WIP # 9
62
(Title Change)

UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Domain Decomposition Benefitial Even Sequentially

Tony F. Chan
Danny Goovaerts

June 1988

CAM Report 88-18

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

Domain Decomposition Beneficial Even Sequentially

TONY F. CHAN[1]
DANNY GOOVAERTS[2]
June 1988

Abstract. In the past several years, domain decomposition has been a very popular topic, motivated by the ease of parallelization. However, the question of whether it is better than parallelizing some standard sequential methods has never been directly addressed.

In this paper we show that the answer is affirmative in the case of iterative solutions of elliptic problems by preconditioned conjugate gradient iteration. Specifically, we show how to construct effective preconditioners based on the domain decomposition principle which, in addition to having all the advantages of domain decomposition, also result in better convergence rates than the analogous preconditioners on the whole domain.

We show some numerical examples to illustrate our point.

1. Introduction

In the past several years, domain decomposition methods for solving elliptic partial differential equations have attracted much attention. The main impulse for the enormous interest in these methods has come from the arrival of parallel computers. Besides the ease of parallelization, domain decomposition allows one to treat complex geometries or to isolate singular parts of the domain.

In the majority of domain decomposition methods, the matching of the solution on the subdomains to an overall solution is done by an iterative process. A large class is based on the preconditioned conjugate gradiant method for solving the reduced equations on the interfaces between the subdomains. The efficiency of these methods is determined by the preconditioner used. This approach involves a solve on each subdomain in each iteration step and the cost could be expensive if the number of iterations is not kept at a minimum. Based on this observation, doubts have been raised on the efficiency of these methods as compared to a parallelization of traditional preconditioned conjugate gradient iterations on the whole domain.

One aspect that has generally been ignored is the gain in sequential computational complexity that domain decomposition can yield as a divide and conquer technique. When the work for solving a problem grows more than linearly with its size, splitting it up in 2 subproblems of half the size will yield a faster method *provided* that the subsolutions can be efficiently combined to obtain the solution of the original problem. In this paper, we propose a method that achieves this goal. We use an approach that performs iterations on the whole domain, not just on the interface.

Specifically, in section 2, given a method for constructing a preconditioner $\widehat{M}$ on the whole domain (such as the ILU and MILU type methods, which have a superlinear computational complexity), we show how to construct a domain decomposed preconditioner based on applying the same method in the subdomains and using an appropriately chosen preconditioner for the interface. We hereby stress the importance of the preconditioner on the interface, for a badly chosen one can affect the overall convergence rate adversely. One that we have found to be very successfull is the Boundary Probe preconditioner [7]. This domain decomposed preconditioner for the original domain can yield a faster convergence rate with roughly the same operation count per iteration step, as compared to traditional preconditioners.

[1]Department of Mathematics, UCLA, Los Angeles, CA 90024.
[2]Department of Computer Science, K. U. Leuven, Celestijnenlaan 200A, B-3030 Leuven. This research was done during a visit to the Department of Mathematics, UCLA, January 1988 – June 1988.
The research of the authors is supported in part by The Department of Energy under contract DE-FG03-87ER25037, by the National Science Foundation under contract NSF-DMS87-14612 and by the Army Research Office under contract No. DAAL03-88-K-0085. The second author is holder of a Research Assistantship of the National Fund for Scientific Research Belgium (NFWO/FNRS). In addition, his visit UCLA was sponsored by a travel grant of the National Fund for Scientific Research Belgium (NFWO/FNRS).

In section 3 we stress the importance of an appropriate factorization of the subdomain preconditioners in order to minimize the number of arithmetic operations per iteration step. In section 4 we show some numerical experiments to illustrate our main points.

Another instance of where a parallel algorithm , when executed sequentially, turns out to be better than the traditional sequential algorithms is the parallel method for solving the symmetric eigenvalue problem as proposed in [12].

2. Domain Decomposed Preconditioners.

We formulate this approach for the simplest case of a domain Ω split into two subdomains Ω_1 and Ω_2 sharing the interface Γ. Consider the problem :

$$\begin{aligned} Lu &= f \quad \text{on} \quad \Omega \\ u &= u_b \quad \text{on} \quad \partial\Omega \end{aligned}$$

where L is a linear second order elliptic operator.

If we order the unknows for the internal points of the subdomains first and those on the interface Γ last, then the discrete solution vector $u = (u_1, u_2, u_3)^T$ satisfies the linear system :

$$Au = \begin{pmatrix} A_{11} & & A_{13} \\ & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} \quad , \tag{2-1}$$

where the discrete vector $f = (f_1, f_2, f_3)^T$ contains the contribution of the right hand side of the differential equation and of the Dirichlet boundary condition.

System (2–1) can be solved by Block Gaussian Elimination which gives the equations for the interface variables u_3 :

$$Su_3 = \hat{f}_3 \quad , \tag{2-2}$$

with

$$S = A_{33} - A_{31}A_{11}^{-1}A_{13} - A_{32}A_{22}^{-1}A_{23}$$

and

$$\hat{f}_3 = f_3 - A_{31}A_{11}^{-1}f_1 - A_{32}A_{22}^{-1}f_2 \quad .$$

The matrix S is the Schur complement of A_{33} in the matrix A. It corresponds to the reduction of the operator L on Ω to an operator on the internal boundary Γ. Constructing the Schur complement would require the solution on n_Γ elliptic problems on each subdomain, where n_Γ is the number of internal points on Γ. Furthermore it is dense, so that factoring would be expensive.

Instead of solving the system (2–2) directly, iterative methods such as preconditioned conjugate gradient (PCG) can be applied in which only matrix vector product Sy are required.This product can be computed by one solve on each subdomain with boundary condition on Γ determined by y.

Several preconditioners have been proposed in the literature. A large class of preconditioners have been derived for the splitting of a rectangle and are intimately related to the underlying properties of the differential operator [13], [16], [3], or are based on symmetry properties of the operator and of the domain [1]. Therefore, these preconditioners may prove unsatisfactory for general elliptic operators on general geometries. For a thorough discussion of these preconditioners we refer to [3] and [18].

Since each iteration involves the solution of problems on the subdomains, keeping the number of iterations small is very important for the efficiency of the method. Even requiring only a modest number of iterations, this can make the method slower than analogous methods on the whole domain. An obvious remedy for easing this disadvantage is using inexact solves on the subdomains.

One point of view is to consider the PCG iteration on the Schur complement as a combination of an outer and an inner iteration [14] [17]. Another approach is to combine preconditioners on the subdomains

and on the interface into a preconditioner on the whole domain, and thus to iterate on the subdomains and the interface simultanuously. We will use the latter approach in this paper.

The matrix A can be factored in block form :

$$A = \begin{pmatrix} A_{11} & & \\ & A_{22} & \\ A_{31} & A_{32} & S \end{pmatrix} \begin{pmatrix} I & & A_{11}^{-1}A_{13} \\ & I & A_{12}^{-1}A_{23} \\ & & I \end{pmatrix} \quad . \tag{2-3}$$

This is the LU factorization corresponding to the block Gaussian elimination which led to the Schur complement. We can derive a preconditioner for A by replacing A_{ii} in (2–3) by approximations B_{ii} and replacing the Schur complement by a preconditioner M. For the latter, we can take any of the preconditioners that were derived for PCG on the Schur complement. We therefore arrive at the following preconditioner :

$$\widetilde{M} = \begin{pmatrix} B_{11} & & \\ & B_{22} & \\ A_{31} & A_{32} & M \end{pmatrix} \begin{pmatrix} I & & B_{11}^{-1}A_{13} \\ & I & B_{12}^{-1}A_{23} \\ & & I \end{pmatrix} \quad . \tag{2-4}$$

Preconditioners of this form were first proposed in [2] and were also mentioned in [8], and [24].

Carrying out the matrix multiplication in (2–4) gives :

$$\widetilde{M} = \begin{pmatrix} B_{11} & & A_{13} \\ & B_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} + M - \tilde{S} \end{pmatrix} \tag{2-5}$$

with

$$\tilde{S} = A_{33} - A_{31}B_{11}^{-1}A_{13} - A_{32}B_{22}^{-1}A_{23} \quad . \tag{2-6}$$

Equation (2–5) shows that if M is a good approximation to $\tilde{S}$, we end up with with a preconditioner $\widetilde{M}$ which approximates the problem on the subdomains but keeps the original coupling between the subproblems as expressed by the matrices A_{i3} and A_{3i}, $i = 1, 2$. $\tilde{S}$ incorporates all the information on the operator, the geometry of the problem and of the approximations on the subdomains. The preconditioner for $\tilde{S}$ thus should take into account these aspects. A badly chosen preconditioner for $\tilde{S}$ can affect the overall convergence rate adversely, as will be illustrated in section 4.

In order to get a good preconditioner for A, it is not enough that the condition numbers of $B_{ii}^{-1}A_{ii}$ and $M^{-1}\tilde{S}$ are close to one, but also the eigenvalues of $B_{ii}^{-1}A_{ii}$ and $M^{-1}\tilde{S}$ have to be clustered around 1. This possibly asks for a scaling of B_{ii}.

We have found that a good overall preconditioner for $\tilde{S}$ is the Boundary Probe Preconditioner [7]. Several experiments with this preconditioner have been performed [18]. The main motivation for this approach is the observation that, in the case of the Laplace operator, the elements of the matrix S decay rapidly away from the main diagonal [16]. It is therefore reasonable to consider a k diagonal approximation to S. However, it would not be efficient to calculate the elements of S in order to do this.

Instead, as proposed in [7], a $2k+1$ diagonal approximation to S can be constructed by multiplying S by $2k+1$ "probing" vectors $v_j, j = 1, \ldots, 2k+1$. The idea is motivated by sparse Jacobian evaluation techniques [11]. For the case $k = 0$ and $k = 1$ the probing vectors are the following :

$$\begin{aligned} k = 0 : v_1 &= (1, 1, 1, 1, 1, 1, 1, \ldots)^T \\ k = 1 : v_1 &= (1, 0, 0, 1, 0, 0, 1, \ldots)^T \\ v_2 &= (0, 1, 0, 0, 1, 0, 0, \ldots)^T \\ v_3 &= (0, 0, 1, 0, 0, 1, 0, \ldots)^T \end{aligned} \quad .$$

The case $k = 0$ corresponds to a scaling of each row of the matrix S by the sum of the elements of the row. For $k = 1$, if S were indeed tridiagonal, all of its elements would be found in the vectors Sv_j, $j = 1, 2, 3$. For the more general case of $k > 1$, we refer to [9].

This approach for probing a matrix is valid for any matrix S and will yield a good banded approximation, provided that the elements of S decay away from the main diagonal. In the case of the Schur complement this is inherently related to the fact that the operator S is predominantly local. If the discretization stencil used extends over 1 gridline in each direction, the most important coupling of a grid point on the interface will be with its immediate neighbours, and a tridiagonal approximation for S will suffice. This idea can also be generalized for situations where the discretization stencil is wider, as for instance in fourth order equations such as the biharmonic equation [4], or in the multidomain case [9].

The probing technique asks for $2k + 1$ products Sv_j. This implies $(2k + 1)$ solves on each subdomain. However, as indicated by (2–5), it suffices to have a preconditioner for $\tilde{S}$ given by (2–6). Since $\tilde{S}$ approximates S it will also have the property that it is dominantly local. When the matrices A_{ii} are replaced by approximate fctorizations, which correspond to more local operators, the diagonals of $\tilde{S}$ will even decay more. So the probing technique can be applied here also. However, the product $\tilde{S}v_j$ now only involves approximate solves and thus can be performed cheaply.

3. Minimizing the operation count per iteration step

In each step of a preconditioned conjugate gradient iteration, the following system has to be solved:

$$\widetilde{M} z^k = r^k \qquad . \tag{3–1}$$

Straightforward implementation of this step using the block LU factorization (3–4) involves 2 solves on each subdomain with the matrix B_{ii} in each iteration step, one in the forward elimination step and one in the backsubstitution. This implies that the work per iteration step roughly doubles as compared to a PCG iteration with the same preconditioner on the whole domain. This is a relatively high price for a domain decomposed method to pay in the sequential case as it would ask for a reduction of the number of iteration steps by a factor of 2, which is not achievable for many problems. In the parallel case, to be competitive, the gain from ease of parallelization must compensate for this factor when compared with a straigthforward parallelization of a traditional PCG method for the whole domain [19]. A similar situation also occurs for the domain decomposed fast Poisson solver on a rectangle [5].

Here we describe a general technique to save this factor of 2 by an appropriate factorization and an appropriate ordering of the unknowns. The matrices B_{ii} can generally be expressed in a LU factorization :

$$B_{ii} = L_i U_i \qquad .$$

This then gives the following factored form for the preconditioner $\widetilde{M}$:

$$\widetilde{M} = \begin{pmatrix} L_1 & & \\ & L_2 & \\ A_{31}U_1^{-1} & A_{32}U_2^{-1} & M \end{pmatrix} \begin{pmatrix} U_1 & & L_1^{-1}A_{13} \\ & U_2 & L_2^{-1}A_{23} \\ & & I \end{pmatrix} \qquad . \tag{3–2}$$

The solution of (3–1) can now be performed as follows :

Forward elimination : Solve

$$\begin{aligned} L_1 y_1 &= r_1^k \\ L_2 y_2 &= r_2^k \\ M y_3 &= r_3^k - A_{31}U_1^{-1}y_1 - A_{32}U_2^{-1}y_2 \end{aligned}$$

Backsubstitution : Solve

$$
\begin{aligned}
z_3^k &= y_3 \\
U_1 z_1^k &= y_1 - L_1^{-1} A_{13} z_3^k \\
U_2 z_2^k &= y_2 - L_2^{-1} A_{23} z_3^k
\end{aligned}
$$

Written this way, the solution of (3–1) still takes 4 solves with the matrices L_i and 4 with U_i, or two solves on each subdomain. However, in this form, it is easy to show how one can save the factor of 2.

Since the matrices A_{3i} are very sparse, the product $A_{3i}U_i^{-1}y_i$ only involves a few components of the vectors $U_i^{-1}y_i$. More specifically, for a 5 point stencil, we only need the components of $U_i^{-1}y_i$ adjacent to the gridpoints on the internal interface.

Ordering the unknowns linewise in the direction of the internal edge, with the lines ordered going towards the interface, we obtain that the product $A_{3i}U_i^{-1}y_i$ only requires the bottom right hand corner of U_i^{-1}. The rest of the solution in the interior is not needed. An analogous assertion holds for the product $L_i^{-1}A_{i3}z_3^k$.

For a rectangular domain, splitted into 2 rectangles with the interface in the x-direction, this ordering of the lines corresponds to a UL factorization on the top and a LU factorization on the bottom domain with the usual ordering on lines from bottom to top.

A similar technique was also applied in the efficient implementation of a domain decomposed fast poisson solver on a rectangle, based on Fourier analysis of the Schur complement [6].

When the domain is split into more than two subdomains, the same technique can be applied using "twisted factorizations" on the subdomains with two internal edges (factoring starting from the middle of the domain towards the two edges simultanuously). However, it is not clear that these twisted factorizations will give approximations to the matrices A_{ii} that are comparable to the approximations by traditional factorizations in LU or UL form. The effect of these factorizations on the domain decomposed preconditioner is being investigated.

Similar techniques to save the factor of two were used by G. Meurant in the construction of domain decomposed preconditioners based on block preconditioners ([21], [22], [23]) using alternating LU and UL factored preconditioners on the subdomains. In the 2 subdomain case, this corresponds to the technique we propose.

4. Numerical experiments

In this section we present some numerical experiments that will illustrate the point of view that we made earlier. We consider second order self adjoint differential operators :

$$
Lu = \frac{\partial}{\partial x}\left(a(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(b(x,y)\frac{\partial u}{\partial y}\right) + c(x,y)u \qquad .
$$

The equation is discretized using a standard 5 point second order discretization stencil [15] on a equidistant grid. In our examples, the domain Ω is the unit square and is divided in two rectangular strips :

$$
\begin{aligned}
\Omega_1 &= [0,1] \times [0,0.5] \\
\Omega_2 &= [0,1] \times [0.5,1] \qquad .
\end{aligned}
$$

Taking n internal gridlines in each direction with gridsize $h = \frac{1}{n+1}$, the system for the whole domain is of the order $n \times n$ and for the subdomains of the order $n \times n/2$. The interface has n internal points. For the preconditioner on the interface, we consider the following choices :

(1) $M = M_D = \delta\sqrt{K}$, where K is the square root of the discrete one dimensional Laplacian on the interface [13] and δ is a scaling factor related to the coefficients of the differential operator ($\delta = a(0.5, 0.5)$),

(2) $M = M_P(\tilde{S}, 1)$, the boundary probe preconditioner on $\tilde{S}$ with $k = 1$ (tridiagonal approximation).

As approximate factorizations on the subdomains we use the ILU preconditioner [20], denoted by $B_{ii} \leftarrow$ ILU, or the MILU preconditioner with $\alpha = h^2$ [15], denoted by $B_{ii} \leftarrow$ MILU. PCG iteration with the ILU preconditioner has a condition number of $O(h^{-2})$ and with the MILU preconditioner it is of $O(h^{-1})$. Therefor, we can expect that the divide and conquer effect in domain decomposition will yield a faster method.

Example 1 :
In this example, we illustrate the importance of using a preconditioner for the interface that captures the characteristics of the approximate Schur complement $\tilde{S}$. The operator, the gridsize and the preconditioners for this example are summarized in the following table.

$$\begin{aligned} a &= e^{5xy} \quad , \quad h = 1/16, \quad M = M_D, \quad B_{ii} \leftarrow \text{MILU} \\ b &= e^{-5xy} \quad , \quad n = 15 \quad , \quad M = M_P \\ c &= \frac{1}{1+x+y} \end{aligned}$$

The rate of convergence of the PCG method is determined by the eigenvalue spectrum of $\widetilde{M}^{-1}A$. The eigenspectra for different blocks appearing in the product $\widetilde{M}^{-1}A$ are plotted in figure 1.

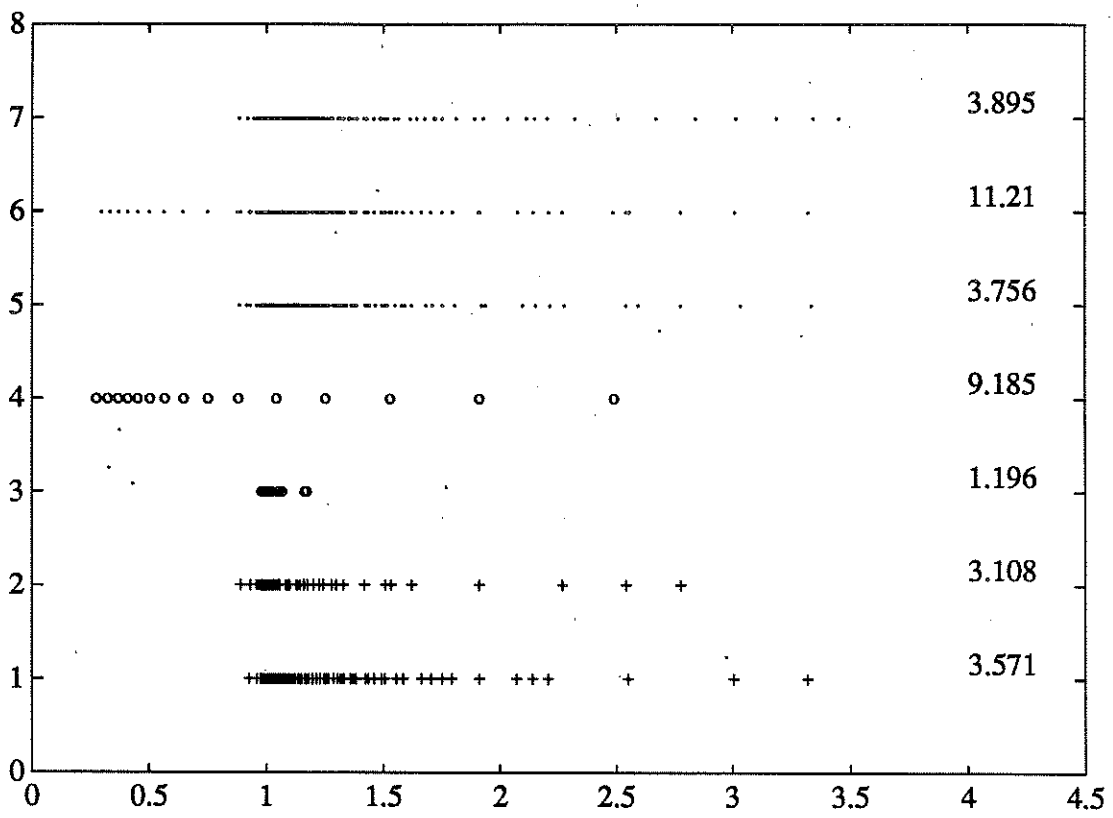
8
7
6
5
4
3
2
1
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
3.895
11.21
3.756
9.185
1.196
3.108
3.571

Figure 1

Number	Matrix
1	$B_{11}^{-1}A_{11}$
2	$B_{22}^{-1}A_{22}$
3	$M_P^{-1}\tilde{S}$
4	$M_D^{-1}\tilde{S}$
5	$\widetilde{M}_P^{-1}A$
6	$\widetilde{M}_D^{-1}A$
7	$\widehat{M}^{-1}A$

$\widetilde{M}_P$ and $\widetilde{M}_D$ are the domain decomposed preconditioners using M_P, and M_D respectively on the interface. $\widehat{M}$ is the MILU preconditioner on the whole domain Ω, with the natural ordering of the unknowns. The numbers to the right of the eigenspectra are the condition numbers, $\kappa = \lambda_{max}/\lambda_{min}$, of these matrices. Comparing lines 3 and 4, one immediately sees that the probing preconditioner yields a very good approximation to $\tilde{S}$, while M_D does not. We also notice that the eigenspectrum of $\widetilde{M}^{-1}A$ is a small perturbation of the union of the eigenspectra of $B_{11}^{-1}A_{11}$, $B_{22}^{-1}A_{22}$ and $\widetilde{M}^{-1}\tilde{S}$. In the case of $M = M_P$, the spectrum of $M_P^{-1}\tilde{S}$ is tightly clustered around 1 and lies fully within the spectra of $B_{11}^{-1}A_{11}$ and $B_{22}^{-1}A_{22}$. This leads to a good spectrum for $\widetilde{M}_P^{-1}A$ with a condition number that is smaller than for the sequential preconditioner. As the rate of convergence of the preconditioned conjugate gradient method is proportional to $(\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$ [10], this gives a faster convergence. In the case of $M = M_D$, the spectrum of $M_D^{-1}\tilde{S}$ is very wide and the condition number of $\widetilde{M}_D^{-1}A$ is very much larger. For a extensive discussion of the eigenvalue spectrum of $\widetilde{M}^{-1}A$ in relation to the spectra of $B_{11}^{-1}A_{11}$ and $B_{22}^{-1}A_{22}$, we refer to [9].

Example 2 :

In this example we show that our domain decomposed preconditioner yields a faster convergence than an analogous preconditioner applied to the whole domain for smooth problems.

The problem is the following :

$$\begin{aligned} a &= \mathrm{e}^{xy} \quad , \quad h = 1/32, \quad M = M_P, \quad B_{ii} \leftarrow \mathrm{ILU} \\ b &= \mathrm{e}^{-xy} \quad , \quad n = 31 \quad , \quad B_{ii} \leftarrow \mathrm{MILU} \\ c &= \frac{1}{1+x+y} \end{aligned}$$

The right hand side of the discrete equations is taken such that the exact discrete solution u_e satisfies :

$$(u_e)_{ij} = x_i \quad .$$

As starting guess we used $u_{ij}^0 = 1$.

In figure 2 we plot $e^k = \left\| u^k - u_e \right\|_2$ versus the iteration count k. We compare the following cases:

(1) full line : $\widehat{M} \leftarrow$ ILU
(2) dashed line : $\widetilde{M}$, with $B_{ii} \leftarrow$ ILU
(3) dotted line : $\widehat{M} \leftarrow$ MILU
(4) dashed-dotted line : $\widetilde{M}$, with $B_{ii} \leftarrow$ MILU,

where $\widehat{M}$ denotes the sequential preconditioner on the whole domain.

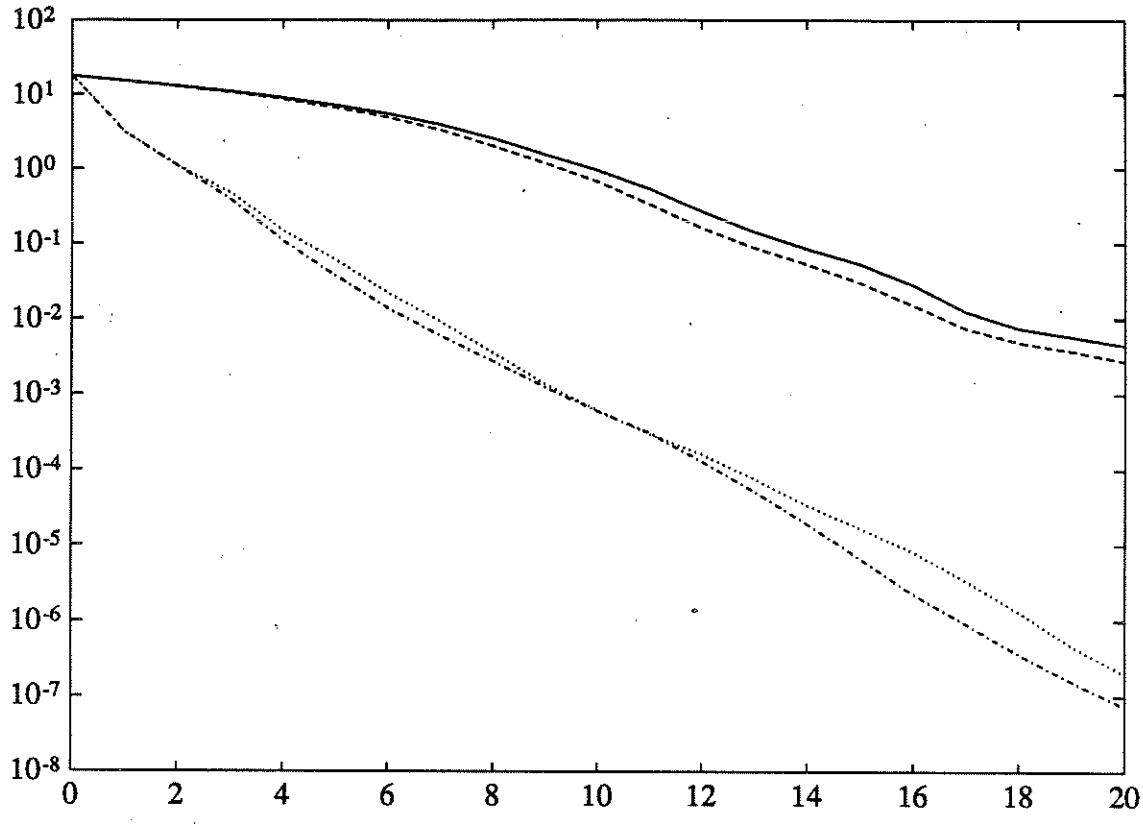
10^2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
0
2
4
6
8
10
12
14
16
18
20

Figure 2

For both choices for the preconditioner on the subdomains, the domain decomposed preconditioner has a slightly faster rate of convergence. Several other experiments that we have done with other operators, have all confirmed this point.

Example 3 :

In this example, we show a more dramatic gain, achieved for a problem with discontinuous coefficients. The coefficients of the equation and the preconditioners are :

$$\begin{aligned} &a = d(x,y), \quad h = 1/16, \quad M = M_P, \quad B_{ii} \leftarrow \text{ILU} \\ &b = d(x,y), \quad n = 15 \\ &c = 0 \quad , \end{aligned}$$

with

$$\begin{cases} d(x,y) = 1000 & y > 0.5 \quad , \\ d(x,y) = 500.5 & y = 0.5 \quad , \\ d(x,y) = 1 & y < 0.5 \quad . \end{cases}$$

The exact discrete solution and the starting guess are as in example 2.
Figure 3 gives $e^k = \left\| u^k - u_e \right\|_2$ versus the iteration count k. We compare the following preconditioners :

(1) full line : $\widehat{M} \leftarrow$ ILU
(2) dashed line : $\widetilde{M}$, with $B_{ii} \leftarrow$ ILU
(3) dotted line : ILU PCG on just one subdomain with the exact value of the solution on the interface.

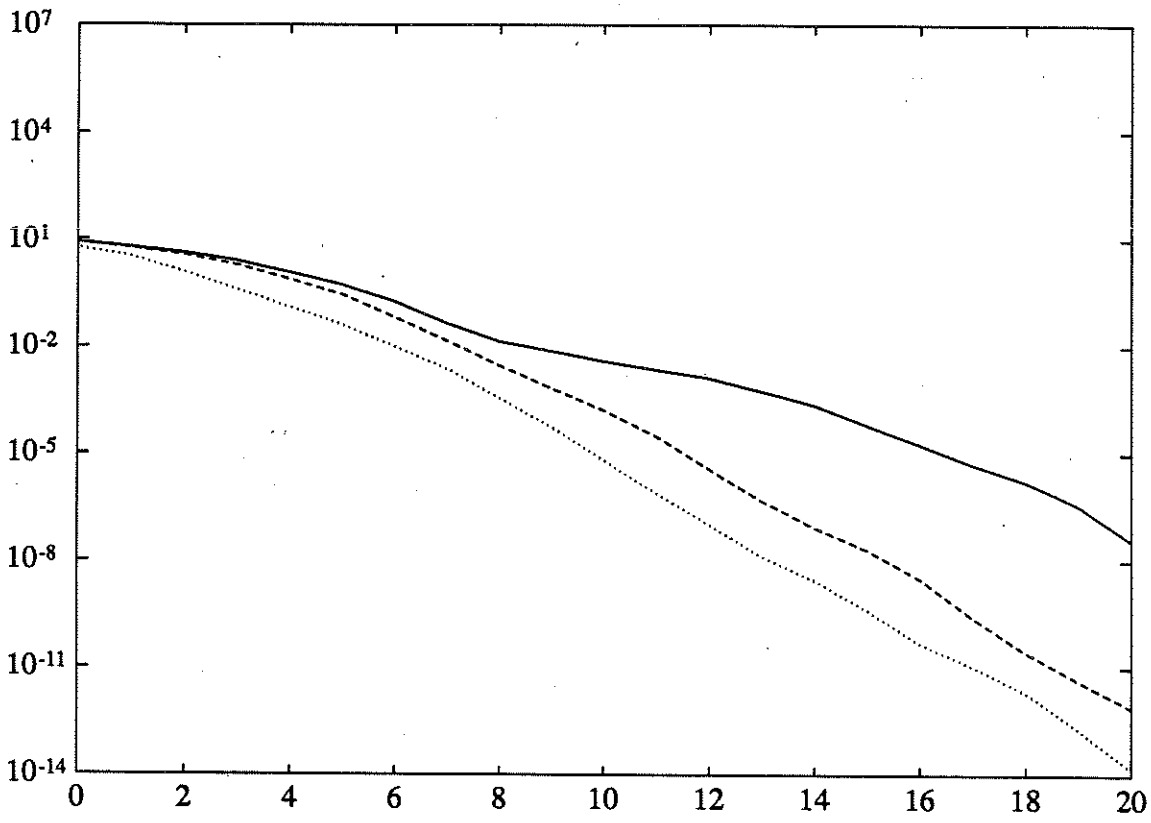

Figure 3

For this problem, the domain decomposed preconditioners yields a much better convergence than the analogous preconditioner on the whole domain. Comparing it with the PCG iteration on just one subdomain we see in the beginning of the iteration it is somewhat slower than PCG on the subdomain. After some iterations, the rate of convergence is about the same. This illustrates clearly how domain decomposition allows us to exploit the smoother coefficients within each subdomain.

5. Acknowledgments

The second author would like to express his profound appreciation for the hospitality and the inspiring environment provided by the Applied Mathematics Group and by prof. T. Chan especially during his stay at UCLA.

REFERENCES

1. P. E. Bjørstad and O. B. Widlund, *Iterative Methods for the Solution of Elliptic Problems on Regions Partitioned into Substructures*, SIAM J. Numer. Anal. **23** (1986), 1097–1120.
2. J. H. Bramble, J. E. Pasciak and A. H. Schatz, *The Construction of Preconditioners for Elliptic Problems by Substructures*, Math. Comp. **47** (1986), 103–134.
3. T. F. Chan, *Analysis of Preconditioners for Domain Decomposition*, SIAM J. Numer. Anal. **24** (1987), 382–390.
4. T. F. Chan, *Operator-Adaptive Domain Decompostion Preconditioners for 4^{th} Order Problems*, in "Proceedings of the Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, UCLA, January 14-16, 1988," T. F. Chan et al.(eds.) (to appear).
5. T. F. Chan and D. Resasco, *A Domain-Decomposed Fast Poisson Solver on a Rectangle*, SIAM J. Sc. Stat. Comp. **8** (1987), s14–s26.
6. T. F. Chan and D. Resasco, *Hypercube Implementation of Domain Decomposed Fast Poisson Solvers*, in "Proceedings of the Second Conference on Hypercube Multiprocessors, Knoxville, August, 1986," M. Heath (ed.), SIAM publ., 1987, pp. 738–746.
7. T. F. Chan and D. Resasco, *A survey of Preconditioners for Domain Decomposition*, in "Proceedings of the IV Coloquio de Matematicas, Taller de Analisis Numerico y sus Aplicaciones, Taxco, Guerrero, Mexico, August, 1985," to be published by Springer Verlag.
8. T. F. Chan and D. Resasco, *A Framework for the Analysis and Construction of Domain Decomposition Preconditioners*, in "Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Paris, France, January 7-9, 1987," R. Glowinski, G. H. Golub, G. A. Meurant and J. Périaux (eds.), SIAM, Philadelphia, 1988, pp. 217–230.
9. T. F. Chan and D. Goovaerts, *Domain Decomposition Methods with Inexact Subdomain Solves*, UCLA Computational and Applied Math Report (to appear).
10. P. Concus, G. H. Golub and D. O'Leary, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, in "Sparse Matrix Computation," J. Bunch and D. Rose (eds.), Academic Press, New-York, 1976, pp. 309–322.
11. A. R. Curtis, M. J. D. Powell and J. K. Reid, *On the Estimation of Sparse Jacobian Matrices*, J. Inst. Maths. Applics. **13** (1974), 117–119.
12. J. Dongarra and D. Sorensen, *A Fully Parallel Algorithm for the Symmetric Eigenvalue Problem*, SIAM J. Sc. Stat. Comp. **8** (1987), s139–s154.
13. M. Dryja, *A Capacitance Matrix Method for Dirichlet Problem on Polygonal Region*, Numer. Math. **39** (1982), 51–64.
14. R. S. Dembo, S. C. Eisenstat and T. Steihaug, *Inexact Newton Methods*, SIAM J. Numer. Anal. **19** (1982), 400–408.
15. T. Dupont, R. Kendall and H. H. Rachford, *An Approximate Factorization Procedure for Solving Self-Adjoint Elliptic Difference Equations*, SIAM J. Numer. Anal. **5** (1968), 559–573.
16. G. H. Golub and D. Mayers, *The Use of Pre-Conditioning over Irregular Regions*, in "Proceedings of the 6th Internl. Conf. Comput. Meth. Sci. Engng., Versailles, France, December 12-16, 1983."
17. G. H. Golub and M. L. Overton, *The Convergence of Inexact Chebyshev and Richardson Iterative Methods for Solving Linear Systems*, Research Report CMA-R46-86, Centre for Mathematical Analysis, Australian National University, December 1986.
18. D. E. Keyes and W. D. Gropp, *A Comparison of Domain Decomposition Techniques for Elliptic Partial Differential Equations*, SIAM J. Sc. Stat. Comp. **8** (1987), s166–s202.
19. D. E. Keyes and W. D. Gropp, *Domain Decomposition Techniques For Nonsymmetric Systems of Elliptic Boundary Value Problems : Examples from CFD*, in "Proceedings of the Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, UCLA, January 14-16, 1988," T. F. Chan et al. (eds.) (to appear).
20. J. A. Meijerink and H. A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-Matrix*, Math. of Comp. **31** (1977), 148–162.

21. G. Meurant, *Domain Decomposition Versus Block Preconditioning*, in "Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Paris, France, January 7-9, 1987," R. Glowinski, G. H. Golub, G. A. Meurant and J. Périaux (eds.), SIAM, Philadelphia, 1988, pp. 231–249.
22. G. Meurant, *Domain Decomposition Preconditioners for the Conjugate Gradient Method*, submitted for publication to Calcolo.
23. G. Meurant, *Domain Decomposition Methods for Partial Differential Equations on Parallel Computers*, in "Proceedings of the Second International Conference on Vector and Parallel Processing, Tromso, Norway, June 6-10, 1988" (to appear).
24. O. B. Widlund, *Iterative Substructuring Methods: Algorithms and Theory for Elliptic Problems in the Plane*, in "Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Paris, France, January 7-9, 1987," R. Glowinski, G. H. Golub, G. A. Meurant and J. Périaux (eds.), SIAM, Philadelphia, 1988, pp. 113–128.

Keywords. parallel algorithms, domain decomposition, partial differential equations, preconditioned conjugate gradient
1980 *Mathematics subject classifications*: 65N20, 65F10