The effect of orderings on sparse approximate inverse preconditioners for nonsymmetric problems

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

Department of Mathematics, University of Las Palmas de Gran Canaria, Edif. de Informática y Matemáticas, Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria, SPAIN.

Abstract

We experimentally study how 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 and computation required for a given accuracy. These properties are illustrated with several numerical experiments taken from the discretization of PDEs by a finite element method and from a standard matrix collection.

Key words: Nonsymmetric linear systems, preconditioning, sparse approximate inverse, reordering techniques, iterative solvers.

1 Introduction

The solution of sparse linear systems $Ax = b,\ A \in \mathcal{M}_n(\mathbb{R})\ ;\ x, b \in \mathbb{R}^n$, by preconditioned Krylov methods is studied. Here A is a large, sparse, nonsymmetric and nonsingular matrix. We focus our work on sparse approximate inverse preconditioners. These preconditioners are specially interesting in a 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, sparse approximate inverse preconditioners may be useful for solving those problems for which other types of preconditioner such as ILU fail. The approximate inverse here considered, is a version of the SPAI proposed by Grote and Huckle [1], and both theoretical and computational aspects have been analyzed in [2]. We present results of the effect of reordering not only on the amount

[*] Corresponding author.

of entries in the sparse approximate inverse, 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 inverse is dependent on the ordering of A. A similar study has been carried out by Benzi and Tůma [3] as well as Bridson and Tang [4] for factorized approximate inverses. Also for interesting results on the effect of ordering for incomplete factorization in the convergence of Krylov subspace methods see [5–7].

The computation of the sparse approximate inverse is presented in section 2. In section 3, some considerations about the reordering techniques, Minimum Degree [8], Reverse Cuthill-Mckee [9] and Minimum Neighboring [10], are discussed. In section 4 we describe the preconditioned BiCGSTAB algorithm [11]. Numerical experiments are presented in order to show the effect of the ordering algorithms on the convergence of BiCGSTAB for the solution of nonsymmetric systems of linear equations when such sparse approximate inverses are used as preconditioners. Similar numerical problems which were carried out for other iterative solvers like GMRES [12,13] and QMRCGSTAB [14] have not been included here since they led to the same conclusions. Results of systems with a matrix belonging to the Harwell-Boeing collection [15] and others arising from the finite element discretization of different problems are presented and commented upon in section 5. Finally, our conclusions are exposed in section 6.

2 Sparse approximate inverse computation

We consider a left preconditioning with matrix M, such that MA is close to the identity. Hence, using Frobenius matrix norm, the problem to solve is

$$\min_{M \in \mathcal{S}} ||MA - I||_F = ||NA - I||_F = \varepsilon \tag{1}$$

$\mathcal{S}$ being an arbitrary vectorial subspace of the linear space $\mathcal{M}_n(\mathbb{R})$ of all $n \times n$ real matrices. The parameter ε represents the accuracy of the approximate inverse M in the subspace S. In [2] a non-factorized approximate inverse M is proposed for a fixed sparsity pattern which is automatically captured for a given number of nonzero entries. Let K be a subset of $\{1, 2, ..., n\} \times \{1, 2, ..., n\}$ and $\mathcal{S} = \{M \in \mathcal{M}_n(\mathbb{R}) / m_{i,j} = 0; \forall (i,j) \notin K\}$. Denote by p_k the number of nonzero entries in the k-th row of $M \in \mathcal{S}$ and by $i_1^k < i_2^k < ... < i_{p_k}^k$ the column indices of these nonzero entries. We have $K = \bigcup_{k=1}^{n} \left\{ \left(k, i_1^k\right), \left(k, i_2^k\right), ..., \left(k, i_{p_k}^k\right) \right\}$, where $1 \leq i_1^k < i_2^k < ... < i_{p_k}^k \leq n$. Let $E_{i,j}$ be the $n \times n$ matrix whose only nonzero entry is $e_{i,j} = 1$. With this notation, the solution to problem (1) for the sparsity pattern K is summarized in the following result.

Theorem 2.1 *Let $A \in \mathcal{M}_n(\mathbb{R})$ be nonsingular and let $\mathcal{S}$ be the subspace of matri-*

ces with sparsity pattern K. Then, the solution to problem (1) is

$$M = \sum_{k=1}^{n}\sum_{l=1}^{p_k} \frac{\det\left(D_l^k\right)}{\det\left(G_{l-1}^k\right)\det\left(G_l^k\right)} \widetilde{M}_{k,l} \tag{2}$$

$$||MA - I||_F^2 = n - \sum_{k=1}^{n}\sum_{l=1}^{p_k} \frac{\left[\det\left(D_l^k\right)\right]^2}{\det\left(G_{l-1}^k\right)\det\left(G_l^k\right)} \tag{3}$$

where, for all $k \in \{1, 2, ..., n\}$, $\det\left(G_0^k\right) = 1$ 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 is the matrix which results from replacing the last row of G_l^k by $a_{i_1^k}, k, a_{i_2^k}, k, ..., a_{i_l^k}, k$, and $\widetilde{M}_{k,l}$ is the matrix obtained evaluating the symbolic determinant which results from replacing the last row of $\det\left(G_l^k\right)$ by $E_{k,i_1^k}, E_{k,i_2^k}, ..., E_{k,i_l^k}$, with $1 \leq l \leq p_k$.

The proof of this theorem is similar to that developed in [2] for the case of right preconditioning.

In [1,16,17], the authors present different algorithms to obtain an approximate inverse as close to A^{-1} as required. We have followed such techniques but changing the selection method for entries in M and the solver for problem (1). Also in [19,20,18], different strategies for choosing the sparsity pattern of M and different ways for solving (1) are proposed. Taking into account equation (2), here the computation of M is carried out in parallel for each row k. Our algorithm allows us to start from any entry of row k. However, since the diagonals of the selected matrices have nonzero entries and the expression of the optimal diagonal preconditioner is well known, we have started from the diagonal approach in the applications. The next entry to be considered is selected from a set of candidates which is defined following the criterion proposed in [1]. Thus, our algorithm automatically searches one by one the non null entries in matrix M, starting from the diagonal structure. Denote by $r_k^t = m_k^t A - e_k^t$, the residual corresponding to the vector m_k^t which represents the k-th row of M, and let $\mathcal{I}_k$ be the set of indices of the non null entries in r_k^t, i.e., $\mathcal{I}_k = \{i \in \{1, 2, ..., n\} \;/\; r_{ki} \neq 0\}$. Here, e_k^t is the k-th row of the identity matrix. If $\mathcal{L}_k = \{l \in \{1, 2, ..., n\} \;/\; m_{kl} \neq 0\}$, then the new entry is searched in the set $\mathcal{J}_k = \{j \in \mathcal{L}_k^c \,/\, a_{ji} \neq 0, \forall i \in \mathcal{I}_k\}$, with $\mathcal{L}_k^c$ being the complementary set of $\mathcal{L}_k$. 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 $j \in \mathcal{J}_k$. Frobenius matrix norm may be parallelized as

$$||MA - I||_F^2 = \sum_{k=1}^{n} ||m_k^t A - e_k^t||_2^2 \tag{4}$$

Thus, taking into account equations (3) and (4), for each j we compute

$$||m_k^t A - e_k^t||_2^2 = 1 - \sum_{l=1}^{p_k} \frac{\left[\det\left(D_l^k\right)\right]^2}{\det\left(G_{l-1}^k\right)\det\left(G_l^k\right)} \tag{5}$$

where, for all k, $\det\left(G_0^k\right)$, $\det\left(G_l^k\right)$ and $det\left(D_l^k\right)$ have the same above definitions, 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 $\mathcal{L}_k$ and leading us to an actual optimum where all the entries given by the indices of $\mathcal{L}_k$ are updated. Therefore, 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

$$m_k^t = \sum_{l=1}^{p_k} \frac{\det\left(D_l^k\right)}{\det\left(G_{l-1}^k\right)\det\left(G_l^k\right)} \widetilde{m}_l^t \tag{6}$$

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 $1 \leq l \leq p_k$. Equation (6) is obtained identifying columns in (2).

Evidently, the computation of $||m_k^t A - e_k^t||_2$ and m_k^t is updated adding the contribution of the last entry $j \in \mathcal{J}_k$ to the previous sum from 1 to $p_k - 1$. In practice, $\det\left(G_l^k\right)$ is computed using Cholesky decomposition since G_l^k is a symmetric and positive definite matrix. It only involves the factorization of the last row and column if we take advantage of G_{l-1}^k decomposition. Moreover, note that $\det\left(D_l^k\right) / \det\left(G_l^k\right)$ is the value of the last unknown of the system $G_l^k d_l = \left(a_{i_1^k k}, a_{i_2^k k}, ..., a_{i_l^k k}\right)^t$ and only a complete forward-substitution is carried out. Finally, to obtain $\widetilde{m}_l$ is to solve the system $G_l^k v_l = e_l$, with $\widetilde{m}_{i_h^k l} = v_{hl}$ $(1 \leq h \leq l)$.

Sparse approximate inverse algorithm.

```
Start from the optimal diagonal preconditioner
for $k = 1$ to $n$
    while $||m_k^t A - e_k^t||_2 > \varepsilon_k$ and $p_k < n_k$ do
        Compute $r_k^t = m_k^t A - e_k^t$
        Define $\mathcal{I}_k = \{i \in \{1, 2, ..., n\} \,/\, r_{ki} \neq 0\}$
        Define $\mathcal{L}_k = \{l \in \{1, 2, ..., n\} \,/\, m_{kl} \neq 0\}$
        Define $\mathcal{J}_k = \{j \in \mathcal{L}_k^c \,/\, a_{ji} \neq 0, \forall i \in \mathcal{I}_k\}$
        Select $j_k \in \mathcal{J}_k$ for which $||m_k^t A - e_k^t||_2$ is minimal
        Update $m_k^t$ using (6)
    end
end
```

3 Some remarks on reordering

We have considered several reordering techniques in order to illustrate the effect of ordering on the iterative solution of linear systems of equations using SPAI preconditioners. The original ordering corresponds to matrices directly arising from the Finite Element Method with unstructured meshes and adaptive mesh refinement. The minimum degree ordering has been used to reduce the fill-in in factorization of matrices with symmetric sparsity pattern. Its performance is discussed in [8]. The Reverse Cuthill-Mckee algorithm [9] is a modification of the Cuthill-Mckee algorithm [22] that simply takes the reverse ordering of that obtained by the latter. The general characteristics of these algorithms are the reduction of bandwidth and profile. The Minimum Neighbouring algorithm [10] is a variant of the Minimum 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. These reordering algorithms are summarized in [21,7]. In all the algorithms, we start from a pseudo-peripheral node searched by George's algorithm [23].

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?
Let P be the permutation matrix corresponding to a reordering scheme. Since $(P^TAP)^{-1} = P^TA^{-1}P$, when we reorder a matrix, its approximate inverse should tend to the reordered inverse. Then, since P is orthogonal and the Frobenius norm is invariant with respect to the product by orthogonal matrices, it yields

$$\begin{aligned} \left\| N'P^TAP - I \right\|_F &= \min_{M' \in P^TSP} \left\| M'P^TAP - I \right\|_F \\ &= \min_{M \in S} \left\| MA - I \right\|_F = \left\| NA - I \right\|_F \end{aligned} \tag{7}$$

and also

$$N' = P^TNP \tag{8}$$

Equation (7) means that the accuracy of the best preconditioner N of matrix A in the subspace S is equal to the accuracy of the best preconditioner N' of matrix P^TAP in the subspace P^TSP (for any permutation matrix P).

Now, let S_0, S_0' be the patterns generated by our SPAI algorithm corresponding to the original matrix A and the reordered matrix P^TAP, respectively, for the same accuracy. Our SPAI algorithm always obtains the best approximate inverse in each subspace of $M_n(\mathbb{R})$ that it generates. Thus, if the reordering is efficient in such a way that

$$\min_{M \in S_0'} \left\| MP^T AP - I \right\|_F \leq \min_{M' \in P^T S_0 P} \left\| M' P^T AP - I \right\|_F \tag{9}$$

then, from equations (7) and (9), we obtain

$$\min_{M \in S_0'} \left\| MP^T AP - I \right\|_F \leq \min_{M \in S_0} \left\| MA - I \right\|_F \tag{10}$$

that is, if (9) is satisfied, the application of our algorithm to the reordered matrix $P^T AP$ improves the accuracy of the result obtained for the original matrix A.

So we will look for reordering techniques which allow the generation of patterns S_0' satisfying (9). Evidently, if the permutation matrix P is such that $S_0' \supseteq P^T S_0 P$, then we can ensure that relation (9) is satisfied and thus also inequality (10) is held, but in this case more nonzero entries would be involved and the computational cost would increase. Nevertheless, relation (9) may be satisfied even if $S_0' \not\supseteq P^T S_0 P$. For example, suppose that S_0' contains less nonzero entries than $P^T S_0 P$ but in better locations. The behaviour of different reordering techniques in the reduction of the amount of entries in the approximate inverse for a given accuracy ε is illustrated in the numerical experiments.

If (9) is assumed, the reordered SPAI preconditioners acquire better properties for the performance of the iterative solver [2]. To study the closeness of the condition number (related to the spectral norm) of $N' P^T AP$ to 1 we proceed as follows. On one hand we have

$$\left\| N' P^T AP \right\|_2 = \left\| (N' P^T AP - I) + I \right\|_2 \leq 1 + \left\| N' P^T AP - I \right\|_2 \tag{11}$$

and on the other hand, assuming that $\left\| N' P^T AP - I \right\|_2 < 1$ we can use the Banach lemma [24, chapter 1, lemma 1.2.1] to obtain

$$\left\| (N' P^T AP)^{-1} \right\|_2 = \left\| (I - (I - N' P^T AP))^{-1} \right\|_2 \leq \frac{1}{1 - \left\| N' P^T AP - I \right\|_2} \tag{12}$$

Finally, equations (11) and (12) yield,

$$\kappa_2(N' P^T AP) \leq \frac{1 + \left\| N' P^T AP - I \right\|_2}{1 - \left\| N' P^T AP - I \right\|_2} \tag{13}$$

From the well-known characterization of normality (see e.g. [25, chapter 3, pp. 156-157]: $A \in M_n(\mathbb{R})$ is normal if and only if $|\lambda_k(A)| = \sigma_k(A)$; $\forall i = 1, 2, ..., n$, the $(N' P^T AP)$'s departure from normality may be estimated with the quantity

$$\frac{1}{n} \sum_{k=1}^{n} \left(|\lambda_k| - \sigma_k \right)^2 \tag{14}$$

with $\{\lambda_k\}_{k=1}^n$, $\{\sigma_k\}_{k=1}^n$ being the eigenvalues and singular values of $N'P^TAP$ (non increasing modules sequence). To evaluate (14) we consider the following relation

$$\sum_{k=1}^{n} |\lambda_k|^2 \leq \sum_{k=1}^{n} \sigma_k^2 \tag{15}$$

which is satisfied for all square matrix [25, chapter3, theorem 3.3.13]. Furthermore, take into account that N' is the solution to the optimization problem

$$\left\| N'P^TAP - I \right\|_F = \min_{M' \in S_0'} \left\| M'P^TAP - I \right\|_F \tag{16}$$

the orthogonal projection theorem yields

$$\left\| N'P^TAP \right\|_F^2 = tr\left(N'P^TAP\right) \tag{17}$$

that is

$$\sum_{k=1}^{n} \sigma_k^2 = \sum_{k=1}^{n} \lambda_k \tag{18}$$

Hence, using (15) and (18) we obtain

$$\begin{aligned} \frac{1}{n}\sum_{k=1}^{n} \left(|\lambda_k| - \sigma_k\right)^2 &= \frac{1}{n}\left[\sum_{k=1}^{n} |\lambda_k|^2 - 2\sum_{k=1}^{n} |\lambda_k|\,\sigma_k + \sum_{k=1}^{n}\sigma_k^2\right] \\ &\leq \frac{2}{n}\left[\sum_{k=1}^{n}\sigma_k^2 - \sigma_n\sum_{k=1}^{n} |\lambda_k|\right] \leq \frac{2}{n}\left[\sum_{k=1}^{n}\sigma_k^2 - \sigma_n\sum_{k=1}^{n}\lambda_k\right] \\ &= \frac{2}{n}\left(1 - \sigma_n\right)\sum_{k=1}^{n}\sigma_k^2 = \frac{2}{n}\left(1 - \sigma_n\right)\left\| N'P^TAP \right\|_F^2 \end{aligned} \tag{19}$$

Finally, the eigenvalues and singular values are clustered at 1 (see [1]),

$$\sum_{k=1}^{n} (1 - \sigma_k)^2 \leq \left\| N'P^TAP - I \right\|_F^2 \tag{20}$$

$$\sum_{k=1}^{n} |1 - \lambda_k|^2 \leq \left\| N'P^TAP - I \right\|_F^2 \tag{21}$$

4 Iterative solver

The BiCGSTAB algorithm proposed in [11] is a variant of the BiCG algorithm which provides a smoother convergence behaviour than CGS algorithm [26]. 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 [27]. A preconditioned version of this algorithm is presented below. When we use an incomplete factorization, the preconditioning involves the classical procedures of backward and forward substitution twice in each iteration. On the other hand, if we use the SPAI preconditioner, only two matrix-vector products are required, which is the main feature from the point of view of parallelization.

5 Numerical experiments

We begin with a study of a problem in the Harwell-Boeing matrix collection: *orsreg1*. This is an oil reservoir simulation matrix for a $21 \times 21 \times 5$ full grid of size $n = 2205$ with $nz = 14133$ nonzero entries. Tables 1-3 show the results obtained with preconditioned BiCGSTAB after Original, Minimum Degree and Reverse Cuthill-Mckee reordering of matrix A, respectively. The performance of ILU(0) is compared with several SPAI preconditioners corresponding to different levels of fill-in. We present the number of iterations, the number of entries in M and the Frobenius norm of the residual matrix.

The values of nz for SPAI in the reordering cases become lower than those in the original ordering from $\varepsilon_k = 0.2$. However, in the other cases the norm of residual matrix is always reduced with reordering and thus it does not contradict the above theoretical results. On the other hand, the number of BiCGSTAB iterations always decreased when we used reordering except for the first SPAI which gave instabilities to the convergence behaviour of the iterative method. We also observed in this experiment that with similar storage requirements to the ones of ILU(0), a faster convergence with SPAI ($\varepsilon_k = 0.3$, or even $\varepsilon_k = 0.2$ if reordered by Reverse Cuthill-Mckee) is produced. Figure 1 represents the performance of BiCGSTAB algorithm when ILU(0) and SPAI(0.3) are constructed after reordering. The main conclusion is that SPAI might compete with ILU in a parallel framework. Furthermore, if we apply a suitable reordering algorithm to both strategies this competitiveness still continues.

The second example is a convection-diffusion problem (*convdifhor* [28]) 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 $\{(0.5, 0),$ $(1, 0.5), (0.5, 0.5)\}$ 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 on the boundary we set null Neumann conditions. The matrix corresponds to an unstructured triangular mesh of finite elements with $n = 1960$ and $nz = 13412$.

Tables 4-7 are similar to those of the previous problem. 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 iterations 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 2(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 outside the profile. This is clearly illustrated in Figure 2(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 3 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.

Table 1
orsreg1: *Original Ordering* and left preconditioned BiCGSTAB

Preconditioner	Iter.	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA - I\|\|_F$
Unprecond.	548	2205	0.16	854454
ILU(0)	47	14133	1.00	–
SPAI $\varepsilon_k = 0.6$	299	3087	0.22	26.77
SPAI $\varepsilon_k = 0.5$	169	6615	0.47	22.31
SPAI $\varepsilon_k = 0.4$	86	10353	0.73	13.67
SPAI $\varepsilon_k = 0.3$	59	11025	0.78	11.45
SPAI $\varepsilon_k = 0.2$	37	31782	2.25	8.56

The third numerical experiment is also a convection-diffusion problem (*cuaref* [28]) given in $[0,1] \times [0,1]$ by the equation,

$$v_1 \frac{\partial u}{\partial x} + v_2 \frac{\partial u}{\partial y} - K \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = 0$$

Table 2
orsreg1: *Minimum Degree* and left preconditioned BiCGSTAB

Preconditioner	Iter.	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Unprecond.	> 2205	2205	0.16	854454
ILU(0)	381	14133	1.00	–
SPAI $\varepsilon_k = 0.6$	1396	3593	0.25	25.62
SPAI $\varepsilon_k = 0.5$	73	7199	0.51	20.93
SPAI $\varepsilon_k = 0.4$	36	10678	0.76	13.02
SPAI $\varepsilon_k = 0.3$	21	13850	0.98	8.24
SPAI $\varepsilon_k = 0.2$	14	19694	1.39	5.36

Table 3
orsreg1: *Reverse Cuthill-McKee* and left preconditioned BiCGSTAB

Preconditioner	Iter.	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Unprecond.	> 2205	2205	0.16	854454
ILU(0)	26	14133	1.00	–
SPAI $\varepsilon_k = 0.6$	810	4152	0.29	23.84
SPAI $\varepsilon_k = 0.5$	88	7005	0.50	19.57
SPAI $\varepsilon_k = 0.4$	27	9304	0.66	11.62
SPAI $\varepsilon_k = 0.3$	19	10608	0.75	8.08
SPAI $\varepsilon_k = 0.2$	12	13322	0.94	5.74

where $v_1 = C\left(y - \frac{1}{2}\right)(x - x^2), v_2 = C\left(\frac{1}{2} - x\right)(y - y^2)$, $K = 1$, and $C = 10^5$. The boundary conditions are the same as the above case. The matrix corresponds to a refinement step of an unstructured mesh of finite elements with $n = 7520$ and $nz = 52120$. Tables 8-11 indicate the performance of ILU(0) and SPAI preconditioners for *cuaref*. The reduction of the amount of entries in SPAI for a given accuracy of the approximate inverse is also evident here (from 20 to 30 per cent approximately for Minimum Degree and Reverse Cuthill-Mckee). Moreover, the number of iterations of BiCGSTAB is drastically reduced from 75 to 85 and from 60 to 70 per cent, respectively. The sparsity pattern of SPAI(0.2) matrices corresponding to the original ordering and the reordering algorithms considered here have similar properties to Figure 2. We conclude, as in the previous problem and also in other experiments carried out, that the sparsity pattern of SPAI seems to start from a structure similar to that typical of A obtained by reordering, and tends to a full matrix as we augment its accuracy. The convergence curves in this case do not show up any significant differences to Figure 3 in the second problem. Minimum Degree and Reverse Cuthill-Mckee are preferable to Minimum Neighbouring or the original ordering. However, we have noticed that if nz is increased, the differences

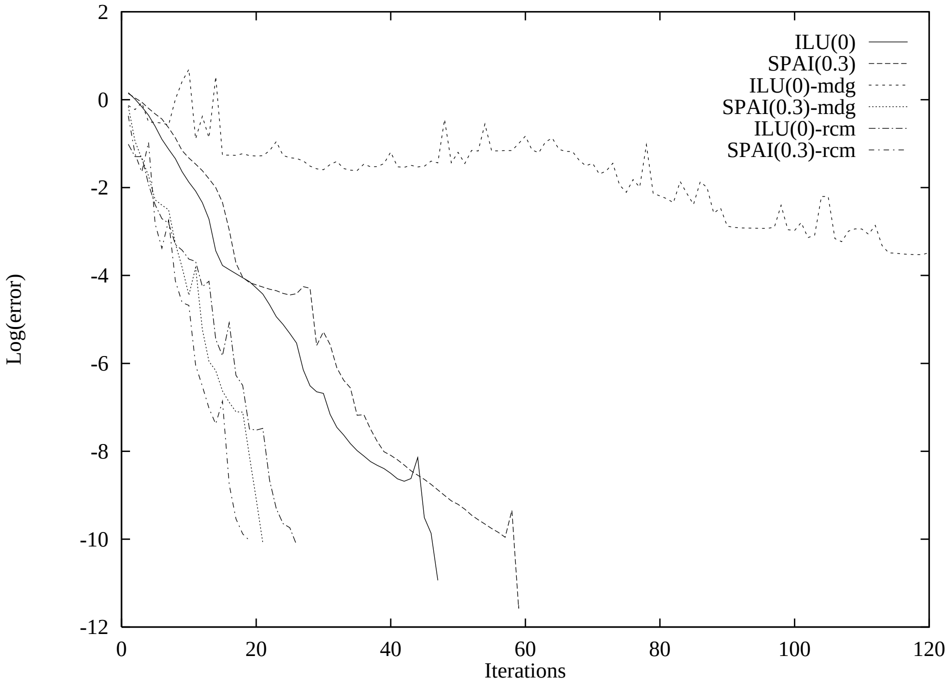


Fig. 1. Comparison of the performance of BiCGSTAB-ILU(0) and BiCGSTAB-SPAI with reordering for *orsreg1*.

between Minimum Degree and Reverse Cuthill-Mckee are more appreciable and that the first one becomes a better choice.

Table 4
convdifhor: *Original Ordering* and left preconditioned BiCGSTAB

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 5
convdifhor: *Minimum Degree* and left preconditioned BiCGSTAB

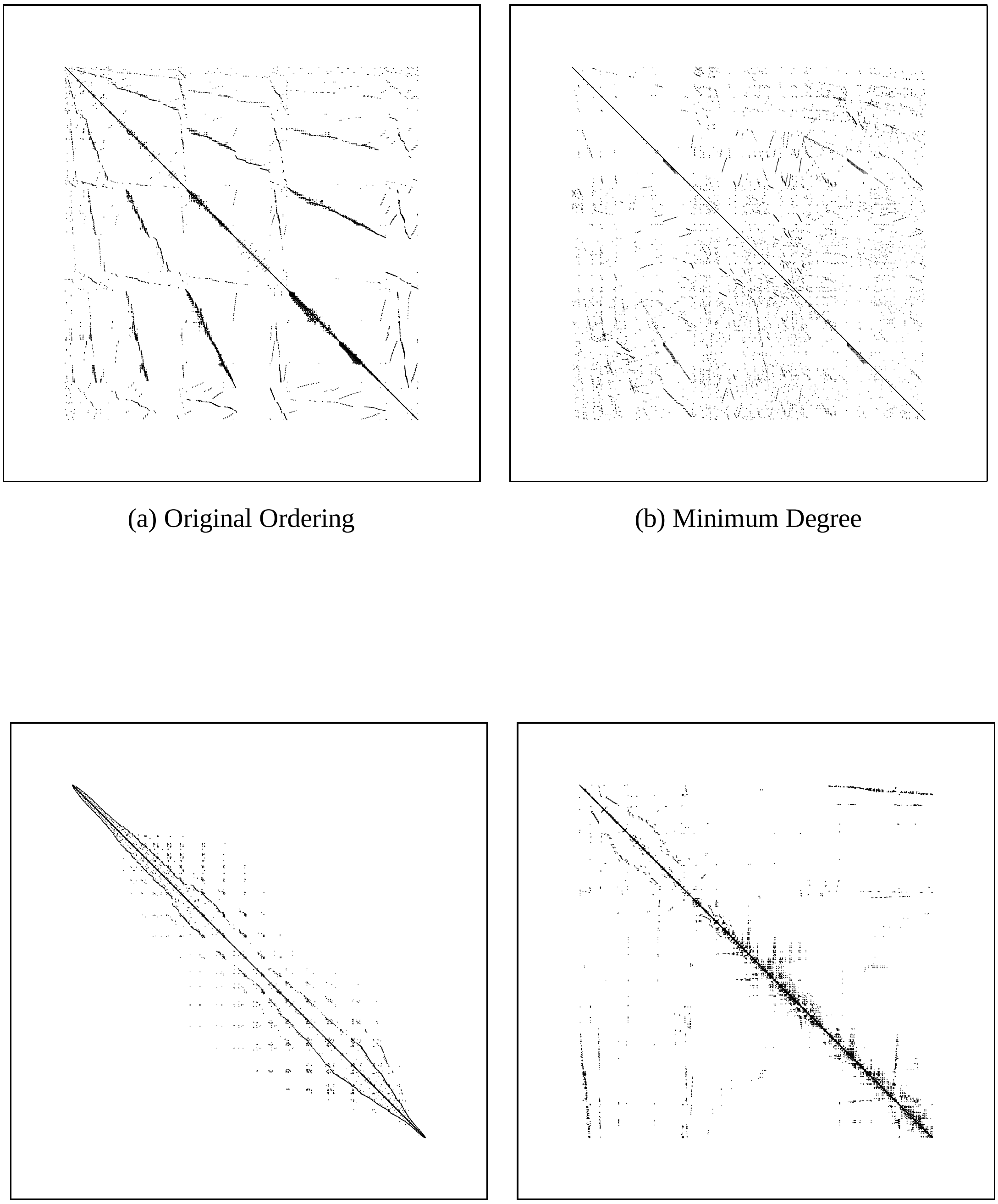
Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA - I\|\|_F$
Unprecond.	> 1960	1960	0.15	13275
ILU(0)	57	13412	1.00	–
SPAI $\varepsilon_k = 0.6$	166	2617	0.20	19.82
SPAI $\varepsilon_k = 0.4$	99	6255	0.47	15.18
SPAI $\varepsilon_k = 0.3$	68	11461	0.85	11.42
SPAI $\varepsilon_k = 0.2$	40	26992	2.01	7.78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

Table 6
convdifhor: *Reverse Cuthill-McKee* 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 7
convdifhor: *Minimum Neighbouring* 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 $\varepsilon_k = 0.1$	21	167678	12.50	4.36



(a) Original Ordering

(b) Minimum Degree

(c) Reverse Cuthill-Mckee

(d) Minimum Neighbouring

Fig. 2. Sparsity pattern of SPAI(0.3) matrix with different orderings for *convdifhor.*

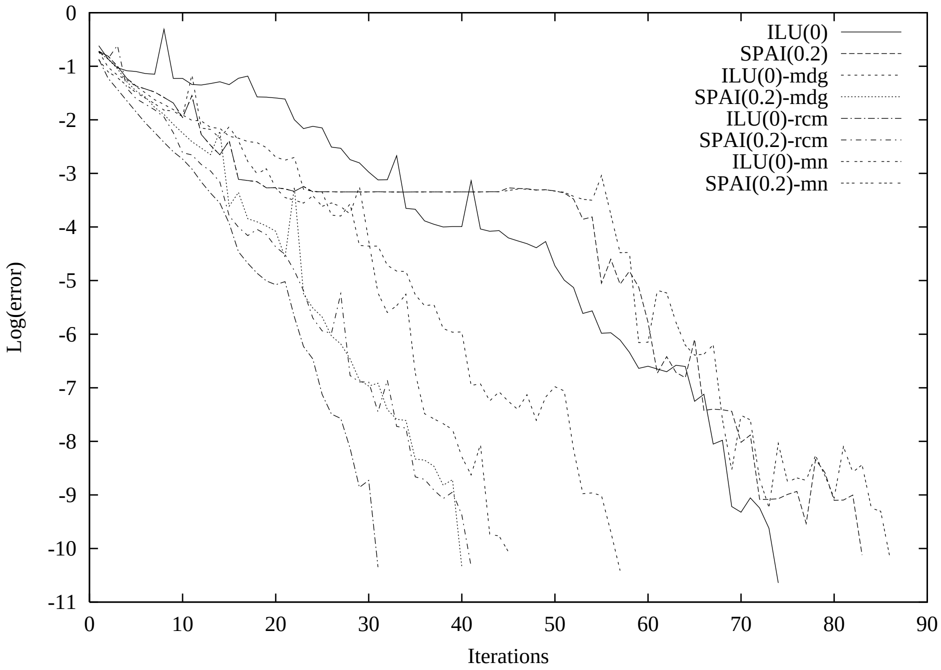


Fig. 3. Comparison of the performance of BiCGSTAB-SPAI with reordering for *convdifhor.*

Table 8

cuaref: *Original Ordering* and left preconditioned BiCGSTAB

Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Unprecond.	1740	7520	0.14	725.81
ILU(0)	378	52120	1.00	–
SPAI $\varepsilon_k = 0.4$	584	32227	0.62	32.78
SPAI $\varepsilon_k = 0.3$	455	61400	1.18	25.24
SPAI $\varepsilon_k = 0.2$	214	152078	2.92	17.12

Table 9

cuaref: *Minimum Degree* and left preconditioned BiCGSTAB

Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Unprecond.	395	7520	0.14	705.14
ILU(0)	69	52120	1.00	–
SPAI $\varepsilon_k = 0.4$	125	25256	0.48	31.86
SPAI $\varepsilon_k = 0.3$	79	47147	0.90	24.66
SPAI $\varepsilon_k = 0.2$	51	106302	2.04	16.77

Table 10
cuaref: *Reverse Cuthill-McKee* and left preconditioned BiCGSTAB

Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Unprecond..	> 7520	7520	0.14	703.54
ILU(0)	20	52120	1.00	–
SPAI $\varepsilon_k = 0.4$	215	24684	0.47	31.69
SPAI $\varepsilon_k = 0.3$	178	45808	0.88	24.56
SPAI $\varepsilon_k = 0.2$	72	103391	1.98	16.67

Table 11
cuaref: *Minimum Neighbouring* and left preconditioned BiCGSTAB

Preconditioner	Iter	$nz(M)$	$nz(M)/nz(A)$	$\|\|MA-I\|\|_F$
Unprecond.	1739	7520	0.14	725.81
ILU(0)	102	52120	1.00	–
SPAI $\varepsilon_k = 0.4$	577	32227	0.62	32.78
SPAI $\varepsilon_k = 0.3$	410	61400	1.18	25.24
SPAI $\varepsilon_k = 0.2$	209	152078	2.92	17.12

6 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 one to obtain sparse approximate inverses with similar accuracy to 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. [29,30]). Though these techniques are usually too expensive, when several linear systems involving the same matrix are solved, they may be a competitive choice for use on parallel machines.

7 Acknowledgments

We would like to thank two anonymous referees for their careful reading of the manuscript. Their comments and suggestions helped us improve the exposition of this paper. We are also grateful to P. Almeida and A. George for providing us with some of the reordering software which were used in the numerical experiments and to E. Rodríguez for his technical assistance. This work was partially support by MCYT, Spain, grant contract: REN2001-0925-C03-02/CLI.

References

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

[2] G. Montero, L. González, E. Flórez, M.D. García and A. Suárez, Approximate inverse computation using Frobenius inner product, *Numer. Lin. Alg. Appl.*, to appear in 2002.

[3] M. Benzi and M. Tůma, Orderings for factorized sparse approximate inverse preconditioners, *SIAM J. Sci. Comput.*, **21**(5) (2000) 1851-1868.

[4] R. Bridson and W.-P. Tang, Ordering, anisotropy, and factored sparse approximate inverses, *SIAM J. Sci. Comput.*, **21**(3) (1999) 867-882.

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

[6] M. Benzi, W. Joubert and G. Mateesen, Numerical experiments with parallel orderings for ILU preconditioners, *Electr. Transact. Numer. Anal.*, **8** (1999) 88-114.

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

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

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

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

[11] 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** (1992) 631-644.

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

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

[14] 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) (1994) 338-347.

[15] I.S. Duff, R.G. Grimes and J.G. Lewis, Sparse matrix test problems, *ACM Trans. Math. Software*, **15** (1989) 1-14.

[16] N.I.M. Gould and J.A. Scott, Sparse approximate-inverse preconditioners using norm minimization techniques, *SIAM J. Sci. Comput.*, **19**(2) (1998) 605-625.

[17] W.-P. Tang, Toward an effective sparse approximate inverse preconditioner, *SIAM J. Matrix Anal. Appl.*, **20**(4) (1999) 970-986.

[18] T.K. Huckle, Efficient computation of sparse approximate inverses, *Numer. Lin. Alg. Appl.*, **5** (1998) 57-71.

[19] E. Chow, A priori sparsity patterns for parallel sparse approximate inverse preconditioners, *SIAM J. Sci. Comput.*,**21**(5) (2000) 1804-1822.

[20] B. Carpentieri, I.S. Duff and L. Giraud, Sparsity pattern selection strategies for robust Frobenius-norm minimization preconditioners in electromagnetism, *Numer. Lin. Alg. Appl.*, **7** (2000) 667-685.

[21] 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).

[22] 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* (Brondon Press, New Jersey, 1969) 157-172.

[23] A. George and J.W. Liu, An implementation of a pseudo-peripheral node finder,*ACM Trans. Math. Software* **5** (1979) 284-295.

[24] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations* (Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995).

[25] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis*, (Cambridge University Press, Cambridge, 1999).

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

[27] 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).

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

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

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