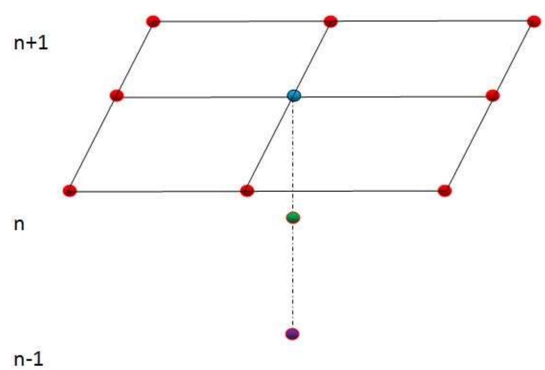

Figure 3: nine point scheme

Denote

$$\begin{aligned} \overline{\Delta} u_{i,j} &= \frac{u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i+1,j+1} - 4u_{i,j}}{2} \\ &= \Delta_+\Delta_- u_{i,j} + \delta_+\delta_- u_{i,j} + \frac{1}{2}\delta_+\delta_-\Delta_+\Delta_- u_{i,j}. \end{aligned}$$

Then (6.21) is equivalent to

$$\begin{aligned} \Delta_\tau u_{i,j}^{n+1} - \frac{\overline{\Delta} u_{i,j}^{n+1} + 2\Delta_+\Delta_- u_{i,j}^{n+1} + 2\delta_+\delta_- u_{i,j}^{n+1}}{3h^2} \\ + \left(\frac{\tau}{2} + \frac{h^2}{12}\right)\frac{u_{i,j}^{n+1} - 2u_{i,j}^{n} + u_{i,j}^{n-1}}{\tau^2} = 0, \end{aligned}$$

which can be used for computing $u_{i,j}^{n+1}$ when $i \neq k$. When $i = k$, $u_{i\pm1,j}^{n+1}$ in (6.21) are replaced by $2u_{i\pm1,j}^{n} - u_{i\pm1,j}^{n-1}$. In this case, we need to solve a tridiagonal matrix using Thomas algorithm for computing $u_{k,j}^{n+1}$.

It is straightforward to extend the two-subdomain results to many subdomains on x direction by cutting the whole domain into vertical strips. Moreover, we can also extend the domain decomposition scheme to three dimensional case by dividing x axis into many subdomains.

7 Two dimensional numerical experiments

We consider the problem defined in equations (6.16)-(6.18) with $U_0(x, y) = \sin(\pi x)\sin(\pi y)$. Obviously the exact solution of the equations is $U(x, y, t) = e^{-2\pi^2 t}\sin(\pi x)\sin(\pi y)$. The computing time and the speed-up of the scheme (6.19-6.20) are shown in Table 3 with $h = 10^{-3}$, $\tau = 10^{-2}$ and $T = 1$.

Table 3: Comparison of time for two dimensional case

Processors n	Time t_n (seconds)	Speed-up t_1/t_n
1	10h23m19s	-
2	6h01m29s	1.72
4	2h52m14s	3.61
8	1h23m54s	7.41
16	42m47s	14.53
32	21m23s	29.06
64	12m58s	47.91

7.1 Lotka-Volterra system

The competitive Lotka-Volterra equations are a simple model of the population dynamics of species competing for some common resource. Given two populations, u and v, with logistic dynamics, the Lotka-Volterra formulation adds an additional term to account for the species' interactions. Thus the competitive Lotka-Volterra equations are:

$$\begin{aligned} u_t &= \Delta u + u(1 - v), (x, y) \in \Omega \\ v_t &= \Delta v - v(1 - u), (x, y) \in \Omega \\ v &= u = 0, (x, y) \in \partial\Omega. \end{aligned}$$

We choose random values for $u(x, y)$ and $v(x, y)$ as initial conditions shown in FIG 4. We take $h = 0.001$, $\tau = 0.01$, $T = 1$ and show the computing time in Table 4. FIG 5 shows the numerical solution for $T = 1$.

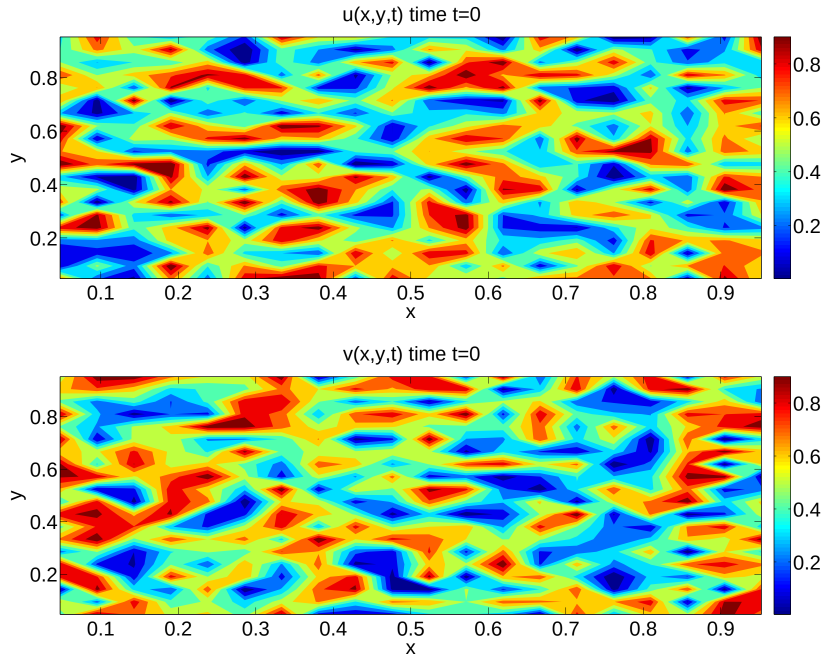

Figure 4: Initial condition for the Lotka-Volterra system

Table 4: Comparison of time for Lotka-Volterra system

Processors n	Time t_n (seconds)	Speed-up t_1/t_n
1	21h51m32s	-
2	11h45m8s	1.86
4	6h12m36s	3.52
8	2h58m57s	7.37
16	1h33m37s	14.16
32	45m28s	28.85
64	28m12s	46.51
128	14m33s	90.12
256	7m28s	175.61

Conclusion

We presented domain decomposition finite difference schemes with unconditional stability and third-order accuracy for the parabolic system. Error estimate and stability of the numerical solutions have been derived for one dimensional case. The scheme is easy to implement the parallelism and is extended in

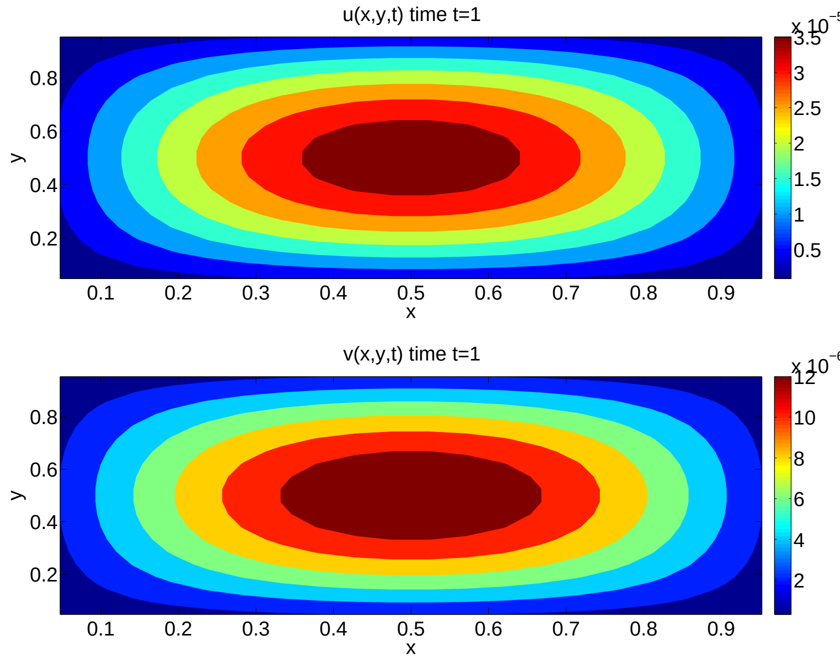

Figure 5: Numerical solutions for $T = 1$

two and three dimensional case. The numerical results demonstrate the good performance of the parallel scheme, namely, unconditional stability, the third order accuracy and high degree of parallelism.

References

[1] C.N. Dawson, Q. Du, and T. F. Dupont, A finite difference domain decomposition algorithm for numerical solution of the Heat equation,*Math. Comp.*, Vol. 57, 63–71 (1991).

[2] G. Yuan, S. Zhu and L. Shen, Domain decomposition algorithm based on the group explicit formula for the heat equation, *Int. J. Comput. Math*, Vol. 82, 1295–1306 (2005).

[3] S. Zhu, Z. Yu and J. Zhao, A high-order parallel finite difference algorithm, *Applied Math. and Comp.*, Vol. 183, 365-372 (2006).

[4] C.N. Dawson and T.F. Dupont, Explicit/implicit conservative domain decomposition procedures for parabolic problems based on block-centered finite differences, *SIAM J. Numer. Anal.*, Vol. 31, 1045–1061 (1994).

[5] G. Yuan and L. Shen, Stability and convergence of the explicit-implicit conservative domain decomposition procedure for parabolic problems, *Computers and Math. with Applications*, Vol. 47, 793-801 (2004).

[6] H. Shi and H. Liao, Unconditional stability of corrected explicit-implicit domain decomposition algorithms for parallel approximation of heat equations, *SIAM J. Numer. Anal.*, Vol. 44, 1584-1611 (2006).

[7] H. Liao, H. Shi and Z. Sun, Corrected explicit-implicit domain decomposition algorithms for two-dimensional semilinear parabolic equations, *Science in China Series A*, Vol. 52, 2362-2388 (2009).

[8] Z. Sheng, G. Yuan and X. Hang, Unconditional stability of parallel difference schemes with second order accuracy for parabolic equation, *Applied Math. and Comp.*, Vol. 184, 1015-1031 (2007).

[9] S. Zhu, Conservative domain decomposition procedure with unconditional stability and second-order accuracy, *Applied Math. and Comp.*, Vol.216, 3275-3282 (2010).

[10] W. Hao and S. Zhu, Parallel iterative methods for parabolic equations, *Int. J. Comput. Math*, Vol. 86, 431–440 (2009).

[11] Y. Zhou, Applications of discrete functional analysis to the finite difference method, *International Academic publishers* (1991).