COMM. MATH. SCI.
Vol. 2, No. 3, pp. 391–402

DOMAIN DECOMPOSITION ALGORITHM FOR THE PARABOLIC EQUATION WITH VARIABLE COEFFICIENT *

ZHIQIANG SHENG †, XINGPING LIU ‡, AND XIA CUI §

Abstract. In this paper, we design a domain decomposition algorithm for the two-dimensional parabolic equation with variable coefficient by using a larger spacing at interface points and the implicit scheme at the interior points, hence get an algorithm with the relaxed stability bounds. Then we prove the stability and analyze the accuracy of the algorithm by using the idea of maximum principle. Some results of numerical experiments are also provided.

Key words. Domain decomposition, parabolic equation, implicit scheme, parallel computation, finite difference.

AMS subject classifications: 65M06, 65M12, 65M55

1. Introduction

Domain decomposition is a powerful tool for devising parallel PDE methods. There is rich literature on domain decomposition methods. [1] has developed the finite difference domain decomposition algorithm for the linear parabolic equation by using the larger spacing $H=mh$(m is a positive integer) in the explicit scheme at the interface points. The algorithm increases the stability bound of the classical explicit scheme by m^2 times. [2] has developed some techniques for the linear parabolic equation by using smaller time step $\Delta\bar{t}=\Delta t/m$ in Saul'yev schemes at the interface points. The algorithm designed with the technique can increase the stability bound of the classical explicit scheme by 2m times. The algorithm in [3] can increase the stability bound of the classical explicit scheme by $2m^2$ times for the linear parabolic equation, using the larger spacing in the x-direction implicit scheme and the y-direction implicit scheme at the interface points. The parallel efficiency is not very high, because the algorithm needs the global communication while solving the tridiagonal linear algebraic equations. [4] has proposed a parallel finite difference method for parabolic PDEs, using either a high-order explicit scheme or a multistep explicit scheme with an intermediate mesh size H lying inside (h, H_D) at the interface points. There are some other algorithms, see [5,6,7,8,9] for related discussions.

However, much of the work has been directed at the linear parabolic equation, and the proof technique is a constructive method, which is unfit for the parabolic equation with variable a coefficient. In this paper, we design a domain decomposition algorithm which can increase the stability bound of the classical explicit scheme by $4m^2$ times for the parabolic equation with a variable coefficient, and prove the stability and analyze the accuracy of the algorithm by using the idea of maximum principle.

The framework of the paper is as follows. In the next section, a domain decomposition algorithm for the parabolic equation with a variable coefficient is constructed. We use a larger spacing at interface points and the implicit scheme at the interior

*Received: December 28, 2003; accepted (in revised version): June 10, 2004. Communicated by Pingwen Zhang. This work was supported by National Natural Science Foundation (No.60373015).
†Graduate School of China Academy of Engineering Physics, P. O. Box 2101, Beijing, 100088, China.
‡Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, P. O. Box 8009, Beijing, 100088, China.
§Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, P. O. Box 8009, Beijing, 100088, China.

points, hence we get an algorithm with the relaxed stability bounds. Then the approximation property is displayed. In section 3, first some Lemmas are provided, then we prove that the algorithm is stable in the sense of L^∞ and analyze the accuracy of the algorithm by these Lemmas. In the last section, we provide some results of numerical experiments and examine numerically the stability, accuracy and parallelism of the algorithm on a certain test problem.

2. Domain decomposition algorithm

In this paper, we consider the two-dimensional parabolic equation with variable coefficient:

$$\begin{aligned} &u_t = a(x,y,t)u_{xx} + b(x,y,t)u_{yy}, && (x,y)\in\Omega,\ t\in[0,T] \\ &u(x,y,t) = 0, && (x,y)\in\partial\Omega,\ t\in[0,T] \\ &u(x,y,0) = u_0(x,y), && (x,y)\in\Omega, \end{aligned} \tag{2.1}$$

where $\Omega=(0,1)\times(0,1)$; u_0 is a known function. a is a continuous function and b is a continuous function in Ω, $0<\delta_1\le a=a(x,y,t)\le K_1<\infty$, $0<\delta_2\le b=b(x,y,t)\le K_2<\infty$.

Divide interval [0,T] and [0,1], [0,1] into N and J, J equal small intervals respectively. Denote $\tau=T/N$, $t_n=n\tau$, $h=1/J$, $x_i=ih$, $y_j=jh$, $r=\tau/h^2$. For a function $\phi(x,y,t)$ defined at mesh points (x_i,y_j,t_n), let $\phi_{ij}^n=\phi(x_i,y_j,t_n)$.

It's well known that there are several discrete schemes for the parabolic equation the explicit scheme:

$$u_{i,j}^{n+1} = a_{ij}^n r u_{i+1,j}^n + a_{ij}^n r u_{i-1,j}^n + b_{ij}^n r u_{i,j+1}^n + b_{ij}^n r u_{i,j-1}^n + (1-2a_{ij}^n r - 2b_{ij}^n r)u_{ij}^n, \tag{2.2}$$

the implicit scheme:

$$\begin{aligned} &-a_{ij}^{n+1} r u_{i+1,j}^{n+1} - a_{ij}^{n+1} r u_{i-1,j}^{n+1} - b_{ij}^{n+1} r u_{i,j+1}^{n+1} - b_{ij}^{n+1} r u_{i,j-1}^{n+1} \\ &\qquad + (1+2a_{ij}^{n+1} r + 2b_{ij}^{n+1} r)u_{ij}^{n+1} = u_{ij}^n, \end{aligned} \tag{2.3}$$

the x-direction implicit scheme:

$$\begin{aligned} &-a_{ij}^{n+1} r u_{i+1,j}^{n+1} + (1+2a_{ij}^{n+1} r)u_{ij}^{n+1} - a_{ij}^{n+1} r u_{i-1,j}^{n+1} \\ &\qquad = (1-2b_{ij}^n r)u_{ij}^n + b_{ij}^n r u_{i,j+1}^n + b_{ij}^n r u_{i,j-1}^n, \end{aligned} \tag{2.4}$$

and the y-direction implicit scheme:

$$\begin{aligned} &-b_{ij}^{n+1} r u_{i,j+1}^{n+1} + (1+2b_{ij}^{n+1} r)u_{ij}^{n+1} - b_{ij}^{n+1} r u_{i,j-1}^{n+1} \\ &\qquad = (1-2a_{ij}^n r)u_{ij}^n + a_{ij}^n r u_{i+1,j}^n + a_{ij}^n r u_{i-1,j}^n. \end{aligned} \tag{2.5}$$

Their truncation errors are $O(\tau+h^2)$.
In another paper we have gotten a new difference scheme for the linear parabolic equation, e.g.

$$\begin{aligned} u_{ij}^n = {} & [r^2 u_{i+2,j}^{n-1} + r(1-r)u_{i+1,j}^{n-1} + r(1-r)u_{i-1,j}^{n-1} + r^2 u_{i-2,j}^{n-1} + r^2 u_{i,j+2}^{n-1} \\ & + r(1-r)u_{i,j+1}^{n-1} + r(1-r)u_{i,j-1}^{n-1} + r^2 u_{i,j-2}^{n-1} + (1-r)u_{i,j}^{n-1}]/(1+3r). \end{aligned}$$

For the same reason, we can get a new difference scheme for the parabolic equation with variable coefficient as follows:

$$
\begin{aligned}
u_{ij}^{n+1}=&\frac{(a_{ij}^n r)^2u_{i-2,j}^n+a_{ij}^n r(1-a_{ij}^n r)u_{i-1,j}^n+a_{ij}^n r(1-a_{ij}^n r)u_{i+1,j}^n+(a_{ij}^n r)^2u_{i+2,j}^n}{1+3a_{ij}^n r}\\
&+\frac{(b_{ij}^n r)^2u_{i,j-2}^n+b_{ij}^n r(1-b_{ij}^n r)u_{i,j-1}^n+b_{ij}^n r(1-b_{ij}^n r)u_{i,j+1}^n+(b_{ij}^n r)^2u_{i,j+2}^n}{1+3b_{ij}^n r}\\
&+[\frac{1+a_{ij}^n r}{1+3a_{ij}^n r}+\frac{1+b_{ij}^n r}{1+3b_{ij}^n r}-1]u_{ij}^n. \qquad (2.6)
\end{aligned}
$$

By Taylor's expansion at (i,j,n) for the solution $u_{i,j}^n$ of $u_t=au_{xx}+bu_{yy}$, the truncation error for (2.6) is

$$O(\tau+h^2),$$

which is the same as the accuracy of the fully implicit scheme.

Next we design a domain decomposition algorithm.

Define the following operators:

$$
\begin{aligned}
L_1u_{ij}^{n+1}=&u_{ij}^{n+1}-[\frac{1+a_{ij}^n R}{1+3a_{ij}^n R}+\frac{1+b_{ij}^n R}{1+3b_{ij}^n R}-1]u_{ij}^n\\
&-\frac{(a_{ij}^n R)^2u_{i-2m,j}^n+a_{ij}^n R(1-a_{ij}^n R)u_{i-m,j}^n+a_{ij}^n R(1-a_{ij}^n R)u_{i+m,j}^n+(a_{ij}^n R)^2u_{i+2m,j}^n}{1+3a_{ij}^n R}\\
&-\frac{(b_{ij}^n R)^2u_{i,j-2m}^n+b_{ij}^n R(1-b_{ij}^n R)u_{i,j-m}^n+b_{ij}^n R(1-b_{ij}^n R)u_{i,j+m}^n+(b_{ij}^n R)^2u_{i,j+2m}^n}{1+3b_{ij}^n R},
\end{aligned}
\qquad (2.7)
$$

$$
\begin{aligned}
L_4u_{ij}^{n+\frac{1}{2}}=&u_{ij}^{n+\frac{1}{2}}-[\frac{1+a_{ij}^n R_1}{1+3a_{ij}^n R_1}+\frac{1+b_{ij}^n R_1}{1+3b_{ij}^n R_1}-1]u_{ij}^n-\\
&\frac{(a_{ij}^n R_1)^2u_{i-2m,j}^n+a_{ij}^n R_1(1-a_{ij}^n R_1)u_{i-m,j}^n+a_{ij}^n R_1(1-a_{ij}^n R_1)u_{i+m,j}^n+(a_{ij}^n R_1)^2u_{i+2m,j}^n}{1+3a_{ij}^n R_1}\\
&-\frac{(b_{ij}^n R_1)^2u_{i,j-2m}^n+b_{ij}^n R_1(1-b_{ij}^n R_1)u_{i,j-m}^n+b_{ij}^n R_1(1-b_{ij}^n R_1)u_{i,j+m}^n+(b_{ij}^n R_1)^2u_{i,j+2m}^n}{1+3b_{ij}^n R_1},
\end{aligned}
\qquad (2.8)
$$

$$
\begin{aligned}
L_2u_{ij}^{n+1}=&-a_{ij}^{n+1}r_1u_{i+1,j}^{n+1}+(1+2a_{ij}^{n+1}r_1)u_{ij}^{n+1}-a_{ij}^{n+1}r_1u_{i-1,j}^{n+1}-(1-2b_{ij}^{n+\frac{1}{2}}R_1)u_{ij}^{n+\frac{1}{2}}\\
&-b_{ij}^{n+\frac{1}{2}}R_1u_{i,j+m}^{n+\frac{1}{2}}-b_{ij}^{n+\frac{1}{2}}R_1u_{i,j-m}^{n+\frac{1}{2}}, \qquad (2.9)\\
L_3u_{ij}^{n+1}=&-b_{ij}^{n+1}r_1u_{i,j+1}^{n+1}+(1+2b_{ij}^{n+1}r_1)u_{ij}^{n+1}-b_{ij}^{n+1}r_1u_{i,j-1}^{n+1}-(1-2a_{ij}^{n+\frac{1}{2}}R_1)u_{ij}^{n+\frac{1}{2}}\\
&-a_{ij}^{n+\frac{1}{2}}R_1u_{i+m,j}^{n+\frac{1}{2}}-a_{ij}^{n+\frac{1}{2}}R_1u_{i-m,j}^{n+\frac{1}{2}}, \qquad (2.10)\\
L_5u_{ij}^{n+\frac{1}{2}}=&-a_{ij}^{n+\frac{1}{2}}r_1u_{i+1,j}^{n+\frac{1}{2}}+(1+2a_{ij}^{n+\frac{1}{2}}r_1)u_{ij}^{n+\frac{1}{2}}-a_{ij}^{n+\frac{1}{2}}r_1u_{i-1,j}^{n+\frac{1}{2}}-(1-2b_{ij}^nR_1)u_{ij}^n\\
&-b_{ij}^nR_1u_{i,j+m}^n-b_{ij}^nR_1u_{i,j-m}^n, \qquad (2.11)\\
L_6u_{ij}^{n+\frac{1}{2}}=&-b_{ij}^{n+\frac{1}{2}}r_1u_{i,j+1}^{n+\frac{1}{2}}+(1+2b_{ij}^{n+\frac{1}{2}}r_1)u_{ij}^{n+\frac{1}{2}}-b_{ij}^{n+\frac{1}{2}}r_1u_{i,j-1}^{n+\frac{1}{2}}-(1-2a_{ij}^nR_1)u_{ij}^n\\
&-a_{ij}^nR_1u_{i+m,j}^n-a_{ij}^nR_1u_{i-m,j}^n, \qquad (2.12)\\
Su_{ij}^{n+1}=&-a_{ij}^{n+1}ru_{i+1,j}^{n+1}-a_{ij}^{n+1}ru_{i-1,j}^{n+1}-b_{ij}^{n+1}ru_{i,j+1}^{n+1}-b_{ij}^{n+1}ru_{i,j-1}^{n+1}+\\
&(1+2a_{ij}^{n+1}r+2b_{ij}^{n+1}r)u_{ij}^{n+1}-u_{ij}^n, \qquad (2.13)
\end{aligned}
$$

where $r_1 = r/2,\ R = \tau/H^2,\ R_1 = R/2 = \tau/(2H^2),\ H = mh$.
One has the truncation error

$$\begin{aligned} L_1 u_{ij}^{n+1} &= O(\tau + H^2), \\ L_k u_{ij}^{n+1} &= O(\tau_1 + H^2)\,(k = 2,3), \\ L_k u_{ij}^{n+\frac{1}{2}} &= O(\tau_1 + H^2)\,(k = 4,5,6), \\ S u_{ij}^{n+1} &= O(\tau + h^2), \end{aligned}$$

where $\tau_1 = \tau/2$.
The domain decomposition algorithm is as follows:
Algorithm:

$$\begin{aligned} &U_{ij}^{n+1} = u_{ij}^{n+1}, \textit{at boundary points}, \\ &L_1 U_{kj}^{n+1} = 0,\ \textit{at interface points}\ (x_k, y_j, t^{n+1})(2m \le j \le J - 2m), \\ &L_1 U_{il}^{n+1} = 0,\ \textit{at interface points}\ (x_i, y_l, t^{n+1})(2m \le i \le J - 2m), \\ &L_4 U_{ij}^{n+\frac{1}{2}} = 0,\ i \in P_1\, \textit{and}\ j \in P_2, \textit{or}\ j \in P_1\, \textit{and}\ i \in P_3, \\ &L_5 U_{ij}^{n+\frac{1}{2}} = 0,\ 0 < i < 2m\ \textit{or}\ J - 2m < i < J,\ \textit{and}\ j \in P_2, \\ &L_6 U_{ij}^{n+\frac{1}{2}} = 0,\ 0 < j < 2m\ \textit{or}\ J - 2m < j < J,\ \textit{and}\ i \in P_3, \\ &L_2 U_{il}^{n+1} = 0,\ \textit{at interface points}\ (x_i, y_l, t^{n+1})(0 < i < 2m\ \textit{or}\ J - 2m < i < J), \\ &L_3 U_{kj}^{n+1} = 0,\ \textit{at interface points}\ (x_k, y_j, t^{n+1})(0 < j < 2m\ \textit{or}\ J - 2m < j < J), \\ &SU_{ij}^{n+1} = 0,\ \textit{at interior points}\ (x_i, y_j, t^{n+1})(i \ne k, j \ne l), \end{aligned} \tag{2.14}$$

where $P_1 = \{2m, J - 2m\},\ P_2 = \{l - m, l, l + m\},\ P_3 = \{k - m, k, k + m\}$.

Figure 1 and 2 illustrate the various regions that used different operators ($J = 14, m = 2, k = l = 7$).

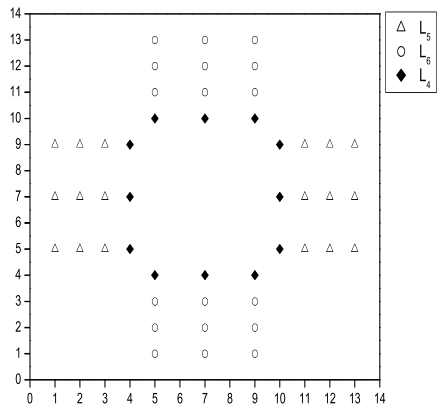

FIG. 2.1. *(n+1/2)*

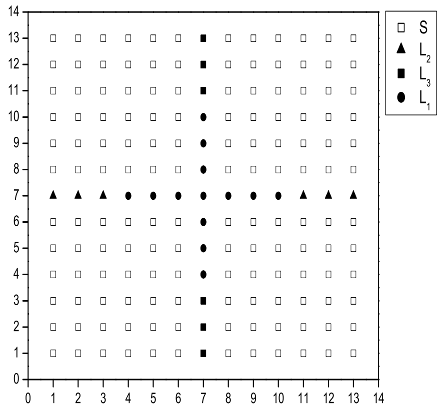

FIG. 2.2. *(n+1)*

Our algorithm and the ones in [1,2,3] all use the classical implicit scheme at the interior points, the difference lying in the scheme used at the interface points. The stability bounds of algorithms relax m^2 in [1], $2m$ in [2] and $2m^2$ in [3] respectively. Our algorithm can relax the stability bound to a further extent ($4m^2$) by combining the larger spacing with the smaller time step.

We can show the algorithm has a feasible accuracy.

THEOREM 2.1. *For the numerical solution U_{ij}^{n+1} of the algorithm and the real solution u_{ij}^{n+1} of (1), if the following conditions are satisfied, e.g.*

$$1-aR\geq 0 \quad and \quad 1-bR\geq 0,$$

then Algorithm is stable and

$$||e^{n+1}||_\infty \leq ||e^0||_\infty + C(\tau+H^2),$$

where $e_{ij}^{n+1}=U_{ij}^{n+1}-u_{ij}^{n+1}$, C is a positive constant independent of τ and H.

It is obvious that our algorithm can increase the stability bound of the classical explicit scheme by $4m^2$ times. The accuracy of the classical implicit scheme is $O(\tau+h^2)$, but τ is generally greater than h^2 in the practical computation, hence we can use an appropriate H instead of h without reducing the accuracy. The accuracy of the algorithm in [1] is $\max_{\mathrm{i,j,n}} |U_{ij}^n - u_{ij}^n| \leq C(\tau+h^2+H^3)$, and the stability condition is $\tau/H^2\leq 1/2$. If we take $\tau = H^2/2$, then the accuracy of algorithm in [1] is $\max_{\mathrm{i,j,n}} |U_{ij}^n - u_{ij}^n| \leq C'H^2$, and the accuracy of our algorithm is $||e^{n+1}||_\infty \leq ||e^0||_\infty + C''H^2$. So the accuracy of our algorithm is feasible. At the same condition, the algorithms in [2, 3] have similar accuracy, e.g. $O(H^2)$.

3. Proof of Theorem

In order to show the theorem, we first provide some lemmas.

LEMMA 3.1. *If v_{ij} satisfies the following relation*

$$\begin{aligned}-a_{ij}^{n+1}rv_{i+1,j}^{n+1}-a_{ij}^{n+1}rv_{i-1,j}^{n+1}-b_{ij}^{n+1}rv_{i,j+1}^{n+1}-b_{ij}^{n+1}rv_{i,j-1}^{n+1}+(1+2a_{ij}^{n+1}r+2b_{ij}^{n+1}r)v_{ij}^{n+1}\\ =v_{ij}^n \ (i=k_1+1,\cdots,k_2-1; j=l_1+1,\cdots,l_2-1),\end{aligned} \tag{3.1}$$

then

$$\max_{\substack{k_1\leq i\leq k_2\\ l_1\leq j\leq l_2}} |v_{ij}^{n+1}| \leq \max\{\max_{\substack{k_1+1\leq i\leq k_2-1\\ l_1+1\leq j\leq l_2-1}} |v_{ij}^n|, \max_{\substack{i\in P_1\\ l_1\leq j\leq l_2}} |v_{ij}^{n+1}|, \max_{\substack{j\in P_2\\ k_1\leq i\leq k_2}} |v_{ij}^{n+1}|\},$$

where $P_1=\{k_1,k_2\}$, $P_2=\{l_1,l_2\}$.

Proof: Let $M=\max_{\substack{k_1\leq i\leq k_2\\ l_1\leq j\leq l_2}} |v_{ij}^{n+1}|$, $P=\{(i,j)||v_{ij}^{n+1}|=M, \quad i\in\{k_1,\cdots,k_2\}; j\in\{l_1,\cdots,l_2\}\}$, and $(i_0,j_0)\in P$, if $i_0\in P_1$ or $j_0\in P_2$, the conclusion is obvious.
Next suppose $i_0\notin P_1$ and $j_0\notin P_2$.
Because $|v_{i_0,j_0}^{n+1}|=M$, first suppose $v_{i_0,j_0}^{n+1}=M$ for convenience, there are

$$v_{i_0-1,j_0}^{n+1}\leq M, \quad v_{i_0+1,j_0}^{n+1}\leq M, \quad v_{i_0,j_0+1}^{n+1}\leq M, \quad v_{i_0,j_0-1}^{n+1}\leq M,$$

from (3.1), we know that

$$\begin{aligned}v_{i_0,j_0}^n &= -a_{i_0,j_0}^{n+1}rv_{i_0+1,j_0}^{n+1}-a_{i_0,j_0}^{n+1}rv_{i_0-1,j_0}^{n+1}-b_{i_0,j_0}^{n+1}rv_{i_0,j_0+1}^{n+1}-b_{i_0,j_0}^{n+1}rv_{i_0,j_0-1}^{n+1}\\ &\quad +(1+2a_{i_0,j_0}^{n+1}r+2b_{i_0,j_0}^{n+1}r)v_{i_0,j_0}^{n+1}\\ &\geq -a_{i_0,j_0}^{n+1}rM-a_{i_0,j_0}^{n+1}rM-b_{i_0,j_0}^{n+1}rM-b_{i_0,j_0}^{n+1}rM+(1+2a_{i_0,j_0}^{n+1}r+2b_{i_0,j_0}^{n+1}r)M\\ &=M.\end{aligned}$$

Then suppose $v^{n+1}_{i_0,j_0}=-M$, there are

$$v^{n+1}_{i_0-1,j_0}\geq -M,\ \ v^{n+1}_{i_0+1,j_0}\geq -M,\ \ v^{n+1}_{i_0,j_0+1}\geq -M,\ \ v^{n+1}_{i_0,j_0-1}\geq -M,$$

from (3.1), we know that

$$\begin{aligned}-v^n_{i_0,j_0}&=a^{n+1}_{i_0,j_0}rv^{n+1}_{i_0+1,j_0}+a^{n+1}_{i_0,j_0}rv^{n+1}_{i_0-1,j_0}+b^{n+1}_{i_0,j_0}rv^{n+1}_{i_0,j_0+1}+b^{n+1}_{i_0,j_0}rv^{n+1}_{i_0,j_0-1}\\&\quad-(1+2a^{n+1}_{i_0,j_0}r+2b^{n+1}_{i_0,j_0}r)v^{n+1}_{i_0,j_0}\\&\geq -a^{n+1}_{i_0,j_0}rM-a^{n+1}_{i_0,j_0}rM-b^{n+1}_{i_0,j_0}rM-b^{n+1}_{i_0,j_0}rM+(1+2a^{n+1}_{i_0,j_0}r+2b^{n+1}_{i_0,j_0}r)M\\&=M,\end{aligned}$$

hence

$$M\leq -v^n_{i_0,j_0}\leq |v^n_{i_0,j_0}|;$$

From above, we know that

$$M\leq \max\{\max_{\substack{k_1+1\leq i\leq k_2-1\\ l_1+1\leq j\leq l_2-1}}|v^n_{ij}|,\ \max_{\substack{i\in P_1\\ l_1\leq j\leq l_2}}|v^{n+1}_{ij}|,\ \max_{\substack{j\in P_2\\ k_1\leq i\leq k_2}}|v^{n+1}_{ij}|\},$$

and the proof is finished.

Lemma 3.2. *If v_{ij} satisfies the following relation*

$$\begin{aligned}&-a^{n+1}_{il}rv^{n+1}_{i+1,l}+(1+2a^{n+1}_{il}r)v^{n+1}_{il}-a^{n+1}_{il}rv^{n+1}_{i-1,l}\\&\qquad=(1-2b^n_{il}r)v^n_{il}+b^n_{il}rv^n_{i,l+1}+b^n_{il}rv^n_{i,l-1}\ (i=k_1+1,\cdots,k_2-1),\end{aligned}\tag{3.2}$$

and

$$1-2b^n_{il}r\geq 0,$$

then

$$\max_{k_1\leq i\leq k_2}|v^{n+1}_{il}|\leq \max\{\max_{\substack{k_1+1\leq i\leq k_2-1\\ j=l-1,l,l+1}}|v^n_{ij}|,\ \ |v^{n+1}_{k_1,l}|,\ \ |v^{n+1}_{k_2,l}|\}.$$

Proof: Let $M=\max_{k_1\leq i\leq k_2}|v^{n+1}_{il}|$, $P=\{i\mid |v^{n+1}_{il}|=M,\ i\in\{k_1,\cdots,k_2\}\}$, and $i_0\in P$, if $i_0=k_1$ or $i_0=k_2$, the conclusion is obvious.
Next suppose $i_0\neq k_1$ and $i_0\neq k_2$.
Because $|v^{n+1}_{i_0,l}|=M$, first suppose $v^{n+1}_{i_0,l}=M$ for convenience, there are

$$v^{n+1}_{i_0-1,l}\leq M,\ \ v^{n+1}_{i_0+1,l}\leq M,$$

from (3.2), we know that

$$\begin{aligned}&-a^{n+1}_{i_0,l}rv^{n+1}_{i_0+1,l}+(1+2a^{n+1}_{i_0,l}r)v^{n+1}_{i_0,l}-a^{n+1}_{i_0,l}rv^{n+1}_{i_0-1,l}\\&\geq -a^{n+1}_{i_0,l}rM+(1+2a^{n+1}_{i_0,l}r)M-a^{n+1}_{i_0,l}rM=M.\end{aligned}$$

Notice that

$$1-2b^n_{i_0,l}r\geq 0,$$

one deduces

$$(1-2b^n_{i_0,l}r)v^n_{i_0,l}+b^n_{i_0,l}rv^n_{i_0,l+1}+b^n_{i_0,l}rv^n_{i_0,l-1}\le \max_{j=l-1,l,l+1}|v^n_{i_0,j}|,$$

hence

$$M\le \max_{\substack{k_1+1\le i\le k_2-1\\ j=l-1,\,l,\,l+1}}|v^n_{ij}|.$$

Then suppose $v^{n+1}_{i_0,l}=-M$, there are

$$v^{n+1}_{i_0-1,l}\ge -M,\quad v^{n+1}_{i_0+1,l}\ge -M,$$

from (3.2), we know that

$$\begin{aligned}&a^{n+1}_{i_0,l}rv^{n+1}_{i_0+1,l}-(1+2a^{n+1}_{i_0,l}r)v^{n+1}_{i_0,l}+a^{n+1}_{i_0,l}rv^{n+1}_{i_0-1,l}\\ \ge&-a^{n+1}_{i_0,l}rM+(1+2a^{n+1}_{i_0,l}r)M-a^{n+1}_{i_0,l}rM=M.\end{aligned}$$

Notice that

$$1-2b^n_{i_0,l}r\ge 0,$$

one deduces

$$|(1-2b^n_{i_0,l}r)v^n_{i_0,l}+b^n_{i_0,l}rv^n_{i_0,l+1}+b^n_{i_0,l}rv^n_{i_0,l-1}|\le \max_{j=l-1,l,l+1}|v^n_{i_0,j}|,$$

hence

$$M\le \max_{\substack{k_1+1\le i\le k_2-1\\ j=l-1,\,l,\,l+1}}|v^n_{ij}|.$$

So we get that

$$M\le \max\{\max_{\substack{k_1+1\le i\le k_2-1\\ j=l-1,\,l,\,l+1}}|v^n_{ij}|,\quad |v^{n+1}_{k_1,l}|,\quad |v^{n+1}_{k_2,l}|\},$$

and the proof is finished.

LEMMA 3.3. *If v_{ij} satisfies the following relation*

$$\begin{aligned}&-b^{n+1}_{kj}rv^{n+1}_{k,j+1}+(1+2b^{n+1}_{kj}r)v^{n+1}_{kj}-b^{n+1}_{kj}rv^{n+1}_{k,j-1}\\ &=(1-2a^n_{kj}r)v^n_{kj}+a^n_{kj}rv^n_{k+1,j}+a^n_{kj}rv^n_{k-1,j}\ (j=l_1+1,\cdots,l_2-1),\end{aligned}\tag{3.3}$$

and

$$1-2a^n_{kj}r\ge 0,$$

then

$$\max_{l_1\le j\le l_2}|v^{n+1}_{kj}|\le \max\{\max_{\substack{l_1+1\le j\le l_2-1\\ i=k-1,\,k,\,k+1}}|v^n_{ij}|,\quad |v^{n+1}_{k,l_1}|,\quad |v^{n+1}_{k,l_2}|\}.$$

The proof is the same as the proof of Lemma3.2.

LEMMA 3.4. *If v_{ij} satisfies the following relation*

$$v_{ij}^{n+1}=\frac{(a_{ij}^n r)^2 v_{i-2,j}^n + a_{ij}^n r(1-a_{ij}^n r)v_{i-1,j}^n + a_{ij}^n r(1-a_{ij}^n r)v_{i+1,j}^n + (a_{ij}^n r)^2 v_{i+2,j}^n}{1+3a_{ij}^n r}$$
$$+\frac{(b_{ij}^n r)^2 v_{i,j-2}^n + b_{ij}^n r(1-b_{ij}^n r)v_{i,j-1}^n + b_{ij}^n r(1-b_{ij}^n r)v_{i,j+1}^n + (b_{ij}^n r)^2 v_{i,j+2}^n}{1+3b_{ij}^n r}$$
$$+[\frac{1+a_{ij}^n r}{1+3a_{ij}^n r}+\frac{1+b_{ij}^n r}{1+3b_{ij}^n r}-1]v_{ij}^n, \tag{3.4}$$

and

$$1-a_{ij}^n r\geq 0 \quad and \quad 1-b_{ij}^n r\geq 0,$$

then

$$\max_{\substack{2m\leq i\leq J-2m\\ 2m\leq j\leq J-2m}}|v_{ij}^{n+1}|\leq \max_{\substack{0\leq i\leq J\\ 0\leq j\leq J}}|v_{ij}^n|.$$

Proof: Notice that $\frac{1+x}{1+3x}$ is a monotonic function for $x\geq 0$, if $1-a_{ij}^n r\geq 0$ and $1-b_{ij}^n r\geq 0$, then

$$\frac{1+a_{ij}^n r}{1+3a_{ij}^n r}+\frac{1+b_{ij}^n r}{1+3b_{ij}^n r}-1\geq 0,$$

Notice (3.4)

$$|v_{ij}^{n+1}|\leq [\frac{2a_{ij}^n r}{1+3a_{ij}^n r}+\frac{2b_{ij}^n r}{1+3b_{ij}^n r}+\frac{1+a_{ij}^n r}{1+3a_{ij}^n r}+\frac{1+b_{ij}^n r}{1+3b_{ij}^n r}-1]\max_{\substack{0\leq i\leq J\\ 0\leq j\leq J}}|v_{ij}^n|$$
$$=\max_{\substack{0\leq i\leq J\\ 0\leq j\leq J}}|v_{ij}^n|,$$

so one deduces

$$\max_{\substack{2m\leq i\leq J-2m\\ 2m\leq j\leq J-2m}}|v_{ij}^{n+1}|\leq \max_{\substack{0\leq i\leq J\\ 0\leq j\leq J}}|v_{ij}^n|$$

and the proof is finished.

Next we show the Theorem.

It's obvious that U_{ij} satisfies the relation in Lemma 3.1, Lemma 3.2, Lemma 3.3, Lemma 3.4, So, if there are

$$1-aR\geq 0 \quad and \quad 1-bR\geq 0,$$

notice that $U_{ij}^{n+1}=U_{ij}^n=0$ for $i=0$ *or* $i=J$ *or* $j=0$ *or* $j=J$, then we have

$$\max_{\substack{0\leq i\leq J\\ 0\leq j\leq J}}|U_{ij}^{n+1}|$$
$$=\max\{\max_{\substack{0\leq i\leq k\\ 0\leq j\leq l}}|U_{ij}^{n+1}|,\max_{\substack{0\leq i\leq k\\ l\leq j\leq J}}|U_{ij}^{n+1}|,\max_{\substack{k\leq i\leq J\\ 0\leq j\leq l}}|U_{ij}^{n+1}|,\max_{\substack{k\leq i\leq J\\ l\leq j\leq J}}|U_{ij}^{n+1}|\}$$
$$\leq\max\{\max_{\substack{1\leq i\leq J-1,\, i\neq k\\ 1\leq j\leq J-1,\, j\neq l}}|U_{ij}^n|,\max_{0\leq j\leq J}|U_{kj}^{n+1}|,\max_{0\leq i\leq J}|U_{il}^{n+1}|\}$$
$$\leq\max_{\substack{0\leq i\leq J_1\\ 0\leq j\leq J_2}}|U_{ij}^n|,$$

e.g.

$$||U^{n+1}||_\infty \le ||U^n||_\infty. \tag{3.5}$$

Then we deduce that

$$||U^n||_\infty \le ||U^{n-1}||_\infty \le \cdots \le ||U^0||_\infty, \tag{3.6}$$

hence the algorithm is stable.

It's obvious that e_{ij} satisfies the relation:

$$\begin{aligned}
e_{ij}^{n+1} &= 0, \qquad \textit{at boundary points},\\
L_1 e_{kj}^{n+1} &= \tau R_{kj}^{n+1}, \ \textit{at interface points } (x_k,y_j,t^{n+1})(2m \le j \le J-2m),\\
L_1 e_{il}^{n+1} &= \tau R_{il}^{n+1}, \ \textit{at interface points } (x_i,y_l,t^{n+1})(2m \le i \le J-2m),\\
L_4 e_{ij}^{n+\frac{1}{2}} &= \tau_1 R_{ij}^{n+\frac{1}{2}}, i \in P_1 \textit{ and } j \in P_2, \textit{or } j \in P_1 \textit{ and } i \in P_3,\\
L_5 e_{ij}^{n+\frac{1}{2}} &= \tau_1 R_{ij}^{n+\frac{1}{2}}, 0 < i < 2m \textit{ or } J-2m < i < J, \textit{ and } j \in P_2,\\
L_6 e_{ij}^{n+\frac{1}{2}} &= \tau_1 R_{ij}^{n+\frac{1}{2}}, 0 < j < 2m \textit{ or } J-2m < j < J, \textit{ and } i \in P_3,\\
L_2 e_{il}^{n+1} &= \tau_1 R_{il}^{n+1}, \ \textit{at interface points } (x_i,y_l,t^{n+1})(0 < i < 2m \textit{ or } J-2m < i < J),\\
L_3 e_{kj}^{n+1} &= \tau_1 R_{kj}^{n+1}, \ \textit{at interface points } (x_k,y_j,t^{n+1})(0 < j < 2m \textit{ or } J-2m < j < J),\\
S e_{ij}^{n+1} &= \tau R_{ij}^{n+1}, \ \textit{at interior points } (x_i,y_j,t^{n+1})(i \ne k, j \ne l),
\end{aligned}$$

where $R_{ij}^{n+1} \le C_{ij}(\tau + h^2)(i \ne k, j \ne l)$, $R_{il}^{n+1} \le C_{il}(\tau + H^2)$, $R_{kj}^{n+1} \le C_{kj}(\tau + H^2)$, $R_{ij}^{n+\frac{1}{2}} \le C'_{ij}(\tau_1 + H^2)$, and $|C_{ij}| \le C$, $|C'_{ij}| \le C$.
Let $P_4 = \{k-2m, k-m, k, k+m, k+2m\}$, $P_5 = \{l-2m, l-m, l, l+m, l+2m\}$ and notice Lemma 3.1, Lemma 3.2, Lemma 3.3, Lemma 3.4, if there are

$$1 - aR \ge 0 \quad \textit{and} \quad 1 - bR \ge 0,$$

then

$$\begin{aligned}
\max_{2m \le j \le J-2m} |e_{kj}^{n+1}| &\le \max_{\substack{i \in P_4\\ 0 \le j \le J}} |e_{ij}^n| + \max_{2m \le j \le J-2m} \tau |R_{kj}^{n+1}|\\
&\le \max_{\substack{i \in P_4\\ 0 \le j \le J}} |e_{ij}^n| + C\tau(\tau + H^2),
\end{aligned}$$

$$\begin{aligned}
\max_{0 \le j \le 2m} |e_{kj}^{n+1}| &\le \max_{\substack{i = k-m, k, k+m\\ 0 \le j \le 2m}} |e_{ij}^{n+\frac{1}{2}}| + \max_{0 \le j \le 2m} \tau_1 |R_{kj}^{n+\frac{1}{2}}|\\
&\le \max_{\substack{i \in P_4\\ 0 \le j \le 2m}} |e_{ij}^n| + \max_{\substack{i = k-m, k, k+m\\ 0 \le j \le 2m}} \tau_1 |R_{ij}^n| + \max_{0 \le j \le 2m} \tau_1 |R_{kj}^{n+\frac{1}{2}}|\\
&\le \max_{\substack{i \in P_4\\ 0 \le j \le 2m}} |e_{ij}^n| + C\tau(\tau + H^2),\\
\max_{J-2m \le j \le J} |e_{kj}^{n+1}| &\le \max_{\substack{i \in P_4\\ J-2m \le j \le J}} |e_{ij}^n| + C\tau(\tau + H^2),
\end{aligned}$$

which means

$$\max_{0 \le j \le J} |e_{kj}^{n+1}| \le \max_{\substack{i \in P_4\\ 0 \le j \le J}} |e_{ij}^n| + C\tau(\tau + H^2).$$

With the same reason

$$\max_{0\le i\le J}|e_{il}^{n+1}|\le \max_{\substack{j\in P_5\\0\le i\le J}}|e_{ij}^{n}|+C\tau(\tau+H^2);$$

then we deduce that

$$\begin{aligned}
&\max_{\substack{0\le i\le J\\0\le j\le J}}|e_{ij}^{n+1}|\\
&\qquad=\max\{\max_{\substack{0\le i\le k\\0\le j\le l}}|e_{ij}^{n+1}|,\ \max_{\substack{0\le i\le k\\l\le j\le J}}|e_{ij}^{n+1}|,\ \max_{\substack{k\le i\le J\\0\le j\le l}}|e_{ij}^{n+1}|,\ \max_{\substack{k\le i\le J\\l\le j\le J}}|e_{ij}^{n+1}|\}\\
&\qquad\le\max\{\max_{\substack{0\le i\le J,\,i\ne k\\0\le j\le J,\,j\ne l}}|e_{ij}^{n}|+C\tau(\tau+h^2),\ \max_{0\le j\le J}|e_{kj}^{n+1}|,\ \max_{0\le i\le J}|e_{il}^{n+1}|\}\\
&\quad\le\max_{\substack{0\le i\le J\\0\le j\le J}}|e_{ij}^{n}|+C\tau(\tau+H^2),
\end{aligned}$$

e.g.

$$||e^{n+1}||_\infty\le||e^n||_\infty+C\tau(\tau+H^2),$$

hence

$$||e^{n}||_\infty\le||e^0||_\infty+CT(\tau+H^2),$$

which finishs the proof.

4. Numerical experiment

In this section we provide some numerical experiments.

For the parabolic equation with a variable coefficient, consider the equation (1) with initial function $u_0(x,y)=x(1-x)y(1-y)$, and

$$a(x,y,t)=x(1-x),$$
$$b(x,y,t)=y(1-y).$$

The real solution of this problem is $u=e^{-4t}x(1-x)y(1-y)$. We give some numerical results calculated by serial procedures and the algorithm in Table 1.

In our experiments the algebraic equations are solved by the biconjugate gradient stabilized algorithm. The control error in the biconjugate gradient stabilized algorithm is 1.0e-5. The last computational time is t=0.1. The max error is $\max_{i,j,n}|u_{ij}^n-U_{ij}^n|$, 2×2 processors are used in the parallel computation. $R=\tau/H^2=\tau/(m^2h^2)=r/m^2$, m is the ratio of the larger spatial step length H compared with the one spatial step length h. Our experiments are implemented on a massively distributed memory computer.

From Table 1 we find the smaller the spacing is, the higher the accuracy is. The accuracy will reduce while R is increasing and h and m are fixed. The same tendency occurs while m is increasing and h and R are fixed.

Table 2 shows the parallel property of the algorithm. Where mesh scale equals $J\times J$, T_s is the run time of a serial implementation, T_p is the parallel run time, speedup is the ratio of T_s and T_p, parallel efficiency is the ratio of speedup and the number of CPUS.

From this table we can see that the parallel efficiency will increase while the number of CPUS is increasing and the scale is fixed. This is because of the fact that our

Mesh scale	R	m	Max error in the Serial	Max error	Average error	Relative error(100%)
150×150	2.0	5	7.4106E-004	1.0117E-003	9.4656E-005	2.4201E-002
150×150	2.0	10	2.8809E-003	2.5180E-003	3.0589E-004	6.3633E-002
150×150	2.0	15	6.4144E-003	3.8527E-003	5.1609E-004	0.1081276
150×150	4.0	5	1.4942E-003	1.2721E-003	1.2464E-004	3.0503E-002
150×150	4.0	10	5.9528E-003	3.3049E-003	3.8478E-004	8.6057E-002
150×150	4.0	15	1.3671E-002	5.0588E-003	6.1974E-004	0.1538021
300×300	2.0	5	1.8596E-004	3.7717E-004	2.4704E-005	8.6821E-003
300×300	2.0	10	7.4107E-004	1.0335E-003	9.6162E-005	2.4789E-002
300×300	2.0	15	1.6570E-003	1.7934E-003	1.9685E-004	4.4316E-002
300×300	2.0	20	2.8808E-003	2.5726E-003	3.0719E-004	6.5256E-002
300×300	4.0	5	3.7146E-004	4.5558E-004	3.4156E-005	1.0531E-002
300×300	4.0	10	1.4942E-003	1.3189E-003	1.2790E-004	3.1738E-002
300×300	4.0	15	3.2775E-003	2.3429E-003	2.4928E-004	5.7677E-002
300×300	4.0	20	5.9528E-003	3.4182E-003	3.8950E-004	8.9445E-002
450×450	2.0	5	8.2748E-005	2.0571E-004	1.1192E-005	4.6827E-003
450×450	2.0	10	3.2979E-005	5.8460E-004	4.4879E-005	1.3630E-002
450×450	2.0	15	7.4106E-004	1.0408E-003	9.6829E-005	2.4987E-002
450×450	2.0	20	1.3025E-003	1.5458E-003	1.6078E-004	3.7694E-002
450×450	4.0	5	1.6560E-004	2.4030E-004	1.5361E-005	5.4801E-003
450×450	4.0	10	6.6199E-004	7.2731E-004	6.1273E-005	1.7081E-002
450×450	4.0	15	1.4942E-003	1.3355E-003	1.2910E-004	3.2175E-002
450×450	4.0	20	2.6437E-003	2.0140E-003	2.1057E-004	4.9550E-002

Table 1

Mesh scale	T_s (s)	CPU	T_p (s)	Speedup	Parallel efficiency(100%)
300×300	186.3031359	9	13.837983	13.4632	1.4959
300×300	186.3031359	25	3.7042720	50.2941	2.0118
300×300	186.3031359	36	2.2475039	82.8934	2.3026
450×450	1047.474992	9	76.897519	13.6217	1.5135
450×450	1047.474992	25	24.197152	43.2892	1.7316
450×450	1047.474992	36	15.337695	68.2942	1.8971
450×450	1047.474992	100	4.0995520	255.5096	2.5551
600×600	3529.428375	9	404.73663	8.7203	0.9689
600×600	3529.428375	25	87.232743	40.4599	1.6184
600×600	3529.428375	36	57.924847	60.9312	1.6925
600×600	3529.428375	100	14.912160	236.6812	2.3668

Table 2

algorithm can be implemented only with communication between nearby processors. Because the smaller the scale of the algebraic equations is, the less the iteration count is when it converges, when we use our algorithm and get some small scale algebraic equations instead of large scale algebraic equations in the serial procedures, the run time can be significantly reduced, so the parallel efficiency is very high.

REFERENCES

[1] Clint N Dawson, Qiang Du, and Todd F Dupont, *A finite difference domain decomposition algorithm for numerical solution of the heat equation*, Math. of Computation, 57:63-71, 1991.

[2] Zhang Bao-lin and Wan Zheng-su, *New techniques in designing finite difference domain decomposition algorithm for the heat equation*, J. Comput. & Math. Appl. 45:1695-1705, 2003.

[3] Cui Xia, *Some simplified domain decomposition algorithns for two-Dimensional heat equation*, Annual Report of Laboratory of Computational Physics Institute of Appl. Physics and Computational Math., Beijing, 2002.

[4] Q. Du, M. Mu, and Z. N. Wu, *Efficient parallel algorithms for parabolic problems*, SIAM J. Numer. Anal. 39:1469-1487, 2001.

[5] Cui Xia, *Some new domain decomposition algorithms for two-dimensional heat equation*, Annual Report of Laboratory of Computational Physics Institute of Appl. Physics and Computational Math., Beijing, 2001.

[6] Zhang Bao-lin, *Difference graphs of block ADI method*, SIAM J. Numer. Anal. 38:742-752, 2002.

[7] Zhang Bao-lin, and Shen Weidong, *Notes on finite difference domain decomposition algorithm for the solution of heat equation*, Journal on Numerical Methods and Computer Applications, 2:81-90, 2002.

[8] Zhang Bao-lin and Chen Jin, *Parallel solution of parabolic equations by finite difference methods*, Journal on Numerical Methods and Computer Applications, 3:187-195, 1995.

[9] Zhang Bao-lin, Gu TongXiang, and Mo ZeYao, *Principles and Methods of Numerical Parallel Computation [M]*, National Defence Industry Press, Beijing, 1999.