Incomplete Domain Decomposition Preconditioners for Nonsymmetric Problems

Gérard Meurant*

Abstract. The aim of this paper is to derive incomplete Domain Decomposition preconditioners that can be used on parallel computers with the Orthomin(1) method for solving non symmetric linear systems. We mainly show how to extend the techniques which have been presented in [13] and we present numerical results that demonstrate the usefulness of the preconditioners described in this paper.

1. Introduction. In the last years, there has been a great development of domain decomposition preconditioners for the conjugate gradient method. This new interest mainly comes from the fact that these methods can be easily and efficiently used on parallel computers with a large number of processors. Up to now, the research has been almost essentially directed towards finding good preconditioners for symmetric linear systems arising from finite difference or finite element discretizations of elliptic partial differential equations in two and three dimensional domains. Several papers adressing these issues have recently appeared : Bjorstad & Widlund [1], Bramble, Pasciak & Schatz [2], Golub & Mayers [9], Chan & Resasco [3], Meurant [13], [14], [15].

In this paper, we will show how to extend the techniques of [13] to the solution of non symmetric problems arising from the finite difference discretization of diffusion convection equations in two dimensional domains. As an acceleration of the basic linear iteration we will use the Orthomin(1) method (see for instance [7]). This method is not the best to solve this problem , but it will be just enough for our main purpose which is constructing preconditioners.

The outline of the paper is as follows. Section 2 introduces the model problem we are solving. In Section 3 we briefly present the tools we are using to construct the preconditioners, mainly how to approximate the inverse of a non symmetric tridiagonal matrix and the generalization of the basic block preconditioner INV (see [5]). Section 4 motivates our derivation exhibiting an exact DD solver and then, Section 5 shows how to derive an incomplete decomposition whose symmetric form was first described in [13]. We conclude in Section 6 with some numerical experiments for model problems with different ratios of the diffusion and convection coefficients.

2. The Model Problem. The problem we want to solve is a linear elliptic PDE,

$$-\frac{\partial}{\partial x}(a(x,y)\frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(b(x,y)\frac{\partial u}{\partial y}) + 2\alpha(x,y)\frac{\partial u}{\partial x} + 2\beta(x,y)\frac{\partial u}{\partial y} + cu = f,$$

* CEA, Centre d'Etudes de Limeil-Valenton, BP 27, 94190 Villeneuve St Georges, France

$$\text{in } \Omega \subset R^2, \quad u|_{\partial\Omega} = 0 \quad \text{or} \quad \frac{\partial u}{\partial n}|_{\partial\Omega} = 0,$$

Ω being a rectangle. With standard finite differences schemes (5 point) and row–wise ordering, this leads to a block tridiagonal linear system :

$$A\,x \;=\; b,$$

with

$$A = \begin{pmatrix} D_1 & B_1 & & & \\ A_2 & D_2 & B_2 & & \\ & \ddots & \ddots & \ddots & \\ & & A_{n-1} & D_{n-1} & B_{n-1} \\ & & & A_n & D_n \end{pmatrix}.$$

With a centered scheme, as we suppose $c \geq 0$, when $a \equiv 1$, $b \equiv 1$ and α, *beta* are constant, D_i is point tridiagonal strictly diagonally dominant if $\alpha h < 2 + \frac{ch^2}{2}$, A is diagonally dominant if $\alpha h < 1 + \frac{ch^2}{2}$ and $\beta h < 1 + \frac{ch^2}{2}$. A_i is a diagonal matrix. Under these conditions A is a non symmetric M-matrix.

The Orthomin(1) algorithm that we are using for solving our problem is the following,

$$x^0, r^0 = b - \overline{A}x^0, p^0 = r^0,$$

$$\alpha_k = \frac{(r^k, \overline{A}p^k)}{(\overline{A}p^k, \overline{A}p^k)},$$

$$x^{k+1} = x^k + \alpha_k p^k,$$

$$r^{k+1} = r^k - \alpha_k \overline{A}p^k,$$

$$\beta_k = -\frac{(\overline{A}r^{k+1}, \overline{A}p^k)}{(\overline{A}p^k, \overline{A}p^k)},$$

$$p^{k+1} = r^{k+1} + \beta_k p^k,$$

$$\overline{A}p^{k+1} = \overline{A}r^{k+1} + \beta_k \overline{A}p^k.$$

We use this method with either $\overline{A} = M^{-1}A$ or $\overline{A} = AM^{-1}$ where M is the preconditioner.

3.Tools. The first tool we use is how to solve tridiagonal linear systems with sparse right hand sides. This has been described in [13]for symmetric matrices. As there is no fundamental difference when the matrix is non symmetric, we refer the reader to this paper.

The second technique, which was developped for the symmetric case in Concus, Golub & Meurant [5] concerns approximating the inverse of a tridiagonal matrix. When T is a symmetric tridiagonal matrix, T^{-1} is approximated by a tridiagonal matrix $trid(T^{-1})$ whose non zero elements are the same as the corresponding ones of T^{-1}. The justification of this technique is given in [6]. Now let T be tridiagonal non symmetric,

$$T = \begin{pmatrix} a_1 & b_1 & & & \\ c_2 & a_1 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-1} & a_{n-1} & b_{n-1} \\ & & & c_n & a_n \end{pmatrix},$$

it is well known that we can symmetrize T by left multiplying with a diagonal matrix D,

$$D = \begin{pmatrix} 1 & & & \\ & \frac{b_1}{c_2} & & \\ & & \ddots & \\ & & & \frac{b_1 \cdots b_{n-1}}{c_2 \cdots c_n} \end{pmatrix}.$$

Then, $\overline{T} = DT$ is symmetric and we can apply the same techniques as before to $\overline{T}$.
If T is line diagonally dominant, so is $\overline{T}$, then the elements of T^{-1} decrease away from the diagonal on each column. If T is column diagonally dominant then the elements of T^{-1} decrease away from the diagonal on each line. We can approximate T^{-1} with $trid(\overline{T}^{-1})D$. For this approximation, we only need to store D and 2 diagonals of $\overline{T}^{-1}$. Another possibility will be to right symmetrize.
The third tool is the block (incomplete) factorization INV of Concus, Golub & Meurant [5] which can be extended to non symmetric matrices as follows . Suppose A is a block tridiagonal matrix, then the block Cholesky factorization of A can be written as

$$A = (\Delta + L)\ \Delta^{-1}\ (\Delta + U)$$

$$\Delta = \begin{pmatrix} \Delta_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \Delta_n \end{pmatrix},$$

$$L = \begin{pmatrix} 0 & & & \\ A_2 & 0 & & \\ & \ddots & \ddots & \\ & & A_n & 0 \end{pmatrix}, U = \begin{pmatrix} 0 & B_1 & & \\ & 0 & B_2 & \\ & & \ddots & \ddots \\ & & & 0 \end{pmatrix},$$

with

$$\begin{cases} \Delta_1 = D_1, \\ \Delta_i = D_i - A_i\ \Delta_{i-1}^{-1}\ B_{i-1}. \end{cases}$$

To construct an INV incomplete decomposition, we simply replace the inverse with a tridiagonal approximation,

$$\Delta_i = D_i - A_i\ trid(\Delta_{i-1}^{-1})\ A_{i-1}.$$

It follows that all the Δ_i are tridiagonal matrices. The stability of this incomplete factorization will be adressed in another paper [16].

4. An exact DD solver. To develop a DD method, as for the symmetric case, we partition the domain Ω into strips $\Omega_i, i = 1, \ldots, k$ and we renumber the unknowns in such a way that the components of x related to the subdomains appear first and then the ones for the interfaces. With this (block) ordering, the system can be written as

$$\begin{pmatrix} B_1 & & & & C_1 & & & \\ & B_2 & & & E_2 & C_2 & & \\ & & B_3 & & & E_3 & \ddots & \\ & & & \ddots & & & \ddots & C_{k-1} \\ & & & B_k & & & & E_k \\ Q_1 & R_2 & & & B_{1,2} & & & \\ & Q_2 & R_3 & & & B_{2,3} & & \\ & & \ddots & \ddots & & & \ddots & \\ & & & Q_{k-1} & R_k & & & B_{k-1,k} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_k \\ x_{1,2} \\ x_{2,3} \\ \vdots \\ x_{k-1,k} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_k \\ b_{1,2} \\ b_{2,3} \\ \vdots \\ b_{k-1,k} \end{pmatrix},$$

where each B_i is related to a subdomain Ω_i,

$$B_i = \begin{pmatrix} D_i^1 & B_i^1 & & & \\ A_i^2 & D_i^2 & B_i^2 & & \\ & \ddots & \ddots & \ddots & \\ & & A_i^{m_i-1} & D_i^{m_i-1} & B_i^{m_i-1} \\ & & & A_i^{m_i} & D_i^{m_i} \end{pmatrix}, \quad i = 1, \ldots, k.$$

D_i^j and $B_{i,j}$ are point tridiagonal, m_i is the number of mesh lines in Ω_i.
The matrices C_i, E_i, Q_i and R_i have a very special structure,

$$C_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ C_i^{m_i} \end{pmatrix}, \quad E_i = \begin{pmatrix} E_i^1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad Q_i = (0 \ \cdots \ 0 \ Q_i^{m_i}), \quad R_i = (R_i^1 \ 0 \ \cdots \ 0).$$

$C_i^{m_i}$, E_i^1, $Q_i^{m_i}$ and R_i^1 are diagonal matrices.
To derive an exact DD solver, we eliminate $x_1, \dots, x_k$ to get a reduced system involving only the unknowns for the interfaces.

$$\begin{pmatrix} B'_{1,2} & G_1 & & & \\ F_2 & B'_{2,3} & G_2 & & \\ & \ddots & \ddots & \ddots & \\ & & F_{k-2} & B'_{k-2,k-1} & G_{k-2} \\ & & & F_{k-1} & B'_{k-1,k} \end{pmatrix} \begin{pmatrix} x_{1,2} \\ x_{2,3} \\ \vdots \\ x_{k-2,k-1} \\ x_{k-1,k} \end{pmatrix} = \begin{pmatrix} b'_{1,2} \\ b'_{2,3} \\ \vdots \\ b'_{k-2,k-1} \\ b'_{k-1,k} \end{pmatrix}$$

It is easy to see that we have the following formulas for $i = 1, \dots, k-1$,

$$B'_{i,i+1} = B_{i,i+1} - Q_i \ B_i^{-1} \ C_i - R_{i+1} \ B_{i+1}^{-1} \ E_{i+1},$$

$$F_i = -Q_i \ B_i^{-1} \ E_i,$$
$$G_{i-1} = -R_i \ B_i^{-1} \ C_i,$$
$$b'_{i,i+1} = b_{i,i+1} - Q_i \ B_i^{-1} \ b_i - R_{i+1} \ B_{i+1}^{-1} \ b_{i+1}.$$

As for the symmetric case, to simplify these expressions, we use 2 factorizations for the matrices B_i coresponding to a subdomain, a block LU one (top–down) which can be written as

$$B_i = (\Delta_i + L_i) \ \Delta_i^{-1} \ (\Delta_i + U_i),$$

$$\begin{cases} \Delta_i^1 = D_i^1, \\ \Delta_i^j = D_i^j - A_i^j \ (\Delta_i^{j-1})^{-1} \ B_i^{j-1}. \end{cases}$$

and a block UL one (bottom–up),

$$B_i = (\Sigma_i + U_i) \ \Sigma_i^{-1} \ (\Sigma_i + L_i),$$

$$\begin{cases} \Sigma_i^{m_i} = D_i^{m_i}, \\ \Sigma_i^j = D_i^j - B_i^j \ (\Sigma_i^{j+1})^{-1} \ A_i^{j+1}. \end{cases}$$

With these notations, we have the following result,

Theorem

For $i = 1, \dots, k-1$,

$$B'_{i,i+1} = B_{i,i+1} - Q_i^{m_i} \ (\Delta_i^{m_i})^{-1} \ C_i^{m_i} - R_{i+1}^1 \ (\Sigma_{i+1}^1)^{-1} \ E_{i+1}^1.$$

If

$$S_i^1 = (\Sigma_i^1)^{-1} \ E_i^1,$$
$$S_i^l = -(\Sigma_i^l)^{-1} \ A_i^l \ S_i^{l-1}, \ l = 2, \dots, m_i,$$

then,

$$F_i = -C_i^{m_i} \ S_i^{m_i},$$

Similarly

$$T_i^{m_i} = (\Delta_i^{m_i})^{-1}\, C_i^{m_i},$$

$$T_i^l = -(\Delta_i^l)^{-1}\, B_i^l\, T_i^{l+1},\ l = m_i - 1, \ldots, 1,$$

and,

$$G_{i-1} = -R_i^1\, T_i^1,$$

5. Domain Decomposition Preconditioners. From this exact factorization we can derive an approximation that will give a Domain Decomposition preconditioner. To do this, we approximate the inverses of tridiagonal matrices in the same way we were doing in INV.
The motivations have been explained in [15] on a two strips example, so here we directly derived the preconditioner for the k strips case, $k > 2$. We choose M as

$$M = L \begin{pmatrix} M_1^{-1} & & & & & & \\ & M_2^{-1} & & & & & \\ & & \ddots & & & & \\ & & & M_k^{-1} & & & \\ & & & & M_{1,2}^{-1} & & \\ & & & & & \ddots & \\ & & & & & & M_{k-1,k}^{-1} \end{pmatrix} U,$$

$$L = \begin{pmatrix} M_1 & & & & & & & & \\ & M_2 & & & & & & & \\ & & \ddots & & & & & & \\ & & & \ddots & & & & & \\ & & & & M_k & & & & \\ Q_1 & R_2 & & & & M_{1,2} & & & \\ & Q_2 & R_3 & & & H_2 & M_{2,3} & & \\ & & \ddots & & & & \ddots & \ddots & \\ & & & Q_{k-1} & R_k & & & H_{k-1} & M_{k-1,k} \end{pmatrix},$$

$$U = \begin{pmatrix} M_1 & & & & C_1 & & & \\ & M_2 & & & E_2 & C_2 & & \\ & & \ddots & & & \ddots & \ddots & \\ & & & \ddots & & & E_{k-1} & C_{k-1} \\ & & & M_k & & & & E_k \\ & & & & M_{1,2} & J_1 & & \\ & & & & & M_{2,3} & J_2 & \\ & & & & & & \ddots & \ddots \\ & & & & & & & M_{k-1,k} \end{pmatrix}.$$

In the lower right corners of L and U are the factors of an incomplete block Cholesky decomposition of the reduced system. As in [13],[15], we choose the approximations

$$M_{1,2} = B_{1,2} - Q_1^{m_1}\, trid[(\Delta_1^{m_1})^{-1}]\, C_1^{m_1} - R_2^1\, trid[(\Sigma_2^1)^{-1}]\, E_2^1.$$

$$M_{i,i+1} = B_{i,i+1} - Q_i^{m_i}\, trid[(\Delta_i^{m_i})^{-1}]\, C_i^{m_i} - R_{i+1}^1\, trid[(\Sigma_{i+1}^1)^{-1}]\, E_{i+1}^1$$
$$- H_i\, trid(M_{i-1,i}^{-1})\, J_{i-1},$$

$$S_i^1 = diag[(\Sigma_i^1)^{-1} E_i^1],$$

$$S_i^l = -diag[(\Sigma_i^l)^{-1} A_i^l \, S_i^{l-1}], \quad l = 2, \ldots, m_i,$$

$$H_i = -Q_i^{m_i} S_i^{m_i},$$

$$T_i^{m_i} = diag[(\Delta_i^{m_i})^{-1} C_i^{m_i}],$$

$$T_i^l = -diag[(\Delta_i^l)^{-1} B_i^l \, T_i^{l+1}], \quad l = m_i - 1, \ldots, 1,$$

$$J_{i-1} = -R_i^1 T_i^1.$$

In these formulas *diag* defines a diagonal approximation. Then, H_i and J_i are diagonal matrices. M_i is chosen as an INV block LU or UL approximation of B_i. Whatever is the approximation, we can solve independantly for the M_i's i.e. for each subdomain, but we have a block recursion for the reduced system i.e. the interfaces. We call this method INVDD.

As for the symmetric case there are many other possibilities giving more parallelism :

1) take $H_i = 0$, $\forall i$; then everything is parallel as there is no more recursion within the interfaces (INVDDH).

2) take only "some" $H_i = 0$, as needed by the number of available processors (INVDDS).

3) use an incomplete twisted factorization for the approximate reduced system [15].

Notice that these approaches are purely algebraic and are feasible for any diagonally dominant block tridiagonal M-matrices regardless of their origins.

Moreover, one can use different approximations (in place of INV) for the subdomains, like FFT-based preconditioners or point preconditioners. Modified (i.e. zero row sums) preconditioners are also possible.

6. Numerical results. We solve the following problem in the domain $\Omega =]0,1[\times]0,1[$,

$$-\Delta u + 2\alpha \frac{\partial u}{\partial x} + 2\beta \frac{\partial u}{\partial y} = f \quad \text{in } \Omega, \quad u\,|_{\partial\Omega} = 0.$$

We discretize with the 5 point scheme and central differences for the first order term. The right hand side is the same as in [15], the starting guess is a random vector, the stopping criterion is $\|r^k\|_2 \leq 10^{-6} \, \|r^0\|_2$ and the value of h is $\frac{1}{101}$.

We are interested in looking, for a fixed mesh size, at the number of iterations of the Orthomin(1) conjugate gradient–like method as a function of the number of subdomains k, for different values of α and β. We give only results for INVDD, the results for more parallel methods (including the number of operations) will be given in another paper [16].

k	$\alpha = 0, \beta = 0$	$\alpha = 1, \beta = 1$	$\alpha = 25, \beta = 50$	$\alpha = 100, \beta = -1$	$\alpha = 300; \beta = -1$
2	24	32	14	18	13
4	25	34	15	19	13
8	25	32	18	19	13
16	27	36	31	20	14
24	32	40	40	24	16
32	33	41	40	27	17
40	36	48	42	31	19
50	34	39	38	28	20

We can see that even if the increase in the number of iterations can be sometimes a little bit larger than for the symmetric case, it is still very slight so the use of these domain decomposition can be useful provided the more parallel techniques developed in [15] are used.

7. Conclusions. The Domain Decomposition methods presented in this paper offer a great deal of parallelism when used with iterative methods. Because of the algebraic nature of these preconditioners they can also be used with discontinuous coefficient problems as they are based on robust approximations.

References

[1] P. BJORSTAD & O.B. WIDLUND, *Iterative methods for the solution of elliptic problems on regions partioned into substructures.* SIAM J. on Numer. Anal. v 23, n 6, (1986) pp 1097–1120.

[2] J.H. BRAMBLE, J.E. PASCIAK & A.H. SCHATZ, *The construction of preconditioners for elliptic problems by substructuring. I.* Math. of Comp. v 47, n 175, (1986) pp 103–104.

[3] T. CHAN & D. RESAŠCO, *A domain decomposition fast Poisson solver on a rectangle.* Yale Univ. report YALEU/DCS/RR 409 (1985).

[4] T. CHAN, *Proceedings of the second international symposium on domain decomposition methods for partial differential equations.* SIAM (1988).

[5] P. CONCUS, G.H. GOLUB & G. MEURANT, *Block preconditioning for the conjugate gradient method.* SIAM J. Sci. Stat. Comp., v 6, (1985) pp 220–252.

[6] P. CONCUS & G. MEURANT, *On computing INV block preconditionings for the conjugate gradient method.* BIT v 26 (1986) pp 493–504.

[7] H. ELMAN, *Iterative methods for large sparse non symmetric systems of linear equations.* Ph.D. thesis, Technical Report 229, Yale University (1982).

[8] R. GLOWINSKI, G.H. GOLUB, G. MEURANT & J. PERIAUX, *Proceedings of the first international symposium on domain decomposition methods for partial differential equations.* SIAM (1988).

[9] G.H. GOLUB & D. MAYERS, *The use of preconditioning over irregular regions.* In "Computing methods in applied science and engineering VI", R. Glowinski & J.L. Lions Eds, North–Holland (1984).

[10] G. MEURANT, *The block preconditioned conjugate gradient method on vector computers.* BIT v 24 (1984) pp 623–633.

[11] G. MEURANT, *The conjugate gradient method on supercomputers.* Supercomputer v 13 (1986) pp 9–17.

[12] G. MEURANT, *Multitasking the conjugate gradient method on the CRAY X-MP/48.* Parallel Computing 5 (1987) pp 267–280.

[13] G. MEURANT, *Domain decomposition vs block preconditioning.* In [8].

[14] G. MEURANT, *Conjugate gradient preconditioners for parallel computers* In "Proceedings of the third SIAM conference on parallel processing for scientific computing", Los Angeles 1987, SIAM (1988).

[15] G. MEURANT, *Domain decomposition preconditioners for the conjugate gradient method.* submitted to Calcolo (1988)

[16] G. MEURANT, *Numerical experiments with domain decomposition preconditioners for non symmetric problems.* to appear