A Domain Decomposition Method for Parabolic Equations Based on Finite Elements

Clint N. Dawson
Qiang Du

CRPC-TR90069
October, 1990

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

A DOMAIN DECOMPOSITION METHOD FOR PARABOLIC EQUATIONS BASED ON FINITE ELEMENTS

CLINT N. DAWSON [*] AND QIANG DU [†]

Abstract. A domain decomposition procedure for solving parabolic partial differential equations is presented. In this scheme, a Galerkin finite element discretization on a rectangular mesh is used. Boundary values at subdomain interfaces are calculated by an implicit/explicit procedure. This solution serves as boundary data for fully implicit subdomain problems, which can be solve simultaneously. Thus, the scheme is non-iterative, and requires non-overlapping subdomains. Numerical results are presented for problems in two space dimensions.

[*] Mathematical Sciences Dept., Rice University, Houston, TX 77251. This research was supported by NSF Grant DMS-8807257, and used the CRAY-2 at the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign, and the Alliant FX/8 Computer at Argonne National Laboratory, Mathematics and Computer Science Division.

[†] Dept. of Mathematics, Michigan State University, East Lansing, Michigan, 48824.

1. Introduction. In this paper, we discuss a domain decomposition algorithm, first presented in [1], for solving equations of the form

$$u_t - \nabla \cdot (D\nabla u) = f, \quad (x,y) \in \Omega \equiv (0,1)^2, \quad t \in (0,T], \tag{1}$$

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

$$u(x,y,t) = g(x,y,t), \qquad (x,y) \in \partial\Omega, \quad t \in (0,T], \tag{3}$$

on parallel processing computers. We assume $D = (D_{ij}(x,y))$ is a smooth, symmetric, two-by-two matrix, satisfying

$$D_* I < D < D^* I, \tag{4}$$

for positive constants D_* and D^*. We restrict attention to two space dimensions; however, the procedure is generalizable to higher dimensions.

Our method is based on a finite element discretization of (1), using continuous, piecewise bilinear, approximating functions. The scheme represents an extension of work discussed in [2], where L^∞ estimates were derived for a finite difference domain decomposition method for solving the heat equation in one and two space dimensions.

The rest of the paper is organized as follows. First, we present the basic algorithm for two subdomains, and state an error estimate. The extension to a multiple strip decomposition or a patch decomposition is discussed in Section 3. In Section 4, numerical results for a variety of two-dimensional problems with strip and box decompositions are presented, and timings for runs made on a parallel processing machine with eight processors are given.

2. The basic method. Denote by $(\cdot,\cdot)$ the L^2 inner product on Ω. Let $\Delta t = T/M$ for some positive integer M, $t^n = n\Delta t$, $n = 0,\ldots,M$, and $g^n = g(t^n)$. Also, let $\partial_t g^n = (g^n - g^{n-1})/\Delta t$.

In order to present the main ideas of the method, we take a simple decomposition. Consider dividing the domain Ω into two subdomains, $(0,\bar{x}) \times (0,1)$ and $(\bar{x},1) \times (0,1)$. Denote by

$$\delta_x : 0 = x_0 < x_1 < \ldots < x_{N_x+1} = 1$$

a partition of $(0,1)$ into intervals of length $h_i^x = x_{i+1} - x_i$, $i = 0,\ldots,N_x$. Similarly, denote by δ_y a partition of $(0,1)$ into intervals of length $h_j^y = y_{j+1} - y_j$, $j = 0,\ldots,N_y$, and let $\delta = \delta_x \otimes \delta_y$ define a partition of Ω into rectangles. Related to $\bar{x}$ we define a parameter $H > 0$ satisfying $H \leq \min(\bar{x}, 1-\bar{x})$, and we assume $\bar{x}$, $\bar{x} - H$, and $\bar{x} + H$ are all points of δ_x, with $\bar{x} = x_k$ for some k. Let $h^x = \max_i h_i^x$, $h^y = \max_j h_j^y$, and $h = \max(h^x, h^y)$.

Denote by $\mathcal{M} \subset \overset{\circ}{H}{}^1_0(\Omega)$ the space of continuous functions on Ω, bilinear on each rectangle defined by δ, and zero on $\partial\Omega$, and note that a basis for $\mathcal{M}$ is the tensor product $\{v_{\delta_x,1}(x),\ldots,v_{\delta_x,N_x}(x)\} \otimes \{v_{\delta_y,1}(y),\ldots,v_{\delta_y,N_y}(y)\}$ of "hat functions;" e.g., $v_{\delta_x,i}(x)$ is given by

$$v_{\delta_x,i}(x) = \begin{cases} (x - x_{i-1})/(x_i - x_{i-1}), & x_{i-1} \leq x \leq x_i, \\ (x_{i+1} - x)/(x_{i+1} - x_i), & x_i \leq x \leq x_{i+1}, \\ 0, & \text{otherwise.} \end{cases}$$

Define spaces $\mathcal{M}_L$, $\mathcal{M}_R$, and $\mathcal{M}_I \subset \mathcal{M}$ by

$$\mathcal{M}_L = \{v \in \mathcal{M} |\ v(x,y) = 0 \text{ for } x \geq \bar{x} = x_k\}, \tag{5}$$

$$\mathcal{M}_R = \{v \in \mathcal{M} |\ v(x,y) = 0 \text{ for } x \leq \bar{x} = x_k\}, \tag{6}$$

$$\mathcal{M}_I = \{v \in \mathcal{M} |\ v(x,y) = 0 \text{ for } x \leq x_{k-1} \text{ and } x \geq x_{k+1}\}, \tag{7}$$

and decompose $W \in \mathcal{M}$ as

$$\begin{aligned} W(x) &= \sum_{j=1}^{N_y}\sum_{i=1}^{k-1} W_{ij} v_{\delta_x,i}(x) v_{\delta_y,j}(y) + \sum_{j=1}^{N_y}\sum_{i=k+1}^{N_x} W_{ij} v_{\delta_x,i}(x) v_{\delta_y,j}(y) \\ &\quad + \sum_{j=1}^{N_y} W_{kj} v_{\delta_x,k}(x) v_{\delta_y,j}(y) \\ &\equiv W_L(x) + W_R(x) + W_I(x). \end{aligned} \tag{8}$$

Hence, $W_L \in \mathcal{M}_L$, $W_R \in \mathcal{M}_R$, and $W_I \in \mathcal{M}_I$, and $\mathcal{M} = \mathcal{M}_L \bigoplus \mathcal{M}_I \bigoplus \mathcal{M}_R$.

Define interface functions $w_{I,j}(x,y)$ by

$$w_{I,j}(x,y) = w_I(x) v_{\delta_y,j}(y), \quad j = 1, \dots, N_y, \tag{9}$$

where $w_I(x)$ is given by

$$w_I(x) = \begin{cases} (x - \bar{x} + H)/H, & \bar{x} - H \le x \le \bar{x}, \\ (\bar{x} + H - x)/H, & \bar{x} \le x \le \bar{x} + H, \\ 0, & \text{otherwise}, \end{cases}$$

and set

$$\mathcal{M}_I^c = \text{span}\,\{w_{I,1}, \dots, w_{I,N_y}\}. \tag{10}$$

For g defined at $(\bar{x}, y)$, $y \in (0,1)$, let

$$\tilde{g}(x,y) = g(\bar{x},y) w_I(x). \tag{11}$$

Hence, for $W \in \mathcal{M}$, $\tilde{W} = \sum_{j=1}^{N_y} W_{kj} w_I(x) v_{\delta_y,j}(y) \in \mathcal{M}_I^c$. Also note that $\mathcal{M}_I^c \subset \mathcal{M}$.

Our domain decomposition approximation $U^n \in \mathcal{M}$ to u^n is given by the following procedure. Approximate u^0 by $U^0 \in \mathcal{M}$, where U^0 is given by the elliptic projection,

$$(D\nabla U^0, \nabla v) = (D\nabla u^0, \nabla v), \quad v \in \mathcal{M}. \tag{12}$$

Then, given U^n, $n = 0, \dots, M-1$, first calculate "interface values" along the line $\bar{x} \times (0,1)$ by solving

$$\langle \partial_t \tilde{U}^{n+1}\,,\, w\rangle + (D\nabla U^n, \nabla w) + \Delta t(D_{22}\partial_t \tilde{U}_y^{n+1}, w_y) = (f^n, w), \quad w \in \mathcal{M}_I^c, \tag{13}$$

where $\langle \cdot\,,\,\cdot\rangle$ is an approximation to $(\cdot,\cdot)$ using the trapezoidal rule in x; that is,

$$\langle \partial_t \tilde{U}^{n+1}\,,\, w\rangle = H \int_0^1 \partial_t U^{n+1}(\bar{x},y) w(\bar{x},y)dy. \tag{14}$$

Thus, (13) determines U_I^{n+1} (see (8)); U_L^{n+1} and U_R^{n+1} are then determined by

$$\begin{aligned}(\partial_t U_L^{n+1}, v) &+ (D\nabla U_L^{n+1}, \nabla v) \\ &= (f^{n+1}, v) - (\partial_t U_I^{n+1}, v) - (D\nabla U_I^{n+1}, \nabla v), \quad v \in \mathcal{M}_L,\end{aligned} \tag{15}$$

and

$$\begin{aligned}(\partial_t U_R^{n+1}, v) &+ (D\nabla U_R^{n+1}, \nabla v) \\ &= (f^{n+1}, v) - (\partial_t U_I^{n+1}, v) - (D\nabla U_I^{n+1}, \nabla v), \quad v \in \mathcal{M}_R.\end{aligned} \tag{16}$$

Note that these two equations decouple, and can be solved in parallel.

Since (13) is partially implicit in y, but fully explicit in x, the interface values U_{kj}^{n+1}, $j = 1, \dots, N_y$, are found by solving a tridiagonal system of equations. One would expect that a relationship involving Δt and H is needed to insure stability; in fact, a sufficient condition is that

$$\frac{\Delta t}{H^2} \|D_{11}\|_\infty \le \lambda \le \frac{5}{12}. \tag{17}$$

However, no relationship between Δt and h^x need be assumed. With a more restrictive constraint than (17), the algorithm satisfies the following *a priori* error estimate, which is proven in [1]:

THEOREM 2.1. *Assume $\lambda = 1/6$ in (17), and assume the true solution u to (1)-(3) is smooth such that $\|u_t\|_{L^2(0,T;W_\infty^2)}$, $\|u\|_{L^\infty(0,T;W_\infty^2)}$ and $\|u_{tt}\|_{L^2(0,T;L^\infty)}$ are bounded. Assume H satisfies $H \le C|ln\ h|^{-1}$, as $h \to 0$. Then U given by (13)-(16) satisfies*

$$\max_n \|u^n - U^n\| \le C(\Delta t + h^2 + H^3), \tag{18}$$

where C depends on the smoothness of u, D_*, and D^*, but not on h, H, or Δt.

Remark: The upper bound of 1/6 on λ used to prove Theorem 1 can be increased to "almost" 5/12, but the constant on the right side of (18) blows up as $\lambda \to 5/12$.

3. Extensions. The extension of the scheme to strip subdomains is detailed in [1]. We will only outline it here.

Suppose a decomposition of Ω into strips $[\bar{x}_l, \bar{x}_{l+1}] \times [0,1]$, $l = 0, \ldots, K$, is constructed, where $\bar{x}_0 = 0$, $\bar{x}_K = 1$, and each $\bar{x}_l$ is in the partition δ_x. Associated with each interface define a parameter $H_l > 0$, and assume $\bar{x}_l \pm H_l$ are in the partition δ_x. We also enforce the condition (on the H_l's) that

$$\bar{x}_{l+1} - H_{l+1} \geq \bar{x}_l, \tag{19}$$

$$\bar{x}_l + H_l \leq \bar{x}_{l+1}, \quad l = 0, \ldots, K-1. \tag{20}$$

Thus, interfaces l and $l+1$ must be at least a distance $\max(H_l, H_{l+1})$ apart. We define spaces $\mathcal{M}^c_{I,l}$ associated with each interface, similar to $\mathcal{M}^c_I$ above, by

$$\mathcal{M}^c_{I,l} = \text{span}\{w^l_{I,j}\},$$

with $w^l_{I,j}$ the analogue of the function $w_{I,j}$ given above. Note that, by (19)-(20),

$$w^l_{I,j}(\bar{x}_{l-1}) = w^l_{I,j}(\bar{x}_{l+1}) = 0, \tag{21}$$

and, by definition,

$$w^l_{I,j}(\bar{x}_l) = 1. \tag{22}$$

At each interior interface, we solve explicit/implicit equations of the form (13) to determine solution values along the interface. These values then serve as boundary data for fully implicit interior equations.

In this case, the algorithm satisfies the following *a priori* error estimate:

THEOREM 3.1. *Assume the true solution u to (1)-(3) satisfies the smoothness assumptions of Theorem 1. Let $\bar{H} = \min_l H_l$, and assume*

$$\frac{\Delta t}{\bar{H}^2}||D_{11}||_\infty \leq \lambda \leq \frac{1}{8}. \tag{23}$$

Assume $H = \max_l H_l$ satisfies $H \leq C|\ln h|^{-1}$, as $h \to 0$. Let U^n be the domain decomposition solution computed on a strip decomposition with $K+1$ subdomains. Then

$$\max_n ||u^n - U^n|| \leq C(\Delta t + h^2 + KH(\Delta t + h^2 + H^2)). \tag{24}$$

where C depends on the smoothness of u, D_*, and D^*, but not on h, H, or Δt.

When considering only stability, an upper bound of 5/24 can be used in (23). In practice, we have used an upper bound of 5/12 without loss of stability. So the constraint does not seem to be tight.

The extension of the algorithm to a patch decomposition is straightforward. We now solve implicit/explicit equations of the form (13) along x interfaces, and solve equations analogous to (13) along y interfaces. At a point where an x and a y interface intersect, we have two point values (one from the x interface calculation, one from the y) representing the value of the approximate solution at the point. In our current implementation, we have taken the value computed by the y interface calculation as the value of the solution at an intersection point. This choice was made arbitrarily.

Denoting by H^x the minimum of the interface parameters H_l associated with $x-$interfaces ($x = const.$), and denoting by H^y the minimum of the $y-$interface parameters, we enforce the constraints

$$\frac{\Delta t}{(H^x)^2}||D_{11}||_\infty \leq \lambda \leq \frac{5}{12},$$

$$\frac{\Delta t}{(H^y)^2}||D_{22}||_\infty \leq \lambda \leq \frac{5}{12}.$$

If D is strongly anisotropic, then one of these constraints is overshadowed by the other. In this case, one is tempted to decompose Ω into strip subdomains in the direction with minimal diagonal entry in D; for example, if $D_{11} >> D_{22}$, one is tempted to use a multi-strip decomposition in the $y-$direction. Numerically, however, this can have adverse effects on the approximation properties of the scheme. We will examine these effects for test problems in the next section.

4. Numerical results. We now present some numerical results for the scheme described in Sections 2 and 3.

First, we verify numerically the rate of convergence of the scheme. We consider test problems of the form (1)-(3), with two coefficient matrices D,

$$D = D_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{25}$$

which reduces (1) to Poisson's equation, and

$$D = D_2 = \begin{bmatrix} 1+x & .5 \\ .5 & 1+y \end{bmatrix}. \tag{26}$$

With these choices of D we consider three test problems with known exact solutions. In Test Problem 1, we choose u^0, g, and f so that, independent of the choice of D, the true solution $u(x,y,t) \equiv u_1(x,y,t) = t + 16x(1-x)y(1-y)$. In Test Problem 2, we choose u^0, g, and f so that $u(x,y,t) \equiv u_2(x,t,y) = 10\,t\,sin(\pi x)sin(\pi y)$. In Test Problem 3, we modify the data in Test Problem 1 slightly so that $u(x,y,t) = u_3(x,y,t) = 10t^2 + 16x(1-x)y(1-y)$. Note that these test problems have solutions which are symmetric about $x = .5$ and $y = .5$. However, depending on the type of decomposition, our approach does not always preserve this symmetry; thus, we will investigate the effects of nonsymmetry on the approximate solution. In all of the runs below, the final time $T = .1$. We compare solutions using a 2x1 strip decomposition ($\bar{x} = .5$), and a 2x2 box decomposition ($\bar{x} = .5$, $\bar{y} = .5$), with the solution obtained using a fully implicit scheme; that is, no domain decomposition.

Examining the error estimate given in Theorem 1, it would appear that, asymptotically, Δt should be on the order of h^2 and H should be $\mathcal{O}(h^{2/3})$, assuming the constants in front of these terms are roughly of the same magnitude. Hence, in these simulations, we set $\Delta t = 4h^2$ and $H = 2h^{2/3}$. With these choices, the constraints given by (17), (23), and (25) are automatically satisfied for small h.

Convergence results for Test Problem 1, with $D = D_1$ and $D = D_2$, are given in Tables 1 and 2, respectively. Examining these results, we see that the error for the domain decomposition solutions is of the same magnitude, and even slightly smaller, than the fully implicit error. Moreover, the convergence rate in h is clearly approaching 2 as h goes to zero in all cases. In Figure 1, the 2x1 domain decomposition solution for $D = D_2$ (at T=.1 with $h^{-1} = 20$), along the lines $y = .5$ and $x = .5$, is compared with the exact solution along $x = .5$. Note that $u_1(.5,y,t) = u_1(x,.5,t)$ when $x = y$, however, the 2x1 domain decomposition solution U does not necessarily satisfy this relationship. Figure 1 demonstrates that even though the domain decomposition procedure is not symmetry-preserving in this case, the solution is virtually symmetric even on a coarse mesh.

Convergence results for Test Problem 2, with $D = D_1$ and $D = D_2$, are given in Tables 3 and 4, respectively. In these cases, the fully implicit solution on each mesh is more accurate than the domain decomposition solutions by roughly an order of magnitude; however, the domain decomposition errors are decreasing at a rate slightly higher than predicted by Theorem 1. In Figure 2, we plot the 2x1 domain decomposition solutions for Test Problem 2, with $D = D_2$, along the line $y = .5$, for $h^{-1} = 20$ and $h^{-1} = 40$, and compare these solutions with the true solution. As the figure and Table 4 demonstrate, the solution generated on the coarser mesh has substantial error, but is improved greatly by one mesh refinement. Moreover, the $h^{-1} = 40$ solution is symmetric to three significant digits.

Convergence results for Test Problem 3 with $D = D_2$ are given in Table 5. In this case the fully implicit solution again gives smaller error, especially on the coarser grids. However, the domain decomposition solution improves greatly as the grid is refined. For $h^{-1} = 160$, the errors for domain decomposition and fully implicit are roughly the same.

One reason the domain decomposition error is substantially larger than the fully implicit error in Test Problem 2 is because the truncation error associated with the interface calculation depends on the second partial derivatives of the solution in a neighborhood of the interface. In Test Problem

Table 1
L^2 errors for Test Problem 1, $D = D_1$.

Decomposition	h^{-1}	$\|\|U-u\|\| * 10^3$	Rate
2x1	20	2.67	-
-	40	.704	-
-	80	.181	-
-	160	.0461	1.95
2x2	20	2.46	-
-	40	.662	-
-	80	.173	-
-	160	.0448	1.93
1x1	20	3.05	-
-	40	.761	-
-	80	.190	2.00

Table 2
L^2 errors for Test Problem 1, $D = D_2$.

Decomposition	h^{-1}	$\|\|U-u\|\| * 10^3$	Rate
2x1	20	2.57	-
-	40	.684	-
-	80	.177	-
-	160	.0452	1.95
2x2	20	2.34	-
-	40	.639	-
-	80	.168	-
-	160	.0438	1.92
1x1	20	3.00	-
-	40	.748	-
-	80	.187	2.00

Table 3
L^2 errors for Test Problem 2, $D = D_1$.

Decomposition	h^{-1}	$\|\|U-u\|\| * 10^3$	Rate
2x1	20	20.0	-
-	40	3.8	-
-	80	.887	-
-	160	.190	2.23
2x2	20	27.6	-
-	40	5.85	-
-	80	1.43	-
-	160	.318	2.14
1x1	20	2.65	-
-	40	.660	-
-	80	.165	2.00

TABLE 4
L^2 errors for Test Problem 2, $D = D_2$.

Decomposition	h^{-1}	$\|\|U-u\|\| * 10^3$	Rate
2x1	20	19.3	-
-	40	3.53	-
-	80	.788	-
-	160	.166	2.27
2x2	20	25.9	-
-	40	5.30	-
-	80	1.24	-
-	160	.270	2.18
1x1	20	2.75	-
-	40	.687	-
-	80	.172	2.00

TABLE 5
L^2 errors for Test Problem 3, $D = D_2$.

Decomposition	h^{-1}	$\|\|U-u\|\| * 10^3$	Rate
2x1	20	2.34	-
-	40	.399	-
-	80	.080	-
-	160	.017	2.36
2x2	20	3.30	-
-	40	.578	-
-	80	.108	-
-	160	.021	2.43
1x1	20	.997	-
-	40	.248	-
-	80	.062	-
-	160	.015	2.00

TABLE 6
L^2 errors as $\bar{x}$ is varied

$\bar{x}$	h^{-1}	$\|\|U-u\|\| * 10^3$
.5	20	11.8
-	40	2.95
-	80	.74
.4	20	26.2
-	40	5.27
-	80	1.25
.3	20	29.9
-	40	7.47
-	80	1.76

2, the second partials in space of u are maximized along the interfaces $x = .5$ and $y = .5$, in fact, the maximum occurs at the point (.5,.5). This suggests that, for a given mesh, the positions of the subdomain interfaces can effect the error in a substantial way. We examine this effect by considering a test problem with the same initial and boundary data as Test Problem 2, with f modified so that the true solution $u(x,y,t) = u_3(x,y,t) = 10\,t\,sin(2\pi x)sin(2\pi y)$. In this case, the second order spatial derivatives of u are zero along $\bar{x} = .5$, and are maximized at $x = .25$, $y = .25$. We consider a 2x1 domain decomposition, and vary the position of the interface $\bar{x}$. In Table 6, we examine the error for different mesh spacings and different values of $\bar{x}$. As expected, for a given mesh spacing, the error grows as $\bar{x}$ approaches .25, and is smallest at $\bar{x} = .5$. The differences in errors are substantial; the errors for $\bar{x} = .3$ are almost three times larger than the errors for $\bar{x} = .5$. For this test problem, the errors for the fully implicit solution were almost identical to the domain decomposition errors for the case $\bar{x} = .5$.

Next, we examine the effects of anisotropy in the matrix D on the domain decomposition algorithm. We consider Test Problems 1 and 3, and take

$$D = D_3 = \begin{bmatrix} 50(1+x) & .5 \\ .5 & 1+y \end{bmatrix}. \tag{27}$$

Hence, $||D_{11}||_\infty = 100$, $||D_{22}||_\infty = 2$, and we enforce the stability constraints

$$100\frac{\Delta t}{(H^x)^2} \leq \frac{5}{12},$$
$$2\frac{\Delta t}{(H^y)^2} \leq \frac{5}{12}.$$

Based on these constraints, it would seem advantageous from an efficiency viewpoint to decompose Ω into strips along the $y-$direction, or to use rectangular subdomains, with fewer interfaces along the x-axis than the $y-$axis. Thus, larger time steps can be taken, without having to take large values of H^x. However, the truncation error associated with each y interface involves D_{11}, while the truncation error associated with each x interface involves D_{22}, which is a much smaller term. Thus, it is not clear how the overall error will be affected. To examine the effect on overall error, we generate solutions using 2x1 ($\bar{x} = .5$), 1x2 ($\bar{y} = .5$), and 1x4 ($\bar{y}_l = l * .25$, l =1,2,3) domain decompositions. We have chosen a "reasonable" time step, and chosen H^x and H^y so that the inequalities above are satisfied. The errors for the different domain decompositions are given in Table 7. In these runs, $h^{-1} = 80$, and $\Delta t = 6.25 * 10^{-4}$. For both test problems, the 2x1 decomposition gives slightly smaller error than the 1x2 decomposition, even though H^x is four times larger than H^y. Moreover, the error for the 1x4 decomposition is substantially increased. These results suggest that some care should be taken in choosing a decomposition when the coefficient matrix is highly anisotropic.

Finally, we present timings of runs made on an Alliant FX/8 computer, which demonstrate the parallelism of the scheme. Here we compare a fully implicit solution generated on a single processor,

TABLE 7
L^2 errors for highly anisotropic D

Test Problem	Decomposition	H^x	H^y	$\|\|U - u\|\| * 10^4$
1	2x1	36h	-	1.37
-	1x2	-	9h	1.75
-	1x4	-	9h	7.69
3	2x1	36h	-	1.39
-	1x2	-	9h	1.69
-	1x4	-	9h	12.80

TABLE 8
Timings for Test Problem 1, $D = D_1$.

Decomposition	CPU	Speed-up factor
1x1	491.2	-
2x1	245.5	2.0
2x2	122.2	4.0
4x1	124.2	3.96

with domain decomposition solutions where the number of processors used equals the number of subdomains. In these runs, $h^{-1} = 80$, and $\Delta t = .001$. The timings for Test Problem 1 with $D = D_1$ and $D = D_2$ are given in Tables 8 and 9, respectively. The times reported were averaged over several runs. As these tables indicate, essentially perfect speed-ups are obtained for two and four processors. When using all eight processors, we have not obtained a factor of eight in speed-up (closer to a factor of seven); however, this seems partly due to the configuration of the computer.

5. Conclusions. In conclusion, we have presented a numerical study of a domain decomposition algorithm for solving parabolic equations. We have demonstrated the convergence of the scheme, and studied the effects of interface position and anisotropy on the error. We have also shown that the scheme can be implemented successfully on a parallel computer.

One drawback of the method presented here which we haven't discussed is that it is not globally conservative. By "not globally conservative" we mean that, in the case of homogeneous Neumann boundary conditions, with $f = 0$, the domain decomposition solution U does not necessarily satisfy

$$\int_\Omega U(t) = \int_\Omega U(0),$$

which the true solution does satisfy. The method presented here also seems constrained by geometry and approximating space, and is only first order in time. We are currently in the process of studying these issues, and in fact, an algorithm has been derived [3] which improves upon this scheme in some areas. This new approach computes flux boundary data on the interfaces, instead of Dirichlet boundary data, as presented here. This scheme is conservative and can also be modified so that it is second order in Δt. Determining the pros and cons of these various approaches will be the subject of future work.

REFERENCES

[1] Dawson, C. N. and Q. Du, *A finite element domain decomposition method for parabolic equations*, Rice Technical Report TR90-21, Department of Mathematical Sciences, Rice University, Houston, Texas.

[2] Dawson, C. N., Q. Du, and T. F. Dupont, *A finite difference domain decomposition procedure for numerical solution of the heat equation*, University of Chicago Dept. of Computer Science Tech. Report 89-09, to appear in Math. Comp.

[3] Dawson, C. N. and T. F. Dupont, *An implicit/explicit, conservative Galerkin domain decomposition procedure for parabolic equations*, to appear.

TABLE 9
Timings for Test Problem 1, $D = D_2$.

Decomposition	CPU	Speed-up factor
1x1	709.5	-
2x1	353.8	2.0
2x2	176.8	4.0
4x1	179.2	3.96

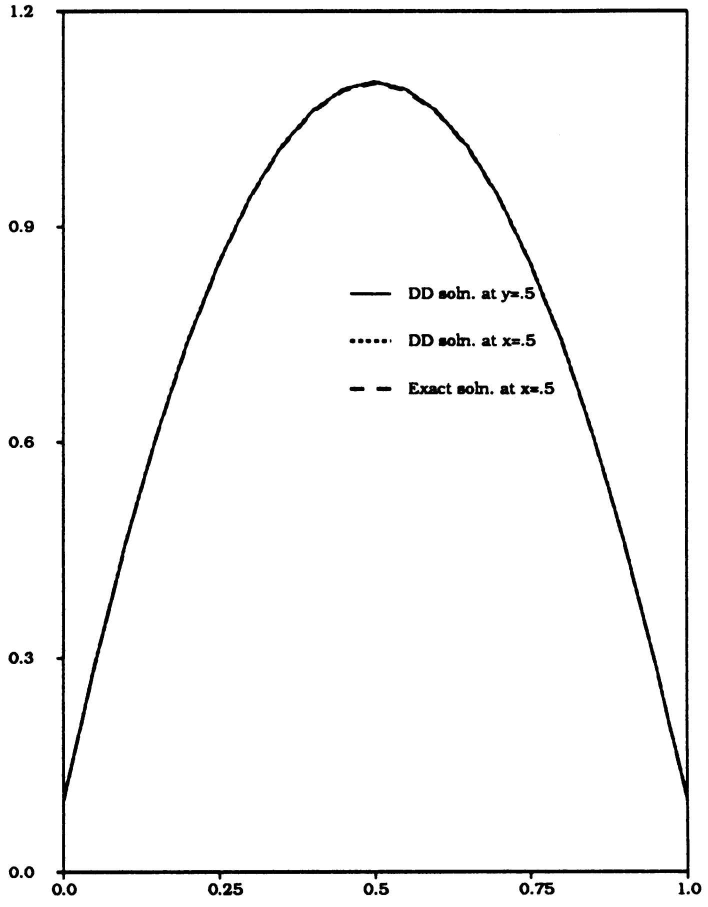

FIG. 1. *True solution vs. approximate solution for Test Problem 1*

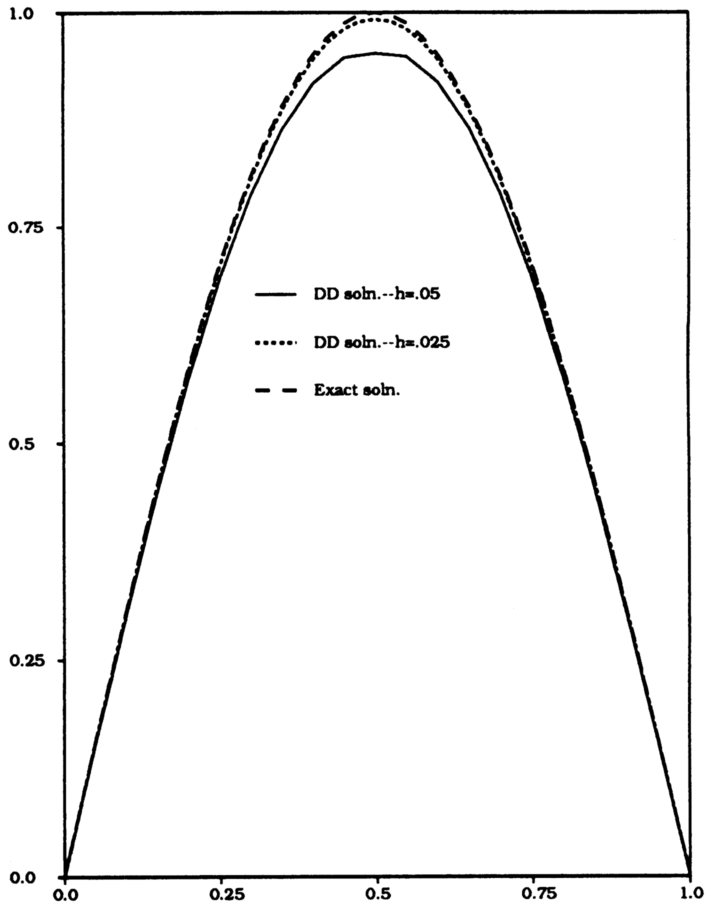

FIG. 2. *True solution vs. approximate solution for Test Problem 2*