35

Additive Schwarz for the Schur Complement Method

Luiz M. Carvalho and Luc Giraud

1 Introduction

Domain decomposition methods for solving elliptic boundary problems have been receiving increasing attention for the last two decades. To a large extent this is due to their potential application on new parallel computers.

We describe here two domain decomposition algorithms based on nonoverlapping subregions for solving self-adjoint elliptic problems in two dimensions and we report on some experimental results. Both alternatives can be viewed as block diagonal preconditioners for the Schur complement matrix. The first is the classical block Jacobi where all but one of the diagonal blocks are related to the interfaces, the remaining block is diagonal and corresponds to the cross-points. The second preconditioner introduces an overlap between the blocks by including the cross-points and their couplings in the diagonal blocks of the block Jacobi preconditioner. In this case, the block related to the cross-points is dropped. We will refer to the latter as Algebraic Additive Schwarz (AAS) since we sum the contributions of each block on the overlap.

The nonzero sub-blocks of the Schur complement are dense matrices, we consider approximations which are inexpensive to construct and invert. The algebraic approximation of the interface operator is constructed by using the probing technique studied in [CM92] for the two-subdomain case and extended in [CMS92] for many subdomains. The proposed preconditioner belongs to the one-level type preconditioners as, for instance, Dirichlet-Neumann [BW86] and Neumann-Neumann [RT91]. Therefore, it does not implement any coarse grid component. Currently, most preconditioners include a coarse correction to propagate the error globally. We refer the reader to [BPS86, Smi90, Man93] for a detailed presentation of some of these preconditioners and to [SrG96, CM94] for complete surveys of domain decomposition methods. We should stress that the AAS approach, which first appeared in a few numerical experiments in [GT93], improves the convergence rate of the well-known block Jacobi on some model problems while retaining the same computational complexity.

First, in Section 2, we describe the AAS preconditioner. In Section 3, we present

Ninth International Conference on Domain Decomposition Methods
Editor Petter E. Bjørstad, Magne S. Espedal and David E. Keyes

the parallel implementation of both preconditioners on distributed memory platforms, and compare their performance in Section 4. We conclude in Section 5.

2 The AAS Preconditioner

We consider the following 2^{nd} order self-adjoint elliptic problem on an open polygonal domain Ω included in $\mathbb{R}^2$:

$$\begin{cases} -\frac{\partial}{\partial x}(a(x,y)\frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(b(x,y)\frac{\partial u}{\partial y}) = f(x,y) & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{2.1}$$

where $a(x,y),\ b(x,y) \in \mathbb{R}^2$ are positive functions on Ω. We assume that the domain Ω is partitioned into N nonoverlapping subdomains $\Omega_1, \dots, \Omega_N$ with boundaries $\partial\Omega_1, \dots, \partial\Omega_N$. We discretise (2.1) by either finite differences or finite elements resulting in a symmetric and positive definite linear system $Au = f$. Grouping the points corresponding to the interfaces between the subdomains (B) in the vector u_B and the ones corresponding to the interior (I) of the subdomains in u_I, we get the reordered problem:

$$\begin{pmatrix} A_{II} & A_{IB} \\ A_{IB}^T & A_{BB} \end{pmatrix} \begin{pmatrix} u_I \\ u_B \end{pmatrix} = \begin{pmatrix} f_I \\ f_B \end{pmatrix} . \tag{2.2}$$

Eliminating u_I from the second block row of (2.2) leads to the following reduced equation for u_B:

$$S u_B = f_B - A_{IB}^T A_{II}^{-1} f_I , \qquad \text{where} \qquad S = A_{BB} - A_{IB}^T A_{II}^{-1} A_{IB} \tag{2.3}$$

is referred to as the Schur complement matrix.

Let

$$B = (\bigcup_{i=1}^{m} E_i) \cup V, \tag{2.4}$$

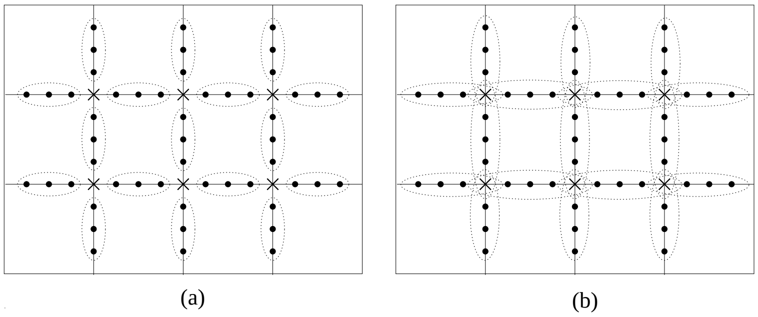
be a partition of the interface B into edge points, E_i, (depicted with $\bullet$ in Figure 1) and vertex points, V, ($\times$ points in Figure 1); E_i can be written as $E_i = (\partial\Omega_j \cap \partial\Omega_l) - V$, with $j \neq l$. Let R_{E_i} and R_V denote the pointwise restriction maps from B onto the nodes on E_i and on V, and let $R_{E_i}^T$ and R_V^T be the corresponding extension maps, respectively. The diagonal block associated with E_i is defined by

$$S_{ii} = R_{E_i} S R_{E_i}^T, \ \ i = 1, \dots, m. \tag{2.5}$$

Let $\tilde{S}_{ii}$ be an approximation for S_{ii}, $\ i = 1, \dots, m$. Then, the approximate block Jacobi preconditioner, bJ, can be represented by

$$bJ = \sum_{i=1}^{m} R_{E_i}^T \tilde{S}_{ii}^{-1} R_{E_i} + R_V^T \tilde{S}_{VV}^{-1} R_V . \tag{2.6}$$

Figure 1 Decomposition of the interface points (a) the block Jacobi and (b) the AAS.



Let

$$B = \bigcup_{i=1}^{m} \hat{E}_i \tag{2.7}$$

be another decomposition of B, where $\hat{E}_i = \partial\Omega_j \cap \partial\Omega_l$ (see Figure 1 (b)). $\hat{E}_i$ and $\hat{E}_j$ may overlap on one point belonging to V, so $\hat{E}_i \cap \hat{E}_j \neq \emptyset$ for some $i \neq j$. Let $\hat{R}_{E_i}$ denote the pointwise restriction map from B onto the nodes on $\hat{E}_i$ and $\hat{R}_{E_i}^T$ the corresponding extension map. We define

$$\breve{S}_{ii} = \hat{R}_{E_i} S \hat{R}_{E_i}^T, \quad i = 1, \dots, m. \tag{2.8}$$

Let $\hat{S}_{ii}$ be an approximation for $\breve{S}_{ii}$, $\; i = 1, \dots, m$. Then, we consider the Algebraic Additive Schwarz (AAS) preconditioner defined as:

$$AAS = \sum_{i=1}^{m} \hat{R}_{E_i}^T \hat{S}_{ii}^{-1} \hat{R}_{E_i} \tag{2.9}$$

3 Parallel Implementation

We use the probe technique to construct $\tilde{S}_{ii}$ and $\hat{S}_{ii}$ [CM92]. Let $P = [p_i]$ be a matrix whose columns are the probe vectors p_i. Therefore, the probe technique can be applied by multiplying S with P. This matrix-matrix approach only requires two communications for the construction and is more efficient than a more classical approach based on a sequence of matrix-probing vector products. Moreover, the matrix-matrix approach allows the algorithm to overlap part of the communication with some computation.

The implementation of $\hat{R}_{E_i}^T \hat{S}_{ii}^{-1} \hat{R}_{E_i}$ in AAS requires two extra neighbour-neighbour communications to exchange information on the cross-points; these communications are not needed for block Jacobi.

The classical PCG requires two synchronisations to compute the inner products; we have implemented the variant proposed in [DER93] where only one synchronisation per iteration is needed.

4 Experimental Results

The elliptic problem (2.1) was discretised by the standard five-point difference stencil on an $(n \times n)$ mesh. The approximations $\tilde{S}_{ii}$ and $\hat{S}_{ii}$ are tridiagonal matrices built using twelve probing vectors [CG97]. The direct solver for the solution of the local Dirichlet problems A_{ii}, $i = 1 \ldots N$, is MA27 [DR83], while the tridiagonal matrices $\tilde{S}_{ii}$ and $\hat{S}_{ii}$ are factorised using LAPACK routines. The stopping criterion is to decrease the relative residual to less than 10^{-5}. The grid is either uniform or nonuniform. The message passing library used is MPI. The parallel platform is a Cray T3D (IDRIS - France). In the parallel experiments, we have used as many processors as subdomains and the computations have been performed in 64 bit precision.

Figure 2 Definition of the function $a(.,.)$ on Ω, the unit square of $\mathbb{R}^2$.

a(.,.) $= 10^{-3}$	a(.,.) $= 10^{3}$	a(.,.) $= 10^{-3}$
a(.,.) $= 10^{3}$	a(.,.) $= 10^{-3}$	a(.,.) $= 10^{3}$

Table 1 gives the number of iterations of the PCG. Table 2 gives the total parallel elapsed time, in seconds, for solving $Ax = b$; that is, the analysis and factorisation steps of MA27, the preconditioner construction, the PCG iterations, and the computation of the global solution by local back substitutions.

Table 1 Number of iterations of the PCG on a 257×257 grid.

	AAS			block Jacobi		
# subdomains	Poisson	Aniso.	Nonunif.	Poisson	Aniso.	Nonunif.
4	17	11	14	18	11	16
16	23	29	27	24	35	37
64	33	52	40	35	82	60

Table 1 shows that AAS performs better than block Jacobi in the presence of anisotropic phenomena; the increase in the number of iterations is less for AAS as the number of subdomains grows larger. We have observed such behaviour for a wide class of model test problems, but we only report two of these results here. For all the experiments the function $b(.,.)$ has been set to one. In the first example, the anisotropy is physically present in the definition of the problem through the function

Table 2 Times (in sec.) on the Cray T3D for solving $Ax = b$ on a 257×257 grid.

	AAS			block Jacobi		
# subdomains	Poisson	Aniso.	Nonunif.	Poisson	Aniso.	Nonunif.
4	10.77	9.64	10.18	10.97	9.64	10.57
16	2.17	2.43	2.33	2.20	2.66	2.72
64	0.58	0.78	0.65	0.59	1.07	0.84

$a(.,.)$ as defined in Figure 2, while in the second example the anisotropy is numerically introduced in the Poisson problem by the discretisation on a nonuniform grid. This grid is used to solve the drift-diffusion equations [GT93] involved in MOSFET device simulation.

As a first remark, we indicate that the time per iteration is almost the same for PCG with both preconditioners; the neighbour-neighbour communications required by AAS do not penalise its performance on the Cray T3D.

We can observe in Table 2 that the reduction in the number of PCG iterations is not directly reflected in the time reduction. This is due to the significant amount of time required by the MA27 analysis and the factorisation routines for the solution of the local problems $A_{ii},\ i = 1 \ldots N$, which are common to both preconditioners. These routines are responsible for 20% of the global solution time in the case of 64 subdomains. Consequently, the global time is not a linear function of the number of iterations but is an affine function; that is, the global parallel solution time is equal to the time for the symbolic and numerical factorisation of the local Dirichlet matrices, plus the time for the back solution (once the interface problem has been solved), plus the number of iterations times the time per iteration (which is the same for both preconditioners).

Table 3 shows the performance of the parallel device simulation code, using PCG with either AAS or block Jacobi for the solution of the linear systems involved in the nonlinear solver. Both numerical and physical anisotropies are encountered and AAS performs better than block Jacobi.

Table 3 Times in seconds and number of iterations for a continuation step of the MOSFET device simulation on a 257×257 grid.

	AAS		block Jacobi	
# subdomains	time	# iterations	time	# iterations
4	857.6	1520	861.6	1543
16	214.3	2731	240.1	3422
64	67.5	4877	88.9	7300

The AAS preconditioner can easily be extended by using a larger overlap between the edges E_i. Experimental results have shown that this does not significantly accelerate the convergence for the MOSFET problem. In Table 4, we vary the size of the overlap and report the total number of linear iterations involved in the solution of the drift-diffusion equations.

Table 4 Iterations using preconditioners with overlap sizes 0 (block Jacobi), 1(AAS), 2 and 3 on a 129×129 grid for the MOSFET problem.

	# points in the overlap between the $\hat{E}_i$			
# subdomains	0	1	3	5
4	1660	1693	1678	1671
16	3610	2842	2797	2720
64	8385	5177	5419	5599

5 Conclusions

We have compared a so-called Algebraic Additive Schwarz preconditioner (AAS) with an approximate block Jacobi preconditioner for the Schur complement domain decomposition method for solving elliptic PDEs. We have shown that AAS performs better both in terms of number of iterations and in computing time, for problems with anisotropic phenomena. This behaviour has been illustrated when solving linear systems that arise from model problems and from the real life application of device modelling.

Although AAS does not introduce a global coupling among the subdomains, the growth of the number of iterations is slower than the one observed with block Jacobi, when the number of subdomains increases. AAS can be a cheap alternative to improve the simple block Jacobi preconditioner when it is difficult to define or to implement a coarse problem in the preconditioner applied to the Schur complement system.

For a detailed presentation of this work we refer to [CG97].

Acknowledgement

We would like to thank the referee whose comments helped to improve the presentation of this paper.

REFERENCES

[BPS86] Bramble J., Pasciak J., and Schatz J. (1986) The construction of preconditioners for elliptic problems by substructuring I. *Math. Comp.* 47(175): 103–134.

[BW86] Bjørstad P. E. and Widlund O. B. (1986) Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM Journal on*

Numerical Analysis 23(6): 1093–1120.

[CG97] Carvalho L. M. and Giraud L. (1997) Parallel domain decomposition for device modelling (in preparation). Technical report, CERFACS, Toulouse, France.

[CM92] Chan T. F. and Mathew T. P. (1992) The interface probing technique in domain decomposition. *SIAM J. Matrix Analysis and Applications* 13.

[CM94] Chan T. F. and Mathew T. P. (1994) *Domain Decomposition Algorithms*, volume 3 of *Acta Numerica*, pages 61–143. Cambridge University Press, Cambridge.

[CMS92] Chan T., Mathew T., and Shao J.-P. (1992) Fourier and probe variants of the vertex space domain decomposition algorithm. In Keyes D., Chan T., Meurant G., Scroggs J., and Voigt R. (eds) *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 236–249. SIAM, Phil.

[DER93] D'Azevedo E., Eijkhout V., and Romine C. (1993) Reducing communication costs in the conjugate gradient algorithm on distributed memory multiprocessors. Technical Report CS-93-185, Computer Science Department, University of Tennessee, Knoxville.

[DR83] Duff I. S. and Reid J. K. (1983) The multifrontal solution of indefinite sparse symmetric linear systems. *TOMS* 9: 302–325.

[GT93] Giraud L. and Tuminaro R. (1993) Domain decomposition algorithms for the drift-diffusion equations. In Sincovec R., Keyes D., Leuze M., Petzold L., and Reed D. (eds) *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing*, pages 719–726. SIAM.

[Man93] Mandel J. (1993) Balancing domain decomposition. *Comm. Numer. Meth. Engrg.* 9: 233–241.

[RT91] Roeck Y.-H. D. and Tallec P. L. (1991) Analysis and test of a local domain decomposition preconditioner. In Glowinski R., Kuznetsov Y., Meurant G., Périaux J., and Widlund O. (eds) *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*. SIAM, Philadelphia, PA, USA.

[Smi90] Smith B. F. (1990) *Domain decomposition algorithms for the partial differential equations of linear elasticity*. PhD thesis, Courant Institute of Mathematical Sciences, New York.

[SrG96] Smith B., rstad P. B., and Gropp W. (1996) *Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, New York, 1st edition.