SIAM J. SCI. COMPUT.
Vol. 21, No. 5, pp. 1851–1868

ORDERINGS FOR FACTORIZED SPARSE APPROXIMATE INVERSE PRECONDITIONERS*

MICHELE BENZI† AND MIROSLAV TŮMA‡

Abstract. The influence of reorderings on the performance of factorized sparse approximate inverse preconditioners is considered. Some theoretical results on the effect of orderings on the fill-in and decay behavior of the inverse factors of a sparse matrix are presented. It is shown experimentally that certain reorderings, like minimum degree and nested dissection, can be very beneficial. The benefit consists of a reduction in the storage and time required for constructing the preconditioner, and of faster convergence of the preconditioned iteration in many cases of practical interest.

Key words. sparse linear systems, sparse matrices, preconditioned Krylov subspace methods, graph theory, orderings, decay rates, factorized sparse approximate inverses, incomplete biconjugation

AMS subject classifications. Primary, 65F10, 65N22, 65F50; Secondary, 15A06

PII. S1064827598339372

1. Introduction. We consider the solution of sparse linear systems $Ax = b$ by preconditioned iterative methods, where the preconditioners are sparse approximate inverses of A. Such preconditioners have particular interest from the point of view of parallel computation since their application at each step of an iterative method requires only sparse matrix–vector products, which are relatively easy to parallelize. Moreover, sparse approximate inverse preconditioners often succeed in solving difficult problems for which ILU-type methods fail, and therefore they can be useful even on sequential computers. A comprehensive survey of sparse approximate inverse preconditioners, together with the results of extensive numerical tests aimed at assessing the performance of the various methods, can be found in [6]. One of the conclusions of that study was that factorized forms, in which the approximate inverse is the product of two sparse triangular matrices, tend to perform better than nonfactorized ones in the sense that they often deliver better convergence rates for the same amount of nonzeros in the preconditioner. Factorized approximate inverses are also much less expensive to compute than other forms, at least in a sequential environment. As mentioned in [6], another potential advantage of the factorized approach is the fact that such preconditioners are sensitive to the ordering of the equations and unknowns. Indeed, for a sparse matrix A the amount of *inverse fill*, which is defined as the number of structurally nonzero entries in the inverse triangular factors, is strongly dependent on the ordering of A. In contrast, the inverse A^{-1} is usually full, regardless of the

*Received by the editors May 26, 1998; accepted for publication (in revised form) December 12, 1998; published electronically April 28, 2000. This work was performed by an employee of the U.S. Government or under U.S. Government contract. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/sisc/21-5/33937.html

†Scientific Computing Group (CIC-19), MS B256, Los Alamos National Laboratory, Los Alamos, NM 87545 (benzi@lanl.gov). The work of this author was supported in part by Department of Energy grant W-7405-ENG-36 with Los Alamos National Laboratory.

‡Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic (tuma@uivt.cas.cz). The work of this author was supported by Grant Agency of the Czech Academy of Sciences grant 2030706 and by Grant Agency of the Czech Republic grant 205/96/0921.

ordering chosen. In this paper we focus on the factorized sparse approximate inverse preconditioner AINV based on incomplete (bi)conjugation, developed in [2], [4]. As already noted in [4], this algorithm can benefit from the minimum degree ordering, which tends to reduce the amount of inverse fill without negatively impacting the rate of convergence. Here we present results with several orderings in addition to minimum degree. We consider the effect of reorderings from the point of view of the induced fill in the inverse factors using tools from graph theory, and we obtain some insight into the concomitant effects on the decay of the entries in the inverse factors. We conclude that reorderings, particularly minimum degree and nested dissection, can significantly enhance the performance of the AINV preconditioner. We also look briefly at other approximate inverse techniques, showing that the effect of ordering can be quite different on different methods. In the present study we restrict our attention to symmetric permutations of the coefficient matrix A, i.e., of the form P^TAP for a permutation matrix P. These permutations do not alter the spectrum of A. Also, we are mostly interested in solving linear systems arising from the discretization of partial differential equations, which typically give rise to matrices which are structurally symmetric or nearly so, and nonsymmetric permutations would destroy the symmetry. If A is structurally symmetric the reorderings are based on the (undirected) graph associated with the structure of A; otherwise, the structure of $A + A^T$ is used. See [13], [18], [36] for basic material on sparse matrix orderings.

This paper complements a recent independent study by Bridson and Tang [7], who also considered the effect of ordering on AINV. Our main conclusions are similar to those reached in [7]; however, several of our results are not found in [7], and conversely, several orderings and new heuristics not considered here can be found in [7]. Where we overlap, our results and those in [7] are in good agreement. See also [17] for related work.

2. The AINV algorithm. The AINV algorithm [2], [4], [6] constructs a factorized sparse approximate inverse of the form

$$M = ZD^{-1}W^T \approx A^{-1},$$

where Z, W are unit upper triangular matrices and D is diagonal. The approximate inverse factors Z and W are sparse approximations of the inverses of the L and U factors in the LDU decomposition of A. The AINV algorithm computes Z, W and D directly from A by means of an incomplete biconjugation process, in which small elements are dropped to preserve sparsity. In order to describe the procedure, let a_i^T and c_i^T denote the ith row of A and A^T, respectively (i.e., c_i is the ith column of A). Also, let e_i denote the ith unit basis vector. The basic A-biconjugation procedure can be written as follows.

Algorithm 2.1. Biconjugation algorithm.

(1) *Let* $w_i^{(0)} = z_i^{(0)} = e_i \quad (1 \le i \le n)$

(2) *For* $i = 1, 2, \ldots, n$ *do*

(3) *For* $j = i, i+1, \ldots, n$ *do*

(4) $p_j^{(i-1)} := a_i^T z_j^{(i-1)}$; $q_j^{(i-1)} := c_i^T w_j^{(i-1)}$

(5) *End do*

(6) *if* $i = n$ *go to* (11)

(7) *For* $j = i+1, \ldots, n$ *do*

(8) $z_j^{(i)} := z_j^{(i-1)} - \left(\dfrac{p_j^{(i-1)}}{p_i^{(i-1)}}\right) z_i^{(i-1)}$; $w_j^{(i)} := w_j^{(i-1)} - \left(\dfrac{q_j^{(i-1)}}{q_i^{(i-1)}}\right) w_i^{(i-1)}$

(9) *End do*

(10) *End do*

(11) *Let* $z_i := z_i^{(i-1)}$, $w_i := w_i^{(i-1)}$ *and* $p_i := p_i^{(i-1)}$, *for* $1 \leq i \leq n$. *Return* $Z = [z_1,\ z_2, \ldots, z_n]$, $W = [w_1,\ w_2, \ldots, w_n]$, *and* $D = \text{diag}(p_1, p_2, \ldots, p_n)$.

Sparsity is preserved by dropping in the z- and w-vectors after the updates at step (8). If $A = A^T$, then $Z = W$ and the columns of Z are (approximately) A-conjugate. The incomplete procedure is well defined, i.e., no breakdown can occur, if A is an H-matrix. In the general case, diagonal shifts may be necessary in order to prevent breakdowns. See [2], [4], [6] for a detailed study of this algorithm and comparisons with other preconditioners.

3. Structural considerations. In this section we review the effect of orderings on the amount of fill occurring in the inverse triangular factors of a sparse matrix A. See also [7] for a treatment along similar lines.

The inverse of a sparse irreducible matrix is structurally full [12], [19], and this property is obviously invariant under permutations. However, A^{-1} may be representable as the product of two sparse triangular matrices. We are interested in finding permutations of A such that the inverse triangular factors Z, W of A preserve a good deal of sparsity. For example, consider an irreducible matrix partitioned in the following block form:

$$A = \begin{bmatrix} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ C_1 & C_2 & A_3 \end{bmatrix}.$$

Then A^{-1} is full, but the inverse triangular factors of A have the same block structure as the lower and upper block triangular parts of A. In particular, fill-in can occur only inside the nonzero blocks. If a similar structure is imposed on the diagonal blocks A_1 and A_2, as is done in nested dissection, then it is clear that the inverse factors will retain a considerable degree of sparsity.

Let A be an unsymmetric $n \times n$ matrix that has a factorization $A = LU$ without pivoting. Let $G(A) = (V(A), E(A))$ be the directed graph of the matrix A, where $V(A) = \{1, \ldots, n\}$ is the vertex set and $E(A)$ is the set of edges $\langle i, j \rangle$ with i, j such that $a_{ij} \neq 0$. Let $x \in V(A)$. The *closure* $\text{cl}_{G(A)}(x)$ of x in $G(A)$ is the set of vertices of $G(A)$ from which there are paths to x. The *structure* of a vector v is defined as $\text{Struct}(v) = \{i | v_i \neq 0\}$. In the following we will state results for the factor L^{-1}. Similar results hold, of course, also for the factor U^{-1}. These results were given in [19], [20]. The usual no cancellation assumption is made throughout. We denote by $L^{-1}(*, i)$ the ith column of L^{-1}.

Proposition 3.1. $\text{Struct}(L^{-1}(*, i)) = \text{cl}_{G(L)}(i)$.

Let $G^\circ(L)$ denote the *transitive reduction* of the directed acyclic graph (dag) $G(L)$. This is a graph with a minimal number of edges which satisfies the following condition: $G^\circ(L)$ has a directed path from i to j if and only if $G(L)$ has a directed path from i to j. Then the following result holds [20].

Proposition 3.2. $\text{Struct}(L^{-1}(*, i)) = \text{cl}_{G^\circ(L)}(i)$.

We mention two simple consequences of this relation.

Proposition 3.3. *Assume that* $\text{cl}_{G^\circ(L)}(i) \cap \text{cl}_{G^\circ(L)}(j) = \emptyset$. *Then*

$$\text{Struct}(L^{-1}(*, i)) \cap \text{Struct}(L^{-1}(*, j)) = \emptyset.$$

Proposition 3.4. *Let* $K = \text{cl}_{G^\circ(L)}(i) \cap \text{cl}_{G^\circ(L)}(j)$. *If* $K \neq \emptyset$, *then all the entries below the main diagonal in* $L^{-1}(K, K)$ *are nonzero.*

Another way to phrase this structural characterization is by saying that the (i, j) element of $W = L^{-T}$ is nonzero if and only if j is an ancestor of i in the elimination tree; see [7]. Incidentally, this characterization is considerably simpler than the one given in [4], which was intended mainly to serve as a guide for the extension of threshold pivoting strategies to the biconjugation process.

The construction of factorized approximate inverse preconditioners is naturally influenced by the inverse fill. Orderings that cause relatively low inverse fill can be expected to result in more sparse approximate inverse factors and possibly in faster computation of the preconditioner. On the other hand, it is difficult to predict the impact that reorderings obtained using unweighted graph information only will have on the rate of convergence of the preconditioned iteration; see the next section for a discussion.

In any event, it makes sense to attempt to keep the number of nonzeros in L^{-1} small, i.e., to try to minimize for each j, with respect to the ordering, the sum

$$\sum_{i<j} |\mathrm{cl}_{G^\circ(L)}(i) \cap \mathrm{cl}_{G^\circ(L)}(j)|. \tag{3.1}$$

This sum represents the overlaps between the closure of vertex j and the remaining closures in the elimination dag. Finding an ordering that minimizes this sum would result in overlaps between individual closures which are as small as possible or, in other words, in an elimination dag which is as bushy as possible. Since the inverse fill corresponds to the closures of the vertices, a bushy dag can be expected to provide less inverse fill. The combinatorial optimization problem (3.1) has close links to that of finding orderings which minimize the height of the elimination tree, which is important in the context of parallel sparse elimination. Previous work in this field was targeted at structurally symmetric problems and attempted to restructure the elimination tree so as to reduce its height as much as possible; see [23], [26], [30], and [31]. The techniques proposed in these papers provide as a side effect a decrease in the overlap between closures of the graph vertices. Note that the problem of finding orderings which result in elimination trees of minimum height for general graphs is NP-hard; see [35].

To illustrate the effect of ordering on the inverse fill-in, consider a matrix (symmetric, for simplicity) before and after the reverse Cuthill–McKee (RCM) reordering. This reordering naturally tends to make the sums (3.1) rather large. Therefore a large amount of inverse fill is to be expected since this is given by the sum of the overlaps of the closures of the vertices. Figure 3.1 shows the patterns of the matrix and the inverse of its factor L. Figure 3.2 shows the patterns of the matrix and the inverse of L after RCM reordering. The inverse factor is much more dense in the second case.

Hence, we can expect that reorderings aimed at reducing the envelope or the band will tend to make the inverses of the triangular factors rather dense. Therefore, the RCM-like orderings do not seem to be advantageous as reordering options before the factorized approximate inverse construction. Minimum degree and nested dissection orderings are in principle more acceptable as reorderings for computing factorized matrix inverses. They provide more bushy elimination trees (or dags) resulting, typically, in less fill-in in the inverse factors. These orderings, particularly nested dissection, can also be used to introduce parallelism in the computation of the AINV preconditioner.

One natural choice of an ordering which keeps the elimination tree reasonably short is nested dissection. It is known that for every graph there exists a nested dissection ordering with minimal separators which produces an elimination tree of

$$
\begin{bmatrix}
* & * & & & * & & & & & & * \\
* & * & & & f & & & & & & f \\
& & * & * & * & & & & & & \\
& & * & * & f & & & & & & \\
* & f & * & f & * & & & & & & f \\
& & & & & * & * & & & * & \\
& & & & & * & * & & & f & \\
& & & & & & & * & * & * & \\
& & & & & & & * & * & f & \\
& & & & & * & f & * & f & * & * \\
* & f & & & f & & & & & * & *
\end{bmatrix}
\quad
\begin{bmatrix}
* & & & & & & & & & & \\
* & * & & & & & & & & & \\
& & * & & & & & & & & \\
& & * & * & & & & & & & \\
* & * & * & * & * & & & & & & \\
& & & & & * & & & & & \\
& & & & & * & * & & & & \\
& & & & & & & * & & & \\
& & & & & & & * & * & & \\
& & & & & * & * & * & * & * & \\
* & * & * & * & * & * & * & * & * & * & *
\end{bmatrix}
$$

FIG. 3.1. *Patterns of the matrix A (left) and of L^{-1}, the inverse of its lower triangular factor (right). Stars denote matrix nonzeros; f is used to denote filled positions in L.*

$$
\begin{bmatrix}
* & & * & & & & & & & & \\
& * & & * & & & & & & & \\
* & & * & & * & & & & & & \\
& * & & * & * & & & & & & \\
& & * & * & * & * & & & & & \\
& & & & * & * & & * & & & \\
& & & & & & * & * & & & \\
& & & & & * & * & * & * & & \\
& & & & & & & * & * & * & \\
& & & & & & & & * & * & * \\
& & & & & & & & & * & *
\end{bmatrix}
\quad
\begin{bmatrix}
* & & & & & & & & & & \\
& * & & & & & & & & & \\
* & & * & & & & & & & & \\
& * & & * & & & & & & & \\
* & * & * & * & * & & & & & & \\
* & * & * & * & * & * & & & & & \\
& & & & & & * & & & & \\
* & * & * & * & * & * & * & * & & & \\
* & * & * & * & * & * & * & * & * & & \\
* & * & * & * & * & * & * & * & * & * & \\
* & * & * & * & * & * & * & * & * & * & *
\end{bmatrix}
$$

FIG. 3.2. *Patterns of the matrix A (left) and of L^{-1}, the inverse of its lower triangular factor (right) after the RCM reordering.*

minimum height; see [31]. However, finding a nested dissection ordering with minimal separators that produces an elimination tree of minimum height is also NP-hard. Next, we state a simple result for the five-point formula discretization of a second-order elliptic partial differential equation with Dirichlet boundary conditions on a two-dimensional regular $k \times k$ grid. Note that the dimension of A is k^2.

THEOREM 3.1. *Consider the matrix A from a $k \times k$ regular grid problem with the nested dissection ordering. The number of nonzeros in the inverse factor L^{-1} is $\mathcal{O}(k^3)$. The number of nonzeros in the inverse factor with the (reverse) Cuthill–McKee ordering is $\mathcal{O}(k^4)$.*

Proof. Consider first nested dissection of a naturally ordered grid. Without loss of generality we may assume that $k = 2^l - 1$ for some $l \geq 2$. Hence, the number of levels in the separator tree is $2(l-1)+1$. We will count nonzeros in columns of L^{-1} by levels in this tree. Columns corresponding to vertices of the first separator in the first level contribute $k^2/2 + \mathcal{O}(k)$ nonzeros. Nodes in the next two separators of the second level contribute together $2(k(k/2) + k^2/(2 \cdot 2^2)) + \mathcal{O}(k)$ nonzeros. There are 2^{s-1} separators at the level s. In general, vertices in the separators at the sth level contribute $2^{s-1}(a_s + b_s) + \mathcal{O}(k)$, where

$$a_s = \left(k + \frac{k}{2} + \frac{k}{2} + \frac{k}{4} + \frac{k}{4} + \cdots + \frac{k}{2^{\frac{s}{2}-2}} + \frac{k}{2^{\frac{s}{2}-2}} + \frac{k}{2^{\frac{s}{2}-1}} + \frac{k}{2^{\frac{s}{2}-1}}\right) \frac{k}{2^{\frac{s}{2}}}$$

Table 3.1
Inverse fill for regular grid problem, $k = 100$, various orderings.

	Lexicographic	RCM	Min. deg.	Nested diss.	Red-black
$\|L^{-1}\|$	50,005,000	50,005,000	3,190,637	2,737,694	25,742,649

and $b_s = k^2/2^{s+1}$ for s even;

$$a_s = \left(k + \frac{k}{2} + \frac{k}{2} + \frac{k}{4} + \frac{k}{4} + \cdots + \frac{k}{2^{\frac{s+1}{2}-2}} + \frac{k}{2^{\frac{s+1}{2}-2}} + \frac{k}{2^{\frac{s+1}{2}-1}} \right) \frac{k}{2^{\frac{s+1}{2}-1}}$$

and $b_s = k^2/2^s$ for s odd. Note that a_s and b_s count nonzeros in rows corresponding to previous levels of separators up to the $(s-1)$th level and nonzeros in rows of the separator vertices from the sth level, respectively, in the submatrix of L^{-1} determined by vertices from the separators considered so far. It is easy to see that summing a_s and b_s separately over $\mathcal{O}(\log_2 k)$ levels we get $\mathcal{O}(k^3)$ behavior for both sums.

(Reverse) Cuthill–McKee ordering of the grid results in a straight elimination tree of depth equal to k^2. This implies $\mathcal{O}(k^4)$ number of nonzeros in L^{-1}. □

More specifically, the inverse fill for the (reverse) Cuthill–McKee ordering is $\frac{n(n+1)}{2}$, where $n = k^2$; it is straightforward to see that the same holds for the lexicographical ordering. For the red-black ordering, the inverse fill is approximately $\frac{n(n+1)}{4}$.

As an illustration, we report in Table 3.1 the number of nonzeros in L^{-1} for a few orderings and $k = 100$.

4. Decay rates. In the previous section we showed that certain orderings result in inverse triangular factors which preserve a good deal of sparsity. These orderings are attractive from the point of view of computing a factorized sparse approximate inverse because intuitively they can be expected to result in significant savings in the inverse factorization process, both in time and space. Furthermore, one might expect that it should be easier to find a good sparse approximation to a sparse inverse factor than to a dense one. However, a weakness of these arguments is that the results in the previous section are purely structural and give little insight into the quality of the approximate inverse factors, which depends on the numerical values of the entries. In other words, the structural results cannot tell the whole story since they disregard the magnitude of the nonzeros and in particular their decay behavior, which is crucial in practice since, in the most effective algorithms, sparsity in the inverse factors is preserved by applying a drop strategy based on value rather than on position.

The decay behavior of the entries of the inverse of a sparse matrix has been investigated by several authors; see, for instance, [11], [15], [32], [34]. Most results concern the case of symmetric banded matrices, but extensions to more general cases also exist. Given an even integer m, a matrix $A = (a_{ij})$ is called m-banded if $a_{ij} = 0$ for $|i-j| > m/2$. A well-known result by Demko, Moss, and Smith [11] gives an exponentially decaying upper bound for the entries of the inverse of a symmetric positive definite (SPD) m-banded matrix A. Let a and b denote the smallest and largest eigenvalue of A, respectively. Let $\kappa = \frac{b}{a}$ denote the spectral condition number of A. Set

$$q = q(\kappa) := \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}$$

and $\lambda := q^{2/m}$. Also, set $K_0 := (1+\sqrt{\kappa})^2/(2a\kappa)$. If $B = (b_{ij})$ denotes the inverse of A, then Demko, Moss, and Smith showed that

$$|b_{ij}| \le K\lambda^{|i-j|}, \tag{4.1}$$

where $K := \max\{a^{-1}, K_0\}$. For extensions and refinements of this result, see [15], [34].

Hence, the entries of A^{-1} are bounded in an exponentially decaying manner away from the main diagonal, along each row and column. The rate of decay is governed by the extreme eigenvalues a and b of A, and by the bandwidth m. Decay can be expected to be fast for matrices which are well conditioned and have a small bandwidth; otherwise it can be very slow. While the constant K is independent of m, symmetric permutations aimed at reducing the bandwidth can be used to reduce the value of λ. Hence, it may be possible to increase the rate of decay away from the main diagonal in the inverse by first applying a band-reducing permutation, such as (reverse) Cuthill–McKee. Since symmetric permutations do not affect the eigenvalues, the constant K is unchanged. Also notice that because $(P^TAP)^{-1} = P^TA^{-1}P$ for any permutation matrix P, applying a band-reducing permutation merely amounts to a redistribution of the entries of the inverse. This redistribution tends to move the larger off-diagonal entries of A^{-1} near the main diagonal and the smaller ones away from it. Of course, because the result by Demko, Moss, and Smith merely provides an upper bound (and a rather loose one in many cases), this need not always be the case. However, if we want to compute a banded approximation to A^{-1}, then it is clear that an ordering like RCM should be used, whereas orderings which result in a large bandwidth, like minimum degree or nested dissection, should be avoided. In contrast, if an adaptive strategy is used to compute a sparse approximate inverse in nonfactorized form, then the ordering is largely immaterial; see the numerical experiments in the following section.

But what about *factorized* approximate inverses? The decay behavior of the entries in the inverse triangular factors of a sparse matrix does not seem to have been investigated before. Yet, it is easy to see that the entries in the inverse Cholesky factor of a banded SPD matrix also obey an exponentially decaying bound away from the main diagonal. This simple result is formalized in the following theorem. Here $A = LL^T$ is the Cholesky factorization of A (with L lower triangular) and we let $Z = (z_{ij}) = L^{-T}$. Note that this matrix is equal to $ZD^{-1/2}$, where now Z and D are the output of the A-conjugation process on which the AINV algorithm is based. Also, the constants K and λ are defined as in the statement of the result (4.1) by Demko, Moss, and Smith.

THEOREM 4.1. *Let A be SPD and m-banded. Suppose that* $\max_{1\le i\le n}\{a_{ii}\} = 1$. *Then for all i, j with $j > i$, the entries z_{ij} in $Z = L^{-T}$ satisfy the following upper bound:*

$$|z_{ij}| \le K_1\lambda^{j-i}, \tag{4.2}$$

where $K_1 = K\frac{1-\lambda^m}{1-\lambda}$.

Proof. Let $A^{-1} = B = (b_{ij})$ and $L = (l_{ij})$. Notice that $l_{ij} = 0$ for $i < j$ and for $i - j > m$. From $Z = L^{-T} = A^{-1}L$ we find that $z_{ij} = \sum_{k=j}^{j+m-1} b_{ik}l_{kj}$ for $i \le j$. Therefore

$$|z_{ij}| \le \sum_{k=j}^{j+m-1} |b_{ik}||l_{kj}| \le K \sum_{k=j}^{j+m-1} \lambda^{k-i}|l_{kj}|,$$

where the second inequality is a consequence of the upper bound (4.1) on the entries of $B = A^{-1}$. Now we observe that the assumption $\max_{1\le i\le n}\{a_{ii}\} = 1$ implies $|l_{ij}| \le 1$ for all i, j, since $|l_{ij}| \le \sqrt{a_{ii}}$ (see [21, p. 147]). Hence,

$$|z_{ij}| \le K(\lambda^{j-i} + \lambda^{j-i+1} + \cdots + \lambda^{j-i+m-1}) = K\lambda^{j-i}(1 + \lambda + \lambda^2 + \cdots + \lambda^{m-1})$$

from which the result immediately follows. □

Hence, the entries in Z are bounded in an exponentially decaying manner away from the main diagonal along rows. This result can be generalized to some extent to nonsymmetric problems using results from [11], [15], and [34]. For special classes of matrices something more precise can be said for the decay in the inverse factors: for instance, in the M-matrix case it is easily seen that the entries of Z decay faster than the corresponding entries of A^{-1}. This cannot be inferred from the bound (4.2) since $0 < \lambda < 1$ implies $\frac{1-\lambda^m}{1-\lambda} > 1$ and therefore $K_1 > K$. We recall that an M-matrix $A = (a_{ij})$ is a matrix such that $a_{ij} \le 0$ for $i \ne j$ and $B = A^{-1} \ge 0$, i.e., $b_{ij} \ge 0$ for all i, j. Furthermore, $b_{ij} > 0$ for all i, j if A is irreducible. Now, in the identity

$$z_{ij} = b_{ij}l_{jj} + \sum_{k=j+1}^{j+m-1} b_{ik}l_{kj}$$

the second summand on the right-hand side is nonpositive (negative if A is irreducible) because the off-diagonal entries of L are also. It follows that $z_{ij} \le b_{ij}l_{jj}$, and if A is normalized so that $\max_{1\le i\le n}\{a_{ii}\} = 1$, then $z_{ij} \le b_{ij}$, the inequality being strict provided that A is irreducible. Notice that this property does not require A to be banded (i.e., m can be arbitrary). If A is banded, then we can conclude that the entries in the inverse Cholesky factor satisfy an exponentially decaying upper bound that is at least as small as the one for the entries of A^{-1}, and the actual decay rate is at least as fast as for A^{-1} (faster if A is irreducible). We mention that refined bounds for the entries of the inverses of M-matrices can be found in [15] and [34].

This result may provide (for the class of M-matrices) a justification of the observed fact [5], [6] that factorized approximate inverses often provide better approximations than nonfactorized forms: because the entries in the inverse factors decay more rapidly than the entries in A^{-1}, it is easier to approximate L^{-1} with a sparse matrix than to approximate A^{-1}.

Concerning the normalization condition $\max_{1\le i\le n}\{a_{ii}\} = 1$ used to derive our results, it is easy to see by examples that it is essential. In order to enforce such a condition, it is clearly sufficient to divide A by its largest diagonal entry. While the spectral condition number and the bandwidth (and therefore λ) are unaffected by this normalization, the constant K is altered. Indeed, K will be increased if the largest diagonal entry of A (prior to the normalization) is greater than 1. However, the qualitative behavior is the same.

Some insight on the effect of reorderings on the decay behavior of the entries of the inverse factors is given by the following argument, which is adapted from an observation by Meurant concerning the effect of ordering on sparse incomplete Cholesky factorizations [33]. Let Z be the inverse Cholesky factor of A, and let $\hat{Z}$ be the inverse Cholesky factor of $\hat{A} := P^TAP$, where P is a permutation matrix. Denoting by $\|\cdot\|_F$ the Frobenius matrix norm, we have

$$\|\hat{Z}\|_F = \sqrt{\text{trace}(\hat{Z}\hat{Z}^T)} = \sqrt{\text{trace}(\hat{A}^{-1})} = \sqrt{\text{trace}(A^{-1})} = \|Z\|_F.$$

This means that an ordering which preserves sparsity in the inverse factors, like nested dissection, will also result in nonzero entries which are larger, on the average, than those corresponding to an ordering which results in a high amount of fill, like reverse Cuthill–McKee. With such an ordering, the use of a drop tolerance in an incomplete inverse factorization scheme (such as AINV) could in principle be problematic. Too small a drop tolerance could result in unacceptably high fill; limiting the number of nonzeros accepted in each column of the approximate inverse factor (in an ILUT-like fashion) could lead to the dropping of many large entries, resulting in a very poor approximation to the inverse. Increasing the drop tolerance may again lead to the dropping of too many large entries. In contrast, the quality of the preconditioner can be more easily tuned if the entries in the inverse factors decay smoothly away from the main diagonal, and a dual threshold approach becomes viable. Notice that this situation is not specific to factorized approximate inverse preconditioners: exactly the same argument applies to standard ILU-type preconditioners. In particular, this suggests a possible explanation of the generally poor performance of minimum degree for ILU(0) and ILUT preconditioning (see [3], [14]).

Hence, we face the following dilemma: graph-theoretical considerations suggest the use of orderings that will cause a small amount of inverse fill, like nested dissection or minimum degree, whereas a look at the decay rates suggests that we use band-reducing orderings (like RCM), which cause large amounts of inverse fill but hopefully result in faster decay. As we shall see in the next section, it turns out that RCM is generally not a good ordering for factorized approximate inverses, whereas nested dissection and minimum degree perform quite well, provided that the number of nonzeros in each column (or row) of the inverse factors is not subject to any a priori upper bound. It appears that for many problems, many of the nonzero entries in the inverse factors corresponding to inverse fill-reducing orderings remain small in absolute value, although they must be larger, on the average, than those corresponding to orderings leading to dense inverse factors.

We illustrate these points with some simple numerical examples computed using MATLAB. Consider the tridiagonal (2-banded) irreducible M-matrix

$$A = \begin{bmatrix} 1 & -1/4 & & & \\ -1/4 & 1 & -1/4 & & \\ & -1/4 & 1 & -1/4 & \\ & & -1/4 & 1 & -1/4 \\ & & & -1/4 & 1 \end{bmatrix}.$$

The upper triangular part of A^{-1} (rounded to four places) is

$$\begin{bmatrix} 1.0718 & 0.2872 & 0.0769 & 0.0205 & 0.0051 \\ & 1.1487 & 0.3077 & 0.0821 & 0.0205 \\ & & 1.1538 & 0.3077 & 0.0769 \\ & & & 1.1487 & 0.2872 \\ & & & & 1.0718 \end{bmatrix}$$

and the inverse factor is

$$Z = L^{-T} = \begin{bmatrix} 1.0000 & 0.2582 & 0.0690 & 0.0185 & 0.0050 \\ & 1.0328 & 0.2760 & 0.0739 & 0.0198 \\ & & 1.0351 & 0.2773 & 0.0743 \\ & & & 1.0353 & 0.2774 \\ & & & & 1.0353 \end{bmatrix},$$

showing that the entries of Z are smaller than the corresponding entries of A^{-1}. The rate of decay in A^{-1} is governed by the following quantities: $K = K_0 = 2.3401$ and $\lambda = 0.2277$. The decay bounds $K\lambda^p$, $0 \le p \le 4$, are

$$[2.3401,\ 0.5329,\ 0.1214,\ 0.0276,\ 0.0063].$$

In this case the decay rate predicted by (4.1) is rather pessimistic, although the estimate becomes more accurate far from the main diagonal.

If a two-domain decomposition is applied to A, the permuted matrix is

$$\hat{A} = \begin{bmatrix} 1 & -1/4 & 0 & 0 & 0 \\ -1/4 & 1 & 0 & 0 & -1/4 \\ 0 & 0 & 1 & -1/4 & -1/4 \\ 0 & 0 & -1/4 & 1 & 0 \\ 0 & -1/4 & -1/4 & 0 & 1 \end{bmatrix}.$$

The corresponding inverse factor is

$$\hat{Z} = \begin{bmatrix} 1.0000 & 0.2582 & 0 & 0 & 0.0716 \\ & 1.0328 & 0 & 0 & 0.2864 \\ & & 1.0000 & 0.2582 & 0.2864 \\ & & & 1.0328 & 0.0716 \\ & & & & 1.0742 \end{bmatrix}.$$

There are 15 nonzeros in Z and 11 in $\hat{Z}$. Note that $\|Z\|_F^2 = \|\hat{Z}\|_F^2 = 5.5949 = \mathrm{trace}(A^{-1}) = \mathrm{trace}(\hat{A}^{-1})$. The elements in $\hat{Z}$ are larger, on the average, than those in Z. Note, however, that the growth is confined to entries in the last column. Clearly, the Frobenius norm does not provide any information about the distribution of the weight among the nonzeros in $\hat{Z}$. Hence, the trace argument outlined above gives only limited insight.

In the previous example A was strictly diagonally dominant and the actual inverse decay was fairly rapid. The next example shows that if A is not strictly diagonally dominant, the constant K could be so large and the constant λ be so close to 1 that there are actually no "small" entries in A^{-1}. Let

$$A = \begin{bmatrix} 1/2 & -1/2 & & & \\ -1/2 & 1 & -1/2 & & \\ & -1/2 & 1 & -1/2 & \\ & & -1/2 & 1 & -1/2 \\ & & & -1/2 & 1 \end{bmatrix}.$$

Then the upper triangular part of the inverse is

$$\begin{bmatrix} 10 & 8 & 6 & 4 & 2 \\ & 8 & 6 & 4 & 2 \\ & & 6 & 4 & 2 \\ & & & 4 & 2 \\ & & & & 2 \end{bmatrix}.$$

The decay bounds from (4.1) are

$$[24.6914,\ 18.3136,\ 13.5832,\ 10.0747,\ 7.4724],$$

corresponding to $K = 24.6914$ and $\lambda = 0.7417$. (See also [8] for similar examples.) In addition, note that the inverse factor $Z = (z_{ij})$ is given by $z_{ij} = \sqrt{2}$ for all $j \geq i$. Although the entries of Z satisfy an exponentially decaying bound, there is no actual decay. For problems of this nature, orderings which preserve sparsity in the inverse factors are better suited than orderings that reduce the bandwidth, although sparse approximate inverse preconditioners may still be ineffective. A possible solution, proposed in [8], is to use wavelet compression techniques in combination with a sparse approximate inverse approach.

5. Numerical experiments. In this section we present the results of numerical tests performed on a variety of matrices, mostly arising from the discretization of partial differential equations. We consider both symmetric and nonsymmetric problems. First we consider the following partial differential equation in $\Omega = (0,1) \times (0,1)$

$$-\varepsilon \Delta u + \frac{\partial e^{xy} u}{\partial x} + \frac{\partial e^{-xy} u}{\partial y} = g \tag{5.1}$$

with Dirichlet boundary conditions. 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 to the discrete system is the vector $(1, 2, \ldots, n)$. The parameter $\varepsilon > 0$ controls the difficulty of the problem—the smaller ε is, the harder it is to solve the discrete problem by iterative methods (see also [3]). For our experiments, we generated 10 linear systems of increasing difficulty, corresponding to $\varepsilon^{-1} = 100, 200, \ldots, 1000$. The coefficient matrix A becomes increasingly nonsymmetric and far from diagonally dominant as ε gets smaller. Moreover, Green function arguments can be used to show that the rate of decay in the inverse of the coefficient matrix becomes slower with decreasing ε.

In Table 5.1 we give the number of Bi-CGSTAB [38] iterations required to reduce the initial residual by at least four orders of magnitude when the preconditioner AINV with drop tolerance $Tol = 0.2$ is used. The initial guess is always the zero vector. In parenthesis, we give the number of nonzeros in the approximate inverse (in thousands). The different orderings considered are the lexicographic, or natural, ordering (denoted no in the tables), Cuthill–McKee (cm), reverse Cuthill–McKee (rcm), multiple minimum degree (mmd) [28], nested dissection (nd), and red-black (rb). A † means that convergence was not attained within 500 iterations.

The best results are obtained with the minimum degree heuristic. Nested dissection is a close second. The natural and Cuthill–McKee-type orderings do poorly, particularly for small ε. The amount of inverse fill grows quickly and convergence is eventually lost. Hence, these orderings are not robust. The red-black ordering is much better, but not as good as minimum degree or nested dissection. It is interesting to compare these results with those obtained with ILU-type preconditioners [3], [14], for which the situation is the opposite of the present one.

It is also interesting to observe that the effect of ordering is completely different for another factorized sparse approximate inverse technique, the FSAI preconditioner [25]. In this method, the sparsity pattern of the incomplete inverse factors must be specified a priori. The simplest choice is to impose the sparsity pattern of the corresponding triangular part of A (or of $\hat{A} = P^T A P$ if a permutation is applied). With this choice, the number of nonzeros in the approximate inverse factors is always the same as the number of nonzeros in the original matrix. The results of our tests, reported in Table 5.2, are somewhat surprising: for this suite of matrices, red-black is

Table 5.1
Number of Bi-CGSTAB iterations and fill-in for AINV(0.2) preconditioning.

ε^{-1}	Ordering					
	no	cm	rcm	mmd	nd	rb
100	14 (8.6)	12 (9.8)	12 (10)	8 (7.8)	9 (7.7)	8 (9.2)
200	14 (11)	19 (14)	16 (14)	8 (9.6)	9 (9.7)	9 (11)
300	24 (14)	43 (18)	29 (18)	9 (12)	10 (12)	11 (14)
400	26 (17)	26 (22)	19 (23)	10 (14)	11 (14)	12 (17)
500	58 (19)	27 (27)	25 (28)	13 (16)	13 (16)	15 (19)
600	73 (22)	33 (31)	29 (32)	13 (17)	15 (17)	20 (21)
700	†(25)	62 (36)	63 (39)	15 (19)	21 (18)	25 (23)
800	†(52)	†(43)	†(47)	18 (20)	23 (20)	19 (26)
900	†(49)	†(51)	†(67)	22 (22)	22 (21)	24 (29)
1000	†(97)	†(61)	†(115)	21 (23)	30 (22)	25 (31)

Table 5.2
Number of Bi-CGSTAB iterations for different orderings, FSAI preconditioner.

ε^{-1}	Ordering					
	no	cm	rcm	mmd	nd	rb
100	†	†	†	36	53	28
200	†	†	†	118	220	28
300	†	†	†	†	†	30
400	†	†	†	†	†	33
500	†	†	†	†	†	34
600	†	†	†	†	†	38
700	†	†	†	†	†	39
800	†	†	†	†	†	42
900	†	†	†	†	†	41
1000	†	†	†	†	†	49

the only robust ordering. A similar behavior was observed on other, more complicated problems: red-black or, more generally, multicoloring seems to have a beneficial effect on the robustness and effectiveness of FSAI. This is especially true for problems which are far from being diagonally dominant. It should be mentioned, however, that good results were reported in [17] using nested dissection with FSAI on SPD matrices from elasticity problems.

In Tables 5.3 and 5.4 we present the results of a few experiments performed in order to check the commonly held view that nonfactorized approximate inverses are largely insensitive to ordering. To this end, we used the SPAI algorithm [22] and the MR algorithm [9] in order to compute sparse approximate inverses to be used as preconditioners for Bi-CGSTAB. We only give results for $\varepsilon^{-1} = 100, 500, 1000$. Due to the increasing difficulty of the problems, the SPAI and MR parameters had to be adjusted so as to allow increasing amounts of fill in the approximate inverse in order to have convergence in a reasonable number of iterations. However, for each value of ε, the same parameters were used for all orderings.

It appears from these results that the amount of fill in the approximate inverse is unaffected by the ordering. The number of iterations, on the other hand, can fluctuate, but not too much. Reorderings cannot do much harm, but they cannot improve performance either. While this is a sign of robustness, it also means that reorderings cannot be used to help solving problems for which SPAI or MR perform poorly. Notice that these two preconditioners, while roughly equivalent to one another, are not as effective as AINV/minimum degree or FSAI/red-black on this set of problems.

The set of matrices used for these experiments is useful because the difficulty of

Table 5.3
Number of Bi-CGSTAB iterations and fill-in for different orderings, SPAI preconditioner.

ε^{-1}	Ordering					
	no	cm	rcm	mmd	nd	rb
100	13 (18)	14 (18)	11 (18)	13 (18)	13 (18)	12 (18)
500	40 (45)	44 (45)	37 (45)	36 (45)	35 (45)	40 (45)
1000	62 (80)	77 (80)	77 (80)	63 (80)	65 (80)	76 (80)

Table 5.4
Number of Bi-CGSTAB iterations and fill-in for different orderings, MR preconditioner.

ε^{-1}	Ordering					
	no	cm	rcm	mmd	nd	rb
100	18 (20)	17 (20)	17 (20)	16 (20)	16 (20)	18 (20)
500	49 (40)	57 (40)	43 (40)	57 (40)	57 (40)	59 (40)
1000	63 (56)	76 (56)	61 (56)	68 (56)	64 (56)	77 (56)

the problem can be easily adjusted by varying ε. On the other hand, it is a somewhat contrived type of problem. Indeed, it is well known that using second-order, centered difference approximations for both the second and first partial derivatives in (5.1) can result in an unstable discretization. Alternative discretizations, such as those which use upwinding for the first-order terms, do not suffer from this problem and give rise to matrices with very nice properties from the point of view of iterative solutions, such as diagonal dominance. However, such approximations are only first-order accurate and in many cases are unable to resolve fine features of the solution, such as boundary layers. A possible solution is to use centered differences, but with a local mesh refinement over regions where the solution is expected to exhibit strong variations. To illustrate this, we take the following example from Elman [16]. Consider the following partial differential equation in $\Omega = (0,1) \times (0,1)$

$$-\Delta u - 2P\frac{\partial u}{\partial x} + 2P\frac{\partial u}{\partial y} = g, \tag{5.2}$$

where $P > 0$, and the right-hand side g and the boundary conditions are determined by the solution

$$u(x,y) = \frac{e^{2P(1-x)} - 1}{e^{2P} - 1} + \frac{e^{2Py} - 1}{e^{2P} - 1}.$$

This function is nearly identically zero in Ω except for boundary layers of width $\mathcal{O}(\delta)$ near $x = 0$ and $y = 1$, where $\delta = 1/2P$. A uniform coarse grid was used in the region where the solution is smooth, and a uniform fine grid was superimposed to the regions containing the boundary layers, so as to produce a stable and accurate approximation; see [16] for details.

We performed experiments with $P = 500$ and $P = 1000$; see Table 5.5. These values are considerably larger than those used in [16]. The resulting matrices are of order 5041 and 7921, with 24921 and 39249 nonzeros, respectively. The convergence criterion used was a reduction of the initial residual norm by at least six orders of magnitude. For the preconditioner, we used the AINV preconditioner with drop tolerance $Tol = 0.1$.

It appears from these experiments that the minimum degree ordering is the most robust among those considered here, as well as the most effective. These are rather challenging problems; see [3] for the performance of ILU preconditioners and different

Table 5.5
Bi-CGSTAB iterations and fill-in for AINV (0.1) preconditioning, Elman's problems.

P	Ordering					
	no	cm	rcm	mmd	nd	rb
500	†(286)	†(346)	†(395)	156 (85)	158 (80)	250 (199)
1000	†(630)	†(2162)	†(2149)	220 (147)	†(143)	†(482)

Table 5.6
Test problems information.

Matrix	n	nnz	Application
1138BUS	1138	2596	Power system network
NASA2146	2146	37198	Structural analysis
BCSSTK21	3600	15100	Structural analysis
FDM2	32010	95738	Computational chemistry
FALC2	4663	57673	Oil reservoir simulation
FALC3	2331	14415	Oil reservoir simulation
MEMPLUS	17758	99147	Digital circuit analysis
3DCD	8000	53600	3D convection-diffusion
UTM1700B	1700	21509	Plasma physics
UTM3060	3060	42211	Plasma physics
ORSIRR1	1030	6858	Oil reservoir simulation
ADD20	2395	17319	Digital circuit analysis

orderings on these two matrices. As with ILU, the ordering of grid points becomes increasingly important as convection becomes stronger. However, whereas for ILU preconditioning (with fill) the RCM-type orderings were found to be highly robust and effective [3], such reorderings are unsuitable for AINV. With AINV, the best results are obtained with minimum degree, which is inferior to RCM when used with ILU when fill is allowed.

Some experiments were performed with a dual threshold variant of AINV, where the maximum number of nonzeros in each row (or column) of the inverse factors is restricted. Even with minimum degree and nested dissection, the results were poor, due to the fact that with these orderings most of the fill in the inverse factors occurs in the last rows (columns). Also, many of these fill-ins are rather large in magnitude (see the examples in section 4), and severely constraining the amount of fill in these rows (columns) results in highly inaccurate approximations to the inverse factors.

While convection-diffusion problems are important, they are a rather narrow class of problems. In the following we report on results obtained with 12 matrices arising in a variety of applications, also including timings. All these problems are part of Davis's collection [10] except for FDM2, FALC2, FALC3, and 3DCD. Matrix FDM2 was provided by Ullrich and is a finite difference discretization of a Kohn–Sham equation in two dimensions. The FALC∗ problems were extracted from FALCON [37], a parallel oil reservoir simulation code developed by Joubert at Los Alamos National Laboratory in collaboration with the Amoco Production Company and Cray Research Inc. Matrix 3DCD is a seven-point finite difference discretization of a diffusion-dominated convection-diffusion equation on the unit cube with Dirichlet boundary conditions. Some information concerning these problems is provided in Table 5.6 above. Here n is the problem size and nnz the number of nonzeros in the matrix. The first four problems are symmetric positive definite, and nnz corresponds to the number of nonzeros in the upper triangular part of the matrix.

In Tables 5.7 and 5.8 we report the results of numerical tests performed with these

Table 5.7
Test results, symmetric positive definite problems.

Matrix	original	mmd	gnd	rcm	mcl
1138BUS	4464/0.063 82/0.107	6422/0.066 57/0.088	4520/0.061 69/0.091	8426/0.074 46/0.085	4092/0.066 79/0.099
NASA2146	31956/0.454 335/5.358	23264/0.660 249/3.446	24825/0.510 323/4.511	35177/0.561 430/6.916	306051/16.45 – / –
BCSSTK21	113104/0.915 –/–	15789/0.291 220/1.656	24229/0.305 2072/18.09	31691/0.315 –/–	14669/0.260 197/1.470
FDM2	169127/1.716 257/32.70	132791/2.216 217/24.05	130839/1.967 210/23.75	166156/1.871 237/29.74	145779/1.826 196/22.11

Table 5.8
Test results, nonsymmetric problems.

Matrix	original	mmd	gnd	rcm	mcl
FALC2	107766/1.941 23/1.318	66885/1.449 25/1.122	93238/1.512 25/1.326	217415/2.036 18/1.596	72639/1.179 33/1.497
FALC3	99384/0.743 29/0.900	58421/0.554 21/0.360	57281/0.547 26/0.442	136928/0.752 20/0.906	66717/0.583 28/0.575
MEMPLUS	59547/2.389 189/17.75	44288/2.145 17/1.923	43914/2.222 264/22.89	55289/2.184 17/1.838	57298/2.371 27/2.770
3DCD	102043/1.128 18/1.204	87288/1.554 13/0.844	88869/1.448 13/0.868	115477/1.255 18/1.275	82946/1.297 13/0.820
UTM1700B	170899/2.511 903/46.66	65930/0.759 232/4.502	65631/0.837 641/11.74	145197/1.258 324/14.46	93301/2.572 659/18.80
UTM3060	403616/5.014 976/117.7	118609/1.683 115/5.526	119956/1.751 124/5.999	378420/3.974 377/42.16	190163/6.342 234/15.11
ORSIRR1	5351/0.124 32/0.078	4819/0.124 33/0.081	4764/0.123 35/0.084	5519/0.127 29/0.072	5049/0.121 34/0.084
ADD20	7525/0.327 7/0.079	5912/0.269 7/0.079	5424/0.284 7/0.076	6892/0.260 7/0.079	6366/0.298 7/0.079

12 matrices and different orderings, performed on one processor of an SGI Origin 2000 with R10000 processors. The results in Table 5.7 are relative to the SPD problems, for which conjugate gradient acceleration was used; in Table 5.8 we give the results for the nonsymmetric problems, where Bi-CGSTAB was the accelerator of choice. In all cases the stopping criterion was a reduction of the initial residual norm by at least eight orders of magnitude. The drop tolerance in AINV was set to 0.1 for all problems except for BCSSTK21, where we used 0.2, and for the FALC$*$ matrices where a drop tolerance equal to 0.01 was used. A "–" means no convergence. In the tables, "gnd" denotes a generalized nested dissection ordering [27] and "mcl" a greedy multicoloring heuristic [36]. For each matrix we provide the number of nonzeros in the approximate inverse factors with the time to compute the preconditioner (above) and the number of iterations with the corresponding timing (below). Timings were measured by the *dtime* function. The codes were compiled by f77 with the -O3 option.

These results indicate that minimum degree is generally the best ordering, resulting in most cases in small inverse fill and good convergence rates. Generalized nested dissection is also good in general. The performance of RCM is not as poor as in the previous set of experiments, although this ordering often results in high amounts of inverse fill. Multicoloring is often better than the original ordering, and sometimes it outperforms all the other orderings, but on the average is not as good as minimum degree or nested dissection. As for the time to compute the preconditioner, it appears that fill-reducing orderings are not always effective at reducing timings. This could be

TABLE 5.9
Results for FDM2, graph partitioning.

p	PCG Its	$\lvert\bar{Z}\rvert$
2	276	168,786
4	276	168,634
8	274	167,850
16	275	164,387
32	274	162,999
64	274	157,032
128	276	149,913
256	276	144,085
512	264	138,954
1024	244	136,960

due to cache effects, considering that minimum degree and nested dissection are global orderings that do not preserve data locality, which is often present with the original ordering or with RCM. It should also be mentioned that a good correlation between inverse fill and timings for the construction of AINV was observed in [7], possibly due to the somewhat different implementation of AINV adopted in [7]. In any event, a reduction in the amount of fill in the approximate inverse factors is important, particularly for large problems. It is also worth mentioning that for symmetric problems preordered with minimum degree, a small but worthwhile reduction in the time for the set-up phase was achieved by applying equivalent postorderings to the computed inverse factors, using techniques described in [23], [29], [30]. In the interest of brevity, we do not report these results here.

Finally, we performed some experiments using the METIS graph partitioning package [24]. Graph partitioning affords a natural way to parallelize sparse matrix computations, and should be a valuable tool for parallelizing the AINV preconditioner, both in the set-up phase and in the iterative application of the approximate inverse. However, partitioning induces a reordering and we are interested in seeing the effect on the convergence rate. Because of the similarity with nested dissection, we expect the performance to be satisfactory.

In Table 5.9 we show the results obtained for matrix FDM2 with the standard PMETIS executable code with default partitioning parameters. Here p denotes the number of subdomains, or graph partitions. In a parallel environment, p would be equal to the number of processors. It is interesting to see that the performance improves with the number of subdomains, both from the point of view of fill in the approximate inverse factor $\bar{Z}$ and from the point of view of convergence rate, at least up until $p = 1024$. These results suggest that efficient parallel implementations of AINV are possible without sacrificing the quality of the preconditioner. For preliminary results obtained with a parallel implementation of AINV based on graph partitioning, see [1].

6. Conclusions. We have presented theoretical results and numerical experiments aimed at assessing the effect of different sparse matrix orderings on the performance of the factorized approximate inverse preconditioner AINV. These experiments appear to confirm the intuition that orderings which produce relatively sparse inverse factors, such as minimum degree and nested dissection, tend to perform better than orderings (like RCM) that result in matrices with a narrow profile but dense inverse factors. This is especially true for difficult problems characterized by slow decay in the inverse. This marks a significant difference between the behavior of factorized sparse approximate inverse preconditioners and that of ILU-type techniques. A good ordering for AINV results not only in reduced storage needs for computing the approximate inverse factorization, but also in better convergence rates for the preconditioned iteration. In several cases the preconditioned iteration failed with the original ordering, but succeeded with an appropriate reordering, usually minimum degree. This is probably due to the fact that with a good ordering, less entries in the inverse factors are discarded, resulting in a more accurate approximation to the exact inverse.

An interesting problem, not considered here, is to look for reorderings which take into account the magnitude of the matrix entries and not just the sparsity structure. Some weighted graph heuristics have been proposed in [7], and the results for anisotropic problems reported in [7] are encouraging. However, more work is needed in this direction.

Preliminary results using permutations induced by graph partitioning suggest that parallelism can be achieved in the construction and application of the AINV preconditioner without compromising its effectiveness at reducing the number of iterations. Tests with a fully parallel implementation of AINV based on graph partitioning, reported in [1], have confirmed this observation.

Acknowledgments. During the preparation of this paper we have benefited from insightful discussions with Robert Bridson, Tim Davis, John Gilbert, and Gérard Meurant. We are also thankful to one of the referees for valuable comments and suggestions.

REFERENCES

[1] M. Benzi, J. Marín, and M. Tůma, *Parallel preconditioning with factorized sparse approximate inverses*, in Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing, B. Hendrickson et al., eds., CD-ROM, SIAM, Philadelphia, PA, 1999.

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

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

[4] M. Benzi and M. Tůma, *A sparse approximate inverse preconditioner for nonsymmetric linear systems,* SIAM J. Sci. Comput., 19 (1998), pp. 968–994.

[5] M. Benzi and M. Tůma, *Numerical experiments with two approximate inverse preconditioners,* BIT, 38 (1998), pp. 234–241.

[6] M. Benzi and M. Tůma, *A comparative study of sparse approximate inverse preconditioners*, Appl. Numer. Math., 30 (1999), pp. 305–340.

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

[8] T. Chan, W.-P. Tang, and W. Wan, *Wavelet sparse approximate inverse preconditioners*, BIT, 37 (1997), pp. 644-660.

[9] E. Chow and Y. Saad, *Approximate inverse preconditioners via sparse-sparse iterations*, SIAM J. Sci. Comput., 19 (1998), pp. 995–1023.

[10] T. Davis, *University of Florida Sparse Matrix Collection*, NA Digest, vol. 94, issue 42, October 1994; also available online from http://www.cise.ufl.edu/∼davis/sparse/.

[11] S. Demko, W. F. Moss, and P. W. Smith, *Decay rates for inverses of band matrices*, Math. Comp., 43 (1984), pp. 491–499.

[12] I. S. Duff, A. M. Erisman, C. W. Gear, and J. K. Reid, *Sparsity structure and Gaussian elimination*, SIGNUM Newsletter, 23 (1988), pp. 2–9.

[13] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, UK, 1986.

[14] I. S. Duff and G. A. Meurant, *The effect of ordering on preconditioned conjugate gradients*, BIT, 29 (1989), pp. 635–657.

[15] V. Eijkhout and B. Polman, *Decay rates of inverses of banded M-matrices that are near to Toeplitz matrices*, Linear Algebra Appl., 109 (1988), pp. 247–277.

[16] H. C. Elman, *Relaxed and stabilized incomplete factorizations for non-self-adjoint linear systems*, BIT, 29 (1989), pp. 890–915.
[17] M. R. Field, *Improving the Performance of Factorised Sparse Approximate Inverse Preconditioner*, Hitachi Dublin Laboratory Technical Report HDL-TR-98-199, Dublin, Ireland, 1998.
[18] A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1981.
[19] J. R. Gilbert, *Predicting structure in sparse matrix computations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 62–79.
[20] J. R. Gilbert and J. W. H. Liu, *Elimination structures for unsymmetric sparse LU factors*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 334–352.
[21] G. H. Golub and C. F. Van Loan, *Matrix Computations,* 3*rd* ed., The Johns Hopkins University Press, Baltimore and London, 1996.
[22] M. Grote and T. Huckle, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), pp. 838–853.
[23] H. Hafsteinsson, *Parallel Sparse Cholesky Factorization*, Ph.D. Thesis, Department of Computer Science, Cornell University, 1988.
[24] G. Karypis and V. Kumar, *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices (Version* 3.0*)*, University of Minnesota, Department of Computer Science and Army HPC Research Center, Minneapolis, MN, 1997.
[25] L. Yu. Kolotilina and A. Yu. Yeremin, *Factorized sparse approximate inverse preconditioning* I: *Theory*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 45–58.
[26] J. G. Lewis, B. W. Peyton, and A. Pothen, *A fast algorithm for reordering sparse matrices for parallel factorization*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1156–1173.
[27] R. J. Lipton, D. J. Rose, and R. E. Tarjan, *Generalized nested dissection*, SIAM J. Numer. Anal., 16 (1979), pp. 346–358.
[28] J. W. H. Liu, *Modification of the minimum degree algorithm by multiple elimination*, ACM Trans. Math. Software, 11 (1985), pp. 141–153.
[29] J. W. H. Liu, *Equivalent sparse matrix reordering by elimination tree rotations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 424–444.
[30] J. W. H. Liu, *Reordering sparse matrices for parallel elimination*, Parallel Comput., 11 (1989), pp. 73–91.
[31] F. Manne, *Reducing the Height of an Elimination Tree through Local Reorderings*, Technical Report CS-51-91, University of Bergen, Norway, 1991.
[32] G. Meurant, *A review of the inverse of symmetric tridiagonal and block tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 707–728.
[33] G. Meurant, *private communication*, December 1997.
[34] R. Nabben, *Decay rates of the inverse of nonsymmetric tridiagonal and band matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 820–837.
[35] A. Pothen, *The Complexity of Optimal Elimination Trees*, Technical Report CS-88-13, Pennsylvania State University, University Park, PA, 1988.
[36] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Co., Boston, 1996.
[37] G. S. Shiralkar, R. E. Stephenson, W. D. Joubert, and B. van Bloemen-Waanders, *FALCON: A production quality distributed memory reservoir simulator*, SPE Paper 37975, presented at the SPE Reservoir Simulation Symposium, Dallas, Texas, 8–11 June 1997.
[38] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems*, SIAM J. Sci. Statist. Comput., 12 (1992), pp. 631–644.