Does the ordering affect to sparse approximate inverse preconditioners?

E. Flórez[1] M.D. García[1] L. González[1] G. Montero[1]

Abstract

We experimentally study how the reordering techniques affect the rate of convergence of preconditioned Krylov subspace methods for nonsymmetric sparse linear systems, where the preconditioner is a sparse approximate inverse. In addition, we show how the reordering reduces the number of entries in the approximate inverse and thus, the amount of storage requirement and computations for a given accuracy. These properties are illustrated with a numerical experiment taken from the discretization of PDE by a finite element method.

Introduction

The solution of sparse linear systems $Ax = b$ by preconditioned Krylov methods is studied. We focus our work on sparse approximate inverse preconditioners. These preconditioners are specially interesting in parallel environment since their construction and application at each step of the iterative method, i.e. matrix-vector products, are highly parallelizable. On the other hand, even on sequential computations, the sparse approximate inverse preconditioners may be useful for solving those problems for which other type of preconditioners such as ILU fail. The approximate inverse here considered, is a version of the SPAI proposed by Grote et al [1], and both theoretical and computational aspects have been analyzed in [2],[3]. We present results of the effect of reordering not only on the amount of the entries in the inverse factors, but also on the number of steps of the iterative solver. Although the inverse A^{-1} is usually full, regardless of the ordering chosen, we experimentally show how the fill-in of the sparse approximate inverses is dependent on the ordering of A. A similar study has been carried out by Benzi et al [4] for factorized approximate inverses. Also, interesting results on the effect of ordering for incomplete factorization in the convergence of Krylov subspace methods may be found in [5].

Sparse approximate inverse computation

Next we consider an explicit preconditioner [13]. Our algorithm automatically searches one by one the non null entries in matrix M, starting from the diagonal structure. Let denote by $r_k^t = m_k^t A - e_k^t$, the residual corresponding to the row k of M and let

$\mathcal{I}_k$ be the set of indices of the non null entries in r_k^t, i.e., $\mathcal{I}_k = \{i \in \{1,2,...,n\} \ / \ r_{ki} \neq 0\}$. If $\mathcal{L}_k = \{l \in \{1,2,...,n\} \ / \ m_{kl} \neq 0\}$, then the new entry is searched in the set $\mathcal{J}_k = \{j \in \mathcal{L}_k^c \ / \ a_{ji} \neq 0, \forall i \in \mathcal{I}_k\}$. Indeed, the only entries considered in m_k^t are those affecting to non null ones of r_k^t. We assume that $\mathcal{L}_k \cup \{j\} = \left\{i_1^k, i_2^k, ..., i_{p_k}^k\right\}$ is not empty, with p_k being the actual number of non null entries of m_k^t and $i_{p_k}^k = j$, for all $j \in \mathcal{J}_k$. For each j, we compute

$$||m_k^t A - e_k^t||_2^2 = 1 - \sum_{l=1}^{p_k} \frac{\left[\det\left(D_l^k\right)\right]^2}{\det\left(G_{l-1}^k\right) \det\left(G_l^k\right)} \tag{1}$$

where, for all k, $\det\left(G_0^k\right) = 1$ and G_l^k is the Gram matrix of the rows $i_1^k, i_2^k, ..., i_l^k$ of matrix A with respect to the euclidean inner product, D_l^k results from replacing the last row of matrix G_l^k by $a_{i_1^k k}, a_{i_2^k k}, ..., a_{i_l^k k}$, with $1 \leq l \leq p_k$. The index j_k which makes minimal the value of $||m_k^t A - e_k^t||_2$ is selected. This strategy defines the new selected index j_k only attending to the set $\mathcal{L}_k$, what leads us to an actual optimum where all the entries corresponding to the indices of $\mathcal{L}_k$ are updated. Thus the row corresponding to m_k^t is searched in the set $\mathcal{S}_k = \{m_k^t \in \mathbb{R}^n / m_{ki} = 0; \forall i \notin \mathcal{L}_k \cup \{j_k\}\}$, of which entries are computed at each stage by

$$m_k^t = \sum_{l=1}^{p_k} \frac{\det\left(D_l^k\right)}{\det\left(G_{l-1}^k\right) \det\left(G_l^k\right)} \widetilde{m}_l^t \tag{2}$$

where $\widetilde{m}_l^t$ is the vector with non null entries i_h^k $(1 \leq h \leq l)$. Each of them is obtained evaluating the corresponding determinant which results from replacing the last row of $\det\left(G_l^k\right)$ by e_h^t, with $1 \leq l \leq p_k$.

Some remarks on reordering

We have considered several reordering techniques in order to illustrate the effect of ordering on the iterative resolution of linear systems of equations using SPAI preconditioners. The original ordering corresponds to matrices directly arising from Finite Element Method with unstructured meshes and adaptive mesh refinement. The reordering algorithms are summarized below (see, e.g., [14] and [15]). Let A be a non-symmetric matrix of dimension n with symmetric pattern and let $G = \langle V, E \rangle$ be the directed graph of the matrix A, where $V = \{1, ..., n\}$ is the set of nodes and E is the set of edges $\langle i, j \rangle$ such that $a_{ij} \neq 0$. The set of nodes adjacent to v in G is denoted by $Adj_G(v)$. The degree of the node v is $|Adj_G(v)|$. $L(G)$ is a partition of the set of nodes V which is known as a level structure rooted at a node v, where $L_1 = \{v\}$ and L_i is the set of the nodes adjacent to nodes of level L_{i-1} which are not yet in a previous level. The width of the level i, $w_i(L)$, is defined by the cardinality of L_i, and the width of the level structure $L(G)$ is $w(L) = \max w_i(L)$.

The minimum degree ordering has been used to reduce the fill-in in factorization of matrices with symmetric sparsity pattern. The performance of the algorithm may be found in [6].

Minimum Degree algorithm.

1. Define the graph $G = \langle V, E \rangle$, associate with the matrix A.
2. While $V \neq \emptyset$
   2.1 Select a node v of minimum degree in G and order v next
   2.2 Define
   $V_v = V - \{v\}$ and $E_v = \{\{i,j\} \in E : i,j \in V_v\} \cup$
   $\{\{i,j\} : i \neq j$ and $i \in \mathrm{Adj}_G(v)$, $j \in \mathrm{Adj}_G(v)\}$.
   Set $V = V_v, E = E_v$ and $G = \langle V, E \rangle$.
3. End

The Reverse Cuthill-Mckee algorithm [8] is a modification of the Cuthill-Mckee algorithm [16] that simply takes the reverse ordering of that obtained by this last one. The general characteristics of these algorithms are the reduction of bandwidth and profile.

Reverse Cuthill-Mckee algorithm.

1. Define the graph $G = \langle V, E \rangle$, associate with the matrix A.
2. Select a node p in V of minimum degree in G and order p the first
3. Generated the level structure rooted at node p $L(G)$.
4. For $1 < i \leq$ number of levels, do
   4.1. For $1 \leq j \leq w_i(L)$, do
   Select a node v in L_i of minimal width and order v next
   Set $L_i = L_i - \{v\}$
5. Set the inverse ordering, i.e., change the numbering $CM(v)$ associated to the node v to $RCM(v) = n + 1 - CM(v)$
6. End

The Minimum Neighbouring [7] is a variant of the Minimun Degree algorithm which works by eliminating the selected nodes in the structure of the graph associated with the matrix A, such that no new edge is defined and inserted in the graph. It chooses a node which has the minimum number of neighbours. This algorithm is specially useful when we look for an incomplete factorization with the same sparsity pattern as matrix A, e.g. the ILU(0) preconditioner which will be used in the numerical experiments of this paper.

Minimum Neighbouring algorithm.

1. Define the graph $G = \langle V, E \rangle$, associate with the matrix A.
2. While $V \neq \emptyset$
   2.1 Select a node v of minimum degree in G and order v next
   2.2 Define
   $V_v = V - \{v\}$ and $E_v = \{\{i,j\} \in E : i,j \in V_v\}$
   Set $V = V_v, E = E_v$ and $G = \langle V, E \rangle$.
3. End

In all the algorithms, we start from a pseudoperipherical node searched by the George's algorithm.

The main objective of this work is to answer the following questions:

1. Does the reordering reduce the amount of entries in the SPAI preconditioners?
2. Does it improve the convergence of iterative solvers with such preconditioners?

Since $(P^TAP)^{-1} = P^TA^{-1}P$, i.e., the inverse of the reordered matrix is the reordering of the inverse matrix, when we reorder a matrix, its approximate inverse should tend to the reordered inverse.

If the accuracy of the approximate inverse is given by ε, in the subspace $S \subset M_n(\mathbb{R})$,

$$\min_{M \in S} \|MA - I\|_F = \|NA - I\|_F < \varepsilon \tag{3}$$

then,

$$\min_{M' \in P^TSP} \left\|M'P^TAP - I\right\|_F = \min_{M \in S} \|MA - I\| < \varepsilon \tag{4}$$

Let S' be a subspace of $M_n(\mathbb{R})$ corresponding to the same number of non null entries as S, where the optimal approximate inverse is obtained for such a number of non null entries. Also note that the number of non null entries of S is the same as P^TSP. In this case, we obtain,

$$\begin{aligned} \left\|N'P^TAP - I\right\|_F &= \min_{M' \in S'} \left\|M'P^TAP - I\right\|_F \\ &\leq \min_{M' \in P^TSP} \left\|M'P^TAP - I\right\|_F < \varepsilon \end{aligned} \tag{5}$$

Evidently, the number of non null entries needed in S' becomes less or equal to that in P^TSP and thus in S. We conclude that the reordering reduces the amount of entries in the approximate inverse for a given accuracy ε, or at least does not augment it.

Due to the result given in (5), the reordered sparse approximate inverse preconditioners adquire better properties from the point of view of the performance of the iterative solver [3]. The closeness of the condition number of $M'P^TAP$ to 1 is characterized by,

$$K_2(M'P^TAP) \leq \frac{1 + \left\|M'P^TAP - I\right\|_2}{1 - \left\|M'P^TAP - I\right\|_2} \tag{6}$$

The $(M'P^TAP)$'s departure from normality is bounded by,

$$\frac{1}{n}\sum_{k=1}^{n}(|\lambda_k| - \sigma_k)^2 \leq \frac{2}{n}\left\|M'P^TAP\right\|_F^2 (1 - \sigma_n) \tag{7}$$

with $\{\lambda_k\}_{k=1}^n, \{\sigma_k\}_{k=1}^n$ being the eingenvalues and singular values of $M'P^TAP$ (non increasing modules sequence). Finally, the clustering of eigenvalues and singular values is defined by,

$$\sum_{k=1}^{n}(1 - \sigma_k)^2 \leq \left\|M'P^TAP - I\right\|_F^2 \tag{8}$$

$$\sum_{k=1}^{n}(1 - \lambda_k)^2 \leq \left\|M'P^TAP - I\right\|_F^2 \tag{9}$$

Numerical experiments

The BiCGSTAB algorithm proposed in [9] is a variant of BiCG algorithm which provides a smoother convergence behaviour than CGS algorithm [17]. We have used it in our numerical experiments since it has proved to be an efficient iterative solver in convection-diffusion problems of the type solved here [18]. The numerical example is a convection-diffusion problem (*convdifhor* [19]) defined in $[0,1] \times [0,1]$ by the equation,

$$v_1 \frac{\partial u}{\partial x} - K\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = F$$

where $v_1 = 10^4\left(y - \frac{1}{2}\right)\left(x - x^2\right)\left(\frac{1}{2} - x\right)$, $K = 10^{-5}$ inside the triangles of vertices $\{(0.5,0),(1,0.5),(0.5,0.5)\}$ and $\{(0,0.5),(0.5,0.5),(0.5,1)\}$, elsewhere $K = 10^2$, and $F = 10^3$ inside the triangles $\{(0.5,0),(1,0),(1,0.5)\}$ and $\{(0,0.5),(0.5,1),(0,1)\}$, elsewhere $F = 1$. Dirichlet boundary conditions, $u = 0$ on $x = 1$ and $u = 1$ on $x = 0$ were considered. Elsewhere of the boundary we set null Neumann conditions. The matrix corresponds to an unstructured mesh of finite elements with $n = 1960$ and $nz = 13412$. Tables 1-4 indicate the performance of ILU(0) and SPAI preconditioners. The reduction of the amount of entries in the SPAI matrices is between 40 and 50 per cent for Minimum Degree and Reverse Cuthill-Mckee. The Minimum Neighbouring does not affect to nz. Furthermore, the number of iteration of BiCGSTAB was reduced by both reorderings from 60 to 70 per cent. Since we are interested in the effect of the reordering of A in the characteristics of the SPAI preconditioners, the sparsity pattern of matrix M with $\varepsilon_k = 0.3$ is shown in Figures 1(a)-(d) for Original, Minimum Degree, Reverse Cuthill-Mckee and Minimum Neighbouring orderings, respectively. Any non null entry is represented by a point. The pattern corresponding to the original ordering represents a full matrix, as expected. However, a certain parallelism with the structure of A is noticed. The bandwidth and profile reduction carried out by the Reverse Cuthill-Mckee algorithm in matrix A are somehow saved in matrix M, even though there is a tendency to exploit some entries out the profile. This is clearly illustrated in Figure 1(c). The patterns of SPAI matrices corresponding to Minimum Degree and Minimum Neighbouring also save in part the structures of the reordered matrix A, respectively, even when our SPAI algorithm should not produce matrices M with symmetric structure. In Figure 2 we compare the convergence behaviour of BiCGSTAB-SPAI(0.2) for all these reorderings. Clearly, the reordering produced by Minimum Degree and Reverse Cuthill-Mckee have a beneficial effect in the rate of convergence of the preconditioned BiCGSTAB-SPAI algorithm. We conclude, as in other experiments carried out, that the sparsity pattern of SPAI starts from a structure similar to the typical of A obtained by reordering, and tends to a full matrix as we augment its accuracy.

Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA - I\|\|_F$
Unprecond.	> 1960	1960	0.15	13279
ILU(0)	74	13412	1.00	–
SPAI $\varepsilon_k = 0.6$	414	3161	0.24	22.42
SPAI $\varepsilon_k = 0.4$	302	10693	0.80	16.99
SPAI $\varepsilon_k = 0.3$	171	21734	1.62	12.78
SPAI $\varepsilon_k = 0.2$	83	54406	4.06	8.70
SPAI $\varepsilon_k = 0.1$	21	167678	12.50	4.36

Table 1: Convergence results for *condifhor* with *Original Ordering* and left preconditioned BiCGSTAB

Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA - I\|\|_F$
Unprecond.	> 1960	1960	0.15	13275
ILU(0)	57	13412	1.00	–
SPAI $\varepsilon_k = 0.6$	166	2617	0.20	19.82
SPAI $\varepsilon_k = 0.4$	99	6255	0.47	15.18
SPAI $\varepsilon_k = 0.3$	68	11461	0.85	11.42
SPAI $\varepsilon_k = 0.2$	40	26992	2.01	7.78
SPAI $\varepsilon_k = 0.1$	21	92864	6.92	3.85

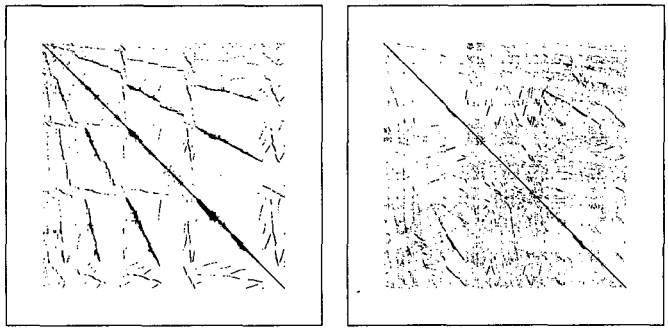
Table 2: Convergence results for *condifhor* with *Minimum Degree* and left preconditioned BiCGSTAB

Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA - I\|\|_F$
Unprecond.	1477	1960	0.15	13272.9
ILU(0)	31	13412	1.00	–
SPAI $\varepsilon_k = 0.6$	144	2510	0.19	19.51
SPAI $\varepsilon_k = 0.4$	92	6126	0.46	15.51
SPAI $\varepsilon_k = 0.3$	66	11355	0.85	11.67
SPAI $\varepsilon_k = 0.2$	41	26270	1.96	7.98
SPAI $\varepsilon_k = 0.1$	18	88093	6.57	4.01

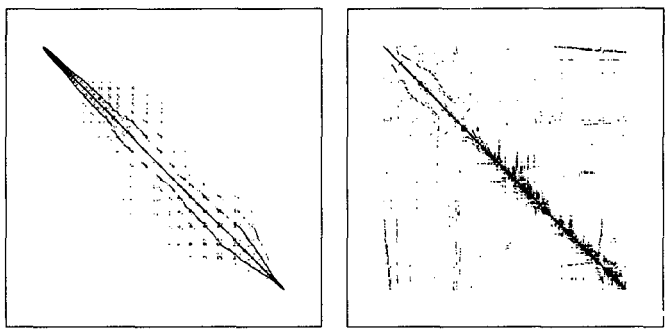
Table 3: Convergence results for *condifhor* with *Reverse Cuthill-McKee* and left preconditioned BiCGSTAB

Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA - I\|\|_F$
Unprecond.	> 1960	1960	0.15	13271.5
ILU(0)	45	13412	1.00	–
SPAI $\varepsilon_k = 0.6$	397	3161	0.23	22.41
SPAI $\varepsilon_k = 0.4$	294	10693	0.80	16.98
SPAI $\varepsilon_k = 0.3$	173	21734	1.62	12.78
SPAI $\varepsilon_k = 0.2$	86	54406	4.06	8.69
SPAI $\varepsilon_k = 0.1$	24	167678	12.50	4.36

Table 4: Convergence results for *condifhor* with *Minimun Neighbouring* and left preconditioned BiCGSTAB



(a) Original Ordering (b) Minimum Degree



(c) Reverse Cuthill-Mckee (d) Minimum Neighbouring

Figure 1: Sparsity pattern of SPAI(0.3) matrix with different orderings

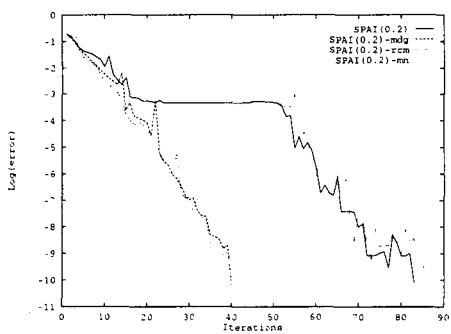


Figure 2: Performance of BiCGSTAB-SPAI with different reorderings.

Conclusions

We have experimentally proved that reordering techniques have beneficial effects on the performance of sparse approximate inverses which are used as preconditioners in iterative solvers based on Krylov subspace methods. The reduction of the number of nonzero entries due to the reordering allows to obtain sparse approximate inverses with similar accuracy as those obtained without reordering, but at a lower storage requirement and computational cost. In addition, the reordering provides better quality preconditioners since the number of steps of an iterative solver for convergence is generally reduced. Further research must be carried out on the effect of other reordering techniques which take into account the numerical values of the entries of A (see e.g. [20], [21]). Though these techniques are usually too expensive, when several linear systems involving the same matrix are solved, they may be a competitive choice in parallel machines.

References

[1] M. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.*, 18 (3) :838-853, 1997.

[2] L. González, G. Montero and E. Flórez. Approximate inverse preconditioners using Frobenius inner product I: Theoretical results. *Lin. Alg. Appl.*, to appear.

[3] G. Montero, L. González, A. Suárez and M.D. García. Approximate inverse preconditioners using Frobenius inner product II: Computational aspects. *Lin. Alg. Appl.*, to appear.

[4] M. Benzi and M. Tůma. Orderings for factorized sparse approximate inverse preconditioners. *SIAM J. Sci. Comput.*, to appear.

[5] M. Benzi, D.B. Szyld and A. van Duin. Orderings for incomplete factorization preconditioning of nonsymmetric problems. *SIAM J. Sci. Comput., 20 (5):1652-1670, 1999.*

[6] A. George and J.W. Liu. The evolution of the minimum degree ordering algorithms. *SIAM Rev.* 31 :1-19 1989.

[7] G. Martin. Méthodes de préconditionnement par factorisation incomplete. *Mémoire de Maîtrise*, Université Laval, Québec, Canada, 1991.

[8] A. George. *Computer implementation of the finite element method.* Ph. D. thesis, Department of Computer Science, Stanford University, Stanford, CA, 1971.

[9] H.A. Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 13: 631-644, 1992.

[10] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Statist. Comput.*, 14: 461-469, 1993.

[11] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Co., Boston, 1996.

[12] T.F. Chan, E. Gallopoulos, V. Simoncini, T. Szeto and C.H. Tong. A Quasi-Minimal residual variant of the Bi-CGSTAB Algorithm for Nonsymmetric Systems. *SIAM J. Sci. Comput., 15 (2):338-347, 1994.*

[13] O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, Cambridge, U.K., 1994.

[14] P. Almeida. *Direct solution of sparse systems by graphs.* Ph. D. thesis, Mathematics Department, University of Las Palmas de Gran Canaria, Las Palmas, Spain, 1990 (in Spanish).

[15] L.C. Dutto. The effect of ordering on preconditioned GMRES algorithm, for solving the compressible Navier-Stokes equations, *Int. J. Num. Meth. Eng.*, 36: 457-497, 1993.

[16] E.H. Cuthill and J.M. Mckee. Reducing the bandwidth of sparse symmetric matrices. In *Proc. 24th National Conference of the Association for Computing Machinery*, pages 157-172. Brondon Press, New Jersey, 1969.

[17] . CGS: a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 10: 36-52, 1989.

[18] A.F. Suárez Sarmiento. *Contribution to Biorthogonalization Algorithms for the Solution of Linear Equations.* Ph. D. thesis, Mathematics Department, University of Las Palmas de Gran Canaria, Las Palmas, Spain, 1995 (in Spanish).

[19] G. Montero and A. Suárez. Left-right preconditioning versions of BCG-like methods. *Neural, Parallel & Scientific Comput.*, 3 (4):487-502, 1995.

[20] R.R. Lewis. Simulated annealing for profile and fill reduction of sparse matrices. *Int. J. Num. Meth. Eng*, 37: 905-925, 1994.

[21] G. Montero, M. Galán, P. Cuesta and G. Winter. Effects of stochastic ordering on preconditioned GMRES algorithm. In B.H.V. Topping & M. Papadrakakis editors,.*Advances in Structural Optimization*, pages 241-246. Civil-Comp Press, Edinburgh,1994.(Proc. Computational Structures Technology, Athens, 1994).

1 Department of Mathematics. University of Las Palmas de Gran Canaria. Edif. de Informática y Matemáticas, Campus Univ. de Tafira, 35017-Las Palmas de Gran Canaria