Department
of
APPLIED MATHEMATICS

A Space Decomposition Method for Parabolic Equations

by

Xue-Cheng Tai

Report no. 106 February 1997

UNIVERSITY OF BERGEN
Bergen, Norway

Department of Mathematics
University of Bergen
5007 Bergen
Norway

ISSN 0084-778x

A Space Decomposition Method for Parabolic Equations

by

Xue-Cheng Tai

Report No. 106

February 1997

A SPACE DECOMPOSITION METHOD FOR PARABOLIC EQUATIONS

Xue–Cheng Tai

Department of Mathematics,
University of Bergen,
Alleg. 55, 5007,
Bergen, Norway.

ABSTRACT. Convergence proof is given for an abstract parabolic equation using general space decomposition techniques. The space decomposition technique may be a domain decomposition method, a multilevel method, or a multigrid method. It is shown that if the Euler or Crank-Nicolson scheme is used for the parabolic equation, then by suitably choosing the space decomposition, only $O(|log\tau|)$ steps of iteration at each time level are needed, where τ is the time step size. Applications to overlapping domain decomposition and to a two-level method are given for a second order parabolic equation. Analysis shows that only a one-element overlap is needed. Discussions about iterative and non-iterative methods for parabolic equations are presented. A method that combines the two approaches together and utilizes some of the good properties of the two approaches is tested numerically.

CONTENTS

§ 1. INTRODUCTION

In this article, we use space decomposition techniques for the abstract parabolic equation

$$\begin{cases} \left(\frac{\partial u}{\partial t}, v\right)_S + a(u,v) = (f,v)_S, \quad \forall v \in V \ , t \in [0,T] \ , \\ u(0) = u_0 \in S. \end{cases} \tag{1.1}$$

Here, S is a Hilbert space and $a(u,v)$ is a bounded bilinear, symmetric, and positive definite form on the Hilbert space V. A space decomposition method refers to a method that decomposes the Hilbert space V into a sum of subspaces, i.e.,

$$V = V_1 + V_2 + ... + V_m, \quad V_i \subset V \ .$$

It was observed by Xu [**28**], see also Tai [**20**], that the domain decomposition methods, multilevel methods, multigrid methods and substructuring methods can all be viewed in some way as space decomposition techniques.

The above mentioned space decomposition methods have been widely used for elliptic problems. When they are used for elliptic problems, they give two benefits. Firstly, by suitably using the space decomposition methods, the original elliptic problem is reduced to a number of smaller problems, and these smaller problems can be computed in parallel. Secondly, the space decomposition

1991 *Mathematics Subject Classification.* 65J10, 65M55, 65Y05.
Key words and phrases. Parallel, domain decomposition, parabolic equation, space decomposition.
The work was supported by the Humboldt scholarship of Germany.
Email: Tai@mi.uib.no. http://www.mi.uib.no/tai.

methods are iterative methods. In the iteration procedure, they produce good preconditioners. So, in order to reach a certain accuracy, the iteration number can be greatly reduced. See [**1**], [**3**], [**4**], [**5**], [**6**], [**8**], [**28**] etc.

It is possible to use space decomposition methods for parabolic problems in two ways. Firstly, one could use a time stepping scheme to discretize the time variable. Then, at each time level an elliptic equation must be solved, and one could use the space decomposition method for this elliptic equation. We shall call this the iterative approach. This approach was used in the pioneering work by Lions [**17**]. Recent advances have been made by Cai [**7**], Dryja [**12**], Ewing, Lazarov, Pasciak and Vassilevski [**13**], [**14**], Kuznetsov [**16**], etc. Secondly, one could combine the space decomposition with the time stepping in such a way that the space decomposition is integrated with the time stepping. This will give a "blockwise implicit" time stepping scheme. We shall call this the non-iterative approach. For overlapping domain decomposition, this approach has been used by Blum, Lisky and Rannacher [**2**], Rannacher [**18**], Jäger, Hebeker and Kuznetsov [**15**]. For nonoverlapping domain decomposition, it has been used by Dawson, Du and Dupont [**11**], Dawson and Dupont [**10**].

For the non-iterative approach, as was shown in [**2**], and [**11**], [**10**], it is not necessary to perform any iterations between the subdomain problems at each time level. The subproblems need to be solved just once and we can then proceed to the next time step. This saves computing time and also communication cost when used with parallel processors. The essential idea in [**2**] is that when a new time level is reached, an extrapolation procedure is used to get the boundary conditions for the subdomain problems from the previous time levels. However, this will produce errors that will explode exponentially. Then by using the property that the errors at the subdomain boundaries will "decay" exponentially into the interior of the subdomains, a convergent algorithm is obtained by taking small time steps or by using large overlapping area. In [**10**] and [**11**], nonoverlapping domain decomposition is used with the Euler scheme, and the boundary conditions of the subdomains are predicted by an explicit Euler method.

Both of the two approaches have advantages and disadvantages, see §6. In §6, we show a method to combine these two approaches together and to utilize some of the good properties of the two approaches. Numerical experiments are presented for this combination.

In this work, the iterative approach for parabolic problems is followed. A convergence proof is given for an abstract parabolic problem for general space decomposition techniques. When this general theory is applied to domain decomposition methods, we obtain algorithms which are the similar to the additive Schwartz method. By suitably choosing the time step, and in a two-level method by suitably choosing the coarse mesh size, the error reduction per iteration of the resulted algorithms is independent of the fine mesh size, the coarse mesh size, the number of subdomains, and the time step size. Numerical tests will be given which support the theoretical predictions.

§ 2. Preliminaries

We consider the following abstract parabolic problem

$$\begin{cases} \left(\frac{\partial u}{\partial t}, v\right) + a(u,v) = (f,v), \quad \forall v \in V\ , \\ u(0) = u_0 \in S\ . \end{cases} \tag{2.1}$$

Above and later, $(\cdot,\cdot)$ denotes the inner product of the Hilbert space S. The Hilbert space V can be imbedded into S. The bilinear form $a(\cdot,\cdot)$ is bounded, symmetric, and positive definite on the Hilbert space V, i.e. $a(u,v) = a(v,u)$ for any $u, v \in V$ and there exists $\lambda > 0$ such that

$$\begin{aligned} &|a(u,v)| \le \lambda \|u\| \|v\|, \quad \forall u, v \in V, \\ &a(u,u) \ge \lambda^{-1} \|u\|^2, \quad \forall u \in V. \end{aligned} \tag{2.2}$$

Here, $\|\cdot\|$ denotes any norm for the Hilbert space V. Due to this special property of $a(\cdot,\cdot)$, we use the following inner product and norm for V

$$\begin{aligned} &(u,v)_V = a(u,v), \quad \forall u, v \in V, \\ &\|v\|_V = \sqrt{a(v,v)}, \quad \forall v \in V. \end{aligned}$$

For space decomposition, it is assumed that V can be decomposed as

$$V = V_1 + V_2 + \ldots + V_m. \tag{2.3}$$

The above decomposition means that if $v_i \in V_i$, $i = 1, 2, \cdots, m$, then $\sum_{i=1}^m v_i \in V$, and on the other hand, for any $v \in V$ there exists $v_i \in V_i$ (which may not be unique) such that $v = \sum_{i=1}^m v_i$. For the space decomposition (2.3), we assume that there exist constants $C_L > 0$, $C_S > 0$ and $C_V > 0$ which satisfy

$$\begin{cases} \forall v \in V, \exists v_i \in V_i \text{ such that } v = \displaystyle\sum_{i=1}^m v_i \text{ and} \\ \displaystyle\sum_{i=1}^m \|v_i\|_S^2 \le C_L \|v\|_S^2\ , \\ \displaystyle\sum_{i=1}^m \|v_i\|_V^2 \le C_S \|v\|_S^2 + C_V \|v\|_V^2\ . \end{cases} \tag{2.4}$$

For any general space decomposition, one can always find such constants; see Lions [**17**, p.7].

After decomposing the space V, we can search for a solution in each subspace V_i iteratively, and in the limit the sum of the solutions in the subspaces will converge to the solution of the original problem. The following two algorithms are a combination of this space decomposition iteration with the Euler and the Crank-Nicolson time stepping.

Algorithm 2.1. (Euler space decomposition).

Step 1. *Set* $u^0 = u_0$, *and choose* α_i *such that* $0 < \alpha \le \alpha_i < 1$ *and* $\sum_{i=1}^m \alpha_i = 1$.

Step 2. *At time level* n, *for* $k = 1, 2, \cdots, s$, *do: for each* k, *compute* $u_i^{n+\frac{k}{s}}$ *in parallel for* $i = 1, 2, \cdots, m$ *such that* $u_i^{n+\frac{k}{s}} - u^{n+\frac{k-1}{s}} \in V_i$ *and*

$$\left(\frac{u_i^{n+\frac{k}{s}} - u^n}{\tau}, v_i\right) + a\left(u_i^{n+\frac{k}{s}}, v_i\right) = (f^{n+1}, v_i), \quad \forall v_i \in V_i\ . \tag{2.5}$$

Step 3. *Set*

$$u^{n+\frac{k}{s}} = \sum_{i=1}^m \alpha_i u_i^{n+\frac{k}{s}}\ . \tag{2.6}$$

If $k = s$, *then set* $k = 0, n = n+1$ *and go to step 2 for the next time step. Otherwise, go to step 2 and compute the new value for* $k = k+1$.

Algorithm 2.2 (Crank-Nicolson space decomposition).

Step 1. *The same.*

Step 2. *At time level n, for $k = 1, 2, \cdots, s$, do: for each k, compute $u_i^{n+\frac{k}{s}}$ in parallel for $i = 1, 2, \cdots, m$ such that $u_i^{n+\frac{k}{s}} - u^{n+\frac{k-1}{s}} \in V_i$ and*

$$\left(\frac{u_i^{n+\frac{k}{s}} - u^n}{\tau}, v_i\right) + a\left(\frac{u_i^{n+\frac{k}{s}} + u^n}{2}, v_i\right) = (f^{n+\frac{1}{2}}, v_i), \quad \forall v_i \in V_i \ . \tag{2.7}$$

Step 3. *The same.*

In the above algorithms, τ is the time step, $f^{n+1} = f((n+1)\tau) \in S$, $f^{n+\frac{1}{2}} = f((n+\frac{1}{2})\tau) \in S$, the number i is the subscript of the subspace V_i, n indicates the time level, s is the number of space decomposition iterations that we shall perform at each time level, and k is the counter for the space decomposition iteration. If $s = 1$, the algorithms are one-step space decomposition algorithms. In the following sections, we analyse how large s should be.

For the sake of analysis, it is convenient to introduce a constant C_d and a τ-dependent norm

$$C_d = \frac{\max(2C_L, 2\tau C_S + C_V)}{\alpha} \ , \tag{2.8}$$
$$\|v\|_{a_\tau}^2 = \|v\|_S^2 + \tau\|v\|_V^2 = \|v\|_S^2 + \tau a(v,v) \ .$$

In (2.8) above, α is the constant in Step 1 of the above two algorithms. Later in the analysis, the generic positive constant C, which does not depend on the time step τ, the time level n and the finite element mesh sizes in §5, may differ from context to context.

Remark 2.1. *All the results in this work can be carried over to the case where $a(v,v)$ only defines a seminorm for V. In such a case, the V-norm in (2.4) should also be replaced by the seminorm $\sqrt{a(v,v)}$.*

§ 3. Error Analysis for the Euler scheme

In order to simplify the error analysis, we shall compare $u_i^{n+\frac{k}{s}}$ with the standard Euler scheme solution V^{n+1} of the problem

$$\begin{cases} \left(\frac{V^{n+1}-V^n}{\tau}, v\right) + a(V^{n+1}, v) = (f^{n+1}, v), \quad \forall v \in V, \\ V^0 = u_0. \end{cases} \tag{3.1}$$

The following error estimate can be easily proved for the Euler scheme (see [**27**])

Theorem 3.1. *Let the solution $u(t)$ of (1.1) be in $W^{2,\infty}((0,T],V)$, and $f \in W^{1,\infty}((0,T],S)$. Then*

$$\|u(t_{n+1}) - V^{n+1}\|_{a_\tau} \le C\tau, \tag{3.2}$$

where C only depends on u and u_0.

Theorem 3.2. *At each time level, assume that for some $0 < \rho \le 1$ the number s of the space decomposition iteration satisfies*

$$s \ge 2|\ln(\rho\tau)| \Big/ \left|\ln \frac{4C_d}{1+4C_d}\right| . \tag{3.3}$$

Then, the error satisfies

$$\|u^n - V^n\|_{a_\tau} \le C\tau . \tag{3.4}$$

Above, $C = C(u)e^{2\rho(n+1)\tau}$, and $C(u)$ depends only on u.

The main part of the proof is described in the following lemma.

Lemma 3.3. *For a given $G^k \in V$, let the function G_i^{k+1} satisfy $G_i^{k+1} - G^k \in V_i$, and*

$$\tau^{-1}(G_i^{k+1}, v_i) + a(G_i^{k+1}, v_i) = 0, \quad \forall v_i \in V_i, \quad i = 1, 2, ..., m. \tag{3.5}$$

If we set

$$G^{k+1} = \sum_{i=1}^m \alpha_i G_i^{k+1}, \tag{3.6}$$

then

$$\|G^{k+1}\|_{a_\tau}^2 \le \frac{4C_d}{1+4C_d}\|G^k\|_{a_\tau}^2 . \tag{3.7}$$

Proof. By assumption (2.4), there exists $\phi_i^{k+1} \in V_i$ such that

$$G^{k+1} = \sum_{i=1}^m \phi_i^{k+1} ,$$

$$\sum_{i=1}^m \|\phi_i^{k+1}\|_S^2 \le C_L \|G^{k+1}\|_S^2 , \tag{3.8}$$

$$\sum_{i=1}^m \|\phi_i^{k+1}\|_V^2 \le C_S\|G^{k+1}\|_S^2 + C_V\|G^{k+1}\|_V^2 . \tag{3.9}$$

Thus

$$\begin{aligned} &\frac{\|G^{k+1}\|_S^2}{\tau} + \|G^{k+1}\|_V^2 \\ &= \sum_{i=1}^m \left[\frac{(G^{k+1}, \phi_i^{k+1})}{\tau} + a(G^{k+1}, \phi_i^{k+1})\right] \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{m} \left[\frac{(G^{k+1} - G_i^{k+1}, \phi_i^{k+1})}{\tau} + a(G^{k+1} - G_i^{k+1}, \phi_i^{k+1}) \right] \qquad \text{(Using relation (3.5))} \\
&\leq \sum_{i=1}^{m} \left[\frac{1}{\tau} \|G^{k+1} - G_i^{k+1}\|_S \|\phi_i^{k+1}\|_S + \|G^{k+1} - G_i^{k+1}\|_V \|\phi_i^{k+1}\|_V \right] \\
&\leq \frac{1}{\tau} \left(\sum_{i=1}^{m} \|G^{k+1} - G_i^{k+1}\|_S^2 \right)^{\frac{1}{2}} \sqrt{C_L} \|G^{k+1}\|_S \\
&\qquad + \left(\sum_{i=1}^{m} \|G^{k+1} - G_i^{k+1}\|_V^2 \right)^{\frac{1}{2}} \left(\sqrt{C_S} \|G^{k+1}\|_S + \sqrt{C_V} \|G^{k+1}\|_V \right) \\
&\qquad \text{(Using (3.8) and (3.9) and the inequality } \sqrt{a+b} \leq \sqrt{a} + \sqrt{b} \text{)} \\
&\leq \frac{C_L}{\alpha} \sum_{i=1}^{m} \alpha_i \frac{\|G^{k+1} - G_i^{k+1}\|_S^2}{\tau} + \frac{1}{4\tau} \|G^{k+1}\|_S^2 + \frac{\tau C_S}{\alpha} \sum_{i=1}^{m} \alpha_i \|G^{k+1} - G_i^{k+1}\|_V^2 \\
&\qquad + \frac{1}{4\tau} \|G^{k+1}\|_S^2 + \frac{C_V}{2\alpha} \sum_{i=1}^{m} \alpha_i \|G^{k+1} - G_i^{k+1}\|_V^2 + \frac{1}{2} \|G^{k+1}\|_V^2 \\
&\qquad \text{(Obtained by the inequality } ab \leq \mu a^2 + \frac{b^2}{4\mu}, \ \forall \mu > 0, \text{ and the fact that } \alpha \leq \alpha_i \text{)} \\
&\leq \frac{\max(C_L, \tau C_S + C_V/2)}{\alpha} \sum_{i=1}^{m} \alpha_i \left(\frac{\|G^{k+1} - G_i^{k+1}\|_S^2}{\tau} + \|G^{k+1} - G_i^{k+1}\|_V^2 \right) \\
&\qquad + \frac{1}{2\tau} \|G^{k+1}\|_S^2 + \frac{1}{2} \|G^{k+1}\|_V^2 \ .
\end{aligned}
$$

This shows that

$$
\begin{aligned}
&\frac{\|G^{k+1}\|_S^2}{\tau} + \|G^{k+1}\|_V^2 \\
&\leq \frac{\max(2C_L, 2\tau C_S + C_V)}{\alpha} \cdot \sum_{i=1}^{m} \alpha_i \left(\frac{\|G^{k+1} - G_i^{k+1}\|_S^2}{\tau} + \|G^{k+1} - G_i^{k+1}\|_V^2 \right) .
\end{aligned}
$$

From (3.6), it follows that

$$
\begin{aligned}
&\sum_{i=1}^{m} \alpha_i \left(\frac{\|G^{k+1} - G_i^{k+1}\|_S^2}{\tau} + \|G^{k+1} - G_i^{k+1}\|_V^2 \right) \\
&\leq 2 \left(\frac{\|G^{k+1} - G^k\|_S^2}{\tau} + \|G^{k+1} - G^k\|_V^2 \right. \\
&\qquad \left. + \sum_{i=1}^{m} \alpha_i \frac{\|G_i^{k+1} - G^k\|_S^2}{\tau} + \|G_i^{k+1} - G^k\|_V^2 \right) \qquad \text{(Implied by } (a+b)^2 \leq 2(a^2+b^2) \text{)} \\
&= 2 \left(\frac{\| \sum_{i=1}^{m} \alpha_i (G_i^{k+1} - G^k) \|_S^2}{\tau} + \left\| \sum_{i=1}^{m} \alpha_i (G_i^{k+1} - G^k) \right\|_V^2 \right) \qquad (3.10) \\
&\qquad + 2 \sum_{i=1}^{m} \alpha_i \left(\frac{\|G_i^{k+1} - G^k\|_S^2}{\tau} + \|G_i^{k+1} - G^k\|_V^2 \right) \qquad \text{(By (3.6) and } \sum_{i=1}^{m} \alpha_i = 1) \\
&\leq 4 \sum_{i=1}^{m} \alpha_i \left(\frac{\|G_i^{k+1} - G^k\|_S^2}{\tau} + \|G_i^{k+1} - G^k\|_V^2 \right) .
\end{aligned}
$$

Therefore, we have arrived at

$$
\begin{aligned}
&\frac{\|G^{k+1}\|_S^2}{\tau} + \|G^{k+1}\|_V^2 \\
&\leq 4C_d \sum_{i=1}^{m} \alpha_i \left(\frac{\|G_i^{k+1} - G^k\|_S^2}{\tau} + \|G_i^{k+1} - G^k\|_V^2 \right) .
\end{aligned} \qquad (3.11)
$$

Next, we estimate the right hand side of (3.11). Taking $v_i = G_i^{k+1} - G^k$ in (3.5), and using the equality

$$(u, u-v) = \frac{1}{2}(\|u\|^2 - \|v\|^2 + \|u-v\|^2) ,$$

we get

$$\begin{aligned} &\frac{1}{2\tau}\left(\|G_i^{k+1}\|_S^2 - \|G^k\|_S^2 + \|G_i^{k+1} - G^k\|_S^2\right) \\ &+\frac{1}{2}(\|G_i^{k+1}\|_V^2 - \|G^k\|_V^2 + \|G_i^{k+1} - G^k\|_V^2) = 0 . \end{aligned} \tag{3.12}$$

Substituting (3.12) into (3.11), we find that

$$\begin{aligned} &\frac{\|G^{k+1}\|_S^2}{\tau} + \|G^{k+1}\|_V^2 \\ \le& 4C_d \sum_{i=1}^{m} \alpha_i \left(\frac{\|G^k\|_S^2 - \|G_i^{k+1}\|_S^2}{\tau} + \|G^k\|_V^2 - \|G_i^{k+1}\|_V^2 \right) \\ \le& 4C_d \left(\frac{\|G^k\|_S^2 - \|G^{k+1}\|_S^2}{\tau} + \|G^k\|_V^2 - \|G^{k+1}\|_V^2 \right) . \end{aligned}$$

This shows that

$$\begin{aligned} &\frac{\|G^{k+1}\|_S^2}{\tau} + \|G^{k+1}\|_V^2 \\ \le& \frac{4C_d}{1+4C_d} \left(\frac{\|G^k\|_S^2}{\tau} + \|G^k\|_V^2 \right) , \end{aligned}$$

which proves the lemma.

Proof of Theorem 3.2. At each time level n, let $U^{n+1} \in V$ be an auxiliary function which satisfies

$$\left(\frac{U^{n+1} - u^n}{\tau}, v \right) + a(U^{n+1}, v) = (f^{n+1}, v), \quad \forall v \in V . \tag{3.13}$$

Comparing (2.5) with (3.13) and using the fact that $V_i \subset V$, we see that

$$\left(\frac{u_i^{n+\frac{k}{s}} - U^{n+1}}{\tau}, v_i \right) + a(u_i^{n+\frac{k}{s}} - U^{n+1}, v_i) = 0, \quad \forall v_i \in V_i \tag{3.14}$$

and

$$u_i^{n+\frac{k}{s}} - u^{n+\frac{k-1}{s}} \in V_i .$$

Let us take

$$\begin{aligned} G_i^{k+1} &= u_i^{n+\frac{k}{s}} - U^{n+1} , \\ G^k &= u^{n+\frac{k-1}{s}} - U^{n+1} . \end{aligned}$$

Note that G_i^{k+1} satisfies (3.5) and $G_i^{k+1} - G^k \in V_i$. Clearly, G_i^{k+1} and G^k also depend on n, but for notational simplicity the index n in G_i^{k+1} and G^k is omitted. Using Lemma 3.3, we obtain by induction that

$$\begin{aligned} &\|u^{n+1} - U^{n+1}\|_{a_\tau}^2 = \|u^{n+\frac{s}{s}} - U^{n+1}\|_{a_\tau}^2 = \|G^{s+1}\|_{a_\tau}^2 \\ \le& \frac{4C_d}{1+4C_d} \|G^s\|_{a_\tau}^2 \le \cdots \\ \le& \left(\frac{4C_d}{1+4C_d} \right)^s \|G^1\|_{a_\tau}^2 = \left(\frac{4C_d}{1+4C_d} \right)^s \|u^n - U^{n+1}\|_{a_\tau}^2 , \end{aligned}$$

which means that

$$\|u^{n+1} - U^{n+1}\|_{a_\tau} \leq \left(\frac{4C_d}{1+4C_d}\right)^{\frac{s}{2}} \|u^n - U^{n+1}\|_{a_\tau} . \tag{3.15}$$

Since

$$\|u^{n+1} - V^{n+1}\|_{a_\tau} \leq \|u^{n+1} - U^{n+1}\|_{a_\tau} + \|U^{n+1} - V^{n+1}\|_{a_\tau} \tag{3.16}$$

and

$$\begin{aligned}\|u^n - U^{n+1}\|_{a_\tau} \leq & \|u^n - V^n\|_{a_\tau} \\ & + \|U^{n+1} - V^{n+1}\|_{a_\tau} + \|V^n - V^{n+1}\|_{a_\tau} ,\end{aligned}$$

we get, assuming that s satisfies (3.3), that

$$\begin{aligned}&\|u^{n+1} - V^{n+1}\|_{a_\tau} \\ \leq & \left(\frac{4C_d}{1+4C_d}\right)^{\frac{s}{2}} \left(\|u^n - V^n\|_{a_\tau} + \|U^{n+1} - V^{n+1}\|_{a_\tau} + \|V^n - V^{n+1}\|_{a_\tau}\right) \\ & + \|U^{n+1} - V^{n+1}\|_{a_\tau} \\ \leq & \rho\tau(\|u^n - V^n\|_{a_\tau} + \|U^{n+1} - V^{n+1}\|_{a_\tau} + \|V^n - V^{n+1}\|_{a_\tau}) \\ & + \|U^{n+1} - V^{n+1}\|_{a_\tau} .\end{aligned} \tag{3.17}$$

Next, we estimate $\|U^{n+1} - V^{n+1}\|_{a_\tau}$ and $\|V^{n+1} - V^n\|_{a_\tau}$. Subtracting (3.1) from (3.13), we find that

$$\left(\frac{V^{n+1} - U^{n+1}}{\tau}, v\right) + a(V^{n+1} - U^{n+1}, v) = \left(\frac{V^n - u^n}{\tau}, v\right), \quad \forall v \in V. \tag{3.18}$$

By taking $v = V^{n+1} - U^{n+1}$, it is easy to deduce that

$$\|V^{n+1} - U^{n+1}\|_{a_\tau}^2 \leq \|V^n - u^n\|_S^2 \leq \|V^n - u^n\|_{a_\tau}^2. \tag{3.19}$$

An application of Theorem 3.1 shows that

$$\|V^{n+1} - V^n\|_{a_\tau} \leq \|V^{n+1} - u(t_{n+1})\|_{a_\tau} + \|u(t_{n+1}) - u(t_n)\|_{a_\tau} + \|u(t_n) - V^n\|_{a_\tau} \leq C\tau. \tag{3.20}$$

Summarizing the estimates (3.16), (3.17), (3.19) and (3.20), we see clearly that

$$\begin{aligned}&\|V^{n+1} - u^{n+1}\|_{a_\tau} \\ \leq & \rho\tau(2\|u^n - V^n\|_{a_\tau} + C\tau) + \|u^n - V^n\|_{a_\tau} \\ \leq & (1 + 2\rho\tau)\|u^n - V^n\|_{a_\tau} + C\rho\tau^2.\end{aligned} \tag{3.21}$$

Hence, by induction, it follows from (3.21) that

$$\begin{aligned}&\|V^n - u^n\|_{a_\tau} \\ \leq & (1 + 2\rho\tau)^n \|u^0 - V^0\|_{a_\tau} \\ & + C\rho\tau^2\Big(1 + (1 + 2\rho\tau) + (1 + 2\rho\tau)^2 + \cdots + (1 + 2\rho\tau)^n\Big) \\ \leq & \frac{C\rho\tau^2}{2\rho\tau}(1 + 2\rho\tau)^{n+1} \leq \frac{C}{2}\tau e^{(n+1)\ln(1+2\rho\tau)} \leq \frac{C}{2}\tau e^{2\rho(n+1)\tau} .\end{aligned}$$

Remark 3.4. *In the proof, we did not assume that V is compactly imbedded into S. This enables us to use the algorithms for a wider class of problems. However, if the embedding $V \subset S$ is compact and if*

$$\|v\|_S \leq c_0\|v\|_V, \quad \forall v \in V , \tag{3.22}$$

then we obtain from (3.18) that

$$\|V^{n+1} - U^{n+1}\|_{a_\tau}^2 \leq \|V^n - u^n\|_S^2 \leq \frac{c_0}{c_0 + \tau}\|V^n - u^n\|_{a_\tau}^2 . \tag{3.23}$$

Combining (3.16), (3.17), (3.23) and (3.20), we get a result similar to (3.21), i.e.

$$\|V^{n+1} - u^{n+1}\|_{a_\tau} \leq \left(\frac{c_0}{c_0 + \tau} + 2\rho\tau\right)\|u^n - V^n\|_{a_\tau} + C\rho\tau^2.$$

By requiring $2(c_0 + \tau)\rho \leq 1$, the above inequality implies

$$\|V^{n+1} - u^{n+1}\|_{a_\tau} \leq C\tau ,$$

where the constant C does not grow exponentially with time t.

§ 4. Error Analysis for the Crank-Nicolson scheme

As before, the solution $u_i^{n+\frac{k}{s}}$ of Algorithm 2.2 is compared with the solution of the standard Crank-Nicolson scheme

$$\begin{cases} \left(\frac{V^{n+1}-V^n}{\tau}, v\right) + a\left(\frac{V^{n+1}+V^n}{2}, v\right) = \left(f^{n+\frac{1}{2}}, v\right) , \quad \forall v \in V , \\ V^0 = u_0 . \end{cases}$$

The following error estimate for the Crank-Nicolson scheme is true (see [**27**]).

Theorem 4.1. *Let $u \in W^{3,\infty}((0,T],V)$ and $f \in W^{2,\infty}((0,T],S)$. Then*

$$\|u(t_{n+1}) - V^{n+1}\|_{a_{\tau/2}} \le C\tau^2 ,$$

where C does not depend on τ or n.

Theorem 4.2. *At each time level, assume that for some $0 < \rho \le 0$ the number s of the space decomposition iteration satisfies*

$$s \ge 4|\ln(\rho\tau)| \Big/ \left|\ln \frac{4C_d}{1+4C_d}\right| . \tag{4.1}$$

Then, the following error estimate holds for Algorithm 2.2

$$\|u^n - V^n\|_{a_{\tau/2}} \le C\tau^2 .$$

Here, $C = C(u)Ce^{2\rho^2(n+1)\tau^2}$.

Remark 4.3. *For the Crank-Nicolson scheme, if the constants C_L, C_S, C_V and τ are the same as for the Euler scheme, then at each time level the iteration number s needs to be doubled to retain the second order convergence.*

Proof. Similar to the proof of Theorem 3.2, at each level n, let U^{n+1} be the solution of

$$\left(\frac{U^{n+1}-u^n}{\tau}, v\right) + a\left(\frac{U^{n+1}+u^n}{2}, v\right) = \left(f^{n+\frac{1}{2}}, v\right), \quad \forall v \in V . \tag{4.2}$$

Comparing (2.7) with (4.2), we get that

$$\left(\frac{u_i^{n+\frac{k}{s}} - U^{n+1}}{\tau}, v_i\right) + a\left(\frac{u_i^{n+\frac{k}{s}} - U^{n+1}}{2}, v_i\right) = 0, \quad \forall v_i \in V_i .$$

By taking $G_i^{k+1} = u_i^{n+\frac{k}{s}} - U^{n+1}$, $G^k = u^{n+\frac{k-1}{s}} - U^{n+1}$ and $\tau = \frac{\tau}{2}$ in Lemma 3.3 and using similar techniques as for (3.15), it follows from the estimate (3.7) that

$$\|u^{n+1} - U^{n+1}\|_{a_{\tau/2}} \le \left(\frac{4C_d}{1+4C_d}\right)^{\frac{s}{2}} \|u^n - U^{n+1}\|_{a_{\tau/2}} .$$

When the space decomposition iteration number s satisfies (4.1), it is true that

$$\begin{aligned} \|u^{n+1} - V^{n+1}\|_{a_{\tau/2}} &\le \rho^2\tau^2\Big(\|u^n - V^n\|_{a_{\tau/2}} \\ &+ \|U^{n+1} - V^{n+1}\|_{a_{\tau/2}} + \|V^{n+1} - V^n\|_{a_{\tau/2}}\Big) + \|U^{n+1} - V^{n+1}\|_{a_{\tau/2}} . \end{aligned}$$

One can prove similarly as in obtaining (3.19) and (3.20) that

$$\begin{aligned} \|U^{n+1} - V^{n+1}\|_{a_{\tau/2}} &\le \|u^n - V^n\|_{a_{\tau/2}} , \\ \|V^{n+1} - V^n\|_{a_{\tau/2}} &\le C\tau , \end{aligned}$$

and so we arrive at the result of Theorem 4.2 by similar induction procedures as in the proof of Theorem 3.2.

However, if we use a second order starting scheme at the first time level and modify the subproblems at the other time levels, we can achieve second order accuracy with the same number of space decomposition iterations as for Algorithm 2.1. From Theorem 4.2, we note that for the first time step, if the iteration number s satisfies (4.1), then the first time step will be of second order accuracy. For the other time levels, s can be taken according to the following Algorithm and Theorem.

Algorithm 4.1. *All the other steps are the same as in Algorithm 2.2. Only (2.7) is modified to be*

$$\begin{cases} \left(\dfrac{u_i^{n+\frac{k}{s}}-u^n}{\tau}, v_i\right) + a\left(\dfrac{u_i^{n+\frac{k}{s}}+u^n}{2}, v_i\right) = (f^{n+\frac{1}{2}}, v_i), \quad \forall v_i \in V_i \ , \\ \text{if } k>1, \text{ let } u_i^{n+\frac{k}{s}} - u^{n+\frac{k-1}{s}} \in V_i \ , \\ \text{if } k=1, \text{ let } u_i^{n+\frac{k}{s}} - 2u^n + u^{n-1} \in V_i \ . \end{cases}$$

Theorem 4.4. *Under the same conditions as in Theorem 3.2, if* $\|u^0 - V^0\|_{a_{\tau/2}} \le C\tau^2$, *and* $\|u^1 - V^1\|_{a_{\tau/2}} \le C\tau^2$, *then the following error estimate holds for Algorithm 4.1*

$$\|u^n - V^n\|_{a_{\tau/2}} \le C\tau^2 \ , \quad \forall n > 1 \ .$$

Here, $C = C(u)e^{4\rho n\tau}$, *and* $C(u)$ *depends only on* u.

Proof. With the above new modification of step 2, we can use Lemma 3.3 similarly as in obtaining (3.15) to get

$$\begin{aligned} \|u^{n+1} - U^{n+1}\|_{a_{\tau/2}} &\le \left(\frac{4C_d}{1+4C_d}\right)^{\frac{s}{2}} \|G^1\|_{a_{\tau/2}} \\ &= \left(\frac{4C_d}{1+4C_d}\right)^{\frac{s}{2}} \|2u^n - u^{n-1} - U^{n+1}\|_{a_{\tau/2}} \ . \end{aligned}$$

Under condition (3.3) , the above inequality gives

$$\begin{aligned} &\|u^{n+1} - V^{n+1}\|_{a_{\tau/2}} \\ \le &\rho\tau\Big(\|U^{n+1} - V^{n+1}\|_{a_{\tau/2}} + 2\|u^n - V^n\|_{a_{\tau/2}} + \|u^{n-1} - V^{n-1}\|_{a_{\tau/2}} \\ &+ \|V^{n+1} - 2V^n + V^{n-1}\|_{a_{\tau/2}}\Big) + \|U^{n+1} - V^{n+1}\|_{a_{\tau/2}} \ . \end{aligned} \tag{4.3}$$

By using a standard energy approach, it can be proved similarly as in obtaining (3.19) and (3.20) that

$$\|U^{n+1} - V^{n+1}\|_{a_{\tau/2}} \le \|u^n - V^n\|_{a_{\tau/2}} \ , \tag{4.4}$$

$$\|V^{n+1} - 2V^n + V^{n-1}\|_{a_{\tau/2}} \le C\tau^2 \ , \tag{4.5}$$

and an application of (4.3), (4.4) and (4.5) gives that

$$\begin{aligned} \|u^{n+1} - V^{n+1}\|_{a_{\tau/2}} &\le (1+3\rho\tau)\|u^n - V^n\|_{a_{\tau/2}} \\ &+ \rho\tau\|u^{n-1} - V^{n-1}\|_{a_{\tau/2}} + C\rho\tau^3 \ . \end{aligned} \tag{4.6}$$

A summation of (4.6) shows that

$$\begin{aligned} \|u^{n+1} - V^{n+1}\|_{a_{\tau/2}} &\le \|u^1 - V^1\|_{a_{\tau/2}} \\ &+ 4\rho\tau\sum_{k=0}^{n}\|u^k - V^k\|_{a_{\tau/2}} + C\rho n\tau^3 \ . \end{aligned}$$

It now follows from the Gronwall inequality that

$$\|u^n - V^n\|_{a_{\tau/2}} \le C\tau^2 e^{4\rho n\tau} \ .$$

Above, we assume that $\|u^1 - V^1\|_{a_{\tau/2}} \le C\tau^2$ and $\|u^0 - V^0\|_{a_{\tau/2}} \le C\tau^2$. This proves the theorem.

§ 5. Applications to domain decomposition

Here, we show the applicability of our algorithms by applying them to the finite element solution of the second order parabolic equation

$$\begin{cases} (u_t, v)_{L^2(\Omega)} + a(u,v) = (f,v)_{L^2(\Omega)}, \ \Omega \subset R^d, \quad \forall v \in H_0^1(\Omega), \\ u(0) = u_0(x) \ , \end{cases} \tag{5.1}$$

with

$$a(u,v) = \sum_{i,j=1}^{d} (a_{ij}(x) D_i u, D_j v)_{L^2(\Omega)}, \quad \forall u, v \in H_0^1(\Omega) \ . \tag{5.2}$$

We can use different space decomposition techniques to solve (5.1). Here, we will only apply the proposed algorithms to an overlapping domain decomposition and to a two-level overlapping domain decomposition method, i.e. an overlapping domain decomposition with a coarse mesh. Application to higher level substructuring methods leads to multigrid-like algorithms. We first show how to decompose the finite element space when solving (5.1). Let $\{\Omega_i\}_{i=1}^M$ be a shape-regular finite element division, or a coarse mesh, of Ω, where Ω_i has diameter of order H. For each Ω_i, we further divide it into smaller simplices with diameter of order h. In the case where Ω has a curved boundary, we also fill the area between $\partial\Omega$ and $\partial\Omega_H$, here $\bar{\Omega}_H = \cup_{i=1}^M \bar{\Omega}_i$, with finite elements with diameters of order h. We assume that the resulting elements form a shape-regular finite element subdivision of Ω, see Ciarlet [**9**]. We call this the fine mesh or the h-level subdivision of Ω with mesh parameter h. We denote by $\Omega_h = \cup\{e \in \mathcal{T}_h\}$ the fine mesh subdivision. Let $S_0^H \subset H_0^1(\Omega)$ and $S_0^h \subset H_0^1(\Omega)$ be the continuous, piecewise linear finite element spaces with zero trace on $\partial\Omega_H$ and $\partial\Omega_h$ over the H-level and h-level subdivisions of Ω, respectively.

For each Ω_i, we consider an enlarged subdomain $\Omega_i^\delta = \{e \in \mathcal{T}_h, dist(e, \Omega_i) \leq \delta\}$. The union of the Ω_i^δ covers $\bar{\Omega}_h$ with overlaps of size δ. Let us denote by $S_0^h(\Omega_i^\delta)$ the piecewise linear finite element space with zero trace on the boundary $\partial\Omega_i^\delta$. Then one can show that

$$S_0^h = \sum S_0^h(\Omega_i^\delta) \ , \tag{5.3}$$

$$S_0^h = S_0^H + \sum S_0^h(\Omega_i^\delta) \ . \tag{5.4}$$

For the overlapping subdomains, assume that there are m colours such that each subdomain Ω_i^δ can be marked with one colour, and that the subdomains with same colour will not intersect each other. By choosing the overlap suitably, one can always get $m = 2$ if $d = 1$, $m \leq 4$ if $d = 2$, and $m \leq 6$ if $d = 3$. Let Ω_i' be the union of the subdomains with the i^{th} colour, and let

$$V_i = \{v \in S_0^h \Big| \quad v(x) = 0, \quad x \notin \Omega_i'\} \ .$$

Let us define $V_0 = S_0^H$, $V = S_0^h$. Now, decompositions (5.3) and (5.4) imply that

$$V = \sum_{i=1}^{m} V_i, \tag{5.5}$$

$$V = V_0 + \sum_{i=1}^{m} V_i. \tag{5.6}$$

Let $\{\theta_i\}_{i=1}^m$ be a partition of unity with respect to $\{\Omega_i'\}_{i=1}^m$, i.e. $\theta_i \in C_0^\infty(\Omega_i' \cap \Omega)$ and $\sum_{i=1}^m \theta_i = 1$. It can be chosen so that $|\nabla\theta_i| \leq C/\delta$. Let I_h be an interpolation operator which uses the function values at the h-level nodes. For any $v \in V$, let $v_i = I_h(\theta_i v) \in V_i$. The functions v_i satisfy $v = \sum_{i=1}^m v_i$ and

$$\sum_{i=1}^{m} \|v_i\|_{L^2(\Omega_i)}^2 \leq C\|v\|_{L^2(\Omega)}^2 \ , \tag{5.7}$$

$$\sum_{i=1}^{m} \|v_i\|_{H^1(\Omega_i)}^2 \leq \frac{C}{\delta^2}\|v\|_{L^2(\Omega)}^2 + C\|\nabla v\|_{L^2(\Omega)}^2 \ , \tag{5.8}$$

The proof of (5.7) and (5.8) can be found in various papers, see e.g. Cai [**7**] [**8**]. When the two-level method is used, let $v_0 \in V_0$ be the solution of $(v_0, \phi_H) = (v, \phi_H), \forall \phi_H \in V_0$, and $v_i = I_h(\theta_i(v - v_0))$. They satisfy $v = \sum_{i=1}^{m} v_i$, and

$$\|v_0\|^2_{L^2(\Omega)} + \sum_{i=1}^{m} \|v_i\|^2_{L^2(\Omega_i)} \leq C\|v\|^2_{L^2(\Omega)} , \tag{5.9}$$

$$\|v_0\|^2_{H^1(\Omega)} + \sum_{i=1}^{m} \|v_i\|^2_{H^1(\Omega_i)} \leq \frac{CH^2}{\delta^2}\|v\|^2_{L^2(\Omega)} + C\|\nabla v\|^2_{L^2(\Omega)}. \tag{5.10}$$

In the literature, the overlapping size δ is often taken to be very large, i.e. $\delta = c_1 H$, see [**7**], [**28**]. In the following, we shall only choose $\delta = h$. The corresponding estimates (5.9) and (5.10) can be obtained with minor modifications of the proof given in Xu [**28**, p. 608].

The estimates (5.7)–(5.10) show that for the overlapping domain decomposition the constants in (2.4) are

$$C_L = C,\ C_V = C,\ C_S = \frac{C}{\delta^2} ,$$

and for the two-level method the constants are

$$C_L = C,\ C_V = C,\ C_S = \frac{CH^2}{\delta^2} .$$

For the Euler scheme, the time step τ is normally taken as $\tau = Ch^2$. Using an overlapping domain decomposition with a one-element overlap and without a coarse grid, the constant C_d is independent of

a) The mesh parameters h and H;
b) The number of subdomains;
c) The dimension d of the physical domain.

This implies that the error reduction factor

$$\beta = \frac{4C_d}{1 + 4C_d}$$

does not depend on the above mentioned parameters. However, for the Crank-Nicolson scheme, the time step τ is often taken as $\tau = Ch$, and so

$$\begin{aligned} C_d &= \frac{max(C_L, \tau C_S + C_V)}{\alpha} \\ &\approx \frac{C}{\alpha}\frac{\tau}{\delta^2} \quad (\delta = h) \\ &\approx \frac{C}{\alpha}\frac{\tau}{h^2} . \end{aligned}$$

In return, the s in (4.1) depends on h and τ, and can be very large. This means that at each time step many space decomposition iterations need to be performed. In order to overcome this, a computation on a coarse mesh should be added. If we use a coarse mesh with the overlapping subdomains that only have a one-element overlap, then

$$C_d \approx \frac{C}{\alpha}\frac{\tau H^2}{h^2} .$$

Since we compute the coarse mesh problem in parallel with the subdomain problems, we shall keep $\dim(V_0) \approx \dim(V_i),\ i = 1, 2, ..., m$. This requires that $h \approx CH^2$, and so

$$C_d \approx \frac{C}{\alpha}\frac{\tau H^2}{h^2} \approx \frac{C}{\alpha} .$$

Hence we also get an error reduction factor which is independent of the parameters h and H with a cost of computing the coarse mesh problem.

Next, we continue with the implementation of the Crank-Nicolson scheme Algorithm 4.1 with the two-level method. Decomposition (5.6) means that we have $m+1$ subspaces, and we take the coarse mesh finite element space V_0 as the "zero colour" subspace. When we know u^n and $u^{n+\frac{k-1}{s}}$, let us define $w_0^{n+\frac{k}{s}} = u_0^{n+\frac{k}{s}} - u^{n+\frac{k-1}{s}} \subset S_0^H$ for the coarse mesh problem. Then from (2.5), $w_0^{n+\frac{k}{s}}$ satisfies

$$\left(\frac{w_0^{n+\frac{k}{s}} + u^{n+\frac{k-1}{s}} - u^n}{\tau}, v_H\right) + a\left(\frac{w_0^{n+\frac{k}{s}} + u^{n+\frac{k-1}{s}} + u^n}{2}, v_H\right) = (f^{n+\frac{1}{2}}, v_H) \ , \quad \forall v_H \in S_0^H \ . \tag{5.11}$$

The problem which needs to be solved in each subdomain is to find $u_i^{n+\frac{k}{s}} \in S^h(\Omega_i') \cap S_0^h$ such that

$$\begin{cases} \left(\frac{u_i^{n+\frac{k}{s}} - u^n}{\tau}, v_i\right) + a\left(\frac{u_i^{n+\frac{k}{s}} + u^n}{2}, v_i\right) = (f^{n+\frac{1}{2}}, v_i) \ , \quad \forall v_i \in S_0^h(\Omega_i') \ , \\ u_i^{n+\frac{k}{s}} = 2u^n - u^{n-1} \quad \text{on } \partial\Omega_i' \backslash \partial\Omega \ , \text{ if } k = 1 \ ; \\ u_i^{n+\frac{k}{s}} = u^{n+\frac{k-1}{s}} \text{ on } \partial\Omega_i' \backslash \partial\Omega \ , \text{ if } k > 1 \ . \end{cases} \tag{5.12}$$

The value of $u_i^{n+\frac{k}{s}}$ is known in $\Omega \backslash \Omega_i'$

$$u_i^{n+\frac{k}{s}} = \begin{cases} 2u^n - u^{n-1} \text{ in } \Omega \backslash \Omega_i' \ , \text{ if } k = 1 \ ; \\ u^{n+\frac{k-1}{s}} \text{ in } \Omega \backslash \Omega_i' \ , \text{ if } k > 1 \ . \end{cases}$$

The equations (5.12) shall be computed in parallel in each of the subdomains and also in parallel with the coarse mesh problem. We make the following remarks.

(1) The matrices and load vectors for (5.11) and (5.12) need only to be assembled once.
(2) From step $k-1$ to k, we only need to update the boundary values for the subdomain problems in solving (5.12) in each Ω_i^δ. So the LU decomposition for the linear algebraic systems needs only to be done once.
(3) Due to the minimum overlap, the communication between the processors is finite.

In [**21**]-[**24**], [**25**], [**26**], the methods proposed here have been used for some nonlinear problems and some variational inequalities.

Example 5.1. We use Algorithm 4.1 to compute the one dimensional problem

$$u_t - (au_x)_x = f \ , \quad x \in [0,1] \ ,$$

with $a = e^x, u = \sin(6\pi xt)$. This solution is oscillatory. The numerical experiment with this solution is representative for the numerous tests that have been done for the proposed algorithms.

In the subsequent tables, N is the number of the elements in each subdomain and M is the number of subdomains. Each subdomain Ω_i is extended by just one element to get overlaps. Uniform mesh size is used both for the fine mesh and the coarse mesh. Since the coarse grid problem is computed in parallel with the problems in the subdomains, we take

$$N = M, \text{ i.e. } h = H^2 \ ,$$

to balance the work, and thus $\Omega_i = [(i-1)H, iH]$. In using parallel machines, the number of subdomains is often determined by the number of processors. Then we may not be able to choose the coarse mesh freely. In such a case we may need to use one processor to compute several subproblems or to adjust the overlapping size and time step to get better convergence.

The overlapping subdomains can be marked by two colours. Together with the coarse mesh finite element space, we have $m = 3$, and so we shall take $\alpha_i = 1/3$ for the under-relaxation parameters. Numerical tests show that the error reduction factor $\beta \approx 0.8$. Thus, in order to show the dependency of the computational error on the iteration number s, we take

$$s_{max} = 2|\ln h|/|\ln 0.8| \ ,$$

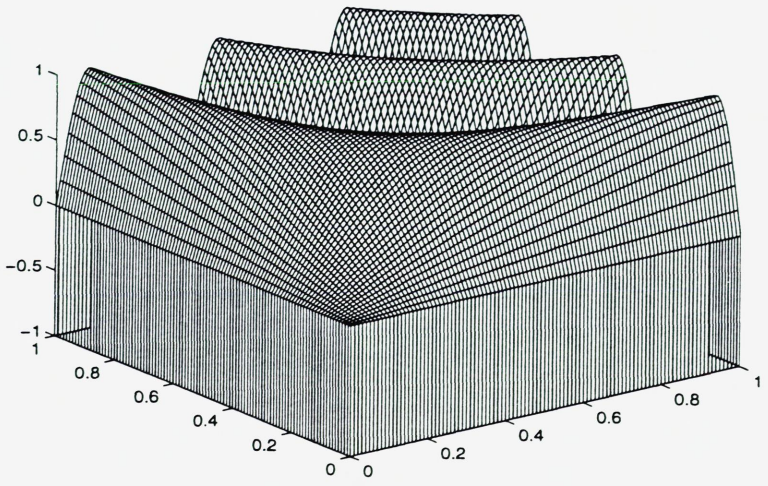

Figure 1: The solution $u = \sin(6\pi xt)$ in the (x,t) plane.

and according to Theorem 4.4, $s = s_{max}$ is sufficient to retain the second order accuracy. In Table 5.1, the solution of the equation is computed using different $s \in [1,\ s_{max}]$. The table contains the maximum errors, i.e. the maximal error over all nodal points and all time levels. In the table e_h^s is the error of Algorithm 4.1 by performing sufficiently many iterations for the first time step and carrying out s steps of iterations for the other time steps. In the last column of the table, e_h^g is the error of the global Crank-Nicolson solution. We can clearly see that for $s << s_{max}$, the solution computed by the domain decomposition method is as accurate as the global Crank-Nicolson solution.

§ 6. A COMBINATION OF TWO DIFFERENT APPROACHES

In this section, we shall discuss some of the advantages and disadvantages of the two different approaches used for parabolic equations and try to combine some of the good properties of both approaches.

In using non-iterative methods for parabolic problems, it is customary to use an explicit method to predict the boundary values for the subdomains. Dawson et al. [**10**] and [**11**] use nonoverlapping domain decomposition and the boundary values of the subdomains are predicted by an explicit Euler method. Blum et al [**2**], Rannacher and Zhou [**19**] and Rannacher [**18**] use an overlapping domain decomposition and the boundary values of the subdomains are predicted by the extrapolation formula

$$u_i^{n+1} = 2u^n - u^{n-1} \tag{6.1}$$

with a suitable starting scheme for the first time level. However, this boundary error will cause the computed solution to explode exponentially. Then by using the property that the boundary error will decay exponentially into the interior of the subdomains, and by cutting off the solution that is "near" the subdomain boundaries, a convergent algorithm is obtained. Due to the explicit manner in which the boundary values of the subdomains are obtained, both the methods of Dawson et al [**11**] and Rannacher [**18**] need to impose the stability condition

$$\tau \leq C\delta^2 \ . \tag{6.2}$$

Here τ is the time step size. In [**11**], δ is the maximum size of the subdomains, and in [**2**], [**18**], δ is the overlapping size.

In using iterative methods for parabolic problems (see Cai [**7**], Dryja [**12**], etc.), one reduces the parabolic equation to an elliptic equation at each time level and then solves this elliptic equation by using an iterative domain decomposition method. For this approach, the advantage that we

N = 5	s =	1	6	11	16	21	26	s_{max}	$e_h^g =$
	$e_h^s =$	1.1e+00	8.7e-02	7.3e-02	7.2e-02	7.2e-02	7.2e-02	29	7.2e-02
N = 7	s =	1	8	15	22	29		s_{max}	$e_h^g =$
	$e_h^s =$	2.4e-01	2.1e-02	1.9e-02	1.8e-02	1.8e-02		35	1.8e-02
N = 9	s =	1	8	15	22	29	36	s_{max}	$e_h^g =$
	$e_h^s =$	1.4e-01	7.6e-03	6.8e-03	6.7e-03	6.7e-03	6.7e-03	39	6.7e-03
N =11	s =	1	9	17	25	33	41	s_{max}	$e_h^g =$
	$e_h^s =$	8.9e-02	3.4e-03	3.0e-03	3.0e-03	3.0e-03	3.0e-03	43	3.0e-03
N =13	s =	1	10	19	28	37	46	s_{max}	$e_h^g =$
	$e_h^s =$	6.3e-02	1.7e-03	1.5e-03	1.5e-03	1.5e-03	1.5e-03	46	1.5e-03
N =15	s =	1	10	19	28	37	46	s_{max}	$e_h^g =$
	$e_h^s =$	4.7e-02	9.7e-04	8.7e-04	8.7e-04	8.7e-04	8.7e-04	49	8.7e-04
N =17	s =	1	11	21	31	41	51	s_{max}	$e_h^g =$
	$e_h^s =$	3.7e-02	5.8e-04	5.3e-04	5.3e-04	5.3e-04	5.3e-04	51	5.3e-04
N =19	s =	1	11	21	31	41	51	s_{max}	$e_h^g =$
	$e_h^s =$	3.0e-02	3.8e-04	3.4e-04	3.4e-04	3.4e-04	3.4e-04	53	3.4e-04

Table 5.1. Maximum error with $h = 1/N^2, \tau = h, H = 1/N$.

can in fact get a better boundary condition for the subdomains from the previous time levels is not utilized.

For the non-iterative approach, the stability condition is not restrictive for the Euler method, but for the Crank-Nicolson scheme and other higher order schemes the stability condition is rather restrictive. In order to remove the time step constraint (6.2), one should use some method to further reduce the error produced by the "explicit prediction" at each time level. In the following, we propose combining the overlapping domain decomposition methods of the last section with the scheme of [**2**]. The combined scheme is absolutely stable and allows the use of larger time steps. Compared with the purely iterative procedure, the combined scheme can reduce the iteration number needed in each time step to reach a prescribed accuracy. Because an extrapolation method is already used in Algorithm 4.1 to get the subdomain boundary conditions, the combined scheme only differs from Algorithm 4.1 in (2.6); i.e. when the solutions in each subdomain have been obtained, $u^{n+\frac{k}{s}}$ is obtained not by the averaging method of (2.6), but by the cutting off technique of [**2**]. When the iterative procedure of the last section is used to further reduce the "explicit prediction" errors and if there is no coarse mesh for the overlapping domain decomposition, the error reduction per iteration will be very small and depend on the overlapping size. Thus an overlapping domain decomposition with a coarse mesh is more appropriate in the combined scheme. We give the scheme below.

Algorithm 6.1.

Step 1. *Set* $u^0 = u_0$.

Step 2. *At time level* n, *for* $k = 1, 2, \cdots, s$, *do: compute* $w_0^{n+\frac{k}{s}} \in S_0^H$ *from (5.11) and* $u_i^{n+\frac{k}{s}}$ *for* $i = 1, 2, \cdots, m$ *from (5.12) in* Ω_i^δ *in parallel.*

Step 3. *Take* $\alpha = 1/2$ *and set*

$$u^{n+\frac{k}{s}} = \begin{cases} (1-\alpha)u_i^{n+\frac{k}{s}} + \alpha(u^{n+\frac{k-1}{s}} + w_0^{n+\frac{k}{s}}) \text{ in } \Omega_i \backslash \partial\Omega_i \ , i = 1, 2, \cdots, m, \\ \dfrac{(1-\alpha)}{2}(u_i^{n+\frac{k}{s}} + u_j^{n+\frac{k}{s}}) + \alpha(u^{n+\frac{k-1}{s}} + w_0^{n+\frac{k}{s}}) \text{ on } \partial\Omega_i \cap \partial\Omega_j \ , \forall i, j, \ i \neq j \ . \end{cases} \tag{6.3}$$

If $k = s$, *then set* $k = 0, n = n + 1$ *and go to the next time level. Otherwise, go to step 2 and compute the new value for* $k = k + 1$.

Above, Ω_i and Ω_i^δ are defined as in §5. When the computations for (5.11) and (5.12) have been computed, a continuous function $u^{n+\frac{k}{s}}$, which is defined in the whole domain Ω, is assembled from the subdomain solutions and the coarse mesh solution. In assembling $u^{n+\frac{k}{s}}$, the values of $u_i^{n+\frac{k}{s}}$ in Ω_i are used. The subdomain solution $u_i^{n+\frac{k}{s}}$ in the overlapping area $\Omega_i^\delta \cap \Omega_j$ contains more errors. This is due to the exponential decay property of the boundary errors, see [**2**, Lemma 1]. This exponential decay property were also used in [**16**].

Example 6.1. In tables 6.1, 6.2 and 6.3, the problem of Example 5.1 is tested by using Algorithm 6.1. As before, sufficiently many space decomposition iterations are performed for the first time step to guarantee that the accuracy is of second order. For the other time steps, only s steps of the space decomposition iterations are performed. The method of [**2**] is equivalent to taking $s = 1$ in Algorithm 6.1 and not using the coarse mesh. Tables 6.1 and 6.2 show the computed maximum errors by Algorithm 6.1 *without a coarse mesh* (i.e. $\alpha = 0$ in (6.3)) for different values of s. We find that the stability of the algorithm of [**2**] has a special relation to the iteration number s. When s is taken to be an even number, the stability is better. When s is taken to be an odd number, the stability is much poorer. In such a case, unless s is sufficient large, oscillation soon appears in the computed solution and is amplified with time integration. Compare Tables 6.1 and 6.2. This special relation between the iteration number and stability does not appear in our two dimensional simulations.

In Table 6.3, the problem is computed by Algorithm 6.1 *with a coarse mesh.* It can be seen from Table 6.3 that the algorithm is stable and that only a few iterations ($s \approx 7$ for $N < 20$) are required at each time level to reach the same accuracy as the global Crank-Nicolson scheme. In Tables 6.1, 6.2 and Table 6.3, e_h^g is the error of the global Crank-Nicolson scheme.

Acknowledgement: The author would like to thank Prof. Dr R. Rannacher and Dr G. H. Zhou for valuable discussions.

References

1. P. Bjørstad and O. Widlund, *Iterative methods for the solution of elliptic problems on regions portioned into substructuring*, SIAM J. Numer. Anal. . **23** (1986), 1097–1120.
2. H. Blum, S. Lisky and R. Rannacher, *A domain splitting algorithm for parabolic problems*, Computing **49** (1992), 11–23.
3. J. H. Bramble, R. Ewing, J. E. Pasciak and A. H. Schatz, *A preconditioning technique for the efficient solution of problems with local grid refinement*, Comput. Meth. Appl. Mech. Eng. . **67** (1988), 149-159.
4. J. H. Bramble, J. E. Pasciak and A. H. Schatz, *The construction of preconditioners for elliptic problems by substructuring. I.*, Math. Comp. . **47** (1986), 103-134.
5. ———, *The construction of preconditioners for elliptic problems by substructuring. II.*, Math. Comp. . **49** (1987), 1–16.
6. J. H. Bramble, J. E. Pasciak and J. Xu, *Parallel multilevel preconditioners*, Math. Comp. . **55** (1990), 1–21.
7. X.-C. Cai, *Additive Schwarz algorithm for parabolic convection diffusion equations*, Numer. Math. . **60** (1991), 41-62.
8. ———, *Multiplicative Schwartz methods for parabolic problems*, SIAM J. Sci. Comp. **15** (1994), 587–604.
9. P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.
10. C. N. Dawson and T. F. Dupont, *Explicit/implicit conservative Galerkin domain decomposition method of a model heat equation*, Math. Comp. . **58** (1992), 21–34.
11. C. N. Dawson, Du and T. F. Dupont, *A finite difference domain decomposition algorithm for the numerical solution of heat equation*, Math. Comp. . **57** (1991), 63-71.
12. M. Dryja, *Substructure methods for parabolic problems*, Fourth international symposium on Domain decomposition methods for partial differential equations (R. Glowinski, Yuri Kuznetsov, G. A. Meurant, J. Periaux, O. Widlund, eds.), SIAM, Philadelphia, 1991, pp. 264–271.

N = 5	s =	1	3	5	7	9	11	s_{max}	e_h^g =
	e_h^s =	4.1e+09	5.2e+05	3.3e+02	1.1e+00	7.9e-02	7.4e-02	29	7.2e-02
N = 7	s =	1	3	5	7	9	11	s_{max}	e_h^g =
	e_h^s =	5.0e+22	4.3e+17	1.4e+13	1.7e+09	7.0e+05	1.0e+03	35	1.8e-02
N = 9	s =	1	3	5	7	9	11	s_{max}	e_h^g =
	e_h^s =	1.3e+41	9.2e+34	2.1e+29	1.4e+24	2.8e+19	1.6e+15	39	6.7e-03
N =11	s =	1	3	5	7	9	11	s_{max}	e_h^g =
	e_h^s =	5.0e+64	3.3e+57	6.0e+50	2.8e+44	3.4e+38	1.1e+33	43	3.0e-03
N =13	s =	1	3	5	7	9	11	s_{max}	e_h^g =
	e_h^s =	2.4e+93	1.4e+85	2.2e+77	8.0e+69	6.9e+62	1.4e+56	46	1.5e-03
N =15	s =	1	3	5	7	9	11	s_{max}	e_h^g =
	e_h^s =	2.2e+126	4.3e+117	9.2e+108	4.0e+100	3.8e+92	7.9e+84	49	8.7e-04
N =17	s =	1	3	5	7	9	11	s_{max}	e_h^g =
	e_h^s =	6.9e+166	5.2e+156	8.6e+146	3.0e+137	2.2e+128	3.4e+119	51	5.3e-04
N =19	s =	1	3	5	7	9	11	s_{max}	e_h^g =
	e_h^s =	2.7e+211	2.1e+200	3.4e+189	1.1e+179	7.5e+168	1.0e+159	53	3.4e-04

Table 6.1. Maximum error with $h = 1/N^2, \tau = h$ and no coarse mesh.
The error for $s = 1$ is the error of the algorithm of [**2**].
The scheme is unstable when s is taken as an odd number.

13. R. Ewing and R. D. Lazarov, J. E. Pasciak and P. S. Vassilevski, *Finite element methods for parabolic problems with time steps variable in space*, preprint (1992).
14. ———, *Domain decomposition type iterative techniques for parabolic problems on locally refined grids*, SIAM J. Numer. Anal. **30** (1993), 1357–1557.
15. J. Jäger, F. K. Hebeker and Yu. A. Kuznetsov, *An overlapping domain decomposition superposition method for a model heat equation*, Technical report of IBM Germany, Heidelberg scientific centre (1992).
16. Y. A. Kuznetsov, *New algorithms for approximate realization of implicit difference schemes*, Sov. J. Numer. Anal. Math. Modelling . **3** (1988), 99–114.
17. P. L. Lions, *On the Schwarz alternating method. I.*, Domain decomposition methods for partial differential equations (R. Glowinski, G. H. Golub, G. A. Meurant and J. Periaux, eds.), SIAM, Philadelphia, 1988, pp. 1-42.
18. R. Rannacher, *Domain decomposition in the non-stationary streamline diffusion finite element method*, Finite element methods, fifty years of the courant element, Lecture notes in pure and applied mathematics, vol. 164, 1994, pp. 367–380.
19. R. Rannacher and G. H. Zhou, *Analysis of a domain-splitting method for non-stationary convection-diffusion problems*, East-west J. Numer. Math. **2** (1994), 151-174.
20. X.–C. Tai, *Parallel function decomposition and space decomposition methods with applications to optimisation, splitting and domain decomposition*, Preprint No. 231–1992, Institut für Mathematik, Technische Universität Graz (http: //www.mi.uib.no/tai) (1992).
21. ———, *Parallel function and space decomposition methods - Part I. Function decomposition*, Beijing Mathematics **1, part 2** (1995), 104–134.
22. ———, *Parallel function decomposition and space decomposition methods: Part II. Space decomposition*, Beijing Mathematics **1, part 2** (1995), 135–152.
23. ———, *Parallel function and space decomposition methods*, Finite element methods, fifty years of the courant element (P. Neittaanmäki, ed.), Lecture notes in pure and applied mathematics, vol. 164, Marcel Dekker inc., 1994, pp. 421–432.
24. ———, *Domain decomposition for linear and nonlinear elliptic problems via function or space decomposition*, Domain decomposition methods in scientific and engineering computing (D. Keyes and J. Xu, eds.), Proc. of the

N = 5	s =	2	4	6	8	10	s_{max}	$e_h^g =$
	$e_h^s =$	5.0e-01	1.9e-01	1.1e-01	8.5e-02	7.6e-02	29	7.2e-02
N = 7	s =	2	4	6	8	10	s_{max}	$e_h^g =$
	$e_h^s =$	4.1e-01	1.6e-01	8.6e-02	5.2e-02	3.5e-02	35	1.8e-02
N = 9	s =	2	4	6	8	10	s_{max}	$e_h^g =$
	$e_h^s =$	3.3e-01	1.3e-01	7.6e-02	4.9e-02	3.4e-02	39	6.7e-03
N =11	s =	2	4	6	8	10	s_{max}	$e_h^g =$
	$e_h^s =$	2.7e-01	1.1e-01	6.7e-02	4.4e-02	3.2e-02	43	3.0e-03
N =13	s =	2	4	6	8	10	s_{max}	$e_h^g =$
	$e_h^s =$	2.3e-01	9.9e-02	5.9e-02	4.0e-02	2.9e-02	46	1.5e-03
N =15	s =	2	4	6	8	10	s_{max}	$e_h^g =$
	$e_h^s =$	2.0e-01	8.7e-02	5.3e-02	3.7e-02	2.7e-02	49	8.7e-04
N =17	s =	2	4	6	8	10	s_{max}	$e_h^g =$
	$e_h^s =$	1.7e-01	7.7e-02	4.8e-02	3.3e-02	2.5e-02	51	5.3e-04
N =19	s =	2	4	6	8	10	s_{max}	$e_h^g =$
	$e_h^s =$	1.5e-01	6.9e-02	4.3e-02	3.0e-02	2.3e-02	53	3.4e-04

Table 6.2. Maximum error with $h = 1/N^2, \tau = h$ and no coarse mesh. Stability is better when s is taken as an even number.

7th international conference on domain decomposition, Penn. State University, 1993, American Mathematical Society, 1995, pp. 355-360.

25. X.-C. Tai and M. Espedal, *A space decomposition method for minimization problems*, Submitted to the proceeding of the 9th domain decomposition conference (1996).
26. ———, *Rate of convergence of a space decomposition method and applications to domain decomposition for linear and nonlinear elliptic problems*, submitted (1996).
27. V. Thomée, *Galerkin finite element method for parabolic problems*, Lecture notes in mathematics, vol. 1054, Springer, Berlin, 1984.
28. J. C. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Review **34** (1992), 581–613.

N = 5	s =	1	3	5	7	9	11	s_{max}	$e_h^g =$
	$e_h^s =$	3.9e-01	8.8e-02	7.5e-02	7.2e-02	7.2e-02	7.2e-02	29	7.2e-02
N = 7	s =	1	3	5	7	9	11	s_{max}	$e_h^g =$
	$e_h^s =$	1.1e-01	2.5e-02	2.0e-02	1.9e-02	1.8e-02	1.8e-02	35	1.8e-02
N = 9	s =	1	3	5	7	9	11	s_{max}	$e_h^g =$
	$e_h^s =$	6.3e-02	1.0e-02	7.3e-03	6.9e-03	6.7e-03	6.7e-03	39	6.7e-03
N =11	s =	1	3	5	7	9	11	s_{max}	$e_h^g =$
	$e_h^s =$	4.2e-02	5.6e-03	3.4e-03	3.1e-03	3.0e-03	3.0e-03	43	3.0e-03
N =13	s =	1	3	5	7	9	11	s_{max}	$e_h^g =$
	$e_h^s =$	3.0e-02	4.1e-03	1.8e-03	1.6e-03	1.6e-03	1.6e-03	46	1.5e-03
N =15	s =	1	3	5	7	9	11	s_{max}	$e_h^g =$
	$e_h^s =$	2.3e-02	3.1e-03	1.1e-03	9.2e-04	8.8e-04	8.7e-04	49	8.7e-04
N =17	s =	1	3	5	7	9	11	s_{max}	$e_h^g =$
	$e_h^s =$	1.8e-02	2.5e-03	7.3e-04	5.6e-04	5.4e-04	5.3e-04	51	5.3e-04
N =19	s =	1	3	5	7	9	11	s_{max}	$e_h^g =$
	$e_h^s =$	1.4e-02	2.0e-03	5.5e-04	3.7e-04	3.5e-04	3.4e-04	53	3.4e-04

Table 6.3. Maximum error by Algorithm 6.1 with $h = 1/N^2, \tau = h, H = 1/N$.