ORDERING FOR FACTORED APPROXIMATE INVERSE PRECONDITIONERS[*]

ROBERT BRIDSON[†] AND WEI-PAI TANG[†]

1. Summary. We highlight how ordering can play an important role in the performance of factored approximate inverse preconditioners, concentrating on the AINV scheme of Benzi and Tůma ([1],[2]). After discussing some theory and heuristics for structure-based ordering schemes, we demonstrate that several practical algorithms can dramatically speed up the calculation of the preconditioner. Furthermore, these orderings generally improve convergence of Krylov subspace methods, especially Minimum Inverse Priority, a new algorithm we propose here.

Recalling the problems anisotropy can pose for ILU (e.g. [6], [7], [8]) and noticing some discrepancies in testing data with highly anisotropic matrices, we are lead to consider the benefit of weighted orderings. These are algorithms that take into account the magnitude of entries in the original matrix as well as its structure. A simple demonstration problem shows the potential of weighted orderings, significantly improving both the speed of preconditioner calculation and the convergence to solutions. We finish by proposing a weighted version of Nested Dissection which shows promise for a robust high-performance ordering.

2. Context. Iterative methods for solving sparse linear systems rely on good preconditioners—the preconditioner not only must speed convergence, but also must be relatively inexpensive to calculate and apply. Sparse approximate inverse preconditioners are of particular interest today, since their application only requires (easily parallelized) matrix-vector multiplication. These methods can be divided into two classes: those that form an approximation to the inverse matrix (e.g. [4], [11], [14]) and those that approximate the inverses of the matrix's LU factors (e.g. [1], [2], [12]). This second class has the benefit of guaranteeing that that the preconditioner is nonsingular, and more importantly it seems that the factored form is more effective per nonzero[3]. However, the inverse factors are critically dependent on the ordering of the matrix—indeed, in general they will not even exist for some orderings.

In this paper we focus our attention on the AINV algorithm[2], which, via implicit Gaussian elimination with small-element dropping, constructs a factored approximate inverse:

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

where Z and W are unit upper triangular, and D^{-1} is diagonal. In [2] Benzi and Tůma began an investigation of ordering schemes for AINV, which we elaborate here. Notice that many of the results found here should apply to other factored approximate inverse schemes (e.g. see [12]).

3. Goals. Intuitively, for a more effective preconditioner we need a more accurate approximation to the true inverse factors. However, sparsity constraints do not allow

[*] The work was supported by the Natural Sciences and Engineering Council of Canada, the Information Technology Research Centre (which is funded by the Province of Ontario), and RIACS/NASA Ames NAS 2-96027.

[†]Department of Computer Science, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada. email: rebridso@yoho.uwaterloo.ca, wptang@yoho.uwaterloo.ca

us to simply retain more nonzeros, and thus there are two basic goals behind the ordering schemes:

1. reduce the number of nonzeros in the true inverse factors, so we will be dropping less of them, and
2. reduce the magnitude of the nonzeros in the true inverse factors, so what we drop will be of less importance.

We address the first goal in section 4, and the second in section 5. The theory in section 4 will also show a more dramatic effect of the first goal: if an ordering reduces the number of nonzeros in the true inverse factors, the calculation of the preconditioner can correspondingly be accelerated.

4. Structural Orderings.

We restrict our discussion to symmetric positive-definite matrices, although our orderings may of course be applied to the symmetric part of any matrix.

DEFINITION 4.1. *Let A be a square matrix with a triangular factorization $A = LU$. The I_F fill of A is defined to be the total number of nonzeros in the inverses of L and U, assuming no cancellation in the forming of those inverses.*

From Gilbert[10] and Liu[13], we have the following graph theoretic characterization of the structure of the inverse Cholesky factor:

THEOREM 4.2. *Let A be a SPD matrix with Cholesky factor L. Then assuming no cancellation the structure of $Z = L^{-T}$ corresponds to the transitive closure of the graph of L^T, that is, for $i < j$ we have $(L^{-1})_{ij} \neq 0$ if and only if there is a dipath from j to i in the (directed acyclic) graph of L^T. Furthermore, this is the same structure as given by the transitive closure of the upper triangular part of A, or as given by the transitive closure of the elimination tree of A.*

Notice that the last structure characterization simply means that $Z_{ij} \neq 0$ if and only if j is an ancestor of i in the elimination tree. This allows us to significantly speed the computation of the preconditioner given a bushy elimination tree, as well as allowing for parallelism—for the calculation of column j in the factors, only the ancestor columns need be considered.

These results suggest orderings that delay long dipaths in the triangular part of A (i.e. paths in A with monotonically increasing node indices), as these cause lots of I_F fill, or alternatively orderings which give short and bushy elimination trees.

Another useful characterization of I_F fill using notions from [9] allows us to calculate the number of nonzeros in each column of the inverse factors very cheaply during a symbolic factorization:

THEOREM 4.3. *$Z_{ij} \neq 0$ if and only if j is reachable from i strictly through nodes eliminated previous to i—or in terms of the quotient graph model, if i is contained in a supernode adjacent to j at the moment when j is eliminated.*

There are several existing ordering methods which might do well in reducing I_F fill, based on the results above: e.g. Red-Black, Minimum Degree, Nested Dissection. Our new ordering is based on theorem 4.3:

Minimum Inverse Penalty (MIP). The minimum degree methods are built around a symbolic Cholesky factorization of the matrix, at each step selecting the node(s) of minimum penalty (meaning degree, external degree, or various estimates) for elimination. In MIP, we base the penalty of a node i on the number of nonzeros created in the inverse factor L^{-T} if i were ordered next. Letting this quantity be

TABLE 4.1

Comparison of unweighted orderings. *I_F fill is how many thousands of nonzeros in the true inverse factors of the symmetric part, Pre-T is the time to compute the preconditioner, Its is the number of iterations for convergence, Sol-T is the time taken by the iterations, and Total-T is the total time taken including ordering. 'N/o' indicates no re-ordering of the given matrix.*

	I_F fill		Pre-T		Its		Sol-T		Total-T	
Matrix	n/o	MIP	n/o	MIP	n/o	MIP	n/o	MIP	n/o	MIP
ADD32	9083	52	33.9	3.7	6	5	0.6	0.5	34.5	4.3
ALE3D	365	404	14.3	15.9	126	33	18.4	4.9	32.7	21.2
BSSTK14	1554	573	6.2	3	77	64	9.3	7.9	15.5	11.3
MEMPLUS	156022	1736	832	54	135	21	67.7	11.8	900	66.5
ORSREG1	2432	643	7.6	3.3	43	42	2.4	2.4	10	5.9
SAYLR4	6352	2182	8.5	4.1	1386	158	71.8	8.3	80.3	12.9
WATT2	1723	583	7.4	2.8	77	7	3.5	0.3	10.9	3.3

$Zdeg_i$, and the current number of uneliminated neighbours (not counting supernodes) of i be $Udeg_i$, we found $Penalty_i = 2Zdeg_i + Udeg_i$ to be a fairly effective penalty function.

We tested these orderings on 25 matrices from a variety of fields, mostly from the Harwell-Boeing collection, with a 180MHz Pentium Pro. As a sample of these results, table 4.1 compares the performance of the given ordering versus MIP on a few matrices. In all cases we chose a threshold for AINV so that the preconditioner would have roughly the same number of nonzeros as the original matrix. Our test solve with BiCGSTAB used the vector of all ones as the solution, stopping when the residual norm was reduced by a factor of 10^{-9}.

Although the improvement in performance is considerable for most of our test matrices, we must warn that ordering didn't seem to solve the robustness problem—a few matrices remained insoluble even after ordering.

5. Weighted Orderings.

We discovered several discrepancies in the general trend that reducing I_F fill gives faster convergence. This happens with anisotropic matrices in particular—hardly surprising recalling the problems anisotropy poses for ILU. Thus a heuristic for handling anisotropy is needed for factored approximate inverses.

For this we concentrate on our second goal, reducing the magnitude of the entries in the true inverse factors. As in [5] we expect that strong connections should be delayed in the ordering, that is, we should order so that large off-diagonal entries appear as far to the end of the matrix as possible. In our research we evaluated the size of an off-diagonal entry A_{ij} simply by the magnitude $M_{ij} = |A_{ij}|/\sqrt{|A_{ii}A_{jj}}$ (assuming A has already been symmetrized).

Before considering weighted ordering algorithms, a simple demonstration of the effect of anisotropy is called for. The matrix SINGLEANISO comes from a 5-point finite difference discretization on a regular 31×31 grid of the PDE:

$$u_{xx} + 1000u_{yy} = F$$

Here the edges of A corresponding to the y-direction are 1000 times heavier than those corresponding to the x-direction. The first ordering ("Strong-first") we consider block-orders the grid columns with nested dissection, and then internally orders each block with nested dissection—this brings the strong connections close to the diagonal. The second ("Weak-first") block-orders the grid rows instead, pushing the strong connections away from the diagonal, delaying them until the last. These are illustrated

FIG. 5.1. **The two orderings of SINGLEANISO, depicted on the domain.** *Lighter shaded boxes indicate nodes ordered later.*

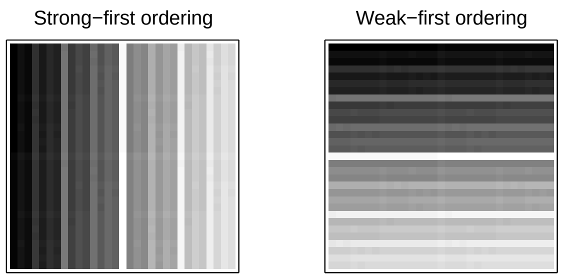

TABLE 5.1
Performance of strong-first ordering versus weak-first ordering for SINGLEANISO. *Pre-T is the time to compute the preconditioner, Its is the number of iterations required for solution, and Sol-T is the time taken by the iterations.*

Ordering	Pre-T	Its	Sol-T
Strong-first	0.51	29	0.4
Weak-first	0.38	25	0.25

in figure 5.1 where each square of the grid is shaded according to its place in the ordering. Both orderings produce a reasonable I_F fill of 103,682, with isomorphic elimination trees. However, they give very different performance—see table 5.1. In all respects the weak-first ordering is significantly better than the strong-first one. In figure 5.2 we plot the decay of the entries in the inverse factors resulting from the two orderings, and show parts of those factors. The much smaller entries from the weak-first ordering confirm our heuristic.

In creating weighted orderings to implement the heuristic on general matrices, we simply tried modifying the successful structural orderings. Our most promising one was based on a multi-level/spectral nested dissection algorithm:

Weighted Nested Dissection (WND). Consider the spectral bipartition algorithm. Finding the Fiedler vector is equivalent to minimizing (over a space orthogonal to the constant vectors):

$$\sum_{(i,j)\ is\ an\ edge} (x_i - x_j)^2$$

We then make the bipartition depending on which side of the median each entry lies. Notice that the closer together two entries are in value—i.e. the smaller $(x_i - x_j)^2$ is—the more likely those nodes will be ordered on the same side of the cut. We would like weakly connected nodes (where M_{ij} is small) to be in the same part and the strong connections to be in the edge cut, so we instead try minimizing the following weighted quadratic sum:

$$\sum_{(i,j)\ is\ an\ edge} \frac{(x_i - x_j)^2}{M_{ij}}$$

FIG. 5.2. **Comparison of the magnitude of entries in inverse factors for the different orderings of SINGLEANISO.** *The close-up images of the actual factors are shaded according to the magnitude of the non-zeros—darker means bigger.*

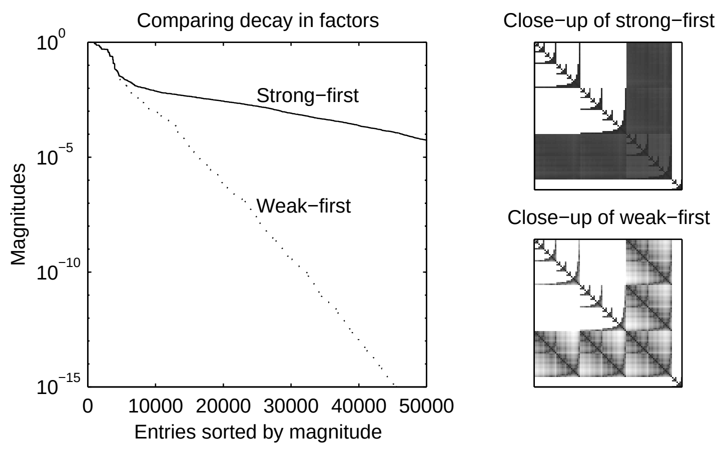

TABLE 5.2

Comparison of weighted orderings. *I_F fill is how many thousands of nonzeros in the true inverse factors of the symmetric part, Pre-T is the time to compute the preconditioner, Its is the number of iterations for convergence, and Sol-T is the time taken by the iterations.*

	I_F fill		Pre-T		Its		Sol-T	
Matrix	ND	WND	ND	WND	ND	WND	ND	WND
ADD32	134	669	3.2	2.9	5	6	0.5	0.6
ALE3D	197	639	6.4	15.7	68	34	10	5
BSSTK14	402	1526	2	3.6	107	85	13	10.3
MEMPLUS	2362	113777	59.4	54.7	17	19	9.7	15
ORSREG1	502	1913	2.7	4.6	39	35	2.2	2
SAYLR4	1153	5091	2.3	2.3	173	92	8.9	4.8
WATT2	353	1251	1.8	1.9	8	6	0.4	0.3

This corresponds to finding the second smallest eigenvalue of the weighted Laplacian matrix for the graph defined by:

$$(i,j) \text{ is an edge if and only if } A_{ij} \neq 0, \qquad \mathit{weight}(i,j) = M_{ij}^{-1}$$

Thus we modify Nested Dissection simply by changing the Laplacian used in the bipartition step (at the coarsest level) to this weighted Laplacian.

In table 5.2 we show the difference in Nested Dissection's performance with this weight scheme. We do not include the total time here, since our ordering algorithms ran in interpreted MATLAB code. Though the comparison is not as dramatic as between no ordering and good structural orderings, WND typically does improve convergence. Sometimes this is offset by the increased time to compute the preconditioner—probably due to WND's poor I_F fill reduction, roughly four times worse than ND. However, the key observation is that WND still performs far better than one would expect based on the I_F fill. This indicates we are getting much faster decay, just as our heuristic suggested. There is clear promise for a more sophisticated version of WND which, while keeping the same superior rate of decay, wouldn't suffer so much I_F fill and thus could outperform any structural ordering.

REFERENCES

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

[2] M. Benzi and M. Tůma, *A sparse approximate inverse preconditioner for nonsymmetric linear systems*, Tech. Rep. TR-PA-96-15, CERFACS, 1996. To appear in SIAM J. Sci. Comput., 19 (1998).

[3] E. Chow and Y. Saad, *Approximate inverse techniques for general sparse matrices*, Colorado Conference on Iterative Methods, April 5–9, (1994).

[4] E. Chow and Y. Saad, *Approximate inverse preconditioners via sparse-sparse iterations*, SIAM J. Sci. Comput., 19 (3), May 1998.

[5] S. Clift, H. Simon, and W.-P. Tang, *Spectral ordering techniques for incomplete LU preconditioners for CG methods*, manuscript.

[6] S. Clift and W.-P. Tang, *Weighted graph based ordering techniques for preconditioned conjugate gradient methods*, BIT, 35 (1995), pp. 30–47.

[7] E. D'Azevedo, P. Forsyth, W.-P. Tang, *Towards a cost-effective ILU preconditioner with high level fill*, BIT, 32 (1992), pp. 442–463.

[8] I. Duff and G. Meurant, *The effect of ordering on preconditioned conjugate gradients*, BIT, 29 (1989), pp. 635–637.

[9] A. George and J. Liu, *The evolution of the minimum degree ordering algorithm*, SIAM Review, 31 (1989), pp. 1–19.

[10] J. Gilbert, *Predicting structure in sparse matrix computations*, SIAM J. Matrix. Anal. Appl., 15 (1994), pp. 62–79.

[11] M. Grote and T. Huckle, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), no. 3, pp. 838–853.

[12] L. Kolotilina and A. Yeremin, *Factorized sparse approximate inverse preconditionings I. theory*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 45–58.

[13] J. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix. Anal. Appl., 11 (1990), pp. 134–172.

[14] W.-P. Tang, *Towards an effective sparse approximate inverse preconditioner.* To appear in SIAM J. Matrix. Anal. Appl., 1998.