

## EFFICIENT PARALLEL ALGORITHMS FOR PARABOLIC PROBLEMS\*

Q. DU<sup>†</sup>, M. MU<sup>†</sup>, AND Z. N. WU<sup>‡</sup>

**Abstract.** Domain decomposition algorithms for parallel numerical solution of parabolic equations are studied for steady state or slow unsteady computation. Implicit schemes are used in order to march with large time steps. Parallelization is realized by approximating interface values using explicit computation. Various techniques are examined, including a multistep second order explicit scheme and a one-step high-order scheme. We show that the resulting schemes are of second order global accuracy in space, and stable in the sense of Osher or in  $L_\infty$ . They are optimized with respect to the parallel efficiency.

**Key words.** parabolic equations, finite difference, parallel efficiency, stability, approximation accuracy

**AMS subject classifications.** 65N10, 65P05

**PII.** S0036142900381710

**1. Introduction.** Domain decomposition is a powerful tool for devising parallel PDE methods. There is rich literature on domain decomposition methods for both elliptic and time-dependent problems [2, 6]. We consider the linear parabolic problem in this paper. Explicit schemes are often naturally parallel and also easy to implement, but they usually require small time steps because of stability constraints. Implicit schemes are necessary for finding steady state solutions or computing slowly unsteady problems where one needs to march with large time steps; however, implicit schemes are not inherently parallel because at each time step essentially an elliptic type of problem needs to be solved.

A conventional approach of parallelizing the implicit schemes is to apply the elliptic-type domain decomposition based preconditioning methods to the problem arising from the semidiscretization at each time step. It is noted [1] that the resulting problem is well conditioned when the time step is small; nevertheless, small step size is not always desirable in situations where implicit schemes become necessary to use.

An alternative approach of parallelization is to take advantage of previous time steps since the problem considered here is time dependent; see [3, 7, 12, 13, 14] for related discussions. Kuznetsov [7] proposed a modified approximation scheme of mixed type, where the standard second order implicit scheme is used inside each subdomain, while the explicit Euler scheme is applied to obtain the interface values on the new time level. Once the interface values are available, the global problem is fully decoupled and can thus be computed in parallel. A similar hybrid scheme was proposed in [3], where instead of using the same spacing  $h$  as for the interior points where the implicit scheme is applied, a larger spacing  $H_D$  is used at each interface point where the explicit scheme is applied. Due to stability and accuracy requirements, both methods do not lead to satisfactory parallel efficiency as shown

---

\*Received by the editors November 29, 2000; accepted for publication (in revised form) June 26, 2001; published electronically December 26, 2001. This work was supported in part by the state major fund for basic research G1999032804, by NNSFC contract 10025210, and by HKRGC.

<http://www.siam.org/journals/sinum/39-5/38171.html>

<sup>†</sup>Department of Mathematics, Hong Kong University of Science and Technology, Kowloon, Hong Kong (madu@ust.hk, mamu@ust.hk).

<sup>‡</sup>Department of Eng. Mech., Tsinghua University, Beijing 100084, People's Republic of China (ziniuwu@tsinghua.edu.cn).

in our analysis, although these methods can be implemented with little extra effort using the original sequential codes.

In this paper we propose new parallel finite difference methods for parabolic PDEs. For simplicity of presentation, we focus on a model problem, namely, the one-dimensional heat equation in a spatial interval  $[0, 1]$ . The algorithms and detailed presentations are given in section 2. For computation on the subdomain interface, we use either a high-order explicit scheme or a multistep explicit scheme, with an intermediate mesh size  $H$  lying inside  $(h, H_D)$ . The stability and error analysis of our two new schemes are given in sections 3 and 4, respectively, using either a maximum principle-type argument or the stability analysis in the sense of Osher [11]. The parallel efficiency is addressed in section 5. Generalizations are considered in section 6, and some comments are given in section 7.

**2. Algorithm presentation.** Let  $u(x, t)$  be the solution of the model problem

$$(2.1) \quad \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad x \in (0, 1), \quad t \in (0, T),$$

together with the initial and boundary conditions

$$(2.2) \quad u(x, 0) = u^0(x), \quad x \in (0, 1),$$

$$(2.3) \quad u(0, t) = u_0(t), \quad t \in (0, T),$$

$$(2.4) \quad u(1, t) = u_1(t), \quad t \in (0, T).$$

The initial and boundary data satisfy the compatibility conditions  $u^0(0) = u_0(0)$  and  $u^0(1) = u_1(0)$ .

We decompose the domain  $(0, 1)$  into  $p$  subdomains. In general,  $p$  is related to the problem size and the number of processors in the computer platform, and the subdomains may be of different lengths. For illustration let us consider only the two-equal-subdomain case with  $p = 2$  at this moment. Assume further that the domain is discretized uniformly with spacing  $h = 1/N$ , where  $N$  is an even integer, although the case of nonuniform discretization may be considered in a similar way. Denote the grid points by  $x_j = \frac{1}{2} + jh$  so that the interface point corresponds to  $j = 0$  and the boundary points to  $j = -\frac{N}{2}$  and  $j = \frac{N}{2}$ .

For the problem of marching the solution of (2.1) toward time  $T$ , we denote  $\Delta t_s$  the time step required by the sequential implicit algorithm and let  $\Delta \tau$  denote the time step used by an explicit scheme.  $\Delta t_s$  is usually large in the situation where the implicit scheme needs to be used. Thus, if we use the standard explicit scheme in the whole domain, due to the stability constraint, more time steps are required to reach the final time  $T$  than using the implicit scheme, although the explicit scheme is perfectly parallel and simple to implement. Overall, the total computational efficiency is still low. Notice that this conclusion is reached using the following criteria for evaluating a parallel algorithm: we are interested in the total parallel time used by an algorithm to find the solution in  $(0, T)$ , instead of the speedup or parallel efficiency when compared with its own sequential version.

Let us start with the simplest case, where the implicit scheme is mixed with an explicit one as proposed by Kuznetsov in [7]. The idea is elegant and natural. To parallelize the computation of the implicit scheme on the new time level, one computes an approximate solution value within certain accuracy at the interface point by using the information computed in the previous level, which is easily done by applying an explicit scheme at the interface point. We note that the resulting finite difference

scheme is an approximation to the original sequential implicit scheme in the whole domain due to the modification at the interface. With this slight change, however, the implicit part can be perfectly parallelized. On the other hand, the explicit part, although introduced only at the interface, causes the limitation on the time step  $\Delta\tau$  due to the stability constraint  $\Delta\tau \leq \frac{1}{2}h^2$ .

To be able to carry out the implicit computation in every interval of the large step size  $O(\Delta t_s)$  while retaining the parallelism and stability as above, a straightforward extension is to apply the explicit scheme  $M$  steps with the step size  $O(\Delta\tau)$ , where  $M \Delta\tau = O(\Delta t_s)$ , until the interface value at the distance of  $O(\Delta t_s)$  is obtained. We note that the overhead due to decoupling the computation for parallelism at every  $O(\Delta t_s)$  step is of  $O(M^2)$  because there are as many points involved in the explicit evaluation. On the other hand, the explicit scheme introduces spatial truncation error at each substep. The accumulated error may affect the global spatial accuracy if  $\Delta t_s$  is too large so that the number of substeps  $M$  is too large. So the spatial accuracy requirement also imposes restriction on the practical number of substeps such that more implicit steps are required to march to the final time  $T$  than the sequential implicit scheme to be compared. We will show that the resulting parallel efficiency is poor because of the high parallel overhead.

To minimize the parallel overhead, Dawson, Du, and Dupont [3] noticed that the stability constraint on the time step  $\Delta\tau$  corresponding to the explicit scheme is caused by the small spacing size  $h$ . However, one does not have to apply the explicit scheme with the same spacing  $h$  as for the implicit scheme in order to approximate the interface value. A standard second order explicit scheme with a larger spacing  $H_D$  was applied only once to compute the new interface value [3]. The corresponding stability constraint now becomes  $\Delta\tau \leq \frac{1}{2}H_D^2$ . However, to make  $\Delta\tau$  comparable with  $\Delta t_s$ ,  $H_D$  could sometimes become too large to use, and the large spacing also affects the spatial approximation accuracy even if the explicit scheme is only local to the interface location. Readers are referred to [3] for the error analysis.

We now present some new algorithms in order to maximize the parallel efficiency and retain the approximation accuracy and stability. They combine the advantages of the above methods. First, we use the second order explicit scheme with an intermediate spacing  $H \in (h, H_D)$  and time step  $\Delta\tau$  which satisfies the stability condition:

$$(2.5) \quad \Delta\tau \leq \frac{1}{2}H^2.$$

Meanwhile, we also apply the explicit scheme for  $m$  substeps with the step size  $\Delta\tau$ , where  $m$  could be taken as large as  $O(\Delta t_s/\Delta\tau)$ . The question is how to choose proper parameters  $H$  and  $m$ . For this purpose, we will present the error analysis to show that the global spatial approximation depends on  $H$ ,  $m$ , and  $p$ . Based on this, we will present the parallel performance analysis for the optimal choice for  $H$  and  $m$  in order to optimize the parallel efficiency while retaining the global accuracy and stability. For simplicity, we assume  $H = M_H h$ , where  $M_H$  is an integer.

Denote  $\Delta t = m \Delta\tau$  and  $t^n = n \Delta t$ ,  $\tau^k = k \Delta\tau$ . Figure 2.1 depicts the grid points involved in a typical time step of the algorithm, where “ $\times$ ” points correspond to where the explicit scheme is applied, “ $\circ$ ” points correspond to where the implicit scheme is applied, and “ $\bullet$ ” points correspond to where the boundary conditions (2.2) and (2.3) are applied. Denote the set of all the “ $\times$ ” points and the set of “ $\circ$ ” points in the whole computational domain by  $\Omega_h^E$  and  $\Omega_h^I$ , respectively. Let  $\Omega_h = \Omega_h^E \cup \Omega_h^I$ . Also denote the set of “ $\bullet$ ” points, including those on  $t = 0$ , by  $\partial\Omega_h$ . Corresponding to the grid points in  $\Omega_h^E$ , denote  $X_J = \frac{1}{2} + JH$  for  $-m+1 \leq J \leq m-1$ . For a function  $f = f(x, t)$

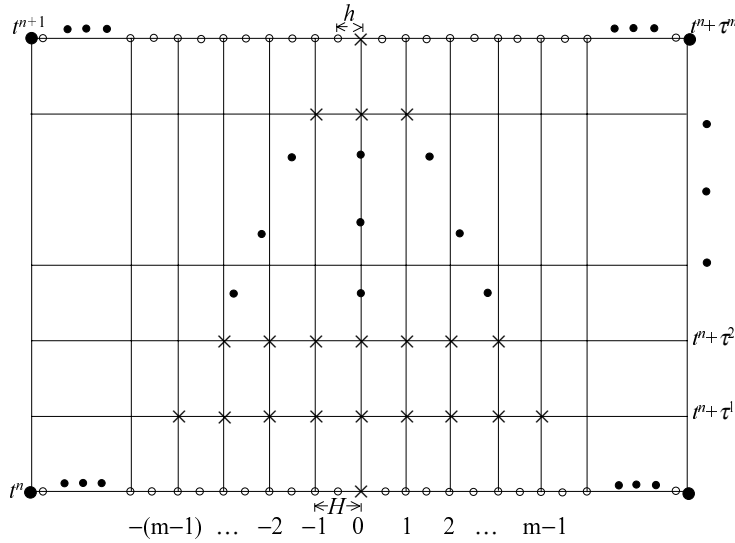


FIG. 2.1. The composite grid  $\Omega_h$ , consisting of the implicit and explicit grid points.

defined at mesh points  $(x_j, t^n)$ , let  $f_j^n = f(x_j, t^n)$ . Similarly, for a function  $f = f(x, t)$  defined at mesh points  $(X_J, t^n + \tau^k)$ , let  $f_J^{n,k} = f(X_J, t^n + \tau^k)$ . The numerical solution to  $u_J^{n,k}$  (or  $u_j^n$ ) will be denoted by  $U_J^{n,k}$  (or  $U_j^n$ ). For convenience, we denote the solution vector at time  $t^{n+1} = t^n + m\Delta\tau$  by  $U^{n+1} = U^{n,m}$  and  $U^{n,0} = U^n$ . Given function  $f$ , we use the notation

$$\partial_{t,\Delta\tau} f(x, t) = \frac{f(x, t) - f(x, t - \Delta\tau)}{\Delta\tau},$$

$$\partial_{x,H}^2 f(x, t) = \frac{f(x - H, t) - 2f(x, t) + f(x + H, t)}{H^2},$$

and similarly for  $\partial_{t,\Delta t}$  and  $\partial_{x,h}^2$ . Our algorithm is now described in detail as follows.

Method 1.

$$(2.6) \quad U_j^{n+1} = u_j^{n+1} \quad (\text{at boundary points}),$$

$$(2.7) \quad \partial_{t,\Delta\tau} U_J^{n,k} = \partial_{x,H}^2 U_J^{n,k-1}, \quad 1 \leq k \leq m, \quad -m + k \leq J \leq m - k \quad (\text{on } \Omega_h^E),$$

$$(2.8) \quad \partial_{t,\Delta t} U_j^{n+1} = \partial_{x,h}^2 U_j^{n+1} \quad (\text{on } \Omega_h^I).$$

The problem is solved as follows. For a typical time step, starting with the solution at time level  $n$ , we know the solutions  $U^{n,0} = U^n$ . Solving (2.7) for  $k = 1, \dots, m$ , we obtain  $U_0^{n,m}$ . For each substep  $k$ , the number of space points involved is  $(2m - 2k + 1)$  so that at  $k = m$ , only one point is involved. Then we put  $U_0^{n+1} = U_0^{n,m}$  at the interface and use (2.6) to solve (2.8) for the interior points. Note that  $H = h$  and  $H = H_D$  correspond to the two special cases in [7] and [3].

As discussed earlier, the choice of  $H$  in the algorithm depends on the spatial approximation accuracy although  $H \gg h$ . In case such an  $H$  is still not large enough

so that a large number of substeps is required in (2.7), the parallel efficiency may still be low. To allow for an even larger spacing  $H$ , we now propose an alternative approach to improve the efficiency. The idea is to increase the spatial approximation accuracy in the explicit scheme by extrapolation.

*Method 2.*

$$(2.9) \quad U_j^{n+1} = u_j^{n+1} \quad (\text{at boundary points}),$$

$$(2.10) \quad \partial_{t,\Delta\tau} U_0^{n+1} = \alpha \partial_{x,H}^2 U_0^n + \gamma \partial_{x,2H}^2 U_0^n \quad (\text{at interface point}),$$

$$(2.11) \quad \partial_{t,\Delta\tau} U_j^{n+1} = \partial_{x,h}^2 U_j^{n+1} \quad (\text{at interior points}),$$

where  $\alpha = \frac{4}{3}$ ,  $\gamma = -\frac{1}{12}$  so that the explicit scheme is of fourth order in space. We note that if  $\Delta\tau$  determined by such an  $H$  is still not large enough compared with  $\Delta t_s$ , multiple substeps could also be applied similar to Method 1. One can expect that the number of substeps should be much smaller than that in Method 1.

We will present a maximum principle analysis for the stability and approximation for Method 1 in section 3. Because the explicit scheme used in Method 2 is of order 4, the maximum principle does not hold there. So we present in section 4 the error analysis for Method 2 based on the normal mode analysis and show that the method is stable in the sense of Osher. The parallel efficiency analysis for all the methods will be given in section 5.

**3. Maximum principle analysis for Method 1.** Define the finite difference operator at a point  $P = (x, t)$  on the composite grid:

$$L_h u_P = \begin{cases} \partial_{t,\Delta\tau} u(x, t) - \partial_{x,H}^2 u(x, t - \Delta\tau) & \forall P \in \Omega_h^E, \\ \partial_{t,\Delta t} u(x, t) - \partial_{x,h}^2 u(x, t) & \forall P \in \Omega_h^I. \end{cases}$$

We have the following maximum principle.

LEMMA 3.1. *Under the stability condition (2.5), the inequality for a grid function  $U$ ,*

$$(3.1) \quad L_h U_P \leq 0 \quad \forall P \in \Omega_h,$$

*implies*

$$(3.2) \quad \max_{P \in \Omega_h} U_P \leq \max \left\{ \max_{P \in \partial\Omega_h} U_P, 0 \right\}.$$

The proof of the lemma is standard (see, for example, Lemma 6.1 in [10].) The main task of the proof is to verify the positivity of the coefficients appearing in the linear operator  $L_h$ , which is ensured by the stability condition (2.5) for the explicit part and otherwise by the implicit part itself. With this maximum principle, we have the error analysis for Method 1.

THEOREM 3.2. *Under the assumption (2.5) there exists a constant  $C$  independent of  $\Delta t$ ,  $h$ ,  $H$ , and  $m$  such that*

$$(3.3) \quad |u_P - U_P| \leq C(\Delta t + h^2 + mH^3) \quad \forall P \in \Omega_h,$$

*where  $u_P$  and  $U_P$  are the true solution of the PDE (2.1)–(2.4) and the approximation generated by Method 1 at  $P \in \Omega_h$ , respectively.*

*Proof.* The proof follows from standard argument based on the discrete maximum principle [3]. We include it here for completeness. Let us define a comparison function corresponding to the implicit part of the scheme as usual:

$$(3.4) \quad \psi^I(x) = \frac{1}{2}x(1-x).$$

For the explicit part, for each “ $\times$ ” point  $X_J, J = 0, \pm 1, \dots, \pm(m-1)$ , on the first level  $t^n + \tau^1$ , we also define a comparison function:

$$(3.5) \quad \psi^J(x) = \begin{cases} Hx(1-X_J), & x \leq X_J, \\ H(1-x)X_J, & x \geq X_J. \end{cases}$$

We can verify that for a grid point  $P = (x, t) \in \Omega_h$ , for  $\psi^I(x)$ ,

$$(3.6) \quad L_h\psi_P^I = 1 \quad \forall P \in \Omega_h;$$

and for  $\psi^J(x)$ , if  $x \neq X_J$ ,

$$(3.7) \quad L_h\psi_P^J = 0,$$

and if  $x = X_J$ , then

$$(3.8) \quad L_h\psi_P^J = \begin{cases} 1 & \text{if } P \in \Omega_h^E, \\ \frac{H}{h} & \text{if } P \in \Omega_h^I. \end{cases}$$

We now define a comparison function corresponding to the explicit part of the scheme as

$$(3.9) \quad \psi^E(x) = \sum_{|J| \leq m-1} \psi^J(x).$$

From (3.7) and (3.8), we have,  $\forall P \in \Omega_h^E$ ,

$$(3.10) \quad L_h\psi_P^E = 1,$$

and if  $P = (x, t) \in \Omega_h^I$ , then

$$(3.11) \quad L_h\psi_P^E = \begin{cases} \frac{H}{h} & \text{if } x = X_J, \text{ for some } |J| \leq m-1, \\ 0 & \text{otherwise.} \end{cases}$$

Let  $e_P = u_P - U_P$  be the error at  $P \in \Omega_h$ . From the standard truncation error analysis, we have

$$(3.12) \quad L_h e_P = \begin{cases} C_P^1 \Delta\tau + C_P^2 H^2 & \forall P \in \Omega_h^E, \\ C_P^1 \Delta t + C_P^2 h^2 & \forall P \in \Omega_h^I. \end{cases}$$

Denote  $C_0 = \max\{\max_{P \in \Omega_h} |C_P^1|, \max_{P \in \Omega_h} |C_P^2|\}$ . Let us now define a comparison function counting for both the explicit and the implicit parts:

$$(3.13) \quad \Phi_P = C_0 [\psi_P^I(\Delta t + h^2) + \psi_P^E(\Delta\tau + H^2)].$$

From (3.6) and (3.10)–(3.12), we have

$$(3.14) \quad L_h(e_P - \Phi_P) \leq 0 \quad \forall P \in \Omega_h.$$

Also note that  $\Phi_P$  is a nonnegative function and

$$(3.15) \quad e_P = 0 \quad \forall P \in \partial\Omega_h.$$

So,

$$(3.16) \quad e_P - \Phi_P \leq 0 \quad \forall P \in \partial\Omega_h.$$

Applying Lemma 3.1, we obtain

$$(3.17) \quad \begin{aligned} \max_{P \in \Omega_h} (e_P - \Phi_P) &\leq \max \left\{ \max_{P \in \partial\Omega_h} (e_P - \Phi_P), 0 \right\} \\ &= 0, \end{aligned}$$

which implies

$$(3.18) \quad e_P \leq \Phi_P \quad \forall P \in \Omega_h.$$

Replacing  $e_P$  by  $-e_P$  in the above argument leads to

$$(3.19) \quad |e_P| \leq \Phi_P \quad \forall P \in \Omega_h.$$

Thus

$$(3.20) \quad \begin{aligned} |e_P| &\leq \max_{P \in \Omega_h} \Phi_P \\ &= C_0 \left[ (\Delta t + h^2) \max_{P \in \Omega_h} \psi_P^I + (\Delta \tau + H^2) \max_{P \in \Omega_h} \psi_P^E \right] \\ &= C_0 \left[ \frac{1}{8}(\Delta t + h^2) + (\Delta \tau + H^2) \sum_{|J| \leq m-1} \max_{P \in \Omega_h} \psi_P^J \right] \\ &= C_0 \left[ \frac{1}{8}(\Delta t + h^2) + (\Delta \tau + H^2) \sum_{|J| \leq m-1} \frac{H}{4} \right] \\ &= C_0 \left[ \left( \frac{1}{8} + \frac{2m+1}{4m} H \right) \Delta t + \frac{1}{8} h^2 + \frac{2m+1}{4} H^3 \right], \end{aligned}$$

which implies (3.3). This completes the proof.  $\square$

The analysis can be extended to the general case of  $p$  subdomains ( $p \geq 2$ ).

**COROLLARY 3.3.** *Under the assumption (2.5), there exists a constant  $C$  independent of  $\Delta t$ ,  $h$ ,  $H$ ,  $p$ , and  $m$  such that*

$$(3.21) \quad |u_P - U_P| \leq C(\Delta t + h^2 + mpH^3) \quad \forall P \in \Omega_h,$$

where  $u_P$  and  $U_P$  are the true solution of the PDE (2.1)–(2.4) and the approximation generated by Method 1 at  $P \in \Omega_h$ , respectively.

**4. Stability and error analysis for Method 2 based on the normal mode analysis.** We use the normal mode analysis of Osher [11] to establish the stability. For convenience, let the solution in the left subdomain be  $u_j^n$  with  $j < 0$ , and the solution at the right subdomain be  $v_j^n$  with  $j > 0$ . Method 2 reduces to

$$(4.1) \quad u_j^{n+1} - \sigma(u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}) = u_j^n, \quad j < 0,$$

$$(4.2) \quad v_j^{n+1} - \sigma(v_{j-1}^{n+1} - 2v_j^{n+1} + v_{j+1}^{n+1}) = v_j^n, \quad j > 0,$$

with

$$(4.3) \quad u_0^{n+1} = u_0^n + \alpha\sigma(u_{-M}^n - 2u_0^n + v_M^n) + \gamma\sigma(u_{-2M}^n - 2u_0^n + v_{2M}^n),$$

$$(4.4) \quad v_0^{n+1} = u_0^{n+1},$$

with  $\alpha = \frac{4}{3}$  and  $\gamma = -\frac{1}{12}$  for fourth order accuracy.

Consider the normal mode representation of the solution

$$(4.5) \quad u_j^n = u_0 z^n k^j.$$

Inserting (4.5) into (4.1) yields

$$[z(1 - \sigma(k^{-1} - 2 + k)) - 1]u_0 = 0.$$

Nontrivial solution exists only for

$$(4.6) \quad z(1 - \sigma(k^{-1} - 2 + k)) - 1 = 0.$$

Equation (4.6) is called the characteristic equation. For  $|z| \geq 1$  and  $z \neq 1$ , the characteristic equation has two roots  $k_1$  and  $k_2$  satisfying

$$\begin{aligned} |k_1| < 1 \text{ and } |k_2| > 1, \\ k_1 &= \frac{1}{k_2}. \end{aligned}$$

For convenience, denote

$$k_1 = \frac{1}{k_2} = \tau$$

so that  $|\tau| < 1$ . Obviously, for  $|z| \geq 1$  and  $z \neq 1$ , the solution belonging to  $l_2$  is

$$(4.7) \quad u_j^n = u_0 z^n k_2^j = u_0 z^n \tau^j.$$

Similarly, and noting that we have used the same scheme for both  $u$  and  $v$ , the normal mode solution in  $l_2$  for  $v_j^n$  is given by

$$(4.8) \quad v_j^n = v_0 z^n k_1^j = v_0 z^n \tau^{-j}.$$

Now inserting (4.7) and (4.8) into the interface conditions (4.3) and (4.4), and noting that  $u_0 = v_0$  by (4.4), we have

$$[z - 1 - \alpha\sigma(\tau^M - 2 + \tau^M) - \gamma\sigma(\tau^{2M} - 2 + \tau^{2M})]u_0 = 0$$

so that nontrivial solution exists for  $|z| \geq 1$  and  $z \neq 1$  only if

$$\begin{aligned} z &= 1 + \alpha\sigma(\tau^M - 2 + \tau^M) + \gamma\sigma(\tau^{2M} - 2 + \tau^{2M}) \\ &= 1 - 2\alpha\sigma(1 - \tau^M) - 2\gamma\sigma(1 - \tau^{2M}) \\ &= 1 - 2\sigma(1 - \tau^M)[\alpha + \gamma(1 + \tau^M)] \\ (4.9) \quad &= 1 - 2\sigma\alpha(1 - \tau^M) \left[ 1 - \frac{|\gamma|}{\alpha}(1 + \tau^M) \right]. \end{aligned}$$



The relation (4.9) is also called the root condition. If we have

$$(4.10) \quad z \neq 1 - 2\sigma\alpha(1 - \tau^M) \left[ 1 - \frac{|\gamma|}{\alpha}(1 + \tau^M) \right]$$

for  $|z| \geq 1$  and  $z \neq 1$ , then it means that there is no (nontrivial) solution with  $|z| \geq 1$  and  $z \neq 1$ , and the problem is stable. Now the left-hand side of (4.10) has a module no smaller than 1 for  $|z| \geq 1$ , it is thus sufficient to prove that its right-hand side has a module smaller than 1, i.e.,

$$(4.11) \quad \left| 1 - 2\sigma\alpha(1 - \tau^M) \left[ 1 - \frac{|\gamma|}{\alpha}(1 + \tau^M) \right] \right| < 1.$$

Let  $\tau^M = re^{i\theta}$ ,  $r < 1$ , and  $w = (1 - re^{i\theta})(15 - re^{i\theta})$ ; then  $z = 1 - \frac{\sigma}{6}w$ , and we want to show that under appropriate conditions on  $\sigma$  we have  $|z| \leq 1$ , thus leading to a contradiction.

This is achieved if we can find a lower bound for

$$\bar{m} = \min \frac{\operatorname{Re}\{w\}}{|w|^2}.$$

In fact, we need  $\sigma \leq 12\bar{m}$  for stability.

Direct calculation shows

$$\operatorname{Re}\{w\} = 15 - r^2 - 16r \cos(\theta) + 2r^2 \cos^2(\theta),$$

$$\operatorname{Im}\{w\} = 2r \sin(\theta)(r \cos(\theta) - 8),$$

$$|w|^2 = (1 + r^2)(225 + r^2) - (480 + 32r^2)r \cos(\theta) + 60r^2 \cos^2(\theta).$$

It is easily seen that

$$\frac{\operatorname{Re}\{w\}}{|w|^2} = \frac{14}{452} < \frac{1}{32}$$

for  $\cos(\theta) = 0$  and  $r = 1$ . Thus,  $\bar{m} < 1/32$ .

On the other hand, we have

$$\begin{aligned} 33\operatorname{Re}\{w\} - |w|^2 &= 495 - 33r^2 - 528r \cos(\theta) + 66r^2 \cos^2(\theta) \\ &\quad - (1 + r^2)(225 + r^2) + (480 + 32r^2)r \cos(\theta) - 60r^2 \cos^2(\theta) \\ &= 270 - 259r^2 - r^4 + (32r^2 - 48)r \cos(\theta) + 6r^2 \cos^2(\theta). \end{aligned}$$

Differentiating the function  $g(x, y) = 270 - 259y - y^2 + (32y - 48)x + 6x^2$  with respect to  $x, y$ , we get

$$\frac{\partial g}{\partial x} = 32y - 36, \quad \frac{\partial g}{\partial y} = -259 - 2y + 32x.$$

We see that for  $0 \leq y \leq 1$  and  $-1 \leq x \leq 1$ , the partial derivatives are all negative; thus,  $g(x, y) \geq g(1, 1) = 0$ .

This implies

$$\bar{m} = \min \frac{\operatorname{Re}\{w\}}{|w|^2} \geq \frac{1}{33}.$$

Thus, we have the stability under the condition that  $0 < \sigma \leq 12/33 = 4/11$ .

**THEOREM 4.1.** *Method 2 is stable in the sense of Osher for  $0 < \sigma \leq 4/11$ .*

Following Gustafsson [4], Theorem 4.1 leads to the following estimate (see also [5, 8, 9]).

**THEOREM 4.2.** *For Method 2, under the stability condition  $0 < \sigma \leq 4/11$ , there exists a constant  $C$  independent of  $\Delta t$ , and  $h$  such that*

$$(4.12) \quad |u_P - U_P| \leq C(\Delta t + h^2 + H^5) \quad \forall P \in \Omega_h,$$

where  $u_P$  and  $U_P$  are the true solution of the PDE (2.1)–(2.4) and the approximation generated by Method 2 at  $P \in \Omega_h$ , respectively.

**5. Parallel efficiency analysis.** Assume that there are  $p$  processors and the whole domain  $(0, 1)$  is uniformly decomposed into  $p$  subdomains with  $p - 1$  interface points. Let  $N$  be the total number of grid points, then each subdomain has  $O(\frac{N}{p})$  grid points. We assume that  $N \gg p \geq 2$  so that the parallel algorithms are of coarse granularity.

There are different measurements to assess parallel algorithms. It is sometimes misleading to use the so-called *relative* speedup or parallel efficiency by comparing a parallel algorithm with its own version running on a single processor. In that sense, an explicit scheme always has the maximal speedup and efficiency. In our context, we assume that for a computational experiment (either for computing the solution for large time or for slow unsteady solution), it is already known that the best sequential algorithm (namely, the one that uses the least time to reach the time  $T$ ) is an implicit scheme of the cost  $t_s^I$  per time step with a large step size  $\Delta t_s$ . Our goal is to find a parallel algorithm which, with a given number of  $p$  processor, takes as little time as possible to reach the final time  $T$  without degrading the order of accuracy.

The per step computational cost  $t_s^I$  depends on which implicit solver is applied. In general, we may have

$$t_s^I = O(C^I N^\alpha \log^\beta N)$$

for certain exponents  $\alpha$ ,  $\beta$ , and constant  $C^I$ , where  $N$  is the size of the problem to which the implicit solver is applied. For the one-dimensional case, the Thomas algorithm is applicable to the tridiagonal matrix problem. Thus the implicit solver has the CPU time complexity

$$(5.1) \quad t_s^I = C^I \cdot N;$$

i.e., we have  $\alpha = 1$  and  $\beta = 0$ . However, the situation becomes more complicated for two-dimensional and three-dimensional problems, depending on the algorithm used in the implicit solver. The available algorithms include band solver, nested dissection, SOR, CG, multigrid, or a general sparse solver. The time complexity varies, of course, and has a different order of magnitude. The best case could be  $t_s^I = O(N)$  if the problem is so nice that, for instance, an effective preconditioner can be constructed for all time steps such that the condition number of the preconditioned problem is independent of the grid size. However, this is not always true in practice. In the rest

of this section, we consider the one-dimensional case. We will extend the analysis to higher-dimensional cases in section 6.

Now, consider Method 1 where the number of “explicit points” involved in a typical step is

$$(5.2) \quad \sum_{l=1}^m (2l - 1) = m^2.$$

Ignoring the communication cost, the parallel time of Method 1 from  $t^n$  to  $t^{n+1}$  is

$$(5.3) \quad t_p = C^E m^2 + C^I \frac{N}{p}.$$

The total parallel run time to reach the final time  $T$  is

$$(5.4) \quad T_p = \left( \frac{T}{\Delta t} \right) t_p.$$

Note that there are two parameters  $m$  and  $H$  to be determined subject to the accuracy condition

$$(5.5) \quad O(mpH^3) = O(h^2)$$

and the stability condition

$$(5.6) \quad \Delta\tau \leq \frac{1}{2}H^2.$$

The speedup relative to the sequential implicit scheme is defined as

$$(5.7) \quad \begin{aligned} S &= \frac{T_s}{T_p} \\ &= \frac{(T/\Delta t_s) \cdot t_s^I}{(T/\Delta t) \cdot t_p} \\ &= \frac{m\Delta\tau}{\Delta t_s} \left[ \frac{C^I N}{C^E m^2 + C^I N/p} \right] \\ &= \frac{mH^2}{2\Delta t_s} \frac{C^I p}{(C^E m^2 hp + C^I)}. \end{aligned}$$

Or the parallel efficiency is given by

$$(5.8) \quad \begin{aligned} E &= \frac{S}{p} \\ &= \frac{C^I}{C^I + C^E m^2 hp} \cdot \frac{mH^2}{2\Delta t_s}. \end{aligned}$$

Let us study different cases as follows.

*Case I.*  $H = h, m = O(\frac{\Delta t_s}{\Delta\tau})$ . This corresponds to one extreme case where one tries to directly parallelize the best sequential implicit scheme by applying Kuznetsov’s technique with enough explicit substeps. Note that the stability restriction (5.6) now reads as

$$\Delta\tau \leq \frac{1}{2}h^2,$$

which implies a large number of substeps

$$m = O\left(\frac{2\Delta t_s}{h^2}\right).$$

On the other hand, the accuracy condition (5.5) reads as

$$m = O\left(\frac{1}{ph}\right).$$

This leads to

$$(5.9) \quad m = \min\left\{O\left(\frac{1}{ph}\right), O\left(\frac{2\Delta t_s}{h^2}\right)\right\},$$

which depends on  $\Delta t_s$ ,  $p$ , and  $h$ . Note that  $O(\frac{1}{ph}) \leq O(\frac{2\Delta t_s}{h^2})$  implies

$$(5.10) \quad \Delta t_s \geq O\left(\frac{h}{2p}\right),$$

which usually holds when an implicit method is applied. So we have

$$\begin{aligned} m &= O\left(\frac{1}{ph}\right) \\ &= C\frac{N}{p}. \end{aligned}$$

Equation (5.10) implies that the implicit scheme cannot be applied with the full step size  $\Delta t_s$  due to the accuracy constraint (5.5) because the actual time step of the implicit scheme in this mixed algorithm is only  $m\Delta\tau = O(\frac{h}{2p}) \leq \Delta t_s$ . In other words, the explicit scheme introduces spatial truncation error at each substep. The accumulated error will degrade the global spatial accuracy if  $\Delta t_s$  is too large so that the number of substeps is too large. So the spatial accuracy requirement also imposes restriction on the practical number of substeps such that more implicit steps are required to march to the final time  $T$  than the sequential implicit scheme to be compared.

We now have the speedup relative to the best sequential algorithm:

$$\begin{aligned} (5.11) \quad S_1 &= \frac{mh^2}{2\Delta t_s} \cdot \frac{C^I p}{(C^E m^2 h p + C^I)} \\ &= \frac{CC^I h^2 \cdot p}{2\Delta t_s (C^E \cdot C^2 + C^I p h)} \\ &= O\left(\frac{C^I}{C^E} \Delta t_s^{-1} h^2 p\right) \\ &\leq O\left(\frac{C^I}{C^E} h p^2\right). \end{aligned}$$

Thus,

$$(5.12) \quad \begin{aligned} E &= \frac{S}{p} \\ &\leq O\left(\frac{C^I}{C^E} p/N\right). \end{aligned}$$

We have two observations from this analysis. First, the parallel efficiency of the algorithm is very poor as  $N \gg p$ , which is caused by the severe stability condition that leads to small  $\Delta\tau$  (or equivalently, large  $m$ ) so that much more extra overhead is introduced in the explicit substep in order to decouple the implicit computation. This shows why we would like to make  $H$  as large as possible. On the other hand, the situation could be improved in higher-dimensional cases where the explicit cost is of lower order than the implicit one. In fact, it is true that the faster the implicit solver is, the more difficult it is to parallelize it by using certain explicit computation.

*Case II.*  $H = H_{\max}$ ,  $m = 1$ . This corresponds to another extreme case as in [3], where only one explicit step is used but  $H$  must be as large as possible. The accuracy condition (5.5) with  $m = 1$  implies

$$(5.13) \quad H = O\left(\left(\frac{h^2}{p}\right)^{1/3}\right) = C\left(\frac{h^2}{p}\right)^{1/3}.$$

Note that  $H \gg h$  because  $N \gg p$ . The stability condition (5.6) now reads as

$$(5.14) \quad \Delta\tau = \frac{C^2}{2} \left(\frac{h^2}{p}\right)^{2/3}.$$

Correspondingly, the speedup is

$$(5.15) \quad \begin{aligned} S_2 &= \frac{\Delta\tau}{\Delta t_s} \cdot \frac{C^I N}{C^E + C^I N/p} \\ &= \frac{C^2}{2\Delta t_s} \cdot \left(\frac{h^2}{p}\right)^{2/3} \cdot \frac{C^I p}{C^I + C^E \cdot hp}. \end{aligned}$$

Comparing the two cases from (5.11) and (5.15), we have

$$(5.16) \quad \frac{S_2}{S_1} = O\left(\frac{C^E}{C^I} \left(\frac{N}{p}\right)^{2/3}\right).$$

We observe from (5.16) that Case II is more efficient than Case I because  $N \gg p$ . On the other hand, Case II is still far less efficient because

$$(5.17) \quad \begin{aligned} E_2 &= \frac{S_2}{p} \\ &= O\left(\Delta t_s^{-1} \cdot \left(\frac{h^2}{p}\right)^{2/3}\right) \\ &\ll 1 \end{aligned}$$

for large  $\Delta t_s$ ,  $p$ , and small  $h$ . This is caused by the fact that  $\Delta\tau \ll \Delta t_s$  from (5.14).

*Case III.*  $H = H_{opt}$ ,  $m = m_{opt}$ . Now let us consider the case where we vary both parameters  $H$  and  $m$  in order to optimize the algorithm in terms of parallel run time. From stability, we have

$$(5.18) \quad \begin{aligned} T_p &= \frac{T}{\Delta t} \cdot t_p \\ &= \frac{T}{m\Delta\tau} \left(C^E m^2 + C^I \frac{N}{p}\right) \\ &= \frac{2T}{mH^2} \left(C^E m^2 + C^I \frac{N}{p}\right). \end{aligned}$$

The accuracy condition requires

$$H^3 = \frac{Ch^2}{mp}.$$

So,

$$(5.19) \quad T_p = \frac{2T}{m \cdot \left(\frac{Ch^2}{mp}\right)^{2/3}} \left( C^E m^2 + C^I \frac{N}{p} \right) = \frac{2Tp^{2/3}}{C^{2/3}h^{4/3}} \left( C^E m^{5/3} + C^I \frac{N}{p} m^{1/3} \right).$$

To minimize  $T_p$  with respect to  $m$ , we solve

$$\frac{\partial T_p}{\partial m} = 0,$$

which yields

$$(5.20) \quad 5C^E m^2 = C^I \frac{N}{p},$$

namely,

$$(5.21) \quad m_{opt} = \sqrt{\frac{C^I N}{5C^E p}}.$$

Correspondingly, we have

$$(5.22) \quad \begin{aligned} T_p^{opt} &= \frac{2Tp^{2/3}}{C^{2/3}h^{4/3} \cdot m_{opt}^{1/3}} \left( C^E m_{opt}^2 + C^I \frac{N}{p} \right) \\ &= \frac{2Tp^{2/3}}{C^{2/3}h^{4/3} \cdot \left(\frac{C^I N}{5C^E p}\right)^{1/6}} \left( \frac{C^I N}{5p} + C^I \frac{N}{p} \right). \end{aligned}$$

Therefore,

$$(5.23) \quad \begin{aligned} S_{opt} &= \frac{\left(\frac{T}{\Delta t_s}\right) C^I N}{T_p^{opt}} \\ &= O\left(\left(\frac{C^I}{5C^E}\right)^{1/6} \Delta t_s^{-1} \cdot h^{7/6} \cdot p^{1/6}\right). \end{aligned}$$

Comparing (5.22) with (5.15), we have

$$(5.24) \quad \frac{S_{opt}}{S_2} = O\left(\left(\frac{C^I}{5C^E}\right)^{1/6} \cdot \left(\frac{1}{hp}\right)^{1/6}\right).$$

Therefore,  $S_{opt} > S_2$  because  $N \gg p$ . Furthermore, the improvement would become more substantial as the cost in the implicit part increases relative to the explicit part like in the two-dimensional and three-dimensional cases, as shown in section 6.

So far, in our parallel efficiency analysis we have compared the parallel algorithms with the best sequential algorithm which is the implicit scheme. It is also of interest to compare them with the explicit scheme which is perfectly parallel. The run time for the parallel explicit scheme is

$$(5.25) \quad T_p^E = \frac{2T}{h^2} \cdot C^E \frac{N}{p}.$$

Comparing with Case III, for instance, from (5.22) we have

$$\begin{aligned} \gamma &= \frac{T_p^{opt}}{T_p^E} \\ &= O\left(\frac{C^I}{C^E} \frac{p}{N}\right)^{5/6}. \end{aligned}$$

We see that  $T_p^{opt} \ll T_p^E$  because  $N \gg p$ .

Similar analysis can be carried out for Method 2, where the accuracy condition (5.5) will be replaced by

$$(5.26) \quad O(mpH^5) = O(h^2).$$

**6. Extension to two and three dimensions.**

**6.1. Presentation of the method.** We now discuss the higher-dimensional generalization. Let us consider the two-dimensional heat equation on a unit square,

$$(6.1) \quad \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \quad x, y \in (0, 1), \quad t \in (0, T),$$

together with the initial and boundary conditions

$$(6.2) \quad u(x, y, 0) = u^0(x, y), \quad x, y \in (0, 1),$$

$$(6.3) \quad u(x, y, t) = 0, \quad x = 0, 1 \text{ or } y = 0, 1, t \in (0, T).$$

For illustration, we decompose the spatial domain into two subdomains, separated by the line  $x = \frac{1}{2}$ , although it is similar to  $p (\geq 2)$  subdomains. Given Cartesian grid points  $\{x_i, y_j\}$ , Method 1 can be extended to

$$(6.4) \quad U_{i,j}^{n+1} = u_{i,j}^{n+1} \quad (\text{at boundary points}),$$

$$(6.5) \quad \partial_{t,\Delta\tau} U_{i,j}^{n,k} = \partial_{x,H}^2 U_{i,j}^{n,k-1} + \partial_{y,h}^2 U_{i,j}^{n,k}, \quad 1 \leq k \leq m \quad (\text{on } \Omega_h^E),$$

$$(6.6) \quad \partial_{t,\Delta t} U_{i,j}^{n+1} = \partial_{x,h}^2 U_{i,j}^{n+1} + \partial_{y,h}^2 U_{i,j}^{n+1} \quad (\text{on } \Omega_h^I).$$

Here,  $\Omega_h^E$  and  $\Omega_h^I$  can be similarly defined.

Similarly, Method 2 can be extended to

$$(6.7) \quad U_{i,j}^{n+1} = u_{i,j}^{n+1} \quad (\text{at boundary points}),$$

$$(6.8) \quad \partial_{t,\Delta t} U_{0,j}^{n+1} = \frac{4}{3} \partial_{x,H}^2 U_{0,j}^n - \frac{1}{12} \partial_{x,2H}^2 U_{0,j}^n + \partial_{y,h}^2 U_{i,j}^{n+1} \quad (\text{at interface point}),$$

$$(6.9) \quad \partial_{t,\Delta t} U_{i,j}^{n+1} = \partial_{x,h}^2 U_{i,j}^{n+1} + \partial_{y,h}^2 U_{i,j}^{n+1} \quad (\text{at interior points}).$$

Note that (6.5) and (6.8) are explicit in the  $x$  direction while implicit in the  $y$  direction. Thus, the Thomas algorithm can be applied in parallel for each tridiagonal system of size  $h^{-1}$  so that the parallel run time for each solution is  $O(h^{-1})$ . Another possible extension is to apply the explicit scheme in both the  $x$  and the  $y$  directions.

**6.2. Stability and convergence of the methods.** For Method 1, maximum principle is still valid in the higher-dimensional case. Thus, similar to Corollary 3.3, we have the following corollary.

COROLLARY 6.1. *Under the assumption (2.5), there exists a constant  $C$  independent of  $\Delta t$ ,  $h$ ,  $H$ ,  $p$ , and  $m$  such that*

$$(6.10) \quad |u_P - U_P| \leq C(\Delta t + h^2 + mpH^3) \quad \forall P \in \Omega_h,$$

where  $u_P$  and  $U_P$  are the true solution of the PDE (6.1)–(6.3) and the approximation generated by Method 1 at  $P \in \Omega_h$ , respectively.

For Method 2, we consider the normal mode

$$U_{i,j}^n = \begin{cases} z^n \tau^i \sin(jh\pi), & i \geq 0, \\ z^n \tau^{-i} \sin(jh\pi), & i < 0. \end{cases}$$

Then, we get

$$\left(1 - \frac{1}{z}\right) - \sigma(\tau + \tau^{-1} - 2) + \sigma 4 \sin^2\left(\frac{h\pi}{2}\right) = 0.$$

Again, for  $|z| \geq 1$ , there are two roots for  $\tau$ , including one with  $|\tau| < 1$ . Using the interface condition, we get

$$(z - 1) - \frac{4}{3}\sigma(2\tau^L - 2) + \frac{1}{12}\sigma(2\tau^{2L} - 2) + 4\sigma z \sin^2\left(\frac{h\pi}{2}\right) = 0.$$

Thus,

$$z = \frac{1 + \frac{4}{3}\sigma(2\tau^L - 2) - \frac{1}{12}\sigma(2\tau^{2L} - 2)}{1 + 4\sigma \sin^2\left(\frac{h\pi}{2}\right)}.$$

Using the analysis made for the one-dimensional problem, we see that for  $0 < \sigma < 4/11$ , we have  $|z| < 1$ . This contradiction implies that  $|z| < 1$ . Thus, the normal mode is stable.

By generalizing the theory of Osher and Gustafsson, we can also obtain the convergence theory for this case as well [4, 5, 8, 9, 11].

**6.3. Parallel efficiency analysis.** Let us examine Method 1, and the analysis is similar for Method 2. Recall that the parallel time for each explicit substep, with the Thomas algorithm applied in parallel, is  $O(h^{-1})$ . So (5.3) is now replaced by

$$(6.11) \quad t_p = C^E m^2 h^{-1} + C^I g(N, p),$$

where  $C^I g(N, p)$  is the run time of the implicit solver applied to each subdomain, and  $N = h^{-2}$  is the total number of unknowns of the global problem. For Jacobi and Gauss–Seidel solvers, we have

$$(6.12) \quad g(N, p) = \frac{N}{p} h^{-2} = \frac{N^2}{p}.$$

For a band solver, because each subdomain has  $h^{-1}$  grid points in the  $y$  direction and  $\frac{h^{-1}}{p}$  points in the  $x$  direction, we can order the grid points so that the bandwidth of the coefficient matrix is  $O(\frac{h^{-1}}{p})$ . Thus, we have

$$(6.13) \quad g(N, p) = \frac{N}{p} \left(\frac{h^{-1}}{p}\right)^2 = \frac{N^2}{p^3}.$$



For SOR solver, we have

$$(6.14) \quad g(N, p) = \frac{N}{p} h^{-1} = N^{1.5}/p.$$

The cost would be further reduced if other more efficient solvers could be applied. Corresponding to (5.18), the total parallel run time becomes

$$(6.15) \quad \begin{aligned} T_p &= \frac{T}{\Delta t} \cdot t_p \\ &= \frac{T}{m\Delta\tau} (C^E m^2 h^{-1} + C^I g(N, p)) \\ &= \frac{2T}{mH^2} (C^E m^2 h^{-1} + C^I g(N, p)). \end{aligned}$$

The conditions (5.5) and (5.6) on accuracy and stability remain the same. Thus, for Case III corresponding to (5.19), we have

$$(6.16) \quad \begin{aligned} T_p &= \frac{2T}{m \cdot \left(\frac{Ch^2}{mp}\right)^{2/3}} (C^E m^2 h^{-1} + C^I g(N, p)) \\ &= \frac{2Tp^{2/3}}{C^{2/3} h^{4/3}} \left( C^E m^{5/3} h^{-1} + C^I g(N, p) m^{-1/3} \right). \end{aligned}$$

Optimizing  $T_p$  with respect to  $m$ , corresponding to (5.21), we have

$$(6.17) \quad m_{opt} = \sqrt{\frac{C^I g(N, p) h}{5C^E}}.$$

Thus,

$$(6.18) \quad \begin{aligned} T_p^{opt} &= \frac{2Tp^{2/3}}{C^{2/3} h^{4/3} \cdot m_{opt}^{1/3}} (C^E m_{opt}^2 h^{-1} + C^I g(N, p)) \\ &= \frac{2Tp^{2/3}}{C^{2/3} h^{4/3} \cdot \left(\frac{C^I}{5C^E} g(N, p) h\right)^{1/6}} \left( \frac{C^I}{5} g(N, p) + C^I g(N, p) \right) \\ &= \frac{12C^I T p^{2/3} g^{5/6}(N, p)}{5C^{2/3} h^{3/2} \left(\frac{C^I}{5C^E}\right)^{1/6}}. \end{aligned}$$

Therefore,

$$(6.19) \quad \begin{aligned} S_{opt} &= \frac{\left(\frac{T}{\Delta t_s}\right) C^I g(N, 1)}{T_p^{opt}} \\ &= O\left(\Delta t_s^{-1} \cdot \frac{h^{3/2} g(N, 1)}{p^{2/3} g^{5/6}(N, p)}\right). \end{aligned}$$

To measure the gain of the new algorithm, we analyze Case II, corresponding to (5.15), to obtain the speedup:

$$(6.20) \quad \begin{aligned} S_2 &= \frac{\Delta\tau}{\Delta t_s} \cdot \frac{C^I g(N, 1)}{C^E h^{-1} + C^I g(N, p)} \\ &= \frac{C^2}{2\Delta t_s} \cdot \left(\frac{h^2}{p}\right)^{2/3} \cdot \frac{C^I g(N, 1)}{C^E h^{-1} + C^I g(N, p)}. \end{aligned}$$

Thus,

$$(6.21) \quad \frac{S_{opt}}{S_2} = O\left((g(N, p)h)^{1/6}\right).$$

This shows that the more expensive the local solver is, the more gain the new algorithm has, because we have managed to maximize the use of cheaper explicit computation while maintaining the stability and global accuracy. For Jacobi and Gauss–Seidel solvers, for example, from (6.12) we have

$$(6.22) \quad S_{opt} = O\left(\Delta t_s^{-1} \cdot h^{5/6} \cdot p^{1/6}\right)$$

and

$$(6.23) \quad \frac{S_{opt}}{S_2} = O\left(\left(\frac{1}{h^3 p}\right)^{1/6}\right).$$

Compared with (5.24) in the one-dimensional case, the improvement in parallel efficiency of the new algorithm is more substantial over the previous approaches. For the case of the band solver, from (6.13), we have

$$(6.24) \quad S_{opt} = O\left(\Delta t_s^{-1} \cdot h^{5/6} \cdot p^{1/6} \cdot p\right).$$

To reach the optimal speedup  $S_{opt} = O(p)$  only requires  $\Delta t_s = O\left(\left(\frac{h^5}{p}\right)^{1/6}\right) > O(h)$ , which is a much weaker restriction on the step size than  $\Delta t = O(h^2)$  is, as in the explicit scheme.

**7. Concluding remarks.** In this paper, we have outlined a few possible improvements for the parallel discretization of the linear parabolic equations so that higher parallel efficiency can be achieved. It is likely that generalizations can be made to cases with nonuniform grids and fractional time steps. We will consider those generalizations in the future.

**Acknowledgments.** The authors would like to thank Tony Chan for the valuable discussions, especially on Method 2, and also to thank Tarek Mathew for the references.

#### REFERENCES

- [1] X. CAI, *Additive Schwarz algorithms for parabolic convection-diffusion equations*, Numer. Math, 50 (1991), pp. 41–52.
- [2] T. F. CHAN AND T. MATHEW, *Domain Decomposition Algorithms*, CAM report CAM 94-2, UCLA, 1994.
- [3] C. N. DAWSON, Q. DU, AND T. F. DUPONT, *A finite difference domain decomposition algorithm for numerical solution of the heat equation*, Math. Comp., 57 (1991), pp. 63–71.
- [4] B. GUSTAFSSON, *The convergence rate for difference approximations to general mixed initial boundary value problems*, SIAM J. Numer. Anal., 18 (1981), pp. 179–190.
- [5] B. GUSTAFSSON, H.-O. KREISS, AND A. SUNDSTRÖM, *Stability theory of difference approximations for mixed initial boundary value problems. II*, Math. Comp., 26 (1972), pp. 649–686.
- [6] D. E. KEYES, *Domain Decomposition: A Bridge between Nature and Parallel Computers*, ICASE report 92-44, Langley, VA, 1992.
- [7] Y. KUZNETSOV, *New algorithms for approximate realization of implicit difference scheme*, Soviet J. Numer. Anal. Math. Modelling, 3 (1988), pp. 99–114.
- [8] D. MICHELSON, *Stability theory of difference approximations for multidimensional initial-boundary value problems*, Math. Comp., 40 (1983), pp. 1–45.

- [9] D. MICHELSON, *Convergence theorem for difference approximations of hyperbolic quasilinear initial-boundary value problems*, Math. Comp. 49, (1987), pp. 445–459.
- [10] K. W. MORTON AND D. F. MAYERS, *Numerical Solution of Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1994.
- [11] S. OSHER, *Stability of parabolic difference approximations to certain mixed initial boundary value problems*, Math. Comp., 26 (1972), pp. 13–39.
- [12] Z. N. WU AND H. ZOU, *Grid overlapping for implicit parallel computations of compressible flows*, J. Comput. Phys., 157 (2000), pp. 2–43.
- [13] G. YUAN, L. SHEN, AND Y. ZHOU, *Unconditional stability of alternating difference schemes with intrinsic parallelism for two-dimensional parabolic systems*, Numer. Methods Partial Differential Equations, 15 (1999), pp. 625–636.
- [14] Y. ZHOU AND G. YUAN, *General difference schemes with intrinsic parallelism for semilinear parabolic systems of divergence type*, J. Comput. Math., 17 (1999), pp. 337–352.