J. Korea Soc. Math. Educ. Ser. B: Pure Appl. Math. ISSN 1226-0657
Volume 13, Number 4 (November 2006), Pages 281–294

RECTANGULAR DOMAIN DECOMPOSITION METHOD FOR PARABOLIC PROBLEMS

YOUNBAE JUN* AND TSUN-ZEE MAI**

ABSTRACT. Many partial differential equations defined on a rectangular domain can be solved numerically by using a domain decomposition method. The most commonly used decompositions are the domain being decomposed in stripwise and rectangular way. Theories for non-overlapping domain decomposition (in which two adjacent subdomains share an interface) were often focused on the stripwise decomposition and claimed that extensions could be made to the rectangular decomposition without further discussions. In this paper we focus on the comparisons of the two ways of decompositions. We consider the unconditionally stable scheme, the MIP algorithm, for solving parabolic partial differential equations. The SOR iterative method is used in the MIP algorithm. Even though the theories are the same but the performances are different. We found out that the stripwise decomposition has better performance.

1. INTRODUCTION

Many partial differential equations defined on a rectangular domain can be solved numerically by a domain decomposition method. Non-overlapping domain decomposition methods [1, 2, 3, 4, 5, 6, 9, 10] have been considered as efficient parallel algorithms for solving parabolic partial differential equations in parallel computers. The basic idea for a non-overlapping decomposition is that the original domain is decomposed into independent stripwise or rectangular subdomains in which two adjacent subdomains share interface lines without overlapping the region. The problem defined in the subdomains is then becoming independent subproblems so that they can be solved independently. In order to accomplish the independency, the values on the interfaces must be estimated before the subproblems can be solved using an implicit scheme. During the last decade and half there are conditionally stable

Received by the editors May 12, 2006 and, in revised form, October 9, 2006.
2000 *Mathematics Subject Classification.* 65M06.
Key words and phrases. domain decomposition method, parabolic problem, Dirichlet condition, finite difference method.

schemes [1, 2] and unconditionally stable schemes [3, 4, 5, 6, 9, 10] being developed for solving the parabolic problems. Theories of those schemes were focused on the stripwise decomposition and claimed that extensions could be made to rectangular decomposition without further discussions. In this paper we focus on the implementation of the unconditionally stable scheme, the modified implicit prediction (MIP) algorithm [4], on rectangular decomposition. The SOR iterative method is used for solving large sparse linear systems which are generated by the discretization of the subdomains. We propose the practical optimum over-relaxation parameter of the SOR scheme. We also prove that the stripwise method is more efficient than the rectangular method, which is demonstrated in numerical experiments.

2. Domain Decomposition Methods

In this paper we consider the parabolic partial differential equation of the form

$$u_t = u_{xx} + u_{yy} + \alpha u_x + \beta u_y + \gamma u + f(x,y,t), \tag{2.1}$$

defined in $\Omega = [0,1] \times [0,1]$ and $0 \le t \le 1$, where α, β, γ are constants, with the initial condition

$$u(x,y,0) = u^0(x,y) \text{ in } \Omega, \tag{2.2}$$

and with Dirichlet boundary conditions on $\partial\Omega$ such as

$$\begin{aligned} &u(0,y,t) = g_1(y,t), u(1,y,t) = g_2(y,t), 0 \le y \le 1, t > 0, \\ &u(x,0,t) = g_3(x,t), u(x,1,t) = g_4(x,t), 0 \le x \le 1, t > 0. \end{aligned} \tag{2.3}$$

Finite difference method is often used to discretize the domain. We choose the positive integers L, M, and N so that $\Delta x = \frac{1}{L}$, $\Delta y = \frac{1}{M}$, and $\Delta t = \frac{1}{N}$. Take $x_i = i\Delta x, y_j = j\Delta y$, and $t_n = n\Delta t$, where $i = 0, \cdots, L, j = 0, \cdots, M$, and $n = 0, \cdots, N$. For the sake of simplicity we let $h = \Delta x = \Delta y$. We let u_{ij}^n be the exact solution that corresponds to $u(x_i, y_j, t_n)$ at the point (x_i, y_j, t_n), and let w_{ij}^n be the approximation to u_{ij}^n. We denote $f(x_i, y_j, t_n)$ by f_{ij}^n. Then the central finite difference operators for the time level n at the point (x_i, y_j) are given by

$$w_t^n = \frac{w_{ij}^n - w_{ij}^{n-1}}{\Delta t}, \; w_{xx}^n = \frac{w_{i+1,j}^n - 2w_{ij}^n + w_{i-1,j}^n}{(\Delta x)^2}, \; w_{yy}^n = \frac{w_{i,j+1}^n - 2w_{ij}^n + w_{i,j-1}^n}{(\Delta y)^2},$$

$$w_x^n = \frac{w_{i+1,j}^n - w_{i-1,j}^n}{2\Delta x}, \; w_y^n = \frac{w_{i,j+1}^n - w_{i,j-1}^n}{2\Delta y}.$$

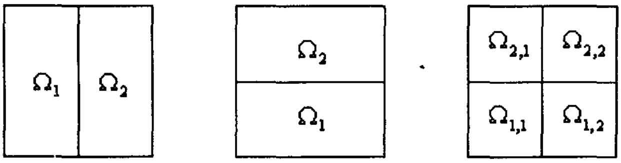

FIGURE 1. Various Decompositions

The main idea of the domain decomposition method is to decompose the original spatial domain into smaller subdomains in either stripwise or rectangular manner, see Fig. 1. Many non-overlapping domain decomposition have been done into vertical strips [1, 2, 3, 4, 5, 6, 10]. In this paper, the rectangular decomposition is considered instead of stripwise decomposition. When we divide the spatial domain into P vertical strips and P horizontal strips, the rectangular domain is decomposed into $P \times P$ subdomains.

The fully implicit scheme, namely BTCS (Backward Time Central Space difference) method, is unconditionally stable, however the spatial domain of the problem is not decomposed, and hence it is not domain decomposition method. There are many domain decomposition methods, for examples Dawson's method[1, 2], the implicit prediction and implicit correction (IPIC) method[3], the modified implicit prediction (MIP) method[4], the explicit prediction and implicit correction (EPIC) method[9], and the stabilized explicit/implicit domain decomposition (SEIDD) method[10]. The main difference among these methods is the way to estimate the values at the interfaces, which is described in [3, 4, 5]. In this paper we adopt the MIP algorithm, in which the spatial domain is decomposed stripwise, namely stripwise MIP algorithm. Since we consider rectangular domain decomposition, we need to estimate not only vertical interface lines, but also horizontal ones. The rectangular MIP algorithm is as follows:

Rectangular MIP Algorithm

(1) At the boundary :
$w_{ij}^n = u_{ij}^n$

(2) Vertical prediction : estimate the values of interface points at $x = x_i$
$w_t^n = \bar{w}_{xx}^n + w_{yy}^n + \alpha \bar{w}_x^n + \beta w_y^n + \gamma w^n + f_{ij}^n$
where $\bar{w}_{xx}^n = \frac{2[iu_{Lj}^n - Lw_{ij}^n + (L-i)u_{0j}^n]}{i(L-i)\Delta x}$ and $\bar{w}_x^n = u_{Lj}^n - u_{0j}^n$

(3) Horizontal prediction : estimate the values of interface points at $y = y_j$
$w_t^n = w_{xx}^n + \bar{w}_{yy}^n + \alpha w_x^n + \beta \bar{w}_y^n + \gamma w^n + f_{ij}^n$

where $\bar{w}_{yy}^n = \frac{2[ju_{iM}^n - Mw_{ij}^n + (M-j)u_{i0}^n]}{j(M-j)\Delta y}$ and $\bar{w}_y^n = u_{iM}^n - u_{i0}^n$

(4) At the interior for each subdomain :
$w_t^n = w_{xx}^n + w_{yy}^n + \alpha w_x^n + \beta w_y^n + \gamma w^n + f_{ij}^n$
which is the fully implicit scheme.

We note that $u_{Lj}^n, u_{0j}^n, u_{iM}^n$, and u_{i0}^n are given as the boundary values. It should be pointed out that Dawson's method is conditionally stable but IPIC, EPIC, and SEIDD methods are unconditionally stable, however the correction procedure is necessary step in the later methods. But there is no correction phase in the MIP method. Since the stripwise MIP prediction has been shown to be unconditionally stable in [4], the vertical prediction and horizontal prediction, and hence rectangular MIP prediction, is also unconditionally stable. We also note that the Gauss-Seidel (GS) iterative method [8] has been employed in [4] to solve the linear system for finding interior point values. In this paper we compare the rectangular MIP algorithm with the stripwise MIP algorithm at the use of the successive over-relaxation (SOR) iterative method [8].

3. Analysis of the Rectangular MIP Algorithm

In this section, we analyze the rectangular MIP algorithm and compare it with the stripwise MIP algorithm. We first describe the coefficient matrix of the linear system generated by the central finite difference method for finding interior point values.

The problem (2.1)–(2.3) can be discretized and represented by a linear system with the coefficient matrix A defined by

$$A = (1 + 4r - \gamma\Delta t)I - 4rR + rS \tag{3.1}$$

where $r = \frac{\Delta t}{h^2}$ and $h = \Delta x = \Delta y$, and R and S are block triangular matrices given by

$$R = \frac{1}{4}\begin{bmatrix} \hat{R} & I & O & \cdots & O \\ I & \hat{R} & I & \ddots & \vdots \\ O & \ddots & \ddots & \ddots & O \\ \vdots & \ddots & I & \hat{R} & I \\ O & \cdots & O & I & \hat{R} \end{bmatrix} \quad \text{with } \hat{R} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & 0 & 1 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}, \text{ and}$$

$$S = \begin{bmatrix} \hat{S} & -\frac{\beta h}{2}I & O & \cdots & O \\ \frac{\beta h}{2}I & \hat{S} & -\frac{\beta h}{2}I & \ddots & \vdots \\ O & \ddots & \ddots & \ddots & O \\ \vdots & \ddots & \frac{\beta h}{2}I & \hat{S} & -\frac{\beta h}{2}I \\ O & \cdots & O & \frac{\beta h}{2}I & \hat{S} \end{bmatrix}$$

$$\text{with } \hat{S} = \begin{bmatrix} 0 & -\frac{\alpha h}{2} & 0 & \cdots & 0 \\ \frac{\alpha h}{2} & 0 & -\frac{\alpha h}{2} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \frac{\alpha h}{2} & 0 & -\frac{\alpha h}{2} \\ 0 & \cdots & 0 & \frac{\alpha h}{2} & 0 \end{bmatrix}$$

The system may be very large and sparse if h is very small, thus iterative methods are often used to solve such system [8]. In this paper, since we use the successive over-relaxation (SOR) iterative method, we first investigate the determination of the optimum value of the over-relaxation parameter ω of the SOR iterative method.

We define MIP($P \times P$) to be the rectangular MIP method with P vertical and P horizontal strips. MIP(P^2) is defined by the stripwise MIP method with P^2 vertical strips. We note that the number of subdomains used in the MIP($P \times P$) is the same as in the MIP(P^2). We let $G_{P\times P}$ and G_{P^2} be the iteration matrices of the Gauss-Seidel (GS) method for the MIP($P \times P$) and the MIP(P^2), respectively. We note that if $P = 1$, then $G_{1\times 1}$ is the GS iteration matrix of the BTCS scheme. We also note that $G_{1\times 1}$ is identical to G_{1^2}. Theorems 3.1 and 3.2 show the spectral radii of $G_{P\times P}$ and G_{P^2}, respectively, for Dirichlet boundary value problem (2.1)–(2.3).

Theorem 3.1. *The spectral radius of $G_{P\times P}$ is approximated to*

$$\rho(G_{P\times P}) \approx \left[\frac{4r\cos P\pi h}{1+4r-\gamma\Delta t}\right]^2.$$

Moreover if $\alpha = \beta = 0$, then the spectral radius of $G_{P\times P}$ is exact to

$$\rho(G_{P\times P}) = \left[\frac{4r\cos P\pi h}{1+4r-\gamma\Delta t}\right]^2.$$

Proof. Consider Equation (3.1). The Jacobi iteration matrix $G_{P\times P,J}$ of the MIP($P\times P$) algorithm can be written as

$$G_{P\times P,J} = I - \frac{1}{1+4r-\gamma\Delta t}\{(1+4r-\gamma\Delta t)I - 4rR + rS\}$$

$$= \frac{4r}{1+4r-\gamma\Delta t}R - \frac{r}{1+4r-\gamma\Delta t}S.$$

Thus

$$\begin{aligned}\rho(G_{P\times P,J}) &\le \frac{4r}{1+4r-\gamma\Delta t}\rho(R) + \frac{r}{1+4r-\gamma\Delta t}\rho(S) \\ &\le \frac{4r\cos P\pi h}{1+4r-\gamma\Delta t} + \frac{r(|\alpha h|+|\beta h|)}{1+4r-\gamma\Delta t}.\end{aligned}$$

When h is very small, the spectral radius of the GS iteration matrix $\rho(G_{P\times P})$ can be approximated by

$$\rho(G_{P\times P}) = [\rho(G_{P\times P,J})]^2 \approx \left[\frac{4r\cos P\pi h}{1+4r-\gamma\Delta t}\right]^2.$$

If $\alpha=\beta=0$, then S is equal to zero matrix, and hence

$$\rho(G_{P\times P}) = \left[\frac{4r\cos P\pi h}{1+4r-\gamma\Delta t}\right]^2.$$

□

Theorem 3.2. *The spectral radius of G_{P^2} is approximated to*

$$\rho(G_{P^2}) \approx \left[\frac{2r(\cos P^2\pi h + \cos\pi h)}{1+4r-\gamma\Delta t}\right]^2.$$

Moreover if $\alpha=\beta=0$, then the spectral radius of $G_{P\times P}$ is exact to

$$\rho(G_{P^2}) = \left[\frac{2r(\cos P^2\pi h + \cos\pi h)}{1+4r-\gamma\Delta t}\right]^2.$$

Proof. See [6]. □

Theorem 3.3.

$$\left[\frac{4r\cos P\pi h}{1+4r-\gamma\Delta t}\right]^2 \ge \left[\frac{2r(\cos P^2\pi h + \cos\pi h)}{1+4r-\gamma\Delta t}\right]^2.$$

The equality holds only if $P=1$.

Proof. Without loss of generosity, we assume that $P\ge 1$ and $P^2 \le \frac{L}{2}$, i.e. $P^2h \le \frac{1}{2}$. It is sufficient to show that

$$2\cos P\pi h - \cos P^2\pi h - \cos\pi h \ge 0.$$

Let $f(P) = 2\cos P\pi h - \cos P^2\pi h - \cos\pi h$ be a function of P. We claim that $f'(P)\ge 0$ and $f(1)=0$. Clearly $f(1)=0$.

$$\begin{aligned}f'(P) &= -2\pi h\sin P\pi h + 2P\pi h\sin P^2\pi h \\ &= -2\pi h\sin P\pi h + 2\pi h\sin P^2\pi h - 2\pi h\sin P^2\pi h + 2P\pi h\sin P^2\pi h\end{aligned}$$

$$= 2\pi h(\sin P^2\pi h - \sin P\pi h) + 2\pi h(P-1)\sin P^2\pi h$$
$$\geq 0.$$

Thus $f(P)$ is an increasing function and has a minimum value of 0, and hence $f(P) \geq 0$. The equality holds only if $P = 1$. □

It should be pointed out that Theorem 3.3 shows the spectral radius of the GS iteration matrix of the stripwise MIP algorithm is less than or equal to that of the rectangular MIP algorithm. This implies that the stripwise MIP method is more efficient than the rectangular MIP method, which will be demonstrated in a later section. Since we employ the SOR iterative method for solving linear systems, the way to determine the over-relaxation parameter ω is important. The optimum over-relaxation parameter can be computed [8] by

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\rho(G)}},$$

where G is the GS iteration matrix. We define the approximated optimum over-relaxation parameters $\omega(P \times P)_{app}$ and $\omega(P^2)_{app}$ by

$$\omega(P \times P)_{app} = \frac{2}{1+\sqrt{1-\left[\frac{4r\cos P\pi h}{1+4r-\gamma\Delta t}\right]^2}} \tag{3.2}$$

and

$$\omega(P^2)_{app} = \frac{2}{1+\sqrt{1-\left[\frac{2r(\cos P^2\pi h+\cos\pi h)}{1+4r-\gamma\Delta t}\right]^2}}, \tag{3.3}$$

for the MIP($P \times P$) and the MIP(P^2), respectively. We pointed out in [6] that the approximated optimum over-relaxation parameter $\omega(P^2)_{app}$ is a very good approximation to ω_{opt} for the stripwise MIP method. It will be seen later that $\omega(P \times P)_{app}$ for the rectangular MIP method is also excellent.

4. Numerical Results

In this section we present the numerical experiments of the rectangular MIP algorithm to solve Dirichlet boundary value problems (2.1)–(2.3). Let $P \times P$ be the number of subdomains decomposed in rectangular manner. We note that when $P = 1$, the MIP algorithm reduces to the fully implicit BTCS method which is very well known as an unconditionally stable scheme. When P is other than 1, we need to estimate the vertical or horizontal interface information in which a tridiagonal linear system is generated and solved by a direct method. After obtaining the information of the interfaces, a large sparse linear system is generated for each subdomain that can be solved independently by the SOR iterative method. The stopping criterion in the iterative procedure is given by

$$\frac{||w^{(n)} - w^{(n-1)}||_2}{||w^{(n)}||_2} < \epsilon$$

where $w^{(n)}$ is the estimate at the nth SOR iteration and ϵ is a preset small value. We choose $\epsilon = 10^{-6}$ for our experiments. All our numerical experiments are carried out on a Sun V880 running Solaris 9.

4.1. Model problems

4.1.1. *Model problem* 1 ($MP1$)

$$u_t = u_{xx} + u_{yy} - u + 2e^{-t} \sin x \cos y,$$

over the region $[0, 1] \times [0, 1]$ and $0 \le t \le 1$. The initial and boundary conditions are given by

$$u(x, y, 0) = \sin x \cos y, 0 < x, y < 1,$$

$$u(0, y, t) = 0, u(1, y, t) = e^{-t} \sin 1 \cos y, 0 < y < 1, t > 0,$$

$$u(x, 0, t) = e^{-t} \sin x, u(x, 1, t) = e^{-t} \sin x \cos 1, 0 < x < 1, t > 0.$$

The exact solution is $u(x, y, t) = e^{-t} \sin x \cos y$.

4.1.2. *Model problem* 2 ($MP2$)

$$u_t = u_{xx} + u_{yy} + u_x + u_y + u + f(x, y, t),$$

where $f(x, y, t) = (1 + xy) \cos t - (1 + x)(1 + y) \sin t$ over the region $[0, 1] \times [0, 1]$ and $0 \le t \le 1$. The initial and boundary conditions are given by

$$u(x, y, 0) = 0, 0 < x, y < 1,$$

$$u(0, y, t) = \sin t, u(1, y, t) = (1 + y) \sin t, 0 < y < 1, t > 0,$$

$$u(x, 0, t) = \sin t, u(x, 1, t) = (1 + x) \sin t, 0 < x < 1, t > 0.$$

The exact solution is $u(x, y, t) = (1 + xy) \sin t$.

We note that MP1 is chosen with $\alpha = \beta = 0$ and MP2 is chosen with nonzero coefficients.

4.2. Stability We first show that the rectangular MIP algorithm is unconditionally stable for solving Dirichlet boundary value problems (2.1)–(2.3). We note that the stable condition for a fully explicit scheme is the value of $\lambda \leq \frac{1}{2}$ where λ is defined by

$$\lambda = \frac{\Delta t}{(\Delta x)^2} + \frac{\Delta t}{(\Delta y)^2}.$$

Thus we use λ as the guide index to illustrate that the rectangular MIP algorithm is stable for large values of λ. Table 1 shows the exact maximum error

$$||w^N - u^N||_\infty$$

at $t = 1$ with various values of λ. We also compare the error in the rectangular MIP(2×2) method of 2×2 subdomains with the stripwise MIP(2^2) method of 4 vertical subdomains.

Table 1. Maximum error at $t = 1$ with various λ

Δt	Δx	λ	MP1			MP2		
	$= \Delta y$		BTCS	MIP(2×2)	MIP(2^2)	BTCS	MIP(2×2)	MIP(2^2)
$\frac{1}{2}$	$\frac{1}{100}$	10000	3e-3	3e-3	6e-3	1e-2	1e-2	1e-2
$\frac{1}{5}$	$\frac{1}{100}$	4000	1e-3	1e-3	4e-3	7e-3	6e-3	7e-3
$\frac{1}{10}$	$\frac{1}{100}$	2000	6e-4	7e-4	3e-3	3e-3	3e-3	3e-3
$\frac{1}{20}$	$\frac{1}{100}$	1000	2e-4	4e-4	3e-3	1e-3	1e-3	1e-3

We note that for the over-relaxation parameter ω of the SOR iterative method in Table 1, we used the approximated optimum over-relaxation parameter ω_{app} in Equations (3.2) and (3.3). As we can see in Table 1, the rectangular MIP algorithm is as unconditionally stable as the stripwise MIP algorithm. In the next section, we will describe how accurate the approximated ω_{app} is and we compare it with the actual ω_{opt}.

4.3. Optimum over-relaxation parameter Approximated optimum over-relaxation parameter of the SOR method to the MP2 at $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.1$ with 2×2 rectangular subdomains is computed by

$$\rho(G_{2\times 2})_{app} = \left[\frac{4r \cos P\pi h}{1 + 4r - \gamma \Delta t}\right]^2 = 0.9956$$

and

$$\omega(2 \times 2)_{app} = \frac{2}{1 + \sqrt{1 - \rho(G_P)}} = 1.8757.$$

Fig. 2 is the graph of CPU time with respect to the various ω to the MP2 at $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.1$ with 2×2 subdomains. As we can see in Fig. 2, the value of $\omega(2 \times 2)_{app}$ is nearly optimum. We note that when $\omega = 1$, the SOR method becomes the GS method.

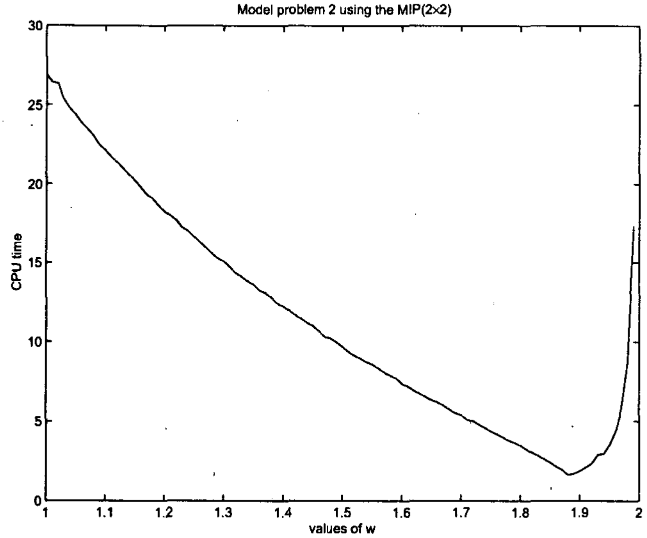

FIGURE 2. CPU time for various ω

Table 2 shows the comparison of ω_{app} with ω_{opt} for the model problems with $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.1$. In order to compute the actual ω_{opt}, Matlab [7] was used in computation of the actual spectral radius. As we can see in Table 2, the approximated optimum over-relaxation parameter ω_{app} is very good approximation to the actual optimum parameter ω_{opt}. It should be pointed out that if $\alpha = \beta = 0$ like MP1, then the approximated $\rho(G_{P \times P})_{app}$ is equal to the actual $\rho(G_{P \times P})_{act}$ which was mentioned in Theorem 3.1. We note that all computation in Table 2 are done in 15 significant digits precision.

Table 2. Optimum over-relaxation parameter

(app=approximate,act=actual,opt=optimum)

MP	MP1				MP2			
$P \times P$	1	2×2	5×5	10×10	1	2×2	5×5	10×10
$\rho(G_{P \times P})_{app}$	0.9985	0.9955	0.9750	0.9040	0.9986	0.9956	0.9751	0.9041
ω_{app}	1.9246	1.8744	1.7269	1.5269	1.9270	1.8757	1.7274	1.5271
$\rho(G_{P \times P})_{act}$	0.9985	0.9955	0.9750	0.9040	0.9985	0.9956	0.9751	0.9041
ω_{opt}	1.9246	1.8744	1.7269	1.5269	1.9264	1.8754	1.7273	1.5271

4.4. Efficiency We now investigate the efficiency of the rectangular MIP algorithm with the SOR method using ω_{app} for the model problems. A common measurement for the efficiency of a parallel algorithm is the speedup which is defined by

$$\text{Speedup} = \frac{\text{Execution time for a single processor}}{\text{Execution time using } Q \text{ processors}}.$$

Since the numerical experiment is simulated with one processor, the true parallel execution time using Q processors is roughly equivalent to the total CPU time, T_Q, obtained by the simulation, being divided by Q. Thus the speedup S_Q is simplified to

$$S_Q = \frac{T_1}{T_Q/Q}.$$

Table 3 shows the maximum error and speedup S_Q of the SOR method of the rectangular MIP algorithm to the model problems at the final time level $t = 1$ with $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.1$ using ω_{app}. We note that the number of processors Q is equal to $P \times P$ for the rectangular MIP algorithm.

Table 3. Speedup of the rectangular MIP algorithm

	$Q(=P \times P)$	1	2×2	4×4	5×5	10×10	25×25	50×50
MP1	Error	6e-4	7e-4	3e-3	3e-3	3e-3	3e-3	2e-3
	T_Q	2.93	1.60	0.72	0.61	0.31	0.13	0.04
	S_Q	1	7.325	65.111	120.08	945.16	14086.5	183125
MP2	Error	3e-3	3e-3	3e-3	3e-3	3e-3	3e-3	3e-3
	T_Q	3.04	1.72	0.72	0.58	0.29	0.12	0.04
	S_Q	1	7.070	67.556	131.03	1048.28	15833.3	190000

Fig. 3 shows the graphs of $\log S_Q$ with respect to the number of processors Q for the model problems with $\Delta x = \Delta y = 0.01$ and $\Delta t = 0.1$. We note that the speedup is much more than linear.

We now describe comparison of the efficiency of the rectangular MIP algorithm with the stripwise MIP algorithm. Table 4 shows total CPU time of the MIP($P \times P$) and the MIP(P^2) for the model problems with various P. As we can see in Table 4, $\rho(G_{P^2}) \leq \rho(G_{P \times P})$ and total CPU time for the MIP(P^2) is less than that for the MIP($P \times P$). We should note that as we mentioned earlier in Theorem 3.3, the stripwise MIP method is faster than the rectangular MIP method.

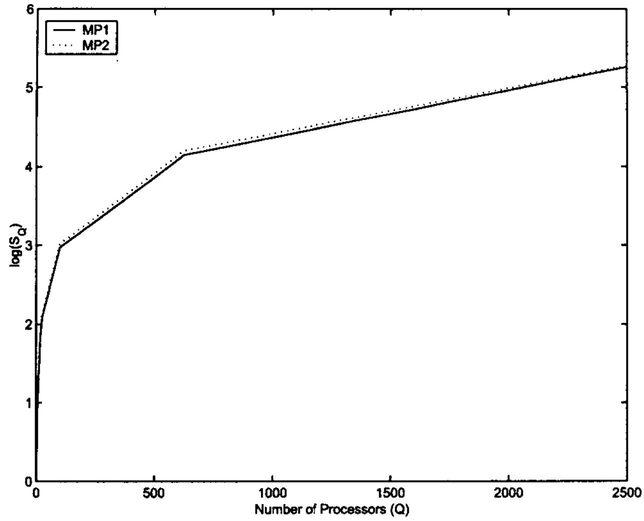

FIGURE 3. Speedup curves for the model problems

Table 4. CPU time for the model problems

MP	MP1			MP2		
$P \times P$	1	2×2	5×5	1	2×2	5×5
$\rho(G_{P \times P})_{app}$	0.9985	0.9955	0.9750	0.9986	0.9956	0.9751
ω_{app}	1.9246	1.8744	1.7269	1.9270	1.8757	1.7274
Error	6e-4	7e-4	3e-4	3e-3	3e-3	3e-3
CPU	2.93	1.60	0.61	3.04	1.72	0.58
P^2	1	4	25	1	4	25
$\rho(G_{P^2})_{app}$	0.9985	0.9911	0.7277	0.9986	0.9912	0.7278
ω_{app}	1.9246	1.8275	1.3142	1.9270	1.8284	1.3143
Error	6e-4	3e-3	3e-3	3e-3	3e-3	3e-3
CPU	2.93	1.49	0.22	3.04	1.45	0.21

We note that when $L = M = 100$, the MIP(P^2) method can use processors up to $P^2 \leq \frac{L}{2}$, i.e. 49 processors. However the MIP($P \times P$) method can use processors up to $P \times P \leq \frac{L}{2} \times \frac{M}{2}$, i.e. 2500 processors. As long as there are enough processors, the MIP($P \times P$) method is useful.

5. CONCLUSION

In this paper, we compared the MIP algorithms solving parabolic partial differential equations with two most common ways of decomposing the spatial domain, stripwise and rectangular decompositions. The iterative method used was the SOR method; the rate of convergence of an iterative method depends on the spectral radius of the iteration matrix as well as the choice of the over-relaxation parameter. We showed that the spectral radius of the iteration matrix is smaller in the case of stripwise decomposition when the same number of processors are used. The smaller spectral radius implies faster convergence. In the numerical experiments the performance using stripwise decomposition is much better than in the rectangular decomposition. Moreover, we have shown how to obtain very good estimations for the optimal over-relaxation parameters for the SOR method in both domain decompositions.

References

1. C. N. Dawson, Q. Du & T. F. Dupont: A finite difference domain decomposition algorithm for numerical solution of the heat equation. *Mathematics of Computation* **57** (1991), no. 195, 63–71.
2. 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), no. 4, 1045–1061.
3. Y. Jun & T. Z. Mai: IPIC domain decomposition algorithm for parabolic problems. *Appl. Math. Comput.* **177** (2006), 352–364.
4. Y. Jun & T. Z. Mai: ADI method–Domain decomposition. *Appl. Numer. Math.* **56** (2006), 1092–1107.
5. Y. Jun & T. Z. Mai: Domain decomposition method for parabolic problems with Neumann conditions. *to appear in Appl. Math. Comput.*
6. Y. Jun & T. Z. Mai: Domain decomposition method for parabolic problems. *PDPTA '06 International Conference* (2006), 389–394.
7. MATHWORKS: Inc. http: www.mathworks.com.
8. D. M. Young: *Iterative Solution of Large Linear Systems.* Dover, New York 2003.
9. J. Zhu & H. Qian: On an efficient parallel algorithm for solving time dependent partial differential equations. *PDPTA '98 International Conference* (1998), 394–401.
10. Y. Zhuang & X. Sun: Stabilized explicit-implicit domain decomposition methods for the numerical solution of parabolic equations. *SIAM J. Sci. Comput.* **24** (2002), 335–358.

*Department of Mathematics, University of West Alabama, Livingston, Alabama 35470, USA
Email address: yjun@uwa.edu

**DEPARTMENT OF MATHEMATICS, UNIVERSITY OF ALABAMA, TUSCALOOSA, ALABAMA 35487, USA
Email address: tmai@gp.as.ua.edu