ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 9, No. 1, 2014, *pp*. 022-030

Conservative domain decomposition procedure for the variable coefficient diffusion equation

Ruiyi Wang [1], Yang Zhang [2]
[1] School of Science, Hebei University of Technology, Tianjin 300401, China
[2] School of Mathematics Science and LPMC, Nankai University, Tianjin, 300071, China.
Corresponding author. zhangyang@nankai.edu.cn
(*Received February 10, 2013, accepted December 23, 2013*)

Abstract. The conservative domain decomposition procedure for solving the variable coefficient diffusion equation is presented. In this procedure, the fluxes at the interface of subdomains are properly defined, which results in the unconditional stability of the procedure. Numerical results examining the stability, the second-order accuracy of solution values as well as fluxes, and parallelism of the procedure are also presented.

Keywords: Diffusion equation; finite difference; domain decomposition; unconditional stability; variable coefficient.

1. Introduction

The diffusion equation is a basic topic, for many equations include the diffusion term, such as Sobolev equation, convection diffusion equations, and so on. The sequent finite difference methods are considered subject to such equations (see [8,9]). There is rich literature on parallel finite difference methods (see [1,2,3,4,5,6,7]). Domain decomposition is a powerful tool for devising parallel methods to solve the diffusion equation. The basic procedure of domain decomposition methods is to first decompose the domain into some subdomains, then define the interface values of subdomains by explicit schemes and the inner values of subdomains by implicit schemes. Once the interface values are available, the global problem is decoupled and parallelization is achieved. Domain decomposition methods with unconditional stability are desired in the application. However, most domain decomposition methods are conditionally stable . The major difficulty devising domain decomposition methods with unconditional stability is defining the suitable interface values of subdomains. It's also an issue to consider conservative domain decomposition methods, for some diffusion problems have the conservation property. Conservative domain decomposition procedures for the constant coefficient diffusion equation are considered in [1,3,6], in which the scheme in [1] is unconditionally stable. The purpose of this paper is to present the conservative domain decomposition procedure with unconditional stability for the following variable coefficient diffusion problem

$$U_t(x,t)-(\alpha(x,t)U_x(x,t))_x=0,\quad (x,t)\in(0,1)\times(0,T], \tag{1.1}$$

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

$$U(x,0)=U_0(x),\quad x\in[0,1], \tag{1.3}$$

where $\alpha(x,t)$ is smooth enough and $0<\alpha(x,t)\le\bar{\alpha}$. Define the flux

$$Q(x,t)=-\alpha(x,t)U_x(x,t). \tag{1.4}$$

Then (1.1) and (1.2) become as

$$U_t+Q_x=0,\quad (x,t)\in(0,1)\times(0,T], \tag{1.5}$$

and

$$Q(0,t)=Q(1,t)=0,\quad t\in(0,T]. \tag{1.6}$$

From (1.5) and (1.6), there is

$$\frac{d}{dt}\int_0^1 U\,dx=0,$$

which expresses conservation of mass. For the above problem, giving the solution U and flux Q the same second-order accuracy approximations, we consider the block-centered finite difference discretization.

The rest of this paper is organized as follows. In the next section, we present the domain decomposition procedure. In Section 3, we prove the unconditional stability. In Section 4, we examine numerically the stability, accuracy, and parallelism of the procedure. In the final section, we give a conclusion.

2. Domain Decomposition Scheme

Divide the domain $[0,1]\times[0,T]$ by a set of lines parallel to the x - and t -axes. The crossing points are

$$0 = x_{1/2} < x_{3/2} < \cdots < x_{I+1/2} = 1,$$

$$0 = t^0 < t^1 < \cdots < t^N = T.$$

Denote

$$\tau^n = t^n - t^{n-1}, \quad 1 \le n \le N,$$
$$h_i = x_{i+1/2} - x_{i-1/2}, \quad 1 \le i \le I,$$
$$x_i = \frac{x_{i-1/2} + x_{i+1/2}}{2}, \quad 1 \le i \le I,$$
$$h_{i+1/2} = x_{i+1} - x_i = \frac{h_i + h_{i+1}}{2}, \quad 1 \le i \le I-1,$$

and

$$h = \max_i h_i, \quad \hbar = \min_i h_i, \quad \Delta t = \max_n \tau^n.$$

Let f_i^n be the discrete function on $\{(x_i, t^n)\}$ and $f_{i+1/2}^n$ be the discrete function on $\{(x_{i+1/2}, t^n)\}$. Define the difference operators

$$\Delta_\tau f_i^n = \frac{f_i^n - f_i^{n-1}}{\tau^n}, \quad \Delta_+ f_i^n = \frac{f_{i+1}^n - f_i^n}{h_{i+1/2}}, \quad \Delta_- f_{i+1/2}^n = \frac{f_{i+1/2}^n - f_{i-1/2}^n}{h_i},$$

and the discrete norms

$$\| f^n \|^2 = \sum_{i=1}^{I} (f_i^n)^2 h_i, \quad ||| f^n |||^2 = \sum_{i=1}^{I-1} (f_{i+1/2}^n)^2 h_{i+1/2}.$$

Denote $U_i^n = U(x_i, t^n)$, $Q_{i+1/2}^n = Q(x_{i+1/2}, t^n)$. And let u_i^n and $q_{i+1/2}^n$ be the numerical approximations of U_i^n and $Q_{i+1/2}^n$. For simplicity, assume a decomposition of the domain $[0,1]\times[0,T]$ into two subdomains $[0,\bar{x}]\times[0,T]$ and $[\bar{x},1]\times[0,T]$, where $\bar{x} = x_{k+1/2}$ for some integer k, $0 < k < I$. It's easy to extend to the case of multiple subdomains. Next we give the domain decomposition procedure.

Approximate the equation (1.5) by

$$\Delta_\tau u_i^n + \Delta_- q_{i+1/2}^n = 0, \quad 1 \le i \le I. \tag{2.1}$$

Enforce the boundary condition (1.6) by

$$q_{1/2}^n = q_{I+1/2}^n = 0, \quad 1 \le n \le N, \tag{2.2}$$

and the initial condition by

$$u_i^0 = U_0(x_i), \quad 1 \le i \le I. \tag{2.3}$$

Sum for (2.1), by the boundary condition (2.2), there is $\sum_{i=1}^{I} u_i^n = \sum_{i=1}^{I} u_i^{n-1}$, which simulates conservation of mass. So, we call the scheme (2.1) conservative scheme.

We further define the approximating values of fluxes. For $1 \le i \le I-1, i \ne k$, approximate $Q_{i+1/2}^n$ by

$$q_{i+1/2}^n = -\alpha_{i+1/2}^n \Delta_+ u_i^n. \tag{2.4}$$

Suppose U and Q are smooth enough, it's easily get from the Taylor expansion that (2.1) and (2.4) have the second-order truncation error $O(h^2 + \Delta t)$ if $\Delta t / \hbar^2$ is any constant. To define $q_{k+1/2}^n$ with the second-order truncation error, from (1.4) and (1.5), we have

$$
\begin{aligned}
Q(\overline{x},t^n) &= [Q+\tau^n Q_t](\overline{x},t^{n-1}) + O((\tau^n)^2) \\
&= [-\alpha U_x + \tau^n(-\alpha_t U_x - \alpha U_{xt})](\overline{x},t^{n-1}) + O((\tau^n)^2) \\
&= [-(\alpha+\tau^n\alpha_t)U_x + \tau^n \alpha Q_{xx}](\overline{x},t^{n-1}) + O((\tau^n)^2) \\
&= -\alpha(\overline{x},t^n)U_x(\overline{x},t^{n-1}) + \tau^n\alpha(\overline{x},t^n)Q_{xx}(\overline{x},t^{n-1}) + O((\tau^n)^2).
\end{aligned}
$$

Then,

$$
\begin{aligned}
Q^n_{k+1/2} &= -\alpha^n_{k+1/2}\Delta_+ U^{n-1}_k \\
&\quad + \frac{\tau^n\alpha^n_{k+1/2}}{h_{k+1/2}}\left(\frac{Q^{n-1}_{k+3/2}-Q^n_{k+1/2}}{h_{k+1}} - \frac{Q^n_{k+1/2}-Q^{n-1}_{k-1/2}}{h_k}\right) \\
&\quad + O(h^2 + \frac{(\Delta t)^2}{\hbar^2} + h\Delta t + (\Delta t)^2).
\end{aligned}
$$

Thus, define $q^n_{k+1/2}$ by

$$
\begin{aligned}
q^n_{k+1/2} &= -\alpha^n_{k+1/2}\Delta_+ u^{n-1}_k \\
&\quad + \frac{\tau^n\alpha^n_{k+1/2}}{h_{k+1/2}}\left(\frac{q^{n-1}_{k+3/2}-q^n_{k+1/2}}{h_{k+1}} - \frac{q^n_{k+1/2}-q^{n-1}_{k-1/2}}{h_k}\right),
\end{aligned}
\tag{2.5}
$$

or

$$
q^n_{k+1/2} = \frac{\alpha^n_{k+1/2}}{1+\frac{2\tau^n}{h_k h_{k+1}}\alpha^n_{k+1/2}}\left[-\Delta_+ u^{n-1}_k + \frac{\tau^n}{h_{k+1/2}}\left(\frac{q^{n-1}_{k+3/2}}{h_{k+1}} + \frac{q^{n-1}_{k-1/2}}{h_k}\right)\right].
$$

Now we get the domain decomposition procedure (2.1)-(2.5). Since $q^n_{k+1/2}$ is defined by the solution values and fluxes in the former time level, we are able to identifying it with the Neumann boundary data of subdomains. Substituting (2.4) into (2.1) and using (2.2), we get two disjoint systems of equations

$$
\mathbf{A}^n_i \mathbf{u}^n_i = \mathbf{b}^n_i, \qquad i = 1,2, \quad 1 \le n \le N,
$$

in which

$$
\mathbf{u}^n_1 = (u^n_1, u^n_2, \cdots, u^n_k)^T, \ \mathbf{u}^n_2 = (u^n_{k+1}, u^n_{k+2}, \cdots, u^n_I)^T,
$$

$$
\mathbf{b}^n_1 = (u^{n-1}_1, u^{n-1}_2, \cdots, u^{n-1}_{k-1}, u^{n-1}_k - \frac{\tau^n}{h_k} q^n_{k+1/2})^T,
$$

$$
\mathbf{b}^n_2 = (u^{n-1}_{k+1} + \frac{\tau^n}{h_{k+1}} q^n_{k+1/2}, u^{n-1}_{k+2}, \cdots, u^{n-1}_{I-1}, u^{n-1}_I)^T,
$$

and we have

$$
\mathbf{A}^n_1 = \begin{pmatrix}
d^n_1 & -c^n_1 & & & \\
-e^n_2 & d^n_2 & -c^n_2 & & \\
 & \ddots & \ddots & \ddots & \\
 & & -e^n_{k-1} & d^n_{k-1} & -c^n_{k-1} \\
 & & & -e^n_k & d^n_k
\end{pmatrix},
$$

$$\mathbf{A}_2^n = \begin{pmatrix} d_{k+1}^n & -c_{k+1}^n & & & \\ -e_{k+2}^n & d_{k+2}^n & -c_{k+2}^n & & \\ & \ddots & \ddots & \ddots & \\ & & -e_{I-1}^n & d_{I-1}^n & c_{I-1}^n \\ & & & -e_I^n & d_I^n \end{pmatrix},$$

where

$$c_i^n = \frac{\tau^n \alpha_{i+1/2}^n}{h_i h_{i+1/2}}, i \neq k, I, \quad c_k^n = c_I^n = 0,$$

$$e_i^n = \frac{\tau^n \alpha_{i-1/2}^n}{h_i h_{i-1/2}}, i \neq 1, k+1, \quad e_1^n = e_{k+1}^n = 0,$$

$$d_i^n = 1 + c_i^n + e_i^n.$$

Thus, the two systems of equations can be solved in parallel.

3. Unconditional stability

For the domain decomposition procedure presented in the previous section, we have the following theorem of stability.

Theorem 3.1. *Assume* $\tau^{n+1} \leq \tau^n$ *and* $\Delta t / \hbar^2 = r$, *where* r *is any positive number. Then the procedure* (2.1)-(2.5) *is* L_2 *stable, i.e.,*

$$\max_n \| u^n \|^2 + \frac{2}{\bar{\alpha}} \sum_{n=1}^{N} \| q^n \|^2 \tau^n \leq (1 + 2r^2 \bar{\alpha}^2) \| u^0 \|^2 .$$

To prove the result, we need the following lemma.

Lemma 3.2. *Let* u_i *and* v_i *be the discrete functions defined on* $\{x_i\}, i = 0,1,\cdots,I$, *then*

$$\sum_{i=0}^{I-1} u_i (v_{i+1} - v_i) = -\sum_{i=1}^{I-1} (u_i - u_{i-1}) v_i - u_0 v_0 + u_{I-1} v_I .$$

Proof of Theorem 3.1 Multiplying (2.1) by $u_i^n h_i$ and summing up the resulting equations for $i = 1, 2, \cdots, I$ yields

$$\sum_{i=1}^{I} u_i^n \Delta_\tau u_i^n h_i + \sum_{i=1}^{I} u_i^n \Delta_- q_{i+1/2}^n h_i = 0. \tag{3.1}$$

Using lemma 3.2 and the boundary condition (2.2), we get

$$\sum_{i=1}^{I} u_i^n \Delta_- q_{i+1/2}^n h_i = -\sum_{i=1}^{I-1} q_{i+1/2}^n \Delta_+ u_i^n h_{i+1/2} . \tag{3.2}$$

Noticing that

$$\sum_{i=1}^{I} u_i^n \Delta_\tau u_i^n h_i = \frac{1}{2\tau^n} \left(\| u^n \|^2 - \| u^{n-1} \|^2 \right) + \frac{\tau^n}{2} \| \Delta_\tau u^n \|^2, \tag{3.3}$$

and substituting (3.3) and (3.2) into (3.1), we have

$$\frac{1}{2\tau^n} \left(\| u^n \|^2 - \| u^{n-1} \|^2 \right) + \frac{\tau^n}{2} \| \Delta_\tau u^n \|^2 - \sum_{i=1}^{I-1} q_{i+1/2}^n \Delta_+ u_i^n h_{i+1/2} = 0. \tag{3.4}$$

From the definition of $q_{i+1/2}^n$ and (2.1), we get

$$-\Delta_+ u_i^n = \frac{q_{i+1/2}^n}{\alpha_{i+1/2}^n}, \quad i \neq k,$$

and

$$
\begin{aligned}
-\Delta_+ u_k^n &= -\Delta_+ u_k^{n-1} - \tau^n \Delta_+(\Delta_\tau u_k^n) \\
&= \frac{q_{k+1/2}^n}{\alpha_{k+1/2}^n} + \frac{\tau^n}{h_{k+1/2}}\left(\frac{q_{k+3/2}^n - q_{k+3/2}^{n-1}}{h_{k+1}} + \frac{q_{k-1/2}^n - q_{k-1/2}^{n-1}}{h_k} \right).
\end{aligned}
$$

Hence, the last term in (3.4) is changed as

$$
\begin{aligned}
&-\sum_{i=1}^{I-1} q_{i+1/2}^n \Delta_+ u_i^n h_{i+1/2} \\
&= \sum_{i=1}^{I-1} \frac{(q_{i+1/2}^n)^2}{\alpha_{i+1/2}^n} h_{i+1/2} \\
&+ q_{k+1/2}^n \frac{\tau^n}{h_{k+1/2}}\left(\frac{q_{k+3/2}^n - q_{k+3/2}^{n-1}}{h_{k+1}} + \frac{q_{k-1/2}^n - q_{k-1/2}^{n-1}}{h_k} \right) h_{k+1/2} \\
&\ge \frac{1}{\bar{\alpha}} |||q^n|||^2 + \tau^n q_{k+1/2}^n \left(\frac{q_{k+3/2}^n - q_{k+3/2}^{n-1}}{h_{k+1}} + \frac{q_{k-1/2}^n - q_{k-1/2}^{n-1}}{h_k} \right).
\end{aligned}
$$

Substitute the above equality into (3.4) to get

$$
\begin{aligned}
&\frac{1}{2\tau^n}\left(\| u^n \|^2 - \| u^{n-1} \|^2 \right) + \frac{\tau^n}{2} \| \Delta_\tau u^n \|^2 \\
&+ \frac{1}{\bar{\alpha}} |||q^n|||^2 + \tau^n q_{k+1/2}^n \left(\frac{q_{k+3/2}^n}{h_{k+1}} + \frac{q_{k-1/2}^n}{h_k} \right) \\
&\le \tau^n q_{k+1/2}^n \left(\frac{q_{k+3/2}^{n-1}}{h_{k+1}} + \frac{q_{k-1/2}^{n-1}}{h_k} \right) \\
&\le \left(\frac{\tau^n}{2h_{k+1}} + \frac{\tau^n}{2h_k} \right)(q_{k+1/2}^n)^2 + \frac{\tau^n}{2h_{k+1}}(q_{k+3/2}^{n-1})^2 + \frac{\tau^n}{2h_k}(q_{k-1/2}^{n-1})^2. \qquad (3.5)
\end{aligned}
$$

Denote

$$
\begin{aligned}
F = &\frac{\tau^n}{2} \| \Delta_\tau u^n \|^2 + \tau^n q_{k+1/2}^n \left(\frac{q_{k+3/2}^n}{h_{k+1}} + \frac{q_{k-1/2}^n}{h_k} \right) \\
&- \left(\frac{\tau^n}{2h_{k+1}} + \frac{\tau^n}{2h_k} \right)(q_{k+1/2}^n)^2 - \frac{\tau^n}{2h_{k+1}}(q_{k+3/2}^n)^2 - \frac{\tau^n}{2h_k}(q_{k-1/2}^n)^2.
\end{aligned}
$$

Rewrite (3.5) as

$$
\begin{aligned}
&\frac{1}{2\tau^n}\left(\| u^n \|^2 - \| u^{n-1} \|^2 \right) + \frac{1}{\bar{\alpha}} |||q^n|||^2 + F + \frac{\tau^n}{2h_{k+1}}(q_{k+3/2}^n)^2 + \frac{\tau^n}{2h_k}(q_{k-1/2}^n)^2 \\
&\le \frac{\tau^n}{2h_{k+1}}(q_{k+3/2}^{n-1})^2 + \frac{\tau^n}{2h_k}(q_{k-1/2}^{n-1})^2. \qquad (3.6)
\end{aligned}
$$

By (2.1), put F in order, we have

$$
\begin{aligned}
F &= \frac{\tau^n}{2} \| \Delta_\tau u^n \|^2 - \frac{\tau^n}{2h_{k+1}}(q_{k+3/2}^n - q_{k+1/2}^n)^2 - \frac{\tau^n}{2h_k}(q_{k+1/2}^n - q_{k-1/2}^n)^2 \\
&= \frac{\tau^n}{2} \sum_{\substack{i=1 \\ i \ne k,k+1}}^{I} (\Delta_\tau u_i^n)^2 h_i \ge 0.
\end{aligned}
$$

Multiplying (3.6) by $2\tau^n$ and summing on $n = 1, 2, \cdots, m$ for any m yields

$$\| u^m \|^2 - \| u^0 \|^2 + \frac{2}{\bar{\alpha}} \sum_{n=1}^{m} \| q^n \|^2 \tau^n + \sum_{n=1}^{m} \left[\frac{(\tau^n)^2}{h_{k+1}} (q^n_{k+3/2})^2 + \frac{(\tau^n)^2}{h_k} (q^n_{k-1/2})^2 \right]$$
$$\le \sum_{n=1}^{m} \left[\frac{(\tau^n)^2}{h_{k+1}} (q^{n-1}_{k+3/2})^2 + \frac{(\tau^n)^2}{h_k} (q^{n-1}_{k-1/2})^2 \right].$$

From $\tau^{n+1} \le \tau^n$ for $n \ge 1$, there is

$$\| u^m \|^2 + \frac{2}{\bar{\alpha}} \sum_{n=1}^{m} \| q^n \|^2 \tau^n$$
$$\le \|u^0\|^2 + \frac{(\tau^1)^2}{h_{k+1}} (q^0_{k+3/2})^2 + \frac{(\tau^1)^2}{h_k} (q^0_{k-1/2})^2$$
$$\le \|u^0\|^2 + \frac{(\Delta t)^2}{\hbar} \left(\alpha^0_{k+3/2} \frac{u^0_{k+2} - u^0_{k+1}}{h_{k+3/2}} \right)^2 + \frac{(\Delta t)^2}{\hbar} \left(\alpha^0_{k-1/2} \frac{u^0_k - u^0_{k-1}}{h_{k-1/2}} \right)^2$$
$$\le \| u^0 \|^2 + \frac{2(\Delta t)^2 \bar{\alpha}^2}{\hbar^3} \left[(u^0_{k+2})^2 + (u^0_{k+1})^2 + (u^0_k)^2 + (u^0_{k-1})^2 \right]$$
$$\le (1 + 2r^2 \bar{\alpha}^2) \|u^0\|^2.$$

Thus, the theorem is proved.

4. Numerical results

In this section, we show numerical results examining the stability, accuracy and parallelism of the domain decomposition procedure. Consider the problem

$$U_t(x,t) - (\alpha(x,t) U_x(x,t))_x = f(x,t), \quad (x,t) \in (0,1) \times (0,T], \tag{4.1}$$
$$U_x(0,t) = U_x(1,t) = 0, \quad t \in (0,T], \tag{4.2}$$
$$U(x,0) = \cos \pi x, \quad x \in [0,1], \tag{4.3}$$

where $\alpha(x,t) = 1 + e^{\pi^2 t} \sin \pi x$ and $f(x,t) = \pi^2 \sin 2\pi x$. The solution is $U(x,t) = e^{-\pi^2 t} \cos(\pi x)$.

4.1 The stability of the procedure

Consider the domain decomposition procedure on uniform mesh $h = 0.001$ and $\tau^n = \Delta t$ with two subdomains $\Omega_1 = [0, 0.5] \times [0, T]$ and $\Omega_2 = [0.5, 1] \times [0, T]$. We show the numerical results $\| u^n \|$ for $r = 10$ and 100 in Table 1, and plot $\| u^n \|$ versus t in Fig. 1. It can be seen that the L^2 -norm $\| u^n \|$ doesn't occur blowing up even if r is large enough. This explains the unconditional stability of the domain decomposition procedure.

$r=10$			$r=100$		
t	n	$\| u^n \|$	t	n	$\| u^n \|$
0.1	10000	2.63553e-01	0.1	1000	2.63467e-01
0.2	20000	9.82296e-02	0.2	2000	9.81735e-02
0.3	30000	3.66111e-02	0.3	3000	3.65837e-02
0.4	40000	1.36452e-02	0.4	4000	1.36333e-02
0.5	50000	5.08565e-03	0.5	5000	5.08565e-03
0.6	60000	1.89545e-03	0.6	6000	1.89350e-03
0.7	70000	7.06446e-04	0.7	7000	7.05677e-04
0.8	80000	2.63297e-04	0.8	8000	2.62997e-04
0.9	90000	9.81325e-05	0.9	9000	9.80160e-05
1.0	100000	3.65750e-05	1.0	10000	3.65298e-05

Table 1. $\| u^n \|$ versus t

4.2 The errors of solution values and fluxes

Denote

$$\eta_h = \| u^n - U^n \|, \quad \varepsilon_h = \left(\sum_{n=1}^{N} \| \| q^n - Q^n \| \|^2 \ \tau^n \right)^{1/2}.$$

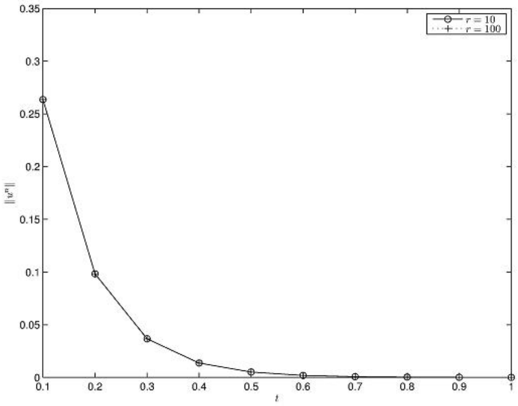

Fig. 1: $\| u^n \|$ versus t

To observe the accuracy of the domain decomposition procedure, we compute errors η_h and ε_h for subsequent mesh refinements. Let $\tau^n = \Delta t$ and $r = \Delta t / \hbar^2 = 1$. We will look at the following three scenarios:

1. Fully implicit finite difference scheme on uniform mesh h.
2. Domain decomposition procedure on uniform mesh h with two subdomains Ω_1 and Ω_2, where $\Omega_1 = [0, 0.5] \times [0, T]$ and $\Omega_2 = [0.5, 1] \times [0, T]$.
3. Domain decomposition procedure with two subdomains $\Omega_1 = [0, 0.6] \times [0, T]$ and $\Omega_2 = [0.6, 1)] \times [0, T]$. Use uniform mesh for each subdomain. The coarsest mesh on Ω_1 and Ω_2 consist of 60 blocks and 50 blocks respectively. All subsequent mesh refinements are obtained by halving the mesh. Denote $\hbar_i$ to be the mesh spacing for subdomain Ω_i, $i = 1, 2$, then $h = \max(\hbar_1, \hbar_2) = \hbar_1$ and $\Delta t = \hbar^2 = \hbar_2^2$.

The errors η_h of solutions at $T = 0.1$ for Scenarios 1-3 are compared in Table 2. Three mesh refinements were used. As can be seen in this table, the errors for each scenario are roughly of the same order of magnitude, and the errors appear to be $O(h^2)$ in each case.

The errors ε_h of fluxes at $T = 0.1$ are given in Table 3. We see virtually the same phenomena in Table 3 as in Table 2. The errors are all of roughly the same magnitude, and the converge rate is like $O(h^2)$.

h^{-1}	*Scenario* 1		*Scenario* 2		*Scenario* 3	
	η_h	η_h / h^2	η_h	η_h / h^2	η_h	η_h / h^2
100	1.247e-04	1.25	1.107e-04	1.11	7.908e-05	.791
200	3.119e-05	1.25	2.943e-05	1.18	2.039e-05	.815
400	7.798e-06	1.25	7.577e-06	1.21	5.173e-06	.828

Table 2. Convergence of solution

h^{-1}	*Scenario* 1		*Scenario* 2		*Scenario* 3	
	ε_h	ε_h / h^2	ε_h	ε_h / h^2	ε_h	ε_h / h^2
100	1.393e-04	1.39	1.477e-04	1.47	9.460e-05	9.46
200	3.483e-05	1.39	3.577e-05	1.43	2.342e-05	9.46
400	8.706e-06	1.39	8.816e-06	1.41	5.809e-06	9.30

Table 3. Convergence of flux

4.3 The parallelism of the procedure

The method has been implemented on the Nankai Star cluster system. In Table 4, we show running timings, speed-up and errors η_h for different decompositions. The notation n refers to the domain decomposition scheme with n subdomains, as well as the number of processors. When $n=1$, this represents the fully implicit finite difference scheme, which is without the domain decomposition. T_n refers to the total running time with n processors. The speed-up of n processors is defined as $S_n = T_1 / T_n$. In these runs an uniform mesh is used with $h = 1/10000$. The time step $\tau^n = \Delta t = 10^{-6}$, and $T = 0.1$. The Thomas algorithm is used to solve the tridiagonal systems on each subdomain.

The table shows that, for the test problem, the errors resulting from the domain decomposition algorithm are slightly less than that from the fully implicit finite difference scheme, and they decrease with the increase of the processors. This may be due to the fact that the rounding error in each subdomain decreases with the increase of the number of subdomains. These results also indicate that the speed-up is slightly better than linear. Thus, the scheme is of high parallelism.

n	η_h	T_n (sec.)	S_n
1	9.947e-07	126.44	
2	8.229e-07	64.90	1.95
3	7.361e-07	49.33	2.56
4	6.495e-07	36.56	3.46

Table 4. The parallelism of the scheme

5. Conclusion

We have presented a conservative domain decomposition scheme with unconditional stability for variable coefficient parabolic problems. The constructed scheme satisfies the discrete mass conservation. The unconditional stability are proved. Numerical results demonstrate the unconditional stability, the second-order accuracy to solution values as well as fluxes and high parallelism. The numerical results also show that the errors resulting from the domain decomposition algorithm are less than that from the fully implicit finite difference scheme, which is without the domain decomposition.

6. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under the grants 11071132 and 11171168.

7. References

[1] S. Zhu, *Conservative domain decomposition procedure with unconditional stability and second-order accuracy*, Appl. Math. Comput. 216 (2010), pp. 3275-3282.

[2] S. Zhu, Z.Yu, J. Zhao, *A high-order parallel finite difference algorithm*, Appl. Math. Comput. 183 (2006), pp. 365-372.

[3] G. Yuan, L. Shen, *Stability and convergence of the explicit-implicit conservative domain decomposition procedure for parabolic problems,* Comput. Math. Appl. 47 (2004), pp. 793-801.

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

[5] C. N. Dawson, Q. Du, T. F, Dupont, *A finite difference domain decomposition algorithm for numerical solution of the heat equation, Math. Compu*. 57 (1991), pp. 63-71.

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

[7] W. Hao, S. Zhu, *Parallel iterative methods for parabolic equations*, Int. J. Comput. Math. 86 (2009), pp. 431-440.

[8] Yang Zhang, *A new skew linear interpolation characteristic difference method for Sobolev equation*, J. of Information and Computing Science, 6 (2011), pp. 109-116.

[9] Yang Zhang, *AD-FDSD for convection diffusion problems*, Appl. Math. Comput. 206 (2008), pp. 257-271.