Threshold Ordering for Preconditioning Nonsymmetric Problems

Michele Benzi[1], Hwajeong Choi[2], Daniel B. Szyld[2*]

[1] CERFACS, 42 Ave. G. Coriolis, 31057 Toulouse Cedex, France (benzi@cerfacs.fr)
[2] Department of Mathematics, Temple University, Philadelphia, Pennsylvania 19122-6094, USA (choi@math.temple.edu, szyld@math.temple.edu)

Dedicated to Gene H. Golub on the occasion of his 65th birthday

Abstract. The effect of a threshold variant TPABLO of the permutation (and partitioning) algorithm PABLO on the performance of certain preconditionings is explored. The goal of these permutations is to produce matrices with dense diagonal blocks, and in the threshold variant, with large entries in the diagonal blocks. Experiments are reported using matrices arising from the discretization of elliptic partial differential equations. The iterative solvers used are GMRES, QMR, BiCGStab and CGNR. The preconditioners are different incomplete factorizations. It is shown that preprocessing the matrices with TPABLO has a positive effect on the overall performance, resulting in better convergence rates for highly nonsymmetric problems.

1 Introduction

For the solution of sparse nonsymmetric linear systems of equations of the form

$$Av = b, \tag{1}$$

it is customary nowadays to use Krylov-based iterative methods, such as GMRES [16], QMR [9], etc. Usually, one needs to precondition the system, e.g., using an incomplete factorization [10], [13], [14]. As is well known, the ordering of the variables influences the effect of incomplete factorization preconditioners, and thus the number of steps the iterative methods takes to converge [6].

In this contribution, we consider a symmetric permutation of the matrix A in (1), i.e., of the form P^TAP, and then solve the equivalent system

$$P^TAPw = P^Tb, \quad \text{with } v = Pw$$

by several iterative methods and preconditionings.

The symmetric permutation we use is performed by the algorithm TPABLO (Threshold PArameterized BLock Ordering) [3]. This algorithm produces a permuted matrix with dense diagonal blocks, while the entries outside the diagonal

* Supported by the National Science Foundation grant DMS-9625865.

blocks have magnitude below a prescribed threshold. It is based on the algorithm PABLO [11]. We briefly describe these algorithms in Sec. 2, but refer the reader to [11] and [3] for their full description and complexity analysis. Both algorithms are linear in the order of the matrix and the number of nonzeros, but in the threshold variant, the constant could be large.

The algorithm PABLO, with some modifications, has been used to establish diagonal blocks used as a block diagonal (or block Jacobi) preconditioner in several contexts [3], [7], [12], [18]. This is the first time where other types of preconditioners are studied.

The effect of the threshold reordering on the performance of preconditioned Krylov subspace solvers is illustrated in Sec. 3, using matrices arising from the discretization of certain elliptic PDEs in two dimensions. Additional experiments will be reported in a forthcoming paper.

2 The Permutation Algorithm

We begin by reviewing the algorithm PABLO [11] on which TPABLO is based. These algorithms work with the graph of the matrix, choosing one node at a time, adding it to a group of nodes which would form the blocks along the diagonal, if the new node satisfies certain criteria. The amount of work is controlled by not searching through all the available nodes, but only through a subset of eligible nodes, those which are adjacent to some nodes in the current set, i.e., in the group of nodes of the block along the diagonal being formed. This feature makes the algorithms linear in the order of A and in the number of its nonzeros [3], [11], adding relatively little computational time to the overall solution method.

Given an $n \times n$ matrix $A = (a_{ij})$, let $G = (V, E)$ be its associated graph, i.e., $V = \{v_1, \ldots, v_n\}$ is the set of n vertices and E is the set of edges, where $(v_i, v_j) \in E$ if and only if $a_{ij} \neq 0$; see, e.g., [5]. Given this graph, PABLO constructs q subgraphs $G_k = (V_k, E_k)$, $k = 1, \ldots, q$. The number of subgraphs, q, i.e., the number of the corresponding diagonal blocks, is not known a priori, but is determined by the algorithm and it depends on the structure of the graph G and the input parameters. In the version presented in [7], q is set by the user to match the number of available processors. In the algorithm PABLO (as well as in the threshold variant used in this paper), a first node is taken from the stack of unmarked nodes and this node starts a new current set of vertices P, then additional nodes are taken from the stack and added to the set P if they satisfy certain criteria, or sent back to the stack if not. Two criteria are used in PABLO to determine if a node v should be added to a current set $P \subset V$, corresponding to a diagonal block, by measuring how full $\{v\} \cup P$ is, and how much the vertices in $\{v\} \cup P$ are connected to each other. The first criterion is measured by the ratio α of the total number of edges corresponding to $\{v\} \cup P$ to the number of edges that subgraph would have if it were complete (corresponding to a full submatrix). If it is satisfied then the new node is added. The second test is that the new node v must be adjacent to at least a certain proportion β of nodes in

the subgraph corresponding to P and more than outside it. For further details, see [11].

In the algorithm TPABLO used in this paper, a third additional criterion is used to decide if a new vertex v is added to the current subgraph being formed. Let γ be the given threshold, let P be the set of nodes of the current subgraph, and v_j be the vertex being tested for addition to P (corresponding to the jth row and column of the matrix). The node v_j is added to P if, in addition to either of the two criteria in PABLO, the following holds:

$$|a_{ij}| > \gamma \quad \text{or } |a_{ji}| > \gamma \quad \text{for at least one } i \in P.$$

The use of this additional criterion produces a permuted matrix in which every entry in the off-diagonal blocks is smaller than the threshold in absolute value. Entries in the diagonal blocks may still have magnitude smaller than the threshold γ, these are not discarded. In addition, the user can specify a minimum and a maximum block size; see further [3].

We point out that the three main parameters of TPABLO, i.e., the two connectivity parameters α and β, and the threshold γ, together with the parameter for maximum size of each block, provide the user with great flexibility. Thus, one may wish to find permutations such that the resulting matrix has denser or sparser diagonal blocks, or ones whose off-diagonal blocks have smaller and smaller entries.

We conclude this brief description of TPABLO by noting that there is no guarantee that the blocks along the diagonal obtained by this algorithm be nonsingular. In fact, it would be easy to construct examples for which the resulting block diagonal matrix is singular. Nevertheless, we should note that this problem did not arise in our experiments.

3 Numerical Experiments

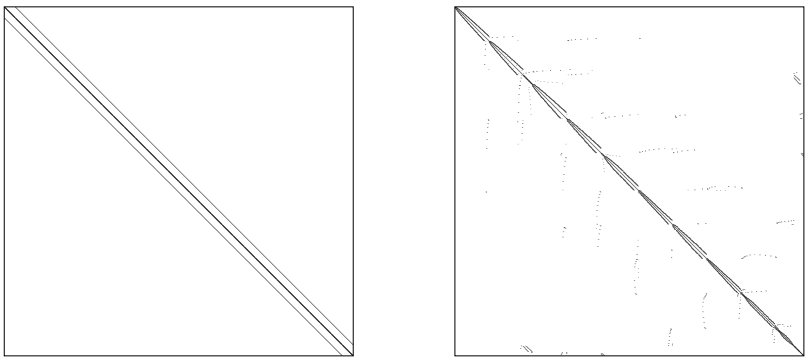
We consider the following partial differential equation in $\Omega = (0,1) \times (0,1)$

$$-\epsilon \Delta u + \frac{\partial e^{xy} u}{\partial x} + \frac{\partial e^{-xy} u}{\partial y} = g \tag{2}$$

with Dirichlet boundary conditions. Equation (2) has been used many times as a model problem in the literature; see, e.g., [13]. The problem is discretized using centered differences for both the second order and first order derivatives with grid size $h = 1/33$, leading to a block tridiagonal linear system of order $N = 1024$ with $NZ = 4992$ nonzero coefficients. The right-hand side is chosen so that the solution v to the discrete system is one everywhere. The parameter $\epsilon > 0$ controls the difficulty of the problem —the smaller is ϵ, the harder it is to solve the discrete problem by iterative methods. For our experiments, we generated ten linear systems of increasing difficulty, corresponding to $\epsilon^{-1} = 100, 200, \ldots, 1000$. The coefficient matrix A becomes increasingly nonsymmetric as ϵ gets smaller.

The Krylov subspace methods tested were GMRES(20) [16], BiCGStab [17], QMR [9] and CGNR (conjugate gradient method applied to the normal equations $A^T Av = A^T b$). The preconditioners used were ILU(0) [10], ILUT [14] and, for use with CGNR, a drop tolerance-based incomplete QR factorization (see, e.g., [13] and [1] for information on incomplete QR preconditioning). In all our experiments we used $v_0 = 0$ as initial guess and we stopped the iterations when the 2-norm of the 'true' residual $b - Av_k$ had been reduced to less than 10^{-9}, with a maximum of 1000 iterations allowed.

For the TPABLO ordering, we set $\alpha = \beta = 0$, a maximum block size of 100, and threshold $\gamma = 0.005$ for $\epsilon^{-1} = 100, 200, 300$ and $\gamma = 0.01$ for $\epsilon^{-1} > 300$. We show in Fig. 1 the sparsity patterns corresponding to (a) the natural ordering and (b) the TPABLO ordering for $\epsilon^{-1} = 1000$.



(a) Natural ordering *(b) TPABLO ordering*

Fig. 1. Matrix patterns

In Table 1 we show the amount of fill-in incurred by the incomplete factorizations ILUT and IQR for typical values of the parameters, using the natural ordering and TPABLO. The figures refer to the number of nonzeros in the incomplete L and U factors for ILUT, and in the incomplete R factor for QR. Note that the amount of fill-in increases almost monotonically with ϵ^{-1}, and that TPABLO reduces the amount if fill-in in all cases, except one.

In Tables 2 and 3 we show the number of iterations for various combinations of Krylov solvers and preconditioners, with the natural ordering and with TPABLO. It is clear that as ϵ decreases, the rates of convergence tend to deteriorate with both orderings, but the deterioration is much slower in the TPABLO ordering. This phenomenon is particularly striking for CGNR/IQR. As a result, TPABLO, which gives worse results than the natural ordering for the 'easy' problems (ϵ large), becomes clearly better for the 'hard' problems (ϵ small).

Additional experiments where performed with CGNR/IQR where the columns of the coefficient matrix A were permuted according to the minimum degree heuristic applied to the structure of $A^T A$. The purpose of this heuristic is to

Table 1. Fill-in for the drop tolerance-based ILUT and IQR preconditioners (n/o = natural ordering, TP = TPABLO)

	ILUT(5,10^{-2})		ILUT(10,10^{-3})		IQR(10^{-2})	
ϵ^{-1}	n/o	TP	n/o	TP	n/o	TP
100	12540	11258	21319	15520	9783	12718
200	14458	13507	22935	19465	16742	14259
300	14706	14139	23640	20850	19108	15050
400	14782	14333	23876	21541	20093	15559
500	14796	14425	24374	21970	20718	16167
600	14807	14439	24571	21986	20697	16360
700	14828	14523	24633	21953	21537	16260
800	14852	14467	24677	22082	24362	16845
900	14863	14002	24773	22023	25680	17135
1000	14856	14041	24855	22039	24727	17429

Table 2. Number of iterations for different preconditioners and iterative solvers (n/o = natural ordering, TP = TPABLO). A † means that convergence was not attained in 1000 iterations.

	ILU(0)						ILUT(5,10^{-2})						ILUT(10,10^{-3})					
	GMR		BST		QMR		GMR		BST		QMR		GMR		BST		QMR	
ϵ^{-1}	n/o	TP	n/o	TP	n/o	TP	n/o	TP	n/o	TP	n/o	TP	n/o	TP	n/o	TP	n/o	TP
100	17	36	12	15	20	27	7	9	4	5	7	10	5	7	3	4	6	8
200	39	90	21	29	37	50	11	11	6	6	11	11	7	8	4	5	7	9
300	113	220	36	44	58	71	14	13	9	7	15	14	8	9	5	6	9	10
400	457	637	63	57	93	94	18	15	10	9	19	16	11	10	6	6	11	10
500	†	†	87	81	198	124	26	19	13	11	23	21	14	12	8	6	14	12
600	†	†	156	102	†	176	33	25	16	12	27	23	16	14	9	8	17	14
700	†	†	333	143	†	210	40	23	20	12	34	22	20	13	13	7	21	14
800	†	†	606	148	†	235	59	50	26	21	41	38	27	15	15	9	23	16
900	†	†	778	177	†	296	†	†	†	†	†	†	32	17	16	9	27	18
1000	†	†	†	205	†	338	†	†	†	†	†	†	38	18	18	10	32	19

preserve sparsity in the triangular factor R (which corresponds to the Cholesky factor of A^TA); see [1]. With this ordering, the convergence rate of CGNR/IQR was about as good as with TPABLO. However, fill-in in the incomplete factor R was consistently found to be about 20% higher with minimum degree than with TPABLO.

4 Concluding Remarks

The results of our experiments show that TPABLO can enhance the robustness and performance of Krylov subspace methods preconditioned with incomplete

Table 3. Number of iterations for CGNR preconditioned with incomplete QR factorization (n/o = natural ordering, TP = TPABLO).

order	100	200	300	400	500	600	700	800	900	1000
n/o	24	22	27	53	84	118	119	112	149	190
TP	23	22	22	24	25	25	23	27	28	29

factorizations. The beneficial effect of TPABLO for the class of problems considered in this paper can be explained as a 'damping' of the instability of the incomplete LU factorizations of such matrices. This instability was first demonstrated in [8]; see also [4] and [15] for some illuminating examples of the same phenomenon. In this context, the 'instability' refers to the fact that the norm of $(LU)^{-1}$ can be extremely large. Hence, the ILU preconditioning matrix $M = LU$ is extremely ill conditioned and so is the preconditioned matrix $M^{-1}A$, although the original problem is actually rather well conditioned. As shown in [8] for the case of ILU(0), the ill conditioning is caused by the long recurrences in the forward and backward triangular solves when the preconditioning is applied. When A is highly indefinite or has large nonsymmetric part, even the more accurate ILUT factorization can exhibit this kind of instability, causing the preconditioned iteration to fail [4]. In this case, a common course of action is to increase the accuracy of the preconditioner by allowing more fill-in in the incomplete factors, which increases the cost of forming and applying the preconditioner. Another possibility would be to switch to sparse approximate inverse preconditioners, which do not involve triangular solves and are therefore immune from the type of instability described above; see, e.g., [2] and [4]. However, these preconditioners are more expensive to compute and use than incomplete factorizations.

Another alternative is to apply some reordering to the original matrix. This leaves the conditioning of the matrix unchanged, but may have a beneficial effect on the conditioning of the preconditioner. It appears that TPABLO can improve the stability of the incomplete factorization without increasing the amount of fill-in, in fact even reducing it. The price for this is the cost of the preprocessing itself, which may be negligible if a sequence of linear systems with the same matrix and different right-hand sides must be solved. A more detailed analysis will be presented, together with additional experiments, in a forthcoming paper.

Acknowledgment. We would like to thank Miroslav Tůma for providing some of the codes which were used for the numerical experiments.

References

1. Benzi M., Tůma M.: A comparison of preconditioning techniques for general sparse matrices. In S. Margenov and P. Vassilevski (Eds.), Iterative Methods in Linear Algebra, II, IMACS Series in Computational and Applied Mathematics Vol. 3, pages 191–203 (1996)

2. Benzi M., Tůma M.: A sparse approximate inverse preconditioner for nonsymmetric linear systems. SIAM J. Sci. Comput. (1998) to appear
3. Choi H., Szyld D. B.: Application of threshold partitioning of sparse matrices to Markov chains. Proceedings of the IEEE International Computer Performance and Dependability Symposium, IPDS'96, Urbana-Champaign, Illinois, September 4–6, 1996, pages 158–165, IEEE Computer Society Press, Los Alamitos, California (1996)
4. Chow E., Saad Y.: Approximate inverse techniques for block-partitioned matrices. SIAM J. Sci. Comput. (1997) to appear
5. Duff I., Erisman A. M., Reid J. K: Direct Methods for Sparse Matrices. Clarendon Press, Oxford, 1986
6. Duff I., Meurant G. A.: The effect of ordering on preconditioned conjugate gradients. BIT **29** (1989) 635–657
7. Dutto, L. C., Habashi W. G., Fortin M.: Parallelizable block diagonal preconditioners for the compressible Navier-Stokes equations. Computer Methods Appl. Mech. Engng. **117** (1994) 15–47
8. Elman H. C.: A stability analysis of incomplete LU factorizations. Math. Comp. **47** (1986) 191–217
9. Freund R., Nachtigal N. M.: QMR: A quasi-minimal residual method for non-Hermitian linear systems. Num. Math. **60** (1991) 315–339
10. Meijerink J. A., and van der Vorst, H.: An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. Math. Comp. **31** (1977) 148–162
11. O'Neil J., Szyld D. B.: A block ordering method for sparse matrices. SIAM J. Sci. Stat. Comput. **11** (1990) 811–823
12. Paloschi J. R.: Testing a new parallel preconditioner on linear systems arising from flowsheeting simulation. Proceedings of ESCAPE7, Trondheim, May 1997, to appear.
13. Saad Y.: Preconditioning techniques for nonsymmetric and indefinite linear systems. J. Comp. Appl. Math. **24** (1988) 89–105
14. Saad Y.: ILUT: A dual threshold incomplete LU factorization. Num. Lin. Alg. Applic. **1** (1994) 387–402
15. Saad Y.: Preconditioned Krylov subspace methods for CFD applications. In W. G. Habashi (Ed.), Solution Techniques for Large-Scale CFD Problems, J. Wiley, Chichester, pages 139–158 (1995)
16. Saad Y., Schultz M. H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7** (1986) 856–869
17. van der Vorst, H.: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **12** (1992) 631–644
18. Yang G., Dutto L. C., Fortin M.: Inexact block Jacobi-Broyden methods for solving nonlinear systems of equations. SIAM J. Sci. Comput. (1997) to appear

This article was processed using the LaTeX macro package with LLNCS style