Parallel Subdomain-Based Preconditioner for the Schur Complement

Luiz M. Carvalho[1]* and Luc Giraud[2]

[1] PESC-COPPE/UFRJ - Caixa Postal 68511
21945-970-Rio de Janeiro, RJ, Brasil
carvalho@cos.ufrj.br
[2] CERFACS - 42, av. Gaspard Coriolis
31057 - TOULOUSE - FRANCE
giraud@cerfacs.fr

Abstract. We present a new parallelizable preconditioner that is used as the local component of a two-level preconditioner similar to BPS. On 2D model problems that exhibit either high anisotropy or discontinuity, we demonstrate its attractive numerical behaviour and compare it with the regular BPS. To alleviate the construction cost of this new preconditioner, that requires the computation of the local Schur complements, we propose a cheap alternative based on Incomplete Cholesky factorization, that reduces the computational cost but retains the good numerical features of the preconditioner.

1 Introduction

The solution of elliptic problems is challenging on parallel distributed memory computers as their Green's functions are global. This problem is often tackled via domain decomposition techniques, using two-level preconditioners. In the framework of non-overlapping domain decomposition techniques, we refer for instance to BPS (Bramble, Pasciak and Schatz) [2], Vertex Space [7, 13], and to some extent Balancing Neumann-Neumann [10, 11], as well as FETI [8], for the presentation of major two-level preconditioners. We refer to [5] and [14] for a more exhaustive overview of domain decomposition techniques.

In this work, we consider non-overlapping domain decomposition techniques, and two-level preconditioners for the conjugate gradient method. These preconditioners can be written similarly to BPS [2], that is, as the sum of local and global components. We focus on a new local preconditioner that solves the assembled local Schur complement on the whole interface of each subdomain.

In Section 2, we briefly describe non-overlapping domain decomposition techniques and the class of two-level preconditioners we considered here. The main goal of that section is to formulate algebraically the sub-domain based preconditioner. In the next section, we introduce the 2D model problems used to

* The work of this author was partially supported by FAPERJ-Brazil under grant 130.117/98.

P. Amestoy et al. (Eds.): Euro-Par'99, LNCS 1685, pp. 1032–1039, 1999.

benchmark the preconditioners, those model problems exhibit high anisotropy and high discontinuity. Numerical experiments are reported in Section 3 and, finally, some concluding remarks are reported.

2 Preconditioner Description

This section is two-fold. First, we formulate a two-dimensional elliptic model problem. Then, we introduce the preconditioners we studied.

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

$$\begin{cases} -\frac{\partial}{\partial x}(a(x,y)\frac{\partial v}{\partial x}) - \frac{\partial}{\partial y}(b(x,y)\frac{\partial v}{\partial y}) = F(x,y) \; in \quad \Omega, \\ \qquad\qquad\qquad\qquad\qquad\qquad v = 0 \qquad \text{on} \qquad \partial\Omega \end{cases} \tag{1}$$

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 non-overlapping subdomains $\Omega_1, \ldots, \Omega_N$ with boundaries $\partial\Omega_1, \ldots, \partial\Omega_N$; this defines a coarse mesh, τ^H, with mesh size H. We discretize (1) either by finite differences or finite elements resulting in a symmetric and positive definite linear system $Au = f$.

Let B be the set of all the indices of the discretized points which belong to the interfaces between the subdomains. Grouping the points corresponding to 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:

$$\begin{pmatrix} A_{II} & A_{IB} \\ A_{IB}^T & A_{BB} \end{pmatrix} \begin{pmatrix} u_I \\ u_B \end{pmatrix} = \begin{pmatrix} f_I \\ f_B \end{pmatrix} . \tag{2}$$

Eliminating u_I from the second block row of (2) leads to the following reduced equation for u_B:

$$Su_B = f_B - A_{IB}^T A_{II}^{-1} f_I \ , \quad \text{where} \quad S = A_{BB} - A_{IB}^T A_{II}^{-1} A_{IB} \tag{3}$$

is the Schur complement of the matrix A_{II} in A, and is usually referred to as the Schur complement matrix. For a stiffness matrix A arising from finite elements discretization the Schur complement matrix (3) can also be written as: $S = \sum_{i=1}^{N} S^{(i)}$, where $S^{(i)} = A_{BB}^{(i)} - (A_{IB}^{(i)})^T (A_{II}^{(i)})^{-1} A_{IB}^{(i)}$ only involves matrices computed locally on the finite elements in Ω_i. In Figure 1, we depicted a subdomain Ω_i with its edge interfaces E_m, E_j, E_k, E_ℓ; the local Schur complement matrix is dense and has the following block structure (for the sake of clarity, we do not consider the corner points):

$$S^{(i)} = \begin{pmatrix} S_{mm}^{(i)} & S_{mg} & S_{mk} & S_{m\ell} \\ S_{gm} & S_{gg}^{(i)} & S_{gk} & S_{g\ell} \\ S_{km} & S_{kg} & S_{kk}^{(i)} & S_{k\ell} \\ S_{\ell m} & S_{\ell g} & S_{\ell k} & S_{\ell\ell}^{(i)} \end{pmatrix} \tag{4}$$

The diagonal blocks represent the coupling between nodes on an edge interface. The off-diagonal blocks represent the coupling between each edge interface of Ω_i. Notice that the off-diagonals of $S^{(i)}$ are blocks actually existing in S, while the diagonal blocks are contributions to the diagonal blocks of the complete Schur complement matrix S. For instance, the diagonal block of the complete matrix S associated with the edge interface E_k is $S_{kk} = S_{kk}^{(i)} + S_{kk}^{(n)}$. We then obtain the local Schur complement assembled on the interface edges by:

$$\hat{S}^{(i)} = \begin{pmatrix} S_{mm} & S_{mg} & S_{mk} & S_{m\ell} \\ S_{gm} & S_{gg} & S_{gk} & S_{g\ell} \\ S_{km} & S_{kg} & S_{kk} & S_{k\ell} \\ S_{\ell m} & S_{\ell g} & S_{\ell k} & S_{\ell\ell} \end{pmatrix}. \tag{5}$$

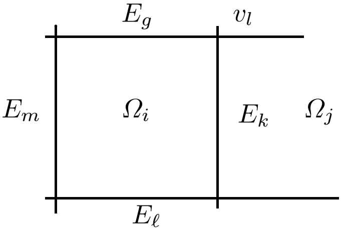

Fig. 1. Example of a regular 2D decomposition

We now describe the preconditioners, starting with BPS, followed by our new preconditioner. In this respect, we define a series of projection and interpolation operators. Specifically, for each E_i we define R_{E_i} as the standard pointwise restriction of nodal values on E_i. Its transpose extends grid functions in E_i by zero in the rest of the interface. Thus, $S_{ik} \equiv R_{E_i} S R_{E_k}^T$. Additionally, we define grid transfer operators between the interfaces and the coarse grid points in τ_H. R_0^T is an interpolation operator which corresponds to using interpolation between each set of edge endpoints (i.e. adjacent points in τ_H) to define values on the edge between the endpoints (i.e. edge interface E_i). R_0 is a projection operator and is the transpose of the interpolation operator. Finally, A_0 is a coarse grid approximation of the Schur complement operator on τ_H computed with to the Galerkin formula $A_0 = R_0 S R_0^T$. We refer to [3] and to the references therein for a more detailed description for the coarse grid component for this type of preconditioner.

With the above notation a close variant of the BPS preconditioner is given by $M_{BPS-E} = M_E + R_0^T A_0^{-1} R_0$ where $M_E = \sum_{E_i} R_{E_i}^T S_{ii}^{-1} R_{E_i}$ (in the original BPS A_0 is built from A and not from S as in our case). It can be interpreted as a generalized block Jacobi preconditioner for (3) augmented with a residual correction used on a coarse grid. The coarse grid correction term $R_0^T A_0^{-1} R_0$ allows global coupling to be incorporated between the interfaces. This global coupling is essential for scalability. In particular, it has been shown in [2] that, when applying the original BPS technique to a uniformly elliptic operator, the preconditioned system has a condition number

$$\kappa(M_{BPS}S) = \mathcal{O}(1 + \log^2(H/h)), \tag{6}$$

where h is the mesh size. This implies that the condition number depends only weakly on the mesh spacing and on the number of processors. Therefore, such a preconditioner is appropriate for large systems of equations on computers with a large number of processors.

The new preconditioner can be described in a similar way by defining another series of projection and interpolation operators. Specifically, for each subdomain Ω_i we define R_{Ω_i} as the standard pointwise restriction of nodal values on the interface of Ω_i. Its transpose extends grid functions on $\partial\Omega_i$ (the interface of Ω_i) by zero on the rest of the interface. Thus, $\hat{S}^{(i)} \equiv R_{\Omega_i} S R_{\Omega_i}^T$. With the above notation, we define the new preconditioner by $M_{BPS-S} = M_S + R_0^T A_0^{-1} R_0$ where $M_S = \sum_{\Omega_i} R_{\Omega_i}^T (\hat{S}^{(i)})^{-1} R_{\Omega_i}$.

In a distributed memory environment, the proposed preconditioner can be constructed almost at the same cost as regular BPS. More precisely, it requires the same amount of communication and a slight increase in the number of operations due to factorization of the dense assembled local Schur complement matrices rather than only their diagonal blocks. We address to [4] for details on the parallel implementation and time comparisons between regular BPS and our alternative.

Notice that the sub-domain based preconditioner M_S can be viewed as a Neumann-Neumann preconditioner [6] except that in our case the block diagonal coefficients of the local Schur complement matrices are assembled on each subdomain. Another difference with Neumann-Neumann is that the contribution of each subdomain is simply summed-up on each interface; for Neumann-Neumann a weighted sum is computed.
In fact, with the above notations, the Neumann-Neumann preconditioner, M_{NN}, can be written as: $M_{NN} = \sum_{\Omega_i} D_i (R_{\Omega_i}^T (S^{(i)})^{-1} R_{\Omega_i}) D_i$, where the matrices D_i are weight matrices defining a partition of unity (i.e. $\sum\limits_{\Omega_i} D_i = I$). For internal domains, $S^{(i)}$ is singular and pseudo-inverses $(S^{(i)})^+$ should be used instead. Assembling the diagonal blocks of the local Schur complement matrices $S^{(i)}$ removes this singularity.

Finally, this new local preconditioner can also be viewed as an Algebraic Additive Schwarz preconditioner for the Schur complement, since it corresponds to a block diagonal preconditioner with an overlap between the blocks.

3 Numerical Experiments

With a first part dedicated to the description of the problems we are dealing with, this section presents the numerical behaviour of the local preconditioners introduced in Section 1. The central issue is to compare the number of iterations of a preconditioned conjugate gradient method when solving (3).

3.1 Model Problems

We mainly address the solution of Equation (1) discretized by linear finite elements. Convergence of the preconditioned conjugate gradient method is attained when the 2-norm of the residual of the current iteration, normalized by the 2-norm of the right hand side, is less than 10^{-5}. The grid is uniform.

The background of our study is the numerical solution of drift-diffusion equations for the numerical simulation of semi-conductor devices. In this respect, we intend to evaluate the sensitivity of the preconditioners to anisotropy and to discontinuity. With this in mind, we consider the following 2D model problems. In Figure 2, we represent the unit square divided into five regions where piecewise constant functions are defined. Let a and b be the functions of the elliptic problem as described in Equation (1). With these notations, we can define different problems with different degrees of difficulty:

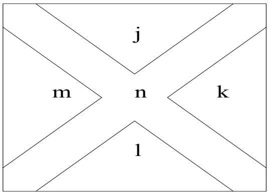

Fig. 2. A region where anisotropy and discontinuity are combined.

- Poisson problem: $a = 1$ and $b = 1$,
- anisotropic and discontinuous problems: $a = 1$ and $b = j, k, l, m, n$.
  - Scot-flag1 (*SF1*): j=l= 10^{-2}, k=m= 10^{2} and n=1.
  - Scot-flag2 (*SF2*): j=l= 10^{-3}, k=m= 10^{3} and n=1.

In addition, we have considered another set of problems where we have only introduced anisotropy not necessarily aligned with the x or y axis but that makes an angle θ with the x-direction. For $\theta = 0$, this corresponds to the classical model anisotropic equation:

$$\varepsilon \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \text{ with } \varepsilon \ll 1.$$

We have tested $\theta \in \{n\pi/8; n = 0, 1, 2\}$; because of the symmetry, the previous tests are actually for $n \in \{0, 1, 2, \ldots, 15\}$.

3.2 Experimental Results

The proposed local preconditioners are computationally expensive to construct as we need to form explicitly the exact local Schur complement $S^{(i)}$. To alleviate the cost of this computation, cheap approximations can be obtained by replacing the exact solution of the local Dirichlet $(A_{II}^{(i)})^{-1}$ problem by some approximation based either on approximated inverse like AINV [1] (that preserves the symmetry of the preconditioner while SPAI [9] usually does not) or by an Incomplete Cholesky factorization ILL^T [12]. In both cases, an approximation of $S^{(i)}$ can be computed by sparse matrix-matrix computation at a lower cost than the one to pay even if an efficient sparse factorization were used. In order to study the effect of this approximation on the quality of the resulting preconditioner, we display in the next tables both the number of iterations using the exact local Schur and the approximation via an incomplete Cholesky factorization (denoted by $\tilde{M}_{BPS-E}$ for the approximated standard BPS preconditioner and $\tilde{M}_{BPS-S}$ for the approximated new preconditioner, respectively).

We first benchmark the preconditioner on the classical Poisson problem to establish that the new preconditioner satisfy the condition number estimate given by Equation (6). In Table 1, we can see that the number of iterations does not depends on the number of subdomains (that is $\frac{1}{H}$) when the size of the subdomain remains constant (that is $\frac{H}{h}$) as predicted by (6). For that problem, a simple ILL^T without fill-in is used for constructing the approximation of the local Schur complement matrices. Furthermore, we observe that the approximation of the local Schur complements do not affect the quality of the preconditioner and that the new preconditioner does not behave better than the regular block Jacobi used for standard BPS.

	Poisson		
# subdomains	4×4	8×8	16×16
M_{BPS-E}	9	11	11
M_{BPS-S}	10	10	11
$\tilde{M}_{BPS-E}$	12	13	13
$\tilde{M}_{BPS-S}$	12	13	13

Table 1. Number of iterations of the preconditioners for the Poisson problem. Each subdomain is discretized using a 16×16 grid.

For non-uniform elliptic operators, the theoretical bound does not work anymore. It can be seen in the next tables where we report the number of iterations for the anisotropic and anisotropic/discontinuous problems. In Table 2 and 3, we observe that the number of iterations increases when the number of sub-domains increases from 16 up-to 256. For both anisotropic and discontinuous problems, the new sub-domain based preconditioner ensures a convergence of the conjugate gradient method in less iterations than M_{BPS-E}. M_{BPS-S} converges in 25% less iterations on the anisotropic problem and reaches 40% less on the discontinuous problem.

Although the new preconditioner is still better when using approximations of the local Dirichlet problem, the figures show a deterioration in the convergence of both $\tilde{M}_{BPS-E}$ and $\tilde{M}_{BPS-S}$. We have used an ILL^T factorization with the fill-in controlled through a threshold strategy to construct cheap approximations of the local Schur complement matrices. This threshold induces some fill-in in the approximated factors; the extra storage is only a factor two or three more than the original matrices. However, this extra fill-in was necessary to avoid an undesirable deterioration in the numerical behaviour of the resulting preconditioner.

	Scot-flag					
# subdomains	4 × 4		8 × 8		16 × 16	
	$SF1$	$SF2$	$SF1$	$SF2$	$SF1$	$SF2$
M_{BPS-E}	25	42	34	69	42	105
M_{BPS-S}	19	30	22	44	27	64
$\tilde{M}_{BPS-E}$	26	47	38	76	48	110
$\tilde{M}_{BPS-S}$	24	40	29	56	36	77

Table 2. # iterations for problems combining high anisotropy and high discontinuity. Each subdomain is discretized using a 16 × 16 grid.

	Anisotropy ($\epsilon = 10^{-3}$)		
# subdomains	4 × 4	8 × 8	16 × 16
M_{BPS-E}	20	28	35
M_{BPS-S}	17	21	26
$\tilde{M}_{BPS-E}$	21	29	35
$\tilde{M}_{BPS-S}$	17	21	25

Table 3. # iterations on the anisotropic problem with $\theta = \pi/4$. Each subdomain is a 16 × 16 grid.

4 Concluding Remarks

We have presented a new local parallelizable preconditioner that can be easily combined with the BPS coarse component to produce an efficient two-level preconditioner. We have shown for two difficult model problems that exhibit high anisotropy and/or discontinuity the effectiveness of this new preconditioner. To overcome its expensive construction, due to the exact explicit computation of the local Schur complements, we have proposed an alternative based on incomplete Cholesky factorization. This alternative enables a cheaper construction and only slightly deteriorates the efficiency of the resulting two-level preconditioners. Finally, the parallel implementation of the new preconditioner does not increase the complexity of the regular BPS, therefore it can be efficiently implemented on parallel distributed memory computers, we refer to [3] and [4] where parallel experiments are reported.

Acknowledgments

Part of this work was completed during a visit of the first author to CERFACS in January 1999. The hospitality and the support provided by CERFACS are greatly appreciated.

References

[1] M. Benzi, C. D. Meyer, and M. Tůma. A sparse approximate inverse preconditioner for the conjugate gradient method. *SIAM J. Sci. Comput.*, 17(5):1135–1149, 1996.

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

[3] L. Carvalho, L. Giraud, and P. Le Tallec. Algebraic two-level preconditioners for the Schur complement method. Tech. Rep. TR/PA/98/18, CERFACS, Toulouse, France, 1998. Submitted to SIAM J. Scientific Computing.

[4] L.M. Carvalho and L. Giraud. Block diagonal preconditioners for the Schur complement method. In M. Papadrakakis and B.H.V. Topping, editors, *Innovative computational methods for structural mechanics*, pages 61–82, Edinburgh, UK, 1999. Saxe-Coburg publications.

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

[6] Y.-H. De Roeck and P. Le Tallec. Analysis and test of a local domain decomposition preconditioner. In R. Glowinski, Y. Kuznetsov, G. Meurant, J. Périaux, and O. Widlund, editors, *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 112–128. SIAM, Philadelphia, PA, 1991.

[7] M. Dryja, B.F. Smith, and O.B. Widlund. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM J. Numer. Anal.*, 31(6):1662–1694, 1993.

[8] C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Int. J. Numer. Meth. Engng.*, 32:1205–1227, 1991.

[9] M. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.*, 18:838–853, 1997.

[10] P. Le Tallec. *Domain decomposition methods in computational mechanics*, volume 1 of *Computational Mechanics Advances*, pages 121–220. North-Holland, 1994.

[11] J. Mandel. Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9:233–241, 1993.

[12] T. A. Manteuffel. Shifted incomplete Cholesky factorization. In I. S. Duff and G. W. Stewart, editors, *Sparse Matrix Proceedings 1978*, Philadelphia, PA, 1979. SIAM Publications.

[13] B.F. Smith. *Domain Decomposition Algorithms for the Partial Differential Equations of Linear Elasticity.* PhD thesis, Courant Institute of Mathematical Sciences, September 1990. Tech. Rep. 517, Department of Computer Science, Courant Institute.

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