

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## New Finite Difference Domain Decomposition Algorithms for Two-Dimensional Heat Equation

<sup>1</sup>Yuanfeng Jin, <sup>1</sup>Lijie Zhao, <sup>2</sup>Jingyuan Li and <sup>3</sup>Tinghuai Ma

<sup>1</sup>Mathematics, Yanbian University, China

<sup>2</sup>Department of Computer Engineering, Kyunghee University, Korea

<sup>3</sup>School of Computer and Software,  
Nanjing University of Information Science and Technology, China

**Abstract:** The domain decomposition scheme is a high efficient and useful method which is widely used in the field of the finite difference parallel computing on parabolic equation. The natural method of parallel solution of the partial differential equation is to divide the solution area into several sub-regions and then independently calculate the problem of each sub-region. To solve the two-dimensional heat equation on parallel computers, we present new domain decomposition algorithms wherein the space domain is divided into two independent sub-regions along with x-axis or divided into four independent sub-regions along with the x-axis and y-axis. The values of the interface points between sub-domains are calculated by Du Fort-Frankel scheme. The values of the interior points are solved by the fully implicit scheme.

**Key words:** Finite difference, domain decomposition method, two-dimensional heat equation, parallel computing

### INTRODUCTION

In recent years, with the need of great scale scientific computing and the maturity of parallel computing environments, domain decomposition method (Quarteroni and Valli, 1999; Dawson and Dupont, 1992; Marini Quarteroni, 1988; Mathew *et al.*, 1998) has been an effective approach to solve partial differential equations numerically. There are a lot of good work in this field: Dawson *et al.* (1991), Wan *et al.* (2002), who made domain decomposition algorithm of the explicit-implicit scheme to let out  $D^2$  times over than the classical explicit scheme on the restriction of the stability and the convergence that is  $\Delta t/h^2 \leq D^2/2$ , but the accuracy achieved  $o(\Delta t + h^2 + H^3)$ . The idea in Zhang (1998) is almost same with Dawson *et al.* (1991), however, the former used explicit scheme of Rodrigue and Wolitzer on the interface point. The stability condition is  $\Delta t/h^2 \leq 2$ ; the error is  $o(\Delta t + h^2)$ . Peaceman and Rachford (1995) and Zhou and Yuan (1999) proposed the parallel difference format for parabolic equation with the error  $o(\Delta t/h^{2/3} + h^{1/2})$ . The reference Zhang and Shen (2002) used the Saul'yev asymmetric difference scheme which is adopt the stride step  $H$  to use in the interface point, furthermore, constructed a new domain decomposition algorithm and give a prior error estimate  $o(\Delta t + h^2 + H^3)$  which is in circumstances of two sub-regions and the stable condition is  $\Delta t/h^2 \leq D^2$ . In

order to improve the accuracy of the interface point, the reference Du *et al.* (2001) considered the multi-step explicit scheme and the one-step high-level explicit scheme.

**Domain decomposition algorithm for one-dimensional heat equation:** Zuo and Yuan (2003) proposed a new domain decomposition algorithm for one-dimensional heat equation by using Du Fort-Frankel scheme at the interface points and the fully implicit scheme at the interior points.

Firstly problem of solving the one-dimensional heat equation is considered:

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0, & x \in (0,1), t \in (0,T] \\ u(x,0) = u^0(x), & x \in (0,1) \\ u(0,t) = u(1,t) = 0, & t \in (0,T] \end{cases} \quad (1)$$

where,  $u^0(x)$  is a given function.

For simplicity, at first we divide domain  $(0, 1)$  into only two sub-domains  $(0, \bar{x})$  and  $(\bar{x}, 1)$ . Suppose  $N, M, D$  are positive integers. Let  $h=1/N$  and  $x_i = i_h, i = 0, \dots, N$ . Take  $\tau = \Delta t = T/M$  and  $t^n = n\Delta t$ , with the assumption that  $\bar{x} = x_k > 0$  for integer  $K$ . A related parameter is  $H = Dh$  and  $H \leq \min(\bar{x}, 1 - \bar{x})$ .

We will refer to points  $(x_i, t^n)$  as boundary points if  $i = 0$  or  $N$ , or if  $n = 0$ . Similarly, we referred to them as interface points if  $x = \bar{x}$  and  $n > 0$ . Otherwise, they are interior points.

We defined a function  $u(x, t)$  at mesh point  $(x_i, t^n)$ . Let  $u_i^n = u(x_i, t^n)$ . Also we defined two difference operators:

$$\partial_{t,\Delta t} u_i^{n-1} = \frac{u_i^n - u_i^{n-1}}{\tau} \quad (2)$$

$$\partial_{x,h}^2 u_i^n = \frac{u_{i+1}^{n-1} - 2u_i^{n-1} + u_{i-1}^{n-1}}{h^2} \quad (3)$$

The one-dimensional finite difference domain decomposition algorithm designed in Zuo and Yuan (2003) is as follows:

**Algorithm 1**

$U_i^n = u_i^n$  at boundary points:

$$\frac{U_i^{n+1} - U_i^n}{2\tau} - \frac{U_{i+D}^n - (U_i^{n+1} + U_i^n) + U_{i-D}^n}{H^2} = 0 \quad (4)$$

at interface points,  $n = 1, 2, \dots, M$

$$\frac{U_i^{n+1} - U_i^n}{\tau} - \frac{U_{i+D}^n - 2U_i^n + U_{i-D}^n}{h^2} = 0$$

at interior points,  $n = 0, 1, 2, \dots, M$

$$\frac{U_i^{n+1} - U_i^n}{\tau} - \frac{U_{i+D}^n - 2U_i^n + U_{i-D}^n}{H^2} = 0$$

at interface points,  $n = 0$

where,  $U_i^n$  is the numerical solution of the exact solution of the problem (1). Algorithm 1 can be shown that the computation from time level  $n-1$  to  $n$  is at first to solve the interface points by using Du Fort-Frankel scheme and then to solve the interior points on the two sub-domains  $(0, \bar{x})$  and  $(\bar{x}, 1)$  independently. For the solution  $U_i^n$  of Algorithm 1 it holds that:

**Theorem 1:** The truncation error of Algorithm 1 is:

$$o(\tau + H^2 + \frac{\tau^2}{H^2})$$

It is generally required to limit the step size  $\tau$  and  $h$ , when,

$$\tau \rightarrow 0, \frac{\tau}{H} \rightarrow 0$$

in the calculation to achieve the accuracy  $O(\Delta t + h^2)$ .

**Theorem 2:** Algorithm 1 is absolutely stable.

In this study, new domain decomposition algorithms are presented for the two-dimensional heat equation based on Algorithm 1 in Zuo and Yuan (2003). The algorithms are unconditional stable and their truncation errors are:

$$o(\tau + H^2 + \frac{\tau^2}{H^2})$$

**New algorithms for the two-dimensional heat equation:**

Consider the following two-dimensional heat conduction equation,  $u(x, y, t)$  is a solution of the heat equation on  $\Omega = (0,1) \times (0, 1)$ . Specifically,  $u$  satisfies:

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 0, & (x, y) \in \Omega, t \in (0,1) \\ u(x, y, 0) = u^0(x, y), & (x, y) \in \Omega \\ u(x, y, t) = 0, & (x, y) \in \partial\Omega. \end{cases} \quad (5)$$

We start a simple two-domain scheme. Take:

$$\Omega_1 = \{(x, y) \in \Omega : x < \bar{x}\}, \quad \Omega_2 = \{(x, y) \in \Omega : \bar{x} < x\} \quad (6)$$

Let  $y_j = jh, j = 0, \dots, N$ . In analogy with the one-dimensional case we will call a points  $(x_i, y_j, t^n)$  as boundary points if  $n = 0$  or if  $(x_i, y_j) \in \partial\Omega$ . Such a point with  $x_i = \bar{x}$  will be an interface point if it is not a boundary point. The remaining points  $(x_i, y_j, t^n)$  are in  $(\Omega_1 \cup \Omega_2) \times (0, T]$  and are interior points. The values  $U_{i,j}^n$  will approximate  $u_{i,j}^n = u(x_i, y_j, t^n)$ , let  $u_i^n = u(x_i, t^n)$ . Defined the difference operators:

$$\partial_{t,\Delta t} u_{i,j}^n = \frac{u_{i,j}^n - u_{i,j}^{n-1}}{\tau} \quad (7)$$

$$\partial_{x,h}^2 u_{i,j}^n = \frac{u_{i+1,j}^{n-1} - 2u_{i,j}^{n-1} + u_{i-1,j}^{n-1}}{h^2} \quad (8)$$

$$\partial_{y,h}^2 u_{i,j}^n = \frac{u_{i,j+1}^{n-1} - 2u_{i,j}^{n-1} + u_{i,j-1}^{n-1}}{h^2} \quad (9)$$

We extended the one-dimensional parabolic equation to two-dimensional one. At interface points we use the improved Du Fort-Frankel scheme; at interior points we use the full-implicit scheme. The algorithm is as follows:

**Algorithm 2:**

$U_{i,j}^n = u_{i,j}^n$  at boundary points

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n-1}}{2\tau} - \frac{U_{i+D,j}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i-D,j}^n}{H^2} - \frac{U_{i,j+1}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i,j-1}^n}{h^2} = 0$$

at interface points,  $n = 1, 2, \dots, M$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\tau} - \frac{U_{i+D,j}^n - 2U_{i,j}^{n+1} + U_{i-1,j}^n}{h^2} - \frac{U_{i,j+1}^n - 2U_{i,j}^{n+1} + U_{i,j-1}^n}{h^2} = 0$$

at interior points,  $n = 0, 1, 2, \dots, M$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\tau} - \frac{U_{i+D,j}^n - 2U_{i,j}^n + U_{i-D,j}^n}{H^2} - \frac{U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n}{h^2} = 0$$

at interface points,  $n = 0$

(10)

**Algorithm 3:**

$U_{i,j}^n = u_{i,j}^n$  at boundary points

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n-1}}{2\tau} - \frac{U_{i+D,j}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i-1,j}^n}{h^2} - \frac{U_{i,j+D}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i,j-D}^n}{H^2} = 0$$

at interface points  $(x_i, y_k, t^n) = (\bar{x}_i, \bar{y}, t^n)$ ,  $n = 1, 2, \dots, M$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n-1}}{2\tau} - \frac{U_{i+D,j}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i-D,j}^n}{H^2} - \frac{U_{i,j+1}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i,j-1}^n}{h^2} = 0$$

at interface points  $(x_k, y_j, t^n) = (\bar{x}, \bar{y}_j, t^n)$ ,  $n = 1, 2, \dots, M$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\tau} - \frac{U_{i+D,j}^n - 2U_{i,j}^{n+1} + U_{i-1,j}^n}{h^2} - \frac{U_{i,j+1}^n - 2U_{i,j}^{n+1} + U_{i,j-1}^n}{h^2} = 0$$

at interior points,  $n = 0, 1, 2, \dots, M$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\tau} - \frac{U_{i+D,j}^n - 2U_{i,j}^n + U_{i-1,j}^n}{h^2} - \frac{U_{i,j+D}^n - 2U_{i,j}^n + U_{i,j-D}^n}{H^2} = 0$$

at interface points  $(x_i, y_k, t^n) = (\bar{x}_i, \bar{y}, t^n)$ ,  $n = 0$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\tau} - \frac{U_{i+D,j}^n - 2U_{i,j}^n + U_{i-D,j}^n}{H^2} - \frac{U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n}{h^2} = 0$$

at interface points  $(x_k, y_j, t^n) = (\bar{x}, \bar{y}_j, t^n)$ ,  $n = 0$

(11)

**Algorithm 4:**

$U_{i,j}^n = u_{i,j}^n$  at boundary points

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n-1}}{2\tau} - \frac{U_{i+D,j}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i-D,j}^n}{H^2} - \frac{U_{i,j+D}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i,j-D}^n}{H^2} = 0$$

at interface points  $(x_i, y_k, t^n) = (\bar{x}_i, \bar{y}, t^n)$ ,  $(x_k, y_j, t^n) = (\bar{x}, \bar{y}_j, t^n)$   
 $D \leq i \leq N - D$ ,  $D \leq j \leq N - D$ ,  $n = 1, 2, \dots, M$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n-1}}{2\tau} - \frac{U_{i+D,j}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i-1,j}^n}{h^2} - \frac{U_{i,j+D}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i,j-D}^n}{H^2} = 0$$

at interface points  $(x_i, y_k, t^n) = (\bar{x}_i, \bar{y}, t^n)$ ,  $0 < i < D$ ,  $N - D < i < N$ ,  $n = 1, 2, \dots, M$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n-1}}{2\tau} - \frac{U_{i+D,j}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i-D,j}^n}{H^2} - \frac{U_{i,j+1}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i,j-1}^n}{h^2} = 0$$

at interface point  $(x_k, y_j, t^n) = (\bar{x}, \bar{y}_j, t^n)$ ,  $0 < j < D$ ,  $N - D < j < N$ ,  $n = 1, 2, \dots, M$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\tau} - \frac{U_{i+D,j}^n - 2U_{i,j}^{n+1} + U_{i-1,j}^n}{h^2} - \frac{U_{i,j+D}^n - 2U_{i,j}^{n+1} + U_{i,j-D}^n}{H^2} = 0$$

at interior points  $n = 0, 1, 2, \dots, M$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n-1}}{2\tau} - \frac{U_{i+D,j}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i-D,j}^n}{H^2} - \frac{U_{i,j+D}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i,j-D}^n}{H^2} = 0$$

at interface points  $(x_i, y_k, t^n) = (\bar{x}_i, \bar{y}, t^n)$ ,  $(x_k, y_j, t^n) = (\bar{x}, \bar{y}_j, t^n)$   
 $D \leq i \leq N - D$ ,  $D \leq j \leq N - D$ ,  $n = 0$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n-1}}{2\tau} - \frac{U_{i+D,j}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i-1,j}^n}{h^2} - \frac{U_{i,j+D}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i,j-D}^n}{H^2} = 0$$

at interface points  $(x_i, y_k, t^n) = (\bar{x}_i, \bar{y}, t^n)$ ,  $0 < i < D$ ,  $N - D < i < N$ ,  $n = 0$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n-1}}{2\tau} - \frac{U_{i+D,j}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i-D,j}^n}{H^2} - \frac{U_{i,j+1}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i,j-1}^n}{h^2} = 0$$

at interface points  $(x_k, y_j, t^n) = (\bar{x}, \bar{y}_j, t^n)$ ,  $0 < j < D$ ,  $N - D < j < N$ ,  $n = 0$

(12)

**Stability and truncation error analysis of the new algorithms:** In order to verify the stability of the Algorithm 2, 3 and 4, we developed the following lemmas:

**Lemma 1:** (Li, 2004) if  $b$  and  $c$  are real numbers and  $x_1, x_2$  are the roots of  $x^2 - bx - c = 0$ , then we have  $|x_i| < 1$ ,  $i = 1, 2$  if and only if  $|b| \leq 1 - 2c < 2$ .

Here, we only discuss the stability and error of Algorithm 4.

**Theorem 3:** Algorithm 4 is unconditional stable.

We only examine the difference scheme at interface points, the formula at interface points of Eq. 12 can be deformed into:

$$\left(1 + \frac{2(D^2 + 1)\tau}{H^2}\right)U_{i,j}^{n+1} = \frac{2\tau}{H^2}(U_{i+D,j}^n + U_{i-D,j}^n) + \left(1 - \frac{2(D^2 + 1)\tau}{H^2}\right)U_{i,j}^{n-1} + \frac{2\tau}{H^2}(U_{i,j+1}^n + U_{i,j-1}^n)$$

Transform it into its equivalent two-level difference equations:

$$\begin{cases} \left(1 + \frac{2(D^2 + 1)\tau}{H^2}\right)U_{i,j}^{n+1} = \frac{2\tau}{H^2}(U_{i+D,j}^n + U_{i-D,j}^n) + \left(1 - \frac{2(D^2 + 1)\tau}{H^2}\right)U_{i,j}^n + \frac{2\tau}{H^2}(U_{i,j+1}^n + U_{i,j-1}^n) \\ V_{i,j}^{n+1} = U_{i,j}^n \end{cases}$$

Let  $U = (U, V)^T$  and make  $U_{j,k}^n = V^n e^{i\alpha(j+k)h}$  to substitute the above formula, then:

$$\begin{pmatrix} 1 + 2(D^2 + 1)\lambda & 0 \\ 0 & 1 \end{pmatrix} V^{n+1} e^{i\alpha(j+k)h} = \begin{pmatrix} 0 & 1 - 2(D^2 + 1)\lambda \\ 1 & 0 \end{pmatrix} V^n e^{i\alpha(j+k)h} + \begin{pmatrix} 2\lambda & 0 \\ 0 & 0 \end{pmatrix} V^n e^{i\alpha(j+k)h} e^{i\alpha D h} + \begin{pmatrix} 2\lambda & 0 \\ 0 & 0 \end{pmatrix} V^n e^{i\alpha(j+k)h} e^{-i\alpha D h} + \begin{pmatrix} 2\lambda & 0 \\ 0 & 0 \end{pmatrix} V^n e^{i\alpha(j+k)h} e^{-i\alpha h} + \begin{pmatrix} 2\lambda & 0 \\ 0 & 0 \end{pmatrix} V^n e^{i\alpha(j+k)h} e^{i\alpha h} = \begin{pmatrix} 8\lambda \cos \alpha D h & 1 - 2(D^2 + 1)\lambda \\ 1 & 0 \end{pmatrix} V^n e^{i\alpha(j+k)h}$$

$$V^{n+1} = \begin{pmatrix} \frac{8\lambda \cos \alpha D h}{1 + 2(D^2 + 1)\lambda} & \frac{1 - 2(D^2 + 1)\lambda}{1 + 2(D^2 + 1)\lambda} \\ 1 & 0 \end{pmatrix} V^n$$

the growth matrix is:

$$G(\tau, k) = \begin{pmatrix} \frac{8\lambda \cos \alpha D h}{1 + 2(D^2 + 1)\lambda} & \frac{1 - 2(D^2 + 1)\lambda}{1 + 2(D^2 + 1)\lambda} \\ 1 & 0 \end{pmatrix}$$

the characteristic equation of  $G(\tau, k)$  is:

$$\omega^2 - \frac{8\lambda \cos \alpha Dh}{1 + 2(D^2 + 1)\lambda} \omega - \frac{1 - 2(D^2 + 1)\lambda}{1 + 2(D^2 + 1)\lambda} = 0 \quad (18)$$

by Lemma 1, the characteristic roots of  $G(\tau, k)$  satisfy the following condition:

$$|b| = \left| \frac{8\lambda \cos \alpha Dh}{1 + 2(D^2 + 1)\lambda} \right| \leq \left| \frac{4\lambda(1 + D^2)}{1 + 2(D^2 + 1)\lambda} \right| = 1 - c \leq 2 \quad (19)$$

It is able to get the growth matrix  $|G| \leq 1$ , thus proving that the algorithm is unconditionally stable.

Similar as the proof of Theorem 3, the Algorithms 2 and 3 are unconditionally stable.

**Theorem 4:** The truncation error of Algorithm 4 is:

$$o(\tau + H^2 + \frac{\tau^2}{H^2})$$

**Proof:** Here, we analysis the truncation error of Eq. 12 at point  $(i, j, n)$ :

$$\begin{aligned} & \frac{U_{i,j}^{n+1} - U_{i,j}^{n-1}}{2\tau} - \frac{U_{i+D,j}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i-D,j}^n}{H^2} - \frac{U_{i,j+D}^n - (U_{i,j}^{n+1} + U_{i,j}^{n-1}) + U_{i,j-D}^n}{H^2} \\ &= \frac{U_{i,j}^n + \tau \frac{\partial U_{i,j}^n}{\partial t} + o(\tau^2) - (U_{i,j}^n - \tau \frac{\partial U_{i,j}^n}{\partial t} + o(\tau^2))}{2\tau} \\ &= \frac{U_{i,j}^n + H \frac{\partial U_{i,j}^n}{\partial x} + \frac{H^2}{2} \frac{\partial^2 U_{i,j}^n}{\partial x^2} + \frac{H^3}{6} \frac{\partial^3 U_{i,j}^n}{\partial x^3} + o(H^4) - [U_{i,j}^n + \tau \frac{\partial U_{i,j}^n}{\partial t} + o(\tau^2)]}{H^2} \\ &= \frac{-[U_{i,j}^n - \tau \frac{\partial U_{i,j}^n}{\partial t} + o(\tau^2)] + U_{i,j}^n - H \frac{\partial U_{i,j}^n}{\partial x} + \frac{H^2}{2} \frac{\partial^2 U_{i,j}^n}{\partial x^2} - \frac{H^3}{6} \frac{\partial^3 U_{i,j}^n}{\partial x^3} - o(H^4)}{H^2} \\ &= \frac{U_{i,j}^n + H \frac{\partial U_{i,j}^n}{\partial y} + \frac{H^2}{2} \frac{\partial^2 U_{i,j}^n}{\partial y^2} + \frac{H^3}{6} \frac{\partial^3 U_{i,j}^n}{\partial y^3} + o(H^4) - [U_{i,j}^n + \tau \frac{\partial U_{i,j}^n}{\partial t} + o(\tau^2)]}{H^2} \\ &= \frac{-[U_{i,j}^n - \tau \frac{\partial U_{i,j}^n}{\partial t} + o(\tau^2)] + U_{i,j}^n - H \frac{\partial U_{i,j}^n}{\partial y} + \frac{H^2}{2} \frac{\partial^2 U_{i,j}^n}{\partial y^2} - \frac{H^3}{6} \frac{\partial^3 U_{i,j}^n}{\partial y^3} - o(H^4)}{H^2} \\ &= \frac{\partial U_{i,j}^n}{\partial t} - \frac{\partial^2 U_{i,j}^n}{\partial x^2} - \frac{\partial^2 U_{i,j}^n}{\partial y^2} + o(\tau) + o(H^2) + o(\frac{\tau^2}{H^2}) = o(\tau + H^2 + \frac{\tau^2}{H^2}) \end{aligned} \quad (20)$$

Similar as Algorithm 4, the truncation errors of Algorithm 2 and 3 are:

Table 1: The numerical results when  $h = 0.05, \Delta t = 0.0001, r = 0.4$

y	Exact solution	Algorithm 1	Absolute error 1	Algorithm 2	Absolute error 2
0.1	1.5983e-05	1.8040e-05	2.0564e-06	2.1304e-05	5.3213e-06
0.3	4.1845e-05	4.7229e-05	5.3838e-06	5.0783e-05	8.9378e-06
0.5	5.1723e-05	5.8378e-05	6.6548e-06	6.1546e-05	9.7916e-06

Table 2: The numerical results when  $h = 0.05, \Delta t = 0.0002, r = 8$

y	Exact solution	Algorithm 1	Absolute error 1	Algorithm 2	Absolute error 2
0.1	1.5983e-05	Overflow	Overflow	3.6547e-05	2.0564e-05
0.3	4.1845e-05	Overflow	Overflow	9.5683e-05	5.3838e-05
0.5	5.1723e-05	Overflow	Overflow	1.1827e-04	6.6548e-05

$$o(\tau + H^2 + \frac{\tau^2}{H^2})$$

it is generally required to limit the step size  $\tau$  and  $h$ , when  $\tau \rightarrow 0, (\tau/H) \rightarrow 0$ , in the calculation to achieve the accuracy  $O(\Delta t + h^2)$ .

## RESULTS

Let  $u^o(x, y) = \sin(\pi x)\sin(\pi y)$  in the two-dimensional heat Eq. 5. Then the exact solution of Eq. 5 is:

$$u(x, y, t) = \exp(-2\pi^2 t)\sin(\pi x)\sin(\pi y)$$

In Table 1-6,  $N = 20, D = 2, t = 0.5, x = 0.5$  absolute error 1 denotes the absolute error between exact solution and Algorithm 1, absolute error 2 denotes the absolute error between exact solution and Algorithm 2, absolute error 3 denotes the absolute error between exact solution and Algorithm 3, absolute error 4 denotes the absolute error between exact solution and Algorithm 4.

Table 1 and 2 show the comparisons of the calculation results between Algorithm 2 and 1 when the grid ratio  $r$  is 0.4 and 8, respectively for two dimensional heat equation. From the numerical results we conclude that: if the value  $r \leq 2$ , the degrees of accuracy between two methods are comparable. While when the value  $r > 2$ , the numerical solution of the Algorithm 1 begins to show the obvious instability. Algorithm 2 maintains the unconditional stability without degrading the overall approximation order.

Table 3 and 4 show the comparisons of the calculation results between Algorithm 3 and Algorithms 1 when the grid ratio  $r$  is 0.4 and 8 respectively for two-dimensional heat equation. From the numerical results we conclude that: if the value  $r \leq 2$ , the degrees of accuracy between two methods are comparable. While when the value  $r > 2$ , the numerical solution of the Algorithm 1 begins to show the obvious instability. Algorithm 3 not only maintains the unconditional stability, but also does not decrease the overall approximation order.

Table 3: The numerical results when  $h = 0.05, \Delta t = 0.0001, r = 0.4$

y	Exact solution	Algorithm 1	Absolute error 1	Algorithm 3	Absolute error 3
0.1	1.5983e-05	1.8040e-05	2.0564e-06	2.0349e-05	4.3658e-06
0.3	4.1845e-05	4.7229e-05	5.3838e-06	4.9999e-05	8.1546e-06
0.5	5.1723e-05	5.8378e-05	6.6548e-06	6.1056e-05	9.3325e-06

Table 4: The numerical results when  $h = 0.05, \Delta t = 0.0002, r = 8$

y	Exact solution	Algorithm 1	Absolute error 1	Algorithm 3	Absolute error 3
0.1	1.5983e-05	Overflow	Overflow	3.1309e-05	1.5326e-05
0.3	4.1845e-05	Overflow	Overflow	9.2853e-05	5.1008e-05
0.5	5.1723e-05	Overflow	Overflow	1.1619e-04	6.4467e-05

Table 5: The numerical results when  $h = 0.05, \Delta t = 0.0001, r = 0.4$

y	Exact solution	Algorithm 1	Absolute error 1	Algorithm 4	Absolute error 4
0.1	1.5983e-05	1.8040e-05	2.0564e-06	1.8738e-05	2.7549e-06
0.3	4.1845e-05	4.7229e-05	5.3838e-06	4.9729e-05	7.8841e-06
0.5	5.1723e-05	5.8378e-05	6.6548e-06	6.0462e-05	8.7392e-06

Table 6: The numerical results when  $h = 0.05, \Delta t = 0.0002, r = 8$

y	Exact solution	Algorithm 1	Absolute error 1	Algorithm 4	Absolute error 4
0.1	1.5983e-05	Overflow	Overflow	2.5857e-05	9.8743e-06
0.3	4.1845e-05	Overflow	Overflow	7.7709e-05	3.5864e-05
0.5	5.1723e-05	Overflow	Overflow	1.0527e-04	5.3549e-05

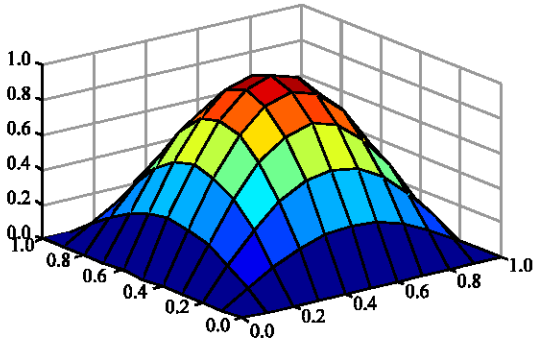


Fig. 1: The exact solution when  $h = 0.01, \Delta t = 0.01, y = 0.01$

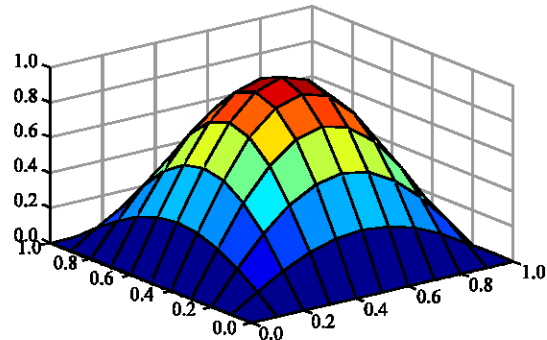


Fig. 2: The approximate solution when  $h = 0.01, \Delta t = 0.01, y = 0.01$

The Table 5 and 6 show the comparisons of the calculation results between Algorithm 4 and Algorithm 1 when the grid ratio  $r$  is 0.4 and 8, respectively for two dimensional heat equation. Based on the numerical results, the following conclusions are drawn: if the value  $r \leq 2$ , the degrees of accuracy between two methods are comparable. When the value  $r > 2$ , the numerical solution of the Algorithm 1 begins to show the obvious instability. Algorithm 4 maintains both the unconditional stability and the overall approximation order.

The real solution and approximate solution obtained from Algorithm 2 are shown in Fig. 1 and 2, respectively. The experimental results are obtained under the following conditions:  $h$  is 0.01;  $\Delta t$  is 0.01;  $y$  is 0.01.

From Fig. 1 and 2, we find that the approximate solution of Algorithm 2 is comparable with the exact solution of Algorithm 1. The truncation error of Algorithm 2 is:

$$O(\tau + H^2 + \frac{\tau^2}{H^2})$$

Moreover, this algorithm is absolute stable. This verifies the superiority of our algorithm.

### CONCLUSION

In this study, we developed a novel domain decomposition method with better stability and calculation accuracy, while we obtain the stability condition and the maximum norm error estimates.

In the two-dimensional case, through integrating the interface, we use Du Fort-Frankel scheme to establish a series of domain decomposition methods. Those methods improve the stability condition when compared to the other methods which are proposed in the previous study.

Analysis results show that appropriately integrating the interface, particularly, the use of Du Fort-Frankel scheme can realize the domain decomposition, meanwhile, can greatly improve the stability conditions of the algorithm without decreasing the overall approximation order of the algorithm.

#### **ACKNOWLEDGMENTS**

This research was also supported by Natural science fund for colleges and universities in Jiangsu Province (08KJD520018), Natural Science Foundation from Nanjing University of Information and Science Technology (20080302).

#### **REFERENCES**

- Dawson, C.N. and T.F. Dupont, 1992. Explicit/implicit conservative Galerkin domain decomposition procedures for parabolic problems. *Mathematics Comput.*, 58: 21-34.
- Dawson, C.N., Q. Du and T.F. Dupont, 1991. A finite difference domain decomposition algorithm for numerical solution of the heat equation. *Mathematics Comput.*, 57: 63-71.
- Du, Q., M. Mu and Z.N. Wu, 2001. Efficient parallel algorithms for parabolic problems. *SIAM J. Numer. Anal.*, 39: 1469-1487.
- Li, A.F., 2004. Domain decomposition difference schemes for a class of variable coefficient parabolic problems. *J. Shandong Univ.*, 5: 1-7.
- Marini, L.D. and A. Quarteroni, 1988. *Domain Decomposition Methods for Partial Differential Equations*. Oxford University Press, New York, UK., ISBN: 0198501781.
- Mathew, T.P., P.L. Polyakov, G. Russo and J. Wang, 1998. Domain decomposition operator splittings for the solution of parabolic equations. *SIAM J. Sci. Comput.*, 19: 912-932.
- Peaceman, A.W. and H.H. Rachford Jr., 1995. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Ind. Applied Mathematics*, 3: 28-41.
- Quarteroni, A. and A. Valli, 1999. *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science Publications, New York, UK., ISBN-10: 0198501781, pp: 376.
- Wan, Z.S., B.L. Zhang and G.N. Chen, 2002. Design and analysis of finite difference domain decomposition algorithms for the two-dimensional heat equation. *Proceeding of the 5th International Conference on Algorithms and Architectures for Parallel Processing*, Oct. 23-25, IEEE Computer Society, Washington, DC, USA., pp: 174-178.
- Zhang, A.L., 1998. A note on explicit-implicit finite difference domain decomposition algorithm for the heat equation. *Aeronautical Comput. Tech.*, 3: 51-54.
- Zhang, A.L. and W.D. Shen, 2002. Notes on finite difference domain decomposition algorithm for the solution of heat equation. *J. Numer. Methods Comp. Applied*, 2: 51-90.
- Zhou, Y.L. and G.W. Yuan, 1999. General difference schemes with Intrinsic parallelism for semilinear parabolic systems of divergence type. *JCM*, 14: 337-352.
- Zuo, A.L. and G.W. Yuan, 2003. A new intrinsic parallel domain decomposition algorithm and its parallel programming. *Int. Parallel Algorithms Comput. Environ. Symp. Memoir*, 25: 70-78.