Computing 49, 11–23 (1992)

Computing

A Domain Splitting Algorithm for Parabolic Problems

H. Blum[1], Dortmund, **S. Lisky**[2] and **R. Rannacher**[2], Heidelberg

Received March 20, 1992

Abstract — Zusammenfassung

A Domain Splitting Algorithm for Parabolic Problems. In the parallel implementation of solution methods for parabolic problems one has to find a proper balance between the parallel efficiency of a fully explicit scheme and the need for stability and accuracy which requires some degree of implicitness. As a compromise a domain splitting scheme is proposed which is locally implicit on slightly overlapping subdomains but propagates the corresponding boundary data by a simple explicit process. The analysis of this algorithm shows that it has satisfactory stability and approximation properties and can be effectively parallelized. These theoretical results are confirmed by numerical tests on a transputer system.

AMS Subject Classifications: 65M55, 65Y05 (primary), 65M60 (secondary)

Key words: Domain decomposition, parallel time stepping, parabolic problems, transputer system

Ein Gebietszerlegungsalgorithmus für parabolische Probleme. Die Implementierung von Lösungsmethoden für parabolische Probleme erfordert eine ausreichende Balance zwischen der parallelen Effizienz voll-expliziter Schemata und der Notwendigkeit von Stabilität und Genauigkeit, welche einen gewissen Grad an Implizitheit bedingt. Als ein Kompromiß wird ein Gebietszerlegungsverfahren vorgeschlagen, welches lokal implizit ist auf leicht überlappenden Teilgebieten, die lokalen Randdaten aber durch einen einfachen expliziten Prozeß fortpflanzt. Die Analyse dieses Algorithmus zeigt, daß er zufriedenstellende Stabilitäts- und Approximationseigenschaften besitzt und effektiv parallelisiert werden kann. Diese theoretischen Resultate werden bestätigt durch numerische Tests auf einem Transputer-System.

1. The Domain Splitting Algorithm

We consider the model diffusion problem

$$\partial_t u - a\Delta u = f, \qquad \text{in } \Omega \times (0, T], \qquad u_{|\partial\Omega} = 0, \qquad u_{|t=0} = u^0, \tag{1}$$

where Ω is a two-dimensional domain (for conceptional simplicity assumed to be convex polygonal), and the driving force f may be time dependent. The spatial discretization of (1), e.g., by a finite element method, results in a system of ordinary differential equations,

$$\dot{u}_h - a\Delta_h u_h = f_h, \qquad \text{on } (0, T], \qquad u_h(0) = u_h^0, \tag{2}$$

where Δ_h stands for the discrete analogue of the Laplacian operator. The discretization of (2) with respect to time has to take into account the problem's strong stiffness, with ratio $O(h^{-2})$, where h is the mesh width. For explicit schemes, the numerical

stability requires a step size restriction, $k \leq ch^2$, which leads to an undesirably large number of time steps to reach the time level $T \sim 1$. Hence, the good parallel efficiency of these schemes may be largely reduced through the losses caused by the frequent communication steps. On the other hand, the implicit schemes, being unconditionally stable, allow for the choice $k \sim h$, but they require the solution of globally coupled linear systems of dimension $n = O(h^{-2})$ in each time step. For very large n, these systems have to be solved iteratively, for example by the well-known SSOR- or IC-CG-method or by an appropriate multigrid algorithm, [1]. However, the efficient parallelization of these methods is not easy to accomplish, because of the recursive nature of the preconditioning procedure and the need for the frequent exchange of small amounts of data within the iteration. As an alternative, we propose a simple domain splitting algorithm, described below, which is only blockwise implicit.

Below, we will use the standard notation $L^2(\Omega)$, $H^k(\Omega)$, and $H_0^k(\Omega)$ for the Lebesgue and Sobolev spaces over Ω, respectively. The inner product and norm of $L^2(\Omega)$ are $(\cdot, \cdot)_\Omega$ and $\|\cdot\|_\Omega$, where the subscript Ω is usually suppressed.

For the spatial discretization of problem (1), we consider a finite element Galerkin method using linear shape functions. Let $\mathcal{T}_h = \{T\}$ be a quasi-regular family of triangulations of $\bar{\Omega}$, consisting of closed triangles T of width h (for a formal definition see [3]). The discrete spaces are

$$S^h(\Omega) = \{v_h \in H^1(\Omega), v_{h|T} \in P_1(T), T \in \mathcal{T}_h\}, \qquad S_0^h(\Omega) = S^h(\Omega) \cap H_0^1(\Omega). \tag{3}$$

The mesh domain $\bar{\Omega} = \bigcup \{T \in \mathcal{T}_h\}$ is divided into finitely many convex subdomains Ω_i, $i = 1, \ldots, N$, each of which is a union of triangles of $\mathcal{T}_h$ and has width $H \sim 1$. Correspondingly, we consider enlarged subdomains $\Omega_i^\delta = \bigcup \{T \in \mathcal{T}_h, \operatorname{dist}(T, \Omega_i) \leq \delta\}$, for some $\delta = Lh$. The union of the Ω_i^δ covers $\bar{\Omega}$ with overlaps of width $\sim \delta$. The corresponding finite element subspaces are denoted by $S^h(\Omega_i)$, $S_0^h(\Omega_i)$, and $S^h(\Omega_i^\delta)$, $S_0^h(\Omega_i^\delta)$, respectively.

Now, fixing the overlap $\delta = Lh$, we formulate the following domain splitting algorithm.

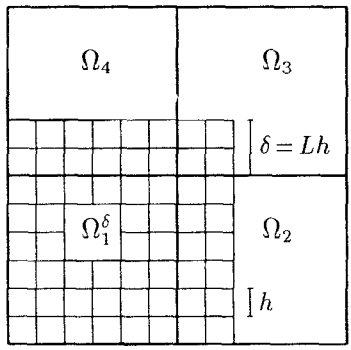

Figure 1. 2 × 2-splitting of a square

Step 0. We choose some approximation $U^0 \in S_h^0(\Omega)$ to the initial value u^0, e.g., its L^2-projection into $S_h^0(\Omega)$.

Assume now that, for some $m > 0$, all approximations $U^\mu \in S_0^h(\Omega)$, $0 \leq \mu < m$, to $u^\mu = u(\cdot, t^\mu)$ have been calculated. Then the approximation U^m is determined through the following sequence of steps.

Step 1. In a predictor step, some approximation $U_*^m = \mathscr{E}_m(\{U^\mu\}_{\mu<m}) \in S_0^h(\Omega)$ is computed through a suitable *explicit* procedure which will be specified below.

Step 2. Then, on each Ω_i^δ, approximations $U_i^m \in S_h(\Omega_i^\delta)$ are obtained through local Crank-Nicolson-steps,

$$\frac{1}{k}(U_i^m - U^{m-1}, \Phi)_{\Omega_i^\delta} + \frac{a}{2}(V[U_i^m + U^{m-1}], V\Phi)_{\Omega_i^\delta} = (\bar{f}^m, \Phi)_{\Omega_i^\delta}, \qquad \forall \Phi \in S_0^h(\Omega_i^\delta), \quad (4)$$

where $\bar{f}^m = \frac{1}{2}(f^m + f^{m-1})$ and

$$U_i^m = U_*^m, \qquad \text{on } \partial\Omega_i^\delta. \quad (5)$$

Step 3. Finally, from the patchwise solutions U_i^m, a global single valued function $U^m = \mathscr{C}(\{U_i^m\}_{1\leq i\leq N}) \in S_0^h(\Omega)$ is constructed through a suitable averaging process.

As a particular realization of Step 1 we consider an explicit Euler step for $\mathscr{E}_1(\{U^\mu\}_{\mu=0})$,

$$\frac{1}{k}(U_*^1 - U^0, \Phi) + a(VU^0, V\Phi) = (\bar{f}^1, \Phi), \qquad \forall \Phi \in S_0^h(\Omega), \quad (6)$$

and linear extrapolation for $\mathscr{E}_m(\{U^\mu\}_{\mu=m-1,m-2})$, $m \geq 2$,

$$U_*^m = 2U^{m-1} - U^{m-2}. \quad (7)$$

In the test calculations, reported below, we actually used quadratic extrapolation,

$$U_*^m = 3U^{m-1} - 3U^{m-2} + U^{m-3}, \quad (8)$$

for $m \geq 3$, in order to increase the accuracy of the predicted boundary values. Note that the prediction step has to be performed only on the artificial boundaries $\partial\Omega_i^\delta$.

There are various possibilities for realizing the operator $\mathscr{C}$ in Step 3. It is natural to set $U^m(P) = U_i^m(P)$ in the interior nodal points P of each Ω_i. For the nodal points at the interfaces or at the cross points of neighboring subdomains, one may take the corresponding arithmetic mean values. However, for our theoretical analysis it is convenient to assume that the operator $\mathscr{C}$ satisfies an L^2-stability estimate with constant 1,

$$\|\mathscr{C}(\{U_i^m\}_{1\leq i\leq N})\|_\Omega^2 \leq \sum_{i=1}^N \|U_i^m\|_{\Omega_i}^2. \quad (9)$$

Further, $\mathscr{C}$ should be the identity operator on those $\{V_i^m\}_{1\leq i\leq N}$ which correspond to a globally continuous function, i.e.,

$$\mathscr{C}(\{V_{|\Omega_i^\delta}^m\}_{1\leq i\leq N}) = V^m, \qquad V^m \in S_0^h(\Omega). \quad (10)$$

The stability property (9) is guaranteed, e.g., if in the definition of $\mathscr{C}$ the nodal values at the interfaces and cross points are taken according to

$$|U^m(P)| = \min\{|U_i^m(P)|, i \in \{j, P \in \Omega_j\}\}. \tag{11}$$

We emphasize that the solution procedure in Step 2 can be performed completely in parallel, without any communication, while in Step 1 and Step 3 only local data from the interfaces and in the overlap strips have to be exchanged between neighboring subdomains. In particular, no *global* communication is required in this algorithm.

Our theoretical analysis will show that the proposed splitting method is numerically stable under a step size condition of the form

$$k \leq \frac{1}{a}\kappa(L)h^2, \tag{12}$$

where $\kappa(L) \sim \kappa_0 L^2(\log L)^{-2}$, independent of the number, N, of subdomains. Furthermore, the global error behavior is of the order $O(h^2 + k^2)$. For a model situation, the constant κ_0 has been found through a numerical experiment to be of the size $\kappa_0 \sim 2$. Hence, for overlap parameters $L = 3$ or 4, the scheme should possess satisfactory stability and convergence properties. This has been confirmed by several test calculations.

The domain splitting algorithm presented in this paper is primarily constructed as a direct time discretization scheme. A similar approach based on a finite difference discretization on non-overlapping subdomains has been proposed in [4], where the global information transfer is realized through an explicit discretization on a coarse mesh; for extensions of this idea to finite elements see [5] and [6]. However, this approach differs from ours as it depends on the coarse grid mesh size H and the number N of subdomains. But, in their underlying philosophy both schemes are similar and differ from the standard approaches for the (implicit) parallel solution of parabolic problems mostly found in the literature. There, the domain splitting is used, in the spirit of elliptic problems, in order to construct a preconditioning procedure for the iterative inversion of the operator $I_h - ak\Delta_h$, see, e.g., [8] for such a method on non-overlapping domains, and [10] and [9] for one using overlapping subdomains. The latter approach is somewhat related to ours as here also the singular perturbation character of the operator $I_h - ak\Delta_h$ is used to suppress the global communication within each time step.

2. Stability and Error Analysis

We shall give an analysis of the stability and the approximation properties of the domain splitting algorithm, described above. The key idea is to compare U^m with the solution of the standard global Crank-Nicolson discretization $V^m \in S_0^h(\Omega)$, successively defined through the equations

$$\frac{1}{k}([V^m - V^{m-1}], \Phi) + \frac{a}{2}(\nabla[V^m + V^{m-1}], \nabla\Phi) = (\bar{f}^m, \Phi), \qquad \forall \Phi \in S_0^h(\Omega), \tag{13}$$

and the initial condition $V^0 = U^0$.

For this, there holds (see, e.g., [13] and [14])

$$\|V^m - u(\cdot, t_m)\| \leq c\{h^2 + t_m^{-1}k^2\}, \qquad t_m > 0, \tag{14}$$

if the data of problem (1) satisfy

$$u^0 \in H_0^1(\Omega) \cap H^2(\Omega), \qquad \partial_t^r f \in L^\infty((0, T]; L^2(\Omega)), r = 0, 1, 2. \tag{15}$$

Theorem. *Suppose that the foregoing assumptions are satisfied. Further assume that the overlap width, $\delta = Lh$, is chosen sufficiently large such that*

$$L^2 \geq \sigma \log^2(\sigma)/\kappa_0, \qquad \sigma = ak/h^2, \tag{16}$$

with a certain constant κ_0. Then the proposed domain splitting scheme is numerically stable and for the difference between the approximation U^m and the global Crank-Nicolson solution V^m, there holds

$$\|V^m - U^m\| \leq C_T t_m^{-1} k^2, \qquad 0 < t_m \leq T. \tag{17}$$

Clearly, the estimate (17) combined with the estimate (14) for the global Crank-Nicolson scheme implies that the proposed domain splitting algorithm is indeed of second order accurate in space and time if the condition (16) is satisfied.

We note that in our numerical tests, the logarithmic growth in the condition (16) was never observed. Instead, we found the simpler relation

$$L^2 \geq \sigma/\kappa_0, \tag{18}$$

for a fixed time step k, or, equivalently, the step size restriction

$$k \leq \frac{\kappa_0}{a} L^2 h^2, \tag{19}$$

for a fixed overlap width Lh. The constant κ_0 turned out to be independent of the size of the domain Ω and of the number N of subdomains.

For the proof of the above theorem, we employ the following decay property for the solutions of a certain class of singularly perturbed problems.

Lemma 1. *For given $\lambda > 0$ and $G \in S^h(\Omega_i^\delta)$, let $U \in S^h(\Omega_i^\delta)$ be defined through*

$$(U, \Phi)_{\Omega_i^\delta} + \lambda(\nabla U, \nabla \Phi)_{\Omega_i^\delta} = 0, \qquad \forall \Phi \in S_0^h(\Omega_i^\delta), \qquad U = G \quad \text{on } \partial\Omega_i^\delta. \tag{20}$$

Then, there holds

$$\|U\|_{\Omega_i}^2 + \lambda \|\nabla U\|_{\Omega_i}^2 \leq c e^{-\gamma\delta/\max\{\sqrt{\lambda}, h\}} \{\|G\|_{\Omega_i^\delta}^2 + \lambda \|\nabla G\|_{\Omega_i^\delta}^2\}, \tag{21}$$

where c and γ are positive constants independent of the size of the subdomain Ω_i.

The estimate (21) states that the effect of the nonhomogeneous boundary data in (20) decays exponentially into the interior of the subdomain Ω_i^δ. This was suggested by the related decay property of the L^2-projection of the point Dirac functional onto finite element spaces (see [7]). A similar result has been shown in the present context in [10] by using a maximum principle for the discrete operator $I_h - \lambda \Delta_h$. The following proof of Lemma 1 will be entirely based on Hilbert space arguments.

Proof of Lemma 1. We only analyze the practically important case $\sqrt{\lambda} > h$. The case $\sqrt{\lambda} \leq h$ can be treated by an analogous argument.

We introduce the distance function $d(\cdot) = \operatorname{dist}(\cdot, \partial\Omega_i^\delta)_I \in S^h(\Omega_i^\delta)$, where here and below the subscript I denotes the piecewise linear nodal interpolation. Since $d - \delta \geq 0$ on Ω_i, there holds

$$\|U\|^2_{\Omega_i} + \lambda\|\nabla U\|^2_{\Omega_i} \leq e^{-\gamma\delta/\sqrt{\lambda}}\{(e^{\gamma d/\sqrt{\lambda}}U, U)_{\Omega_i^\delta} + (e^{\gamma d/\sqrt{\lambda}}\nabla U, \nabla U)_{\Omega_i^\delta}\}, \tag{22}$$

with some constant $\gamma \in (0, 1]$, which will appropriately be chosen below. Subsequently, we shall drop the subscript i and the reference to the set Ω_i^δ in the notation of the inner products.

For abbreviation, we set $\Psi = e^{\gamma d/\sqrt{\lambda}}U$. Note that $d = 0$ on $\partial\Omega^\delta$ and thus, $\Psi_I = G$ on $\partial\Omega^\delta$. In view of (20) and the identity

$$(e^{\gamma d/\sqrt{\lambda}}\nabla U, \nabla U) = (\nabla(e^{\gamma d/\sqrt{\lambda}}U), \nabla U) - (U\nabla e^{\gamma d/\sqrt{\lambda}}, \nabla U), \tag{23}$$

there holds

$$\begin{aligned}(e^{\gamma d/\sqrt{\lambda}}U, U) + \lambda(e^{\gamma d/\sqrt{\lambda}}\nabla U, \nabla U) &= (\Psi - \Psi_I, U) + \lambda(\nabla(\Psi - \Psi_I), \nabla U) + (G, U)\\ &\quad + \lambda(\nabla G, \nabla U) - \lambda(U\nabla e^{\gamma d/\sqrt{\lambda}}, \nabla U).\end{aligned} \tag{24}$$

Since $|\nabla e^{\gamma d/\sqrt{\lambda}}| \leq |\gamma e^{\gamma d/\sqrt{\lambda}}/\sqrt{\lambda}|$, the last term on the right of (24) can be absorbed into the left hand side. Next, we estimate the first term on the right of (24),

$$(\Psi - \Psi_I, U) \leq (e^{-\gamma d/\sqrt{\lambda}}(\Psi - \Psi_I), \Psi - \Psi_I)^{1/2}(e^{\gamma d/\sqrt{\lambda}}U, U)^{1/2}. \tag{25}$$

Using the relation

$$\max_{T \in \mathscr{T}_h}(e^{-\min_T \gamma d/\sqrt{\lambda}}/e^{-\max_T \gamma d/\sqrt{\lambda}}) \leq e^{\gamma h/\sqrt{\lambda}}, \tag{26}$$

the standard L^2-interpolation estimates for finite elements carry over to the corresponding weighted L^2-norms,

$$\begin{aligned}(e^{-\gamma d/\sqrt{\lambda}}(\Psi - \Psi_I), \Psi - \Psi_I) &\leq ch^4 e^{\gamma h/\sqrt{\lambda}} \sum_{T \in \mathscr{T}_h} \int_T e^{-\gamma d/\sqrt{\lambda}} |\nabla^2 \Psi|^2 \, dx\\ &\leq c\gamma^2 h^4 \lambda^{-2} e^{\gamma h/\sqrt{\lambda}}\{(e^{\gamma d/\sqrt{\lambda}}U, U) + \lambda(e^{\gamma d/\sqrt{\lambda}}\nabla U, \nabla U)\}.\end{aligned} \tag{27}$$

Here, we have used that $\nabla^2 U = 0$ and $\nabla^2 d = 0$ on each $T \in \mathscr{T}_h$. By an analogous argument we get

$$\begin{aligned}&\lambda(e^{-\gamma d/\sqrt{\lambda}}\nabla(\Psi - \Psi_I), \nabla(\Psi - \Psi_I))\\ &\quad \leq c\gamma^2 h^2 \lambda^{-1} e^{\gamma h/\sqrt{\lambda}}\{(e^{\gamma d/\sqrt{\lambda}}U, U) + \lambda(e^{\gamma d/\sqrt{\lambda}}\nabla U, \nabla U)\}.\end{aligned} \tag{28}$$

Consequently, choosing γ sufficiently small, we can absorbe also the remaining terms on the right of (24) into the left hand side, obtaining

$$(e^{\gamma d/\sqrt{\lambda}}U, U) + \lambda(e^{\gamma d/\sqrt{\lambda}}\nabla U, \nabla U) \leq c\{\|G\|^2 + \lambda\|\nabla G\|^2\}. \tag{29}$$

Then, combining (29) with (22) yields the desired estimate (21). □

Furthermore, we shall need the following a priori estimate for higher order time difference quotients of the global Crank-Nicolson solution V^m, which can easily be derived by a spectral argument (see, e.g., [13] and [14]).

Lemma 2. *The second order difference quotient of the global Crank-Nicolson solution of problem* (1) *satisfies the estimate*

$$\|d_t^2 V^m\|_\Omega \le c t_m^{-1} \|u^0\|_{H^2(\Omega)} + c\left(\int_0^{t_m} \sum_{i=0}^{2} \|\partial_t^i f\|^2 \, dt\right)^{1/2}. \tag{30}$$

Proof of the Theorem. The proof of estimate (17) for $E^m = \|U^m - V^m\|^2$ is by induction over the time levels t_m. Without loss of generality, we may assume that $\sigma = ak/h^2 \ge 1$. By definition, we have $E^0 = 0$. For any globally defined function $W \in S_0^h(\Omega)$, we introduce the notation $W_i = W_{|\Omega_i^\delta}$. Then, from the assumed properties (9) and (10) of the operator $\mathscr{C}$, we obtain

$$E^m = \|\mathscr{C}(\{U_i^m - V_i^m\}_{1 \le i \le N})\|^2 \le \sum_{i=1}^{N} \|U_i^m - V^m\|_{\Omega_i}^2. \tag{31}$$

We shall compare each U_i^m with the restriction to Ω_i^δ of the auxiliary function $\tilde{V}^m \in S_0^h(\Omega)$, which is obtained through a global Crank-Nicolson step from time level t_{m-1} to t_m with the initial data U^{m-1},

$$\frac{1}{k}(\tilde{V}^m - U^{m-1}, \Phi) + \frac{a}{2}(\nabla(\tilde{V}^m + U^{m-1}), \nabla\Phi) = (\bar{f}^m, \Phi), \qquad \forall \Phi \in S_0^h(\Omega). \tag{32}$$

Then, the difference $Z_i^m = U_i^m - \tilde{V}_i^m \in S^h(\Omega_i^\delta)$ satisfies

$$(Z_i^m, \Phi) + \frac{a}{2} k(\nabla Z_i^m, \nabla\Phi) = 0, \qquad \forall \Phi \in S_0^h(\Omega_i^\delta), \qquad Z_i^m = U_*^m - \tilde{V}^m \quad \text{on } \partial\Omega_i^\delta. \tag{33}$$

In view of Lemma 1, it follows that

$$\|Z_i^m\|_{\Omega_i}^2 \le c e^{-\gamma\delta/\sqrt{ak}} (\|U_*^m - \tilde{V}^m\|_{\Omega_i^\delta}^2 + ak \|\nabla(U_*^m - \tilde{V}^m)\|_{\Omega_i^\delta}^2), \tag{34}$$

with some constant $\gamma \in (0, 1]$. Hence, using the standard inverse property of finite elements,

$$\|\nabla W\|_{\Omega_i^\delta} \le c h^{-1} \|W\|_{\Omega_i^\delta}, \tag{35}$$

and recalling that $\sigma = ak/h^2 \ge 1$ and $\delta = Lh$, we conclude that

$$\|U_i^m - \tilde{V}^m\|_{\Omega_i}^2 \le c e^{-\gamma L/\sqrt{\sigma}} \sigma \|U_*^m - \tilde{V}^m\|_{\Omega_i^\delta}^2. \tag{36}$$

Since $\mathscr{E}_1$ represents an explicit Euler step, we have that

$$\|U_*^1 - \tilde{V}^1\| = \|\mathscr{E}_1(\{V^\mu\}_{\mu=0}) - V^1\| \le ck. \tag{37}$$

This together with (31) implies that

$$E^1 \le c_0 k^4 t_1^{-2}, \tag{38}$$

with a certain constant c_0. Now, we consider the case $m \ge 2$. Suppose that for all $t_\mu < t_m$, there holds

$$E^{\mu} \leq c_0 k^4 t_{\mu}^{-2}. \tag{39}$$

In order to estimate E^m, the estimate (36) is now used for the right hand side of (31). By Young's inequality, we get that

$$E^m \leq c\varepsilon^{-1} e^{-\gamma L/\sqrt{\sigma}} \sigma \sum_{i=1}^{N} \|U_*^m - \tilde{V}^m\|_{\Omega_i^\delta}^2 + (1+\varepsilon)\|\tilde{V}^m - V^m\|^2, \tag{40}$$

where the constant $\varepsilon \in (0, 1]$ will be suitably fixed, below.

To estimate the second term on the right of (40), we observe that $\tilde{V}^m - V^m$ is obtained through one global Crank-Nicolson step with a homogeneous right hand side and the initial data $U^{m-1} - V^{m-1}$. Therefore, again by a standard spectral argument, one sees that

$$\|\tilde{V}^m - V^m\|^2 \leq (1 - \alpha/\sigma)\|U^{m-1} - V^{m-1}\|^2 = (1 - \alpha/\sigma)E^{m-1}, \tag{41}$$

with some numerical constant $\alpha > 0$.

The first term on the right of (40) is now estimated as follows,

$$\begin{aligned} \sum_{i=1}^{N} \|U_*^m - \tilde{V}^m\|_{\Omega_i^\delta}^2 \leq c\{&\|\mathscr{E}_m(\{U^\mu\}_{\mu<m}) - \mathscr{E}_m(\{V^\mu\}_{\mu<m})\|^2 \\ &+ \|\mathscr{E}_m(\{V^\mu\}_{\mu<m}) - V^m\|^2 + \|V^m - \tilde{V}^m\|^2\}. \end{aligned} \tag{42}$$

The last norm on the right has already been estimated in (41). For the second term, for $m \geq 2$, we have that

$$\|\mathscr{E}_m(\{U^\mu - V^\mu\}_{\mu<m})\|^2 \leq c \max\{E^{m-1}, E^{m-2}\}. \tag{43}$$

Notice that, by definition, $\mathscr{E}_1(\{U^\mu\}_{\mu=0}) = \mathscr{E}_1(\{V^\mu\}_{\mu=0})$. To complete the estimate of (42) it remains to bound the term $\|\mathscr{E}_m(\{V^\mu\}_{\mu<m}) - V^m\|$. For $m \geq 2$, we use the standard estimate for linear extrapolation to get

$$\|\mathscr{E}_m(\{V^\mu\}_{\mu<m}) - V^m\| \leq ck^2 \|d_t^2 V^m\|, \tag{44}$$

where d_t^2 stands for the second order difference quotient in time. In virtue of Lemma 2 it follows that

$$\|\mathscr{E}_m(\{V^\mu\}_{\mu<m}) - V^m\| \leq ck^2 t_m^{-1} \|u^0\|_{H^2(\Omega)}. \tag{45}$$

Combining the estimates (41), (43), and (44) with (40), we get the final result

$$\begin{aligned} E^m \leq c\varepsilon^{-1} e^{-\gamma L/\sqrt{\sigma}} \sigma \{&\max\{E^{m-1}, E^{m-2}\} + k^4 t_m^{-2} \|u^0\|_{H^2(\Omega)}^2\} \\ &+ (1+\varepsilon)(1 - \alpha/\sigma)E^{m-1}. \end{aligned} \tag{46}$$

Now, we fix $\varepsilon = \alpha/\sigma$ to obtain,

$$(1+\varepsilon)(1 - \alpha/\sigma) = (1 - \alpha^2/\sigma^2) < 1. \tag{47}$$

The first coefficient in (46) can then be made arbitrarily small by taking the overlap width sufficiently large according to $L^2 \geq \sigma \log^2(\sigma)/\kappa_0$, to obtain

$$E^m \leq c_0 k^4 t_m^{-2}. \tag{48}$$

Now, the assertion follows by induction. □

Remark. The numerical results obtained by the proposed algorithm in its basic form are not satisfactory since the accuracy of the second order boundary extrapolation is insufficient. Therefore, we used quadratic extrapolation in time, for $m \geq 3$, which contributes to the global error by

$$\|\mathscr{E}_m(\{V^\mu\}_{\mu<m}) - V^m\|_\Omega \leq ck^3 \|d_t^3 V^m\|_\Omega, \tag{49}$$

instead of (44). However, the required "smoothing" a priori estimate

$$\|d_t^3 V^m\| \leq ct_m^{-2} \|u^0\|_2 + \left(\int_0^{t_m} \sum_{i=0}^{3} \|\partial_t^i f\|_0^2 \, dt \right)^{1/2}, \tag{50}$$

for the third order time difference of the global Crank-Nicolson solution V^m, similar to Lemma 2, can only be guaranteed if some damping is added to the scheme. For example, one may replace the first time step by one implicit Euler step, see also [13] for a related procedure.

3. Numerical Results

The theoretical results of the preceding section have been verified through various test calculations. It turns out that, for practical step sizes h and k, the simplified conditions (18) and (19) are indeed sufficient to guarantee the stability and the optimum order accuracy of the proposed domain splitting algorithm. Its run time behavior and accuracy is compared with that of a global Crank-Nicolson scheme. Both algorithms have been implemented on a transputer system (Transputer-SuperCluster 128 of the IWR Heidelberg) using up to $p = 64$ processors. This parallel computer is based on the T800 transputer of INMOS which has a measured performance of 0.5-1 MFlops (64 Bit), and 4 MBytes of local memory. The T800 has 4 communication links with a speed of 2.35 MBytes/sec. The coding has been done in FORTRAN 77 under the operating system Helios 1.1. For a comprehensive measurement of the practical performance of this parallel computer, see [2].

All test computations have been performed for rectangular domains using $N = 1 \times p$ (pipe) or $N = r \times s$ subdomains (rectangular splitting). The number p of processors was always chosen according to $p = N$.

First, the stability property of the domain splitting algorithm has been tested on various processor configurations. Table 1 shows the calculated intervals (κ_s, κ_u), where the method is found to be stable for $\kappa_0 < \kappa_s$ and unstable for $\kappa_0 > \kappa_u$. These results are essentially independent of the number of subdomains and of the size of the domain Ω. We note that for problems in one space dimension the stability properties are nearly identical to those for the $1 \times p$-configurations in two dimensions.

Test examples. The parallel performance of the proposed domain splitting method has been tested at the model problem (1), with the data $a = \pi^{-1}$, and $T = 1$, on a rectangular domain $\Omega = \{(x, y): 0 < x < 8, 0 < y < 4\}$. The number of subdomains, $N = 32$, coincides with the number of processors, $p = 32$. The linear systems on each subdomain are solved by the cg-method with SSOR-preconditioning up to the

Table 1. Numerical intervals for the stability constant κ_0

configuration	h	L			
		2	3	4	5
pipe	1/50	(2.75, 2.88)	(2.56, 2.81)	(2.64, 3.17)	(2.53, 3.38)
	1/100	(2.72, 2.75)	(2.62, 2.68)	(2.64, 2.75)	(2.53, 2.70)
rectangle	1/50	(1.98, 2.04)	(2.01, 2.16)	(1.98, 2.26)	(2.03, 2.53)
	1/100	(1.99, 2.01)	(2.20, 2.25)	(2.26, 2.34)	(2.25, 2.38)

residual accuracy of the order 10^{-10}. This algorithm is compared against the "global" Crank-Nicolson method where the linear system in each time step is solved by the conjugate gradient method *without* preconditioning. Clearly, this solution method is slow for $k > 2h$, but may suffice for $k < h/2$. Its parallelization is effectively accomplished by direct data partitioning on the underlying 4×8-domain decomposition. For the tests, the following particular solutions were selected

(a) $$u(x, y, t) = \sin(\pi x/16)\sin(\pi y/16)\sin(\pi t),$$

(b) $$u(x, y, t) = \sin(2.7\pi x)\sin(2.7\pi y)\sin(\pi t).$$

The spatial discretization was set up on a uniform grid of width $h = 1/100$ in each of the subdomains Ω_i, $i = 1, \ldots, 32$, resulting in a discrete problem with about 320.000 unknowns. The local boundary values were predicted by *quadratic* extrapolation, in order to enhance the accuracy of this crucial step. The two considered test cases, (a) and (b), are complementary in the difficulties they may cause for the splitting algorithm. The solution (a) has a very small variation in space. Hence, a balance between the spatial and the time discretization error is achieved only for relatively small time steps, $k \approx h/4$. In this case, the solution performance of the (locally preconditioned) splitting scheme is not expected to be superior to that of the global method. For the solution (b) the optimal error balance is reached for coarse time steps $k \approx 2h$, which causes stability problems for the domain splitting scheme for small overlaps.

In Table 2 the discretization errors, measured in the $L^\infty(\Omega)$-norm at the time $T = 1$, of the domain splitting scheme, $\mathrm{err}_{\mathrm{spli}}$, and that of the global Crank-Nicolson

Table 2. Comparison of the discretization errors (for $h = 1/100$)

(a)				(b)			
k	L	$\mathrm{err}_{\mathrm{spli}}$	$\mathrm{err}_{\mathrm{glob}}$	k	L	$\mathrm{err}_{\mathrm{spli}}$	$\mathrm{err}_{\mathrm{glob}}$
1/130	2	3(−5)	1(−5)	1/30	4	3(−3)	5(−4)
1/180	2	1(−5)	5(−6)	1/40	4	7(−4)	4(−4)
1/260	2	3(−6)	3(−6)	1/50	4	3(−4)	3(−4)

scheme, $\mathrm{err}_{\mathrm{glob}}$, are compared for various time steps k and overlaps $\delta = Lh$. The time step is decreased down to an "optimal" value k_{opt}, at which the time and the space discretization errors appear to be balanced.

The deterioration of the accuracy in the domain splitting algorithm for larger time steps $k > k_{\mathrm{opt}}$ is mainly due to the insufficient accuracy in the prediction of the boundary values on the artificial interior boundaries. It drops below the time discretization error for $k \approx k_{\mathrm{opt}}$. In all tests it turned out that, for overlaps $3h \leq \delta \leq 4h$, the "optimal" time step k_{opt} was well within the stability region of the scheme. Hence, it can be expected that this remains true if the time step size is adaptively chosen according to an a posteriori error control.

Next, we compare the computational efficiency of the domain splitting scheme with that of the "global" Crank-Nicolson method. Table 3 shows the run time performance of the two algorithms on a 2×2-processor array for the solution (a). In this case the time step has to be taken relatively small, $k \approx h/2$, and the global method appears slightly more efficient than the splitting algorithm. This is because the latter has to work on an increased number of unknowns, about 8%, due to the overlap $\delta = 2h$ in the domain decomposition. This additional computational load goes up to 20% in the case $\delta = 5h$. However, the losses by the communication time, t_{comm}, are negligible compared to the total run time, t_{total}.

Table 3. Comparison of the work per time step in the case (a) (for $h = 1/100$)

	global method			splitting method $L = 2$		
k	# cg-iter.	t_{total}	t_{comm}	# cg-iter.	t_{total}	t_{comm}
1/130	35	50 sec	2 sec	7	41 sec	.016 sec
1/180	30	46 sec	2 sec	6	40 sec	.016 sec
1/260	24	37 sec	2 sec	7	40 sec	.016 sec

Clearly, for the solution (b) the global method cannot be competitive with the (locally preconditioned) splitting method, due to the slow convergence of the simple cg-method for large time steps, $k > h$. Therefore, the corresponding results are listed in Table 4 only for the splitting scheme.

Finally, to demonstrate that the proposed splitting method really scales with respect to the number of subdomains, we list the total run times for the model case (b), with the parameters $h = k = 1/100$, $T = 1$, and $L = 3$, for various sizes of problems.

Table 4. Total run time over $[0, T]$ of the splitting method in the case (b) (for $h = 1/100$)

k	L	# cg-iter.	t_{total}	t_{comm}
1/130	2	5	109 min	2 sec
1/60	3	9	60 min	1 sec
1/50	4	10	53 min	1 sec

Table 5. Run time (in min.) for processor configurations of increasing size

Topology	3 × 3	4 × 4	5 × 5	6 × 6	7 × 7	8 × 8
# unknowns	90.000	160.000	250.000	360.000	490.000	640.000
$time_{total}$	90.07	90.11	90.16	90.20	90.22	90.26

Summary

The proposed domain splitting algorithm has satisfactory stability properties and is as accurate as the "global" method, for balanced step sizes h and k. It scales up to large numbers of processors with constant losses through communication overheads, data transfer times, and idle times. The overall communication losses are significantly smaller than those for the considered "global" method. However, for very small time steps this advantage of the "splitting" over the "global" algorithm may be largely compensated by the increase in the computational costs due to the domain overlap. Therefore, this type of domain splitting method seems to be attractive particularly for cases when the computational domain naturally splits into several components and when an effectively parallelized "global" solution algorithm is not available. In the latter case, it may even be used as a preconditioner within a global iteration process.

References

[1] Axelsson, O., Barker, V. A.: Finite element solution of boundary value problems. London: Academic Press 1984.
[2] Bader, G., Gehrke, E.: On the performance of transputer networks for solving linear systems of equations. Parallel Comp. *17*, 1397–1407 (1991).
[3] Ciarlet, Ph. G.: The finite element method for elliptic problems. Amsterdam: North Holland 1978.
[4] Dawson, C. N., Du, Q., Dupont, T. F.: A finite difference domain decomposition algorithm for numerical solution of the heat equation. Report TR90-24, Rice Univ., Houston 1990.
[5] Dawson, C. N., Du, Q.: A domain decomposition method for parabolic equations based on finite elements. Report TR90-25, Rice Univ., Houston 1990.
[6] Dawson, C. N., Dupont, T. F.: Explicit/implicit, conservative, Galerkin domain decomposition procedures for parabolic problems. Report TR90-26, Rice Univ., Houston 1990.
[7] Douglas, Jr., J., Dupont, T., Wahlbin, L.: The stability in L^q of the L^2-projection into finite element function spaces. Numer. Math. *23*, 193–197 (1975).
[8] Dryja, M.: Substructuring methods for parabolic problems. Technical Report 529, New York University, Department of Computer Science, November 1990.
[9] Jäger, J., Hebeker, F. K., Kuznetsov, Y.: Investigation of overlapping in a domain decomposition method for a model heat equation. IBM WZH Technical Report, Heidelberg (in preparation).
[10] Kuznetsov, Y. A.: Domain decomposition methods for time dependent problems. Pubbl. 1st Anal. Numer. Cons. Naz. Ric., Pavia *730*, 261–264 (1989).
[11] Lisky, S.: Eine Gebietszerlegungsmethode zur parallelen Lösung parabolischer Gleichungen auf Transputersystemen. Diplomarbeit, Heidelberg 1992.
[12] Löhner, R., Morgan, K.: Domain decomposition for the simulation of transient problems in CFD. In: R., Glowinski, et al. (eds.) Proc. First Symp. on Domain Decomposition Methods for Part. Diff. Equ. pp. 426–431, SIAM, Philadelphia 1988.
[13] Rannacher, R.: Finite element solution of diffusion problems with irregular data. Numer. Math. *43*, 309–327 (1984).

[14] Thomée, V.: Galerkin finite element methods for parabolic problems. Berlin, New York: Springer 1984 (Lecture Notes in Mathematics 1054).

H. Blum
Fachbereich Mathematik
Universität Dortmund
Vogelpothsweg 87
D-W-4600 Dortmund
Federal Republic of Germany

S. Lisky and R. Rannacher
Institut für Angewandte Mathematik
Universität Heidelberg
Im Neuenheimer Feld 293
D-W-6900 Heidelberg
Federal Republic of Germany