

# Domain decomposition method for parabolic problems

**Younbae Jun**

Department of Mathematics  
University of West Alabama  
Livingston, Alabama 35470, USA

**Tsun-Zee Mai**

Department of Mathematics  
University of Alabama  
Tuscaloosa, Alabama 35487, USA

**Abstract**—Domain decomposition methods are widely used to solve the parabolic partial differential equations with Dirichlet, Neumann, or mixed boundary conditions. Modified implicit prediction (MIP) algorithm is unconditionally stable domain decomposition method. In this paper, the SOR iterative technique is applied to the MIP algorithm and the optimum over-relaxation parameter is provided.

**Keywords:** Domain decomposition method, Parabolic problem, Dirichlet condition, Neumann condition, mixed condition, Finite difference method.

## 1 Introduction

Non-overlapping domain decomposition methods [1], [2], [3], [4], [5], [8], [9] have been considered as efficient parallel algorithms for solving parabolic partial differential equations in parallel computers. The basic idea for a non-overlapping decomposition, in which two adjacent subdomains share an interface, is that the original spatial domain is decomposed into subdomains. All the values at the interfaces are estimated before the interior values of each subdomain is solved by an implicit scheme. Independent subproblem defined on each subdomain is solved in parallel manner. There are conditionally stable [1], [2] and unconditionally stable [3], [4], [5], [8], [9] methods being developed in the last decade and a half. In this paper we focus on the use of the modified implicit prediction (MIP) algorithm [4], [5] for solving parabolic problems with various boundary conditions. We investigate the optimum value of the over-relaxation parameter when the SOR method is used as the iterative algorithm.

## 2 Domain decomposition algorithms

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

$$u_t = u_{xx} + u_{yy} + \alpha u_x + \beta u_y + \gamma u + f(x, y, t), \quad (1)$$

defined in  $\Omega = [0, 1] \times [0, 1]$  and  $0 \leq t \leq 1$ , where  $\alpha, \beta, \gamma$  are constants, with the initial condition

$$u(x, y, 0) = u^0(x, y) \text{ in } \Omega, \quad (2)$$

and with boundary conditions such as

(i) Dirichlet boundary condition

$$u(0, y, t) = g_1(y, t), u(1, y, t) = g_2(y, t), \quad (3)$$

$$u(x, 0, t) = g_3(x, t), u(x, 1, t) = g_4(x, t),$$

or

(ii) pure Neumann boundary condition

$$u_x(0, y, t) = q_1(y, t), u_x(1, y, t) = q_2(y, t), \quad (4)$$

$$u_y(x, 0, t) = q_3(x, t), u_y(x, 1, t) = q_4(x, t),$$

or

(iii) mixed boundary condition

$$u(0, y, t) = g_1(y, t), u(1, y, t) = g_2(y, t), \quad (5)$$

$$u_y(x, 0, t) = q_3(x, t), u_y(x, 1, t) = q_4(x, t),$$

The initial value problem (1–2) with Dirichlet boundary condition (3), with pure Neumann boundary condition (4), and with the mixed boundary condition (5) are referred to as the Dirichlet problem, the Neumann problem, and the Mixed problem, respectively, in this paper.

Finite difference method is often used to discretize the domain. We choose the positive integers  $L, M$ , and  $N$  so that  $h = \Delta x = \frac{1}{L} = \Delta y = \frac{1}{M}$ , and  $\Delta t = \frac{1}{N}$ . We note that  $\Delta x$  and  $\Delta y$  can be

different in the general case. For the sake of simplicity we choose the same values for  $\Delta x$  and  $\Delta y$ . Take  $x_i = i\Delta x, y_j = j\Delta y$ , and  $t_n = n\Delta t$ , where  $i = 0, \dots, L, j = 0, \dots, M$ , and  $n = 0, \dots, N$ . We let  $w_{ij}^n$  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

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

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

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

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. For the domain decomposition method we consider the modified implicit prediction (MIP) algorithm for solving Dirichlet, Neumann, and Mixed problems. The MIP algorithm has been shown to be unconditionally stable [4], [5]. There are many other domain decomposition algorithms [1], [2], [3], [8], [9] and the main difference among these algorithms is the way to estimate the values at the interfaces. The values at the interface  $x = x_i$ , where the domain is decomposed as vertical subdomains, of the MIP algorithm is estimated by

$$w_t^n = \bar{w}_{xx}^n + w_{yy}^n + \alpha \bar{w}_x^n + \beta w_y^n + \gamma w^n + f_{ij}^n,$$

where

$$\bar{w}_{xx}^n = \frac{2[iu_{Lj}^n - Lu_{ij}^n + (L-i)u_{0j}^n]}{i(L-i)\Delta x}$$

and

$$\bar{w}_x^n = u_{Lj}^n - u_{0j}^n.$$

We note that  $u_{Lj}$  and  $u_{0j}$  are boundary values. In this paper we focus on the use of the MIP algorithm with the successive over-relaxation (SOR) iterative method [7].

### 3 Determination of the over-relaxation parameter

In this section, we investigate the determination of the optimum value of the over-relaxation parameter  $\omega$  of the SOR iterative method. We first describe the coefficient matrices of the linear systems generated by the central finite difference method for finding interior point values corresponding to three types of parabolic problems.

#### 3.1 Dirichlet problem

Dirichlet problem can be discretized and represented by a linear system with the coefficient matrix  $A$  defined by

$$A = (1+4r-\gamma\Delta t)I_{K \times K} - 4rR_{K \times K} + rS_{K \times K} \quad (6)$$

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

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

$$\hat{R} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & 0 & 1 \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix}, \quad \text{and}$$

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

$$\hat{S} = \begin{bmatrix} 0 & -\frac{\alpha h}{2} & 0 & \dots & 0 \\ \frac{\alpha h}{2} & 0 & -\frac{\alpha h}{2} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \frac{\alpha h}{2} & 0 & -\frac{\alpha h}{2} \\ 0 & \dots & 0 & \frac{\alpha h}{2} & 0 \end{bmatrix}.$$

### 3.2 Pure Neumann problem

For the Neumann problem, the derivatives are given at the boundaries and the values at the boundaries are unknown. The linear system created for the Neumann problem includes the boundary points as unknown values to be solved. In order to do that a fictitious point has to be created for the central finite difference scheme at the boundary points. For example, at the boundary  $x = 0$ , the fictitious point can be generated by  $u_{-1j}^n = u_{1j}^n - 2\Delta x q_1(y_j, t_n)$ . Fictitious points at the other three boundaries may be created in the same manner. The MIP algorithm is used to estimate the values of derivatives at the interfaces. Details can be found in [5]. After the discretization, the coefficient matrix of a linear system for the Neumann problem is

$$A = (1 - \gamma\Delta t)I_{K \times K} + rT_{K \times K} \quad (7)$$

where  $T$  is a block triangular matrix given by

$$T = \begin{bmatrix} \hat{T} & -2I & O & \cdots & O \\ (-1 + \frac{\beta h}{2})I & \hat{T} & (-1 - \frac{\beta h}{2})I & \ddots & \vdots \\ O & \ddots & \ddots & \ddots & O \\ \vdots & \ddots & (-1 + \frac{\beta h}{2})I & \hat{T} & (-1 - \frac{\beta h}{2})I \\ O & \cdots & O & -2I & \hat{T} \end{bmatrix}$$

with

$$\hat{T} = \begin{bmatrix} 4 & -2 & 0 & \cdots & 0 \\ -1 + \frac{\alpha h}{2} & 4 & -1 - \frac{\alpha h}{2} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 + \frac{\alpha h}{2} & 4 & -1 - \frac{\alpha h}{2} \\ 0 & \cdots & 0 & -2 & 4 \end{bmatrix}.$$

### 3.3 Mixed problem

We assume that the Dirichlet boundary condition occurs at  $x = 0$  and  $x = 1$ . Thus we decompose the domain vertically. The coefficient matrix of the linear system is similar to the one in the Dirichlet problem, i.e.

$$A = (1 + 4r - \gamma\Delta t)I_{K \times K} - 4rR_{K \times K} + rS'_{K \times K} \quad (8)$$

where  $S'$  is a block triangular matrix given by

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

where  $\hat{S}$  is the same in the Dirichlet problem.

We note that we may decompose the domain horizontally using the technique in the Neumann problem for finding the values of the interface. However, the performance in this horizontal decomposition is similar to the Neumann problem which is not as efficient as the Dirichlet problem.

We let  $G_P$  be the iteration matrix of the Gauss-Seidel (GS) method for the MIP algorithm with  $P$  subdomains. We note that if  $P = 1$ , then  $G_1$  is the GS iteration matrix of the BTCS scheme. Following theorems show the spectral radii of  $G_P$  for three types of parabolic problems. The optimum over-relaxation parameter can be computed [7] by

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho(G_P)}}. \quad (9)$$

**Theorem 1:** For the Dirichlet problem, the spectral radius of  $G_P$  is approximated to

$$\rho(G_P) \approx \left[ \frac{2r(\cos P\pi h + \cos \pi h)}{1 + 4r - \gamma\Delta T} \right]^2. \quad (10)$$

*Proof:* Consider Equation (6). The Jacobi iteration matrix  $G_{P,J}$  of the MIP algorithm with  $P$  subdomains can be written as

$$\begin{aligned} G_{P,J} &= I - \frac{1}{1 + 4r - \gamma\Delta t} \{ (1 + 4r - \gamma\Delta t)I - 4rR + rS \} \\ &= \frac{4r}{1 + 4r - \gamma\Delta t} R - \frac{r}{1 + 4r - \gamma\Delta t} S. \end{aligned}$$

Thus

$$\begin{aligned} \rho(G_{P,J}) &\leq \frac{4r}{1 + 4r - \gamma\Delta t} \rho(R) + \frac{r}{1 + 4r - \gamma\Delta t} \rho(S) \\ &\leq \frac{2r(\cos P\pi h + \cos \pi h)}{1 + 4r - \gamma\Delta t} + \frac{r(|\alpha h| + |\beta h|)}{1 + 4r - \gamma\Delta t}. \end{aligned}$$

When  $h$  is small, the spectral radius of the GS iteration matrix  $\rho(G_P)$  can be approximated by

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

**Theorem 2:** For the Neumann problem, the spectral radius of  $G_P$  for any  $P$  is exactly

$$\rho(G_P) = \left[ \frac{4r}{1 + 4r - \gamma\Delta t} \right]^2. \quad (11)$$

*Proof:* See [5].  $\square$

**Theorem 3:** For the Mixed problem, the spectral radius of  $G_P$  is approximated to

$$\rho(G_P) \approx \left[ \frac{2r(\cos P\pi h + \cos \pi h)}{1 + 4r - \gamma\Delta T} \right]^2. \quad (12)$$

*Proof:* Using the same argument in Theorem 1,

$$\rho(G_{P,J}) \leq \frac{2r(\cos P\pi h + \cos \pi h)}{1 + 4r - \gamma\Delta t} + \frac{r}{1 + 4r - \gamma\Delta t} \max\{|\alpha h| + |\beta h|, |\alpha h| + 1\},$$

and hence

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

□

We note that the error using the approximation (12) for the Mixed problem is clearly greater than the error for the Dirichlet problem when  $h$  is very small. This phenomenon can be seen in Table 1.

## 4 Numerical results

### 4.1 Model problem 1 (MP1)

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

The initial condition is given by  $u(x, y, 0) = \sin x \cos y$ . Dirichlet, Neumann, and Mixed problems are artificially created so that the boundary conditions satisfy the exact value if the Dirichlet boundary condition is needed or the value of derivative is satisfied if the Neumann boundary condition is needed. The exact solution to the problems is  $u(x, y, t) = e^{-t} \sin x \cos y$ .

### 4.2 Model problem 2 (MP2)

$$u_t = u_{xx} + u_{yy} + 2u_x + u_y + 3u + f.$$

The initial condition is given by  $u(x, y, 0) = x \sin y$  and  $f = -\sin y - 2(t + x) \sin y - (t + x) \cos y$ . Dirichlet, Neumann, and Mixed problems are created in the same way as described in MP1. The exact solution to the problems is  $u(x, y, t) = (t + x) \sin y$ .

### 4.3 Optimum over-relaxation parameter

Optimum value of the over-relaxation parameter of the SOR method to the Dirichlet problem of MP2 at  $\Delta x = \Delta y = 0.01$  and  $\Delta t = 0.1$  with  $P = 10$  subdomains is computed by

$$\rho(G_P) = \left[ \frac{2r(\cos P\pi h + \cos \pi h)}{1 + 4r - \gamma\Delta T} \right]^2 = 0.95084$$

and

$$\omega = \frac{2}{1 + \sqrt{1 - \rho(G_P)}} = 1.637.$$

Fig. 1 is the graph of CPU time with respect to the over-relaxation parameter  $\omega$  to the Dirichlet problem of MP2 at  $\Delta x = \Delta y = 0.01$  and  $\Delta t = 0.1$  with

$P = 10$  subdomains. As we can see in Fig. 1,  $\omega = 1.637$  is a very good approximation to the optimum over-relaxation parameter of the SOR method. We note that when  $\omega = 1$ , the SOR method becomes the GS method.

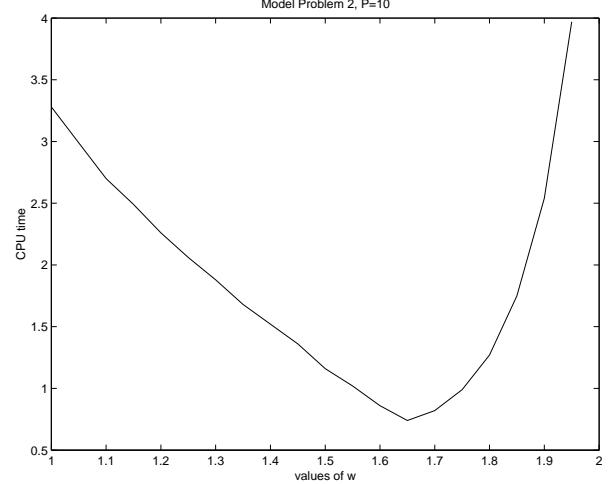


Fig. 1. CPU time for various  $\omega$

Table 1 shows the approximation to the optimum over-relaxation parameter of the SOR method for the model problems with  $\Delta x = \Delta y = 0.01$  and  $\Delta t = 0.1$ . For the Dirichlet and Mixed problems, the approximated spectral radii,  $\rho(G_P)_{app}$ , by Theorems 1 and 3 are compared with the actual ones,  $\rho(G_P)_{act}$ , obtained by using the Matlab [6]. As we can see, the approximated spectral radii are very closed to the actual ones, and hence the approximated optimum over-relaxation parameter  $\omega_{app}$  is very good approximation to the actual optimum parameter  $\omega_{opt}$ . For the Neumann problem, the exact values of the spectral radii by Theorem 2 are used to compute  $\omega_{opt}$ . Table 2 shows the maximum error and total CPU time of the SOR method to the model problems at the final time level  $t = 1$  with  $\Delta x = \Delta y = h = 0.01$  and  $\Delta t = 0.1$  using  $\omega_{app}$  for the Dirichlet and Mixed problems and  $\omega_{opt}$  for the Neumann problem. As we can see in Table 2, with relatively large value of  $\frac{\Delta t}{h^2} = 1000$ , the MIP method using the SOR is stable. More numerical experiments showed that the MIP using the SOR method is unconditionally stable to the Dirichlet, Neumann, and Mixed problems. The value of  $P = 1$  in Tables 1 and 2 indicates the BTCS method. We should point out that the approximated optimum over-relaxation parameters in this paper are very effective.

Table 1. Optimum over-relaxation parameter  
(D=Dirichlet,N=Neumann,M=Mixed,  
app=approximate,act=actual,opt=optimum)

MP		MP1			
$P$		1	2	10	25
D	$\rho(G_P)_{app}$	.99846	.99699	.95065	.72773
	$\omega_{app}$	1.9245	1.8960	1.6365	1.3142
	$\rho(G_P)_{act}$	.99846	.99699	.95065	.72773
	$\omega_{opt}$	1.9245	1.8960	1.6365	1.3142
N	$\rho(G_P)_{act}$	.99945			
	$\omega_{opt}$	1.9542			
M	$\rho(G_P)_{app}$	.99846	.99699	.95065	.72773
	$\omega_{app}$	1.9245	1.8960	1.6365	1.3142
	$\rho(G_P)_{act}$	.99896	.99748	.95113	.72815
	$\omega_{opt}$	1.9375	1.9044	1.6379	1.3146
MP		MP2			
$P$		1	2	10	25
D	$\rho(G_P)_{app}$	.99866	.99719	.95084	.72788
	$\omega_{app}$	1.9294	1.8993	1.6370	1.3144
	$\rho(G_P)_{act}$	.99860	.99712	.95078	.72784
	$\omega_{opt}$	1.9279	1.8981	1.6569	1.3143
N	$\rho(G_P)_{act}$	.99965			
	$\omega_{opt}$	1.9633			
M	$\rho(G_P)_{app}$	.99866	.99719	.95084	.72788
	$\omega_{app}$	1.9294	1.8993	1.6370	1.3144
	$\rho(G_P)_{act}$	.99911	.99763	.95128	.72827
	$\omega_{opt}$	1.9421	1.9072	1.6384	1.3147

Table 2. Maximum error and total CPU time of the SOR method

MP		MP1			
$P$		1	2	10	25
Dirichlet	Error	6e-4	7e-4	3e-3	3e-3
	CPU	8.8	6.2	1.8	0.9
Neumann	Error	7e-3	8e-3	2e-2	3e-2
	CPU	6.4	6.1	6.4	7.3
Mixed	Error	1e-3	1e-3	5e-3	5e-3
	CPU	4.9	3.0	0.5	0.2
MP		MP2			
$P$		1	2	10	25
Dirichlet	Error	4e-5	3e-5	1e-5	2e-6
	CPU	3.3	2.7	0.9	0.4
Neumann	Error	1e-4	6e-4	1e-3	2e-3
	CPU	6.8	6.8	6.7	7.6
Mixed	Error	2e-4	1e-4	1e-5	1e-5
	CPU	5.8	3.2	0.6	0.2

We now investigate the efficiency of the MIP algorithm with the SOR method for the model problems. A common measurement for the efficiency of a parallel algorithm is the speedup which is defined

by

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

Since the numerical experiment is simulated with one processor, the true parallel execution time using  $P$  processors is roughly equivalent to the total CPU time,  $T_P$ , obtained by the simulation, being divided by  $P$ . Thus the speedup  $S_P$  is simplified to  $S_P = \frac{T_1}{T_P/P}$ . Fig. 2 shows the graph of  $\log S_P$  with respect to  $P$  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. The algorithm to the Mixed problem is the most efficient, but the one to the Neumann problem is not as efficient as the one to the Dirichlet or Mixed problem. The reason is that the spectral radius reduces as the number of  $P$  increases at the Dirichlet and Mixed problems, however the spectral radius remains the same as the number of  $P$  increases at the Neumann problem. This phenomenon has been shown in Table 1.

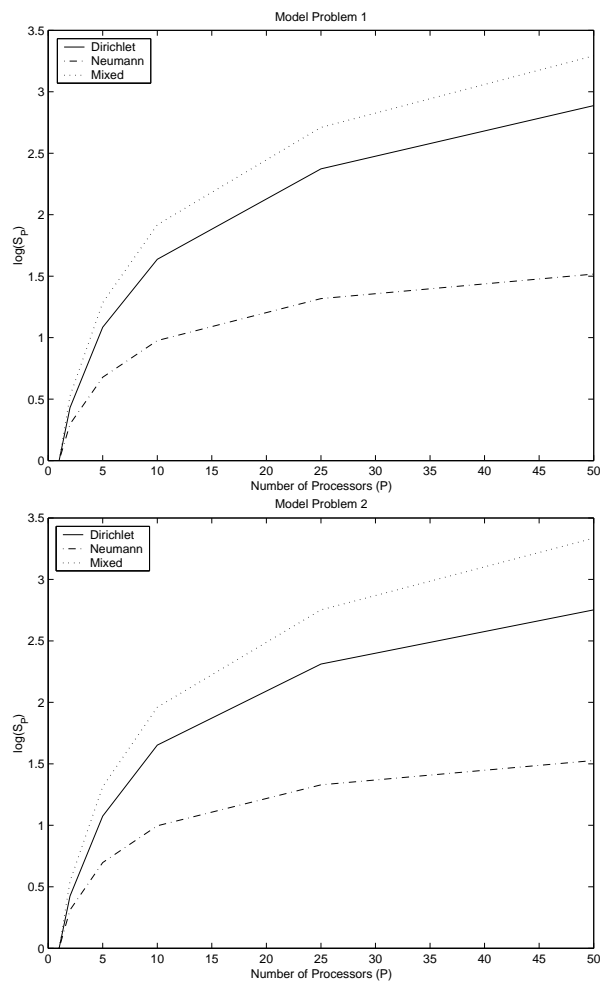


Fig. 2. Speedup curves for MP1 and MP2

## 5 Conclusion

In this paper we have demonstrated that the MIP domain decomposition algorithm with the SOR iterative method for solving parabolic problems with Dirichlet, Neumann, and Mixed boundary conditions is unconditionally stable. We have proposed the practical optimum over-relaxation parameter  $\omega_{app}$  of the SOR method, which has been proved very efficient. The speedup of the algorithm is much more than linear.

## REFERENCES

- [1] C.N. Dawson, Q. Du, and T.F. Dupont, "A finite difference domain decomposition algorithm for numerical solution of the heat equation", *Mathematics of Computation*, Vol 57, No. 195, pp. 63–71, 1991.
- [2] C.N. Dawson and T.F. Dupont, "Explicit/implicit, conservative domain decomposition procedures for parabolic problems based on block-centered finite differences", *SIAM J. Numer. Anal.*, Vol 31, pp. 1045–1061, 1994.
- [3] Y. Jun and T.Z. Mai, "IPIC domain decomposition algorithm for parabolic problems", to appear in *Appl. Math. Comput.*
- [4] Y. Jun and T.Z. Mai, "ADI method – Domain decomposition", to appear in *Appl. Numer. Math.*
- [5] Y. Jun and T.Z. Mai, "Domain decomposition method for parabolic problems with Neumann conditions", Preprint.
- [6] MATHWORKS, Inc. [http: www.mathworks.com](http://www.mathworks.com).
- [7] D.M. Young, *Iterative Solution of Large Linear Systems*, Dover, New York, 2003.
- [8] J. Zhu and H. Qian, "On an efficient parallel algorithm for solving time dependent partial differential equations", *PDPTA '98 International Conference*, pp. 394–401, 1998.
- [9] Y. Zhuang and X. Sun, "Stabilized explicit-implicit domain decomposition methods for the numerical solution of parabolic equations", *SIAM J. Sci. Comput.*, Vol 24, pp. 335–358, 2002.