NASA-CR-199548

RIACS

Research Institute for Advanced Computer Science
NASA Ames Research Center

Spectral Ordering Techniques for Incomplete LU Preconditioners for CG Methods

Simon S. Clift, Horst D. Simon And Wei-Pai Tang

RIACS Technical Report 95.20 September 1995

Spectral Ordering Techniques for Incomplete LU Preconditioners for CG Methods

Simon S. Clift, Horst D. Simon And Wei-Pai Tang

The Research Institute of Advanced Computer Science is operated by Universities Space Research Association, The American City Building, Suite 212, Columbia, MD 21044, (410) 730-2656

Work reported herein was sponsored by NASA under contract NAS 2-13721 between NASA and the Universities Space Research Association (USRA), the Natural Sciences and Engineering Research Council of Canada, the Information Technology Research Center, which is funded by the Province of Ontario, and the Queen's University at Kingston School of Graduate Studies and Faculty of Applied Science.

SPECTRAL ORDERING TECHNIQUES FOR INCOMPLETE LU PRECONDITIONERS FOR CG METHODS *

SIMON S. CLIFT †, HORST D. SIMON ‡ AND WEI-PAI TANG §

Abstract. The effectiveness of an incomplete LU (ILU) factorization as a preconditioner for the conjugate gradient method can be highly dependant on the ordering of the matrix rows during its creation. Detailed justification for two heuristics commonly used in matrix ordering for anisotropic problems is given. The bandwidth reduction, and weak connection following heuristics are implemented through an ordering method based on eigenvector computations. This spectral ordering is shown to be a good representation of the heuristics. Analysis and test cases in two and three dimensional diffusion problems demonstrate when ordering is important, and when an ILU decomposition will be ordering insensitive. The applicability of the heuristics is thus evaluated and placed on a more rigorous footing.

Key words. preconditioned conjugate gradients, preconditioner, ordering methods

AMS(MOS) subject classifications. 65F10

* Work reported herein was supported in part by: NASA under contract NAS 2-13721 between NASA and the Universities Space Research Association (USRA), the Natural Sciences and Engineering Research Council of Canada, the Information Technology Research Center, which is funded by the Province of Ontario, and the Queen's University at Kingston School of Graduate Studies and Faculty of Applied Science.

† Center for Advanced Gas Combustion Technology, Department of Mechanical Engineering, Queen's University at Kingston, Kingston, Ontario, Canada, K7L 3N6, email: clift@me.queensu.ca.

‡ Research Market Development Manager, Silicon Graphics, Mail Stop 7L-580, 2011 N. Shoreline Blvd., Mountain View, CA 94043, U.S.A.

§ Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1.

1. Introduction. Incomplete LU (ILU) matrix decompositions are used both as preconditioners to conjugate gradient (CG) methods, as well as smoothers in multigrid methods (Raw 1994, Stevenson 1994), in solving the linear systems that arise during the numerical solution of PDEs. A number of studies have examined the effect of matrix elimination order on the quality of ILU preconditioners (D'Azevedo, Forsyth and Tang 1992a, D'Azevedo, Forsyth and Tang 1992b, D'Azevedo, Forsyth and Tang 1992c, Duff and Meurant 1989, Dutto 1992, Notay 1993, Doi 1991). In (Duff and Meurant 1989, D'Azevedo et al. 1992a, D'Azevedo et al. 1992b, D'Azevedo et al. 1992c) evidence was presented to demonstrate how matrix ordering can have a profound effect on that quality when solving anisotropic PDEs with preconditioned CG (PCG) methods.

This paper investigates an expression of the matrix ordering heuristic described in (D'Azevedo et al. 1992b, Margenov and Vassilevski 1994) which draws information from the PDE from which the linear system arises, and the grid over which it is discretized. If the factorization is

$$A = LU + E$$

where L and U are the lower and upper factors retained, and E the factors discarded, then we want to minimize some measure of E (such as the matrix norm). The heuristic is, in short, that the matrix rows should be ordered to follow anisotropy in the grid in the direction of weak connections (or close grid spacing), which, in a fundamental case, will be shown to produce low magnitude entries in the discarded fill. An explicit implementation of this heuristic using combinatorial algorithms was investigated in (Clift and Tang 1995). Reducing the bandwidth of the matrix is also known to improve the quality of an ILU factorization (Duff and Meurant 1989, Dutto 1992). Increased fill overlap in the narrow bandwidth means the fraction of the retained fill is higher relative to the complete factorization with such orderings.

One expression of these heuristics can be found in the two-sum of a matrix, which is minimized to find an ordering. Let matrix A be of size $n \times n$, with components a_{ij}. We define the *p-sum* (or *p-discrepancy*), $\sigma_p(A)$ as

$$\sigma_p^p(A, \psi) = \left[\sum_{\substack{1 \le i,j \le n \\ a_{ij} \neq 0}} a_{ij} \, |\psi(i) - \psi(j)|^p \right]^{1/p} \tag{1.1}$$

where and $\psi : V \rightarrow \{1 \ldots |V|\}$ is an ordering, or mapping of the n vertices V of the associated graph (or the rows of the matrix).

If an ordering follows the strongly connected nodes, and reduces the bandwidth, $\sigma_p(A)$ should be low, or minimized. The strong weights will then tend to be near the diagonal of the matrix in the mapping and have a

correspondingly small $|\psi(i) - \psi(j)|$ value, and the small weights will offset the larger distances from the diagonal. Thus we need to compute the problem for some matrix B that that corresponds to A but with inverted weights.

The two-sum, $\sigma_2(B)$, will be used as a measure of the weighted band-width of the matrix. We will be concerned with a *minimum 2-sum* problem associated with matrix B. We wish to find an ordering, a mapping ψ, such that

$$\sigma_2(B) = \min_{\psi} \sigma_2(B, \psi) \ .$$

The fundamental technique we employ to minimize σ_2 is that of (Hall 1970), which involves computing the smallest eigenvector of that associated matrix. The existence of this spectral method dictates the choice of $p = 2$.

The goal of this study was to evaluate the applicability of the basic ordering heuristics to various cases using a more mathematically sound basis than had been previously attempted. The ordering methods presented here are computationally expensive, and thus not particularly practical for day-to-day use. They do, however, indicate where less expensive expressions of the heuristics (such as those described in (Pothen, Simon and Wang 1992, Clift and Tang 1995)) can expect success, and point the way to their sound and appropriate application.

In the sections immediately following we outline a number of the basic concepts used in our research. However, the reader may wish to review (Wallis 1983, Langtangen 1989, D'Azevedo et al. 1992b) for an outline of level based, incomplete, factorization (denoted ILU(l), where l is the level of fill retained in the preconditioner). We will be referring to matrices as weighted graphs, where the matrix rows represent vertices, and the graph edges are encoded in the off-diagonals, the magnitude of the off-diagonal coefficients providing the "strength" of the connections. The reader may wish to review (Parter 1961, Rose 1972, George and Liu 1981, D'Azevedo et al. 1992a) for relevant information on this view of matrices. The concepts connecting σ_2, and the eigenvalues of an associated matrix are to be found in (Hall 1970, Fiedler 1975, Juvan and Mohar 1992, Juvan and Mohar 1993, Mohar and Poljak 1994), the latter being a particularly good survey paper on eigenvalues in combinatorics. These concepts have recently been used in graph decomposition (Pothen et al. 1992, Peyton, Pothen and Yuan 1992, Hendrickson and Leland 1993, Chan, Schlag and Zien 1994), and matrix band-width reduction (Barnard, Pothen and Simon 1993, George and Pothen 1994).

After justifying the heuristic for ordering two dimensional linear diffusion problems, we will link it with the $\min \sigma_2$ problem. This paper presents experimental results and analysis for a set of test cases solved with a PCG method using an ILU preconditioner constructed with the resulting ordering. The exposition will be repeated for three dimensional linear diffusion

problems, with important differences noted. In conclusion we will summarize guidelines about the validity and applicability of the heuristics.

2. Lower bound of fill entries in decomposition. The theoretical justification for the heuristic, noted in the introduction, of following weak connections in the graph requires exploring the connection between the fill in an LU decomposition and the weighted graph representation of the matrix. The following provides an estimate of the lower bound of the fill which supports our heuristic.

Let $G_A = (\mathcal{O}_A, \mathcal{E}_A)$ be the digraph represented by matrix A

$$\mathcal{O}_A = \{o_1, \ldots, o_n\}, \qquad \mathcal{E}_A = \{(o_i, o_j) \mid a_{o_i}(o_j) = a_{ij} \neq 0\}$$

Denote $\mathcal{S}$ be a subset of $\mathcal{O}$ and node $u \notin \mathcal{S}$. Node u is said to be *reachable* from a node v through $\mathcal{S}$ if there exists a path $(v, o_1, o_2, \ldots, o_k, u)$ in graph G_A, such that each $o_i \in \mathcal{S}$, $1 \leq i \leq k$ (George and Liu 1981). Similarly, node u is said to be *contributive* to a node v through $\mathcal{S}$ if there exists a path $(u, o_1, o_2, \ldots, o_k, v)$ in graph G_A, such that each $o_i \in \mathcal{S}$, $1 \leq i \leq k$. If A is self-adjoint, then the statement of "u is reachable from v" is equivalent to "u is contributive to v".

The reachable set and contributive set of node o_i through $\mathcal{S}$ is denoted by

$$\begin{aligned} \mathcal{R} = &\quad \mathit{Reach}(o_i, \mathcal{S}) &&= \{u \mid u \text{ is reachable from } o_i \text{ through } \mathcal{S}\} \\ \mathcal{C} = &\quad \mathit{Contributive}(o_i, \mathcal{S}) &&= \{u \mid u \text{ is contributive to } o_i \text{ through } \mathcal{S}\} \end{aligned}$$

For a self-adjoint matrix, $\mathcal{R} = \mathcal{C}$. If $\mathcal{S} = \{o_1, \cdots, o_{i-1}\}$ is the set of all nodes which are eliminated before o_i , then $\mathcal{R}$ is the set of nodes o_j for which $u_{ij} \neq 0$ and $\mathcal{C}$ is that of $l_{ji} \neq 0$. Notice $j > i$. Let $\mathcal{I} = \mathcal{S} \cup o_i$ and $\mathcal{J} = \mathcal{S} \cup o_j$. Let $A[\mathcal{I}, \mathcal{J}]$ be the submatrix extracted from the rows in $\mathcal{I}$ and columns in $\mathcal{J}$, and $A(\mathcal{I}) = A[\mathcal{I}, \mathcal{I}]$ is the principal submatrix of A in the rows and columns defined by $\mathcal{I}$. It is well known that l_{ij} and u_{ji} can be expressed in terms of determinants:

$$l_{ij} = \frac{\det A[\mathcal{I}, \mathcal{J}]}{\det A[\mathcal{I}, \mathcal{I}]} \; ; \qquad\qquad u_{ij} = \frac{\det A[\mathcal{J}, \mathcal{I}]}{\det A[\mathcal{I}, \mathcal{I}]}$$

Denote $A_p(v \rightarrow u)$ to be the product of elements on the path p from v to u, namely,

$$A_p(v \rightarrow u) = a_{v,o_1} a_{o_1,o_2} \cdots a_{o_k,u}.$$

Denote $V(p, \mathcal{S})$ to be the subset of the nodes of $\mathcal{S}$ not belonging to path p and $A(V(p, \mathcal{S}))$ be the principal submatrix of A in the rows and columns defined by $V(p, \mathcal{S})$, i.e., the nodes not on the path p. We restate a result from (Maybee, Olesky, van den Driessche and Wiener 1989) (Corollary 8.2) here:

LEMMA 2.1.

$$\det A[\mathcal{I},\mathcal{J}] = \sum_{k=1}^{m}(-1)^{l_k} A_{p_k}(i \to j) \det A(V(p_k,\mathcal{S})) \tag{2.1}$$

$$\det A[\mathcal{J},\mathcal{I}] = \sum_{k=1}^{m}(-1)^{l_k} A_{p_k}(j \to i) \det A(V(p_k,\mathcal{S})) \tag{2.2}$$

where the sum is over all distinct paths from o_i to o_j in $\mathcal{I} \cup o_j$; l_k is the length of path p_k from o_i to o_j.

We can have the following result directly from this lemma:

LEMMA 2.2. *Let $\mathcal{S}$ be the subset of all nodes which are eliminated before node o_i in the LU decomposition process and $A = LU$ be the LU decomposition of A. Then*

$$u_{ij} \neq 0 \quad \text{only if } o_j \text{ is reachable through } \mathcal{S} \text{ from } o_i,$$
$$l_{ji} \neq 0 \quad \text{only if } o_i \text{ is contributive to } o_j \text{ through } \mathcal{S}.$$

If A is an M-matrix, each term in the summations of (2.1) and (2.2) is positive. We then have the following lower bound:

THEOREM 2.1. *Let matrix A be an M-matrix, $A = LU$ and $\mathcal{S} = (o_1, o_2, \cdots, o_{i-1})$. If o_j is reachable from o_i through $\mathcal{S}$, then*

$$|u_{ij}| \geq \max_{p_k} \frac{|A_{p_k}(o_i \to o_j)|}{A_{p_k}(\mathcal{I})} > 0 \tag{2.3}$$

where $p_k = (o_i, o_{l_1}, o_{l_2}, \ldots, o_{l_k}, o_j)$, $o_{l_i} \in \mathcal{S}$ is a path from node o_i to o_j and $A_{p_k}(\mathcal{I}) = a_{i,i}a_{l_1,l_1}\cdots a_{l_k,l_k}$.

If o_i is contributive to o_j through $\mathcal{S}$, then

$$|l_{ji}| \geq \max_{p_k} \frac{|A_{p_k}(o_j \to o_i)|}{A_{p_k}(\mathcal{J})} > 0 \tag{2.4}$$

where $p_k = (o_j, o_{l_1}, o_{l_2}, \ldots, o_{l_k}, o_i)$, $o_{l_i} \in \mathcal{S}$ is a path from node o_j to o_i and $A_{p_k}(\mathcal{J}) = a_{j,j}a_{l_1,l_1}\cdots a_{l_k,l_k}$.

Proof. The proofs of (2.3) and (2.4) are similar, only (2.3) is shown. Let $\mathcal{K}$ be the complementary set of $V(p_k,\mathcal{S})$ in set $\mathcal{S} \cup o_i$. It was shown in (Engel and Schneider 1977) that if A is a non-singular M-matrix, then

$$0 < \det A(\mathcal{I}) \leq \det A(V(p_k,\mathcal{S})) \det A(\mathcal{K}). \tag{2.5}$$

Notice, all principal minors of an M-matrix are also M-matrices. From (2.5), a direct generalization can be obtained:

$$\det A(\mathcal{I}) = \det A[\mathcal{I},\mathcal{I}] \leq a_{11}a_{22}\cdots a_{ii} \tag{2.6}$$

Since each term in (2.1) is positive, we have

$$\begin{aligned} |l_{ij}| &= \frac{\det A[\mathcal{I}, \mathcal{J}]}{\det A(\mathcal{I})} \\ &\geq \frac{|A_{p_k}(i \to j)| \det A(V(p_k, \mathcal{S}))}{\det A(\mathcal{I})} \\ &\geq \frac{|A_{p_k}(i \to j)| \det A(V(p_k, \mathcal{S}))}{\det A[V(p_k, \mathcal{S})] \det A(\mathcal{K})} \\ &\geq \frac{|A_{p_k}(i \to j)|}{A_{p_k}(\mathcal{I})} \end{aligned}$$

where p_k is any path from o_i to o_j. □

If a particular ordering would allow a path on which large elements (weights) are located, large magnitude fill elements will result. This supports our heuristics and our observations in testing.

3. Two dimensional grid ordering heuristic. Consider the linear diffusion problem

$$\nabla \cdot (K \nabla U) = -q \tag{3.1}$$

in two dimensions over a square region $0 \leq x, y \leq 1$ where $K = K(x, y)$ is a position dependant coefficient vector, and q is the source term. We used the usual five-point finite difference discretization of this equation over a uniform, Cartesian grid, (a harmonic average is used to deal with the cases where K is discontinuous). This produces a symmetric $n \times n$ linear system

$$Ax = b$$

with between three and five entries per matrix row, and the source terms expressed in the right hand side. The linear system will have positive off-diagonals that correspond to the local values of K, which can be considered the connection strengths between nodes of the problem. Sources $q(x, y)$ are represented by multiplying the diagonal, and source strength in the right hand side of the linear system, by a sufficiently large value, thus retaining matrix symmetry.

Our prototypical anisotropic problem is the case where $K_x = 1000$ and $K_y = 1$. The matrix which results is a symmetric M-matrix. If we scale the matrix so that the diagonal entries are unity, all the edges aligned along the x-axis will have a connection strength of $\mathcal{O}(K_x/K_x + 1)$ and the edges aligned with the y-axis will have a connection strength of $\mathcal{O}(1/K_x)$. If the natural $x - y$ ordering is used, then all new fill entries will be oriented more in the x-direction (see figure 9.1), and by the lower bound given in theorem 2.1, the fill entries will have a slow decay rate. If natural $y - x$ ordering is used, the fill entries will have a more rapid decay rate. Thus, the value of the fill using the $y - x$ ordering will have less of a bearing on the quality

of the preconditioner as the level of fill increases than the fill using $x - y$ ordering. Natural $y - x$ ordering was shown to produce a profoundly more effective preconditioner for the CG method in (D'Azevedo et al. 1992b).

As we noted in the introduction, the quantity we minimize to implement this heuristic is not the two-sum of A, but rather of a related matrix B. We shall use B, with elements b_{ij} such that

$$b_{ij} = \begin{cases} \frac{1}{a_{ij}} & \text{where } a_{ij} \neq 0 \\ 0 & \text{where } a_{ij} = 0 \end{cases} . \tag{3.2}$$

Our case of constant K coefficients over the entire region permits us to demonstrate the following.

LEMMA 3.1. *The two-sum of B under natural ordering is a local minimum with respect to any single permutation of the matrix labelling.*

Proof. The proof is trivial.

If we consider cases where $N_x \approx N_y$ then if $K_x > K_y$, and hence $1/K_x < 1/K_y$ then

$$\begin{aligned} \sigma_2(B, x - y \text{ ordering}) &= \frac{1}{K_x}(N_x - 1)N_y + \frac{1}{K_y}N_y N_x^2 \\ \sigma_2(B, y - x \text{ ordering}) &= \frac{1}{K_y}(N_y - 1)N_x + \frac{1}{K_x}N_x N_y^2 \end{aligned}$$

and we see that of the two local minima associated with natural orderings, the one dictated by the above heuristic, the $y - x$ ordering, corresponds to that with the lowest σ_2. However, if $N_x \gg N_y$ then a natural ordering that reduces bandwidth can be expected to produce the lowest two-sum because $N_x N_y^2 \ll N_y N_x^2$. This will be seen in the experimental results over a long thin problem, even with $K_x = 1000 K_y$ (see the LONGTHIN problem below).

4. Spectral Ordering Algorithm. We now present the spectral ordering algorithm, which will be justified in the next section. Given matrix A we define symmetric matrix C with entries [1]

$$c_{ij} = \begin{cases} |\, 1/a_{ij} \,| & i \neq j \\ 0 & i = j \end{cases}$$

and diagonal matrix D with entries

$$d_{ii} = \sum_j c_{ij} \ .$$

[1] Another definition for C proposed during our research was to set $c_{ij} = 1/\exp(|a_{ij}|)\ \forall\ a_{ij} \neq 0$. This, however, produced matrix entries that varied by too many orders of magnitude, making eigenvalues impossible to compute for many test cases, and producing no favourable effects in general.

We can define the *weighted pseudo-laplacian* of B as

$$B_l = C - D \tag{4.1}$$

which is a symmetric, positive semi-definite matrix of rank $(n-1)$ (Hall 1970). Using B_l as the matrix related to A, with connection strengths inverted, the spectral ordering algorithm proceeds as in figure 9.2.

5. Minimizing σ_2. The spectral ordering algorithm is justified by the proof originally presented by (Hall 1970), and presented in the unweighted context by (Barnard et al. 1993). Detailed analysis of this sort of algorithm is presented in (George and Pothen 1994). The following demonstrates that computing the second weighted pseudo-Laplacian eigenvector solves a continuous relaxation of the discrete problem of minimizing the two-sum of a matrix.

For odd (even) n, let Ψ denote the set of orderings ψ as defined in Equation 1.1. We shall evaluate the mapped row positions $\psi(i)$ as integers such that $1 \leq \psi(i) \leq n$. Thus

$$min_{\psi \in \Psi} \sigma_2(B_l) = \frac{1}{2} min_{\psi \in \Psi} \sum_{c_{ij} \neq 0} c_{ij}(\psi(i) - \psi(j))^2 \quad .$$

Consider a continuous vector $x \in \Re^n$ with elements x_i. We can define a permutation $\psi_x \in \Psi$ induced by x by the rule that $p_i \leq p_j$ if and only if $x_i \leq x_j$. This ordering is unique except where two or more components of x are equal.

We now minimize the two-sum over the class of unit n-vectors $x \in \mathcal{X}$ satisfying $x \neq 0$, $x^t x = 1$, $x^t u = 0$ where $u^t = (1, 1, \ldots, 1)$. This relaxes the condition that the reordering vector must belong to a set of permutation vectors, leaving the continuous optimization problem:

$$\begin{aligned}
\frac{1}{2} min_{x \in \mathcal{X}} \sum_{c_{ij} \neq 0} c_{ij}(x_i - x_j)^2 &= min_{x \in \mathcal{X}} \left(\sum_{c_i = 1}^{n} d_{ii} x_i^2 - \sum_{c_{ij} \neq 0} c_{ij} x_i x_j \right) \\
&= min_{x \in \mathcal{X}} \left(x^t D x - x^t C x \right) \\
&= min_{x \in \mathcal{X}} x^t B x \\
&= \lambda_2 E_2^t E_2 = \lambda_2 \qquad (5.1) \\
& \qquad (5.2)
\end{aligned}$$

Noting that the first eigenvalue of B is zero, and its corresponding eigenvector trivial, the second weighted pseudo-Laplacian eigenvector E_2 solves the continuous approximation to the two-sum problem. The proof that the resulting permutation ψ_x induced by E_2 is the discrete vector that is closest to minimizing σ_2 is to be found in the papers cited above.

As a demonstration we compute the second smallest eigenvector of the discretized 2D anisotropic problem defined by equation 3.1. With the

coefficients $K_x = 1000$, and $K_y = 1$ the eigenvector for the problem over a 3×3 grid is

$$E_2 = \{1, 1, 1, 0, 0, 0, -1, -1, -1\} \tag{5.3}$$

and we see that the natural ordering is permitted under the procedure for inducing the ordering ψ. For the problem over a 30×30 grid the eigenvalue is plotted as a surface in figure 9.3. Again, natural ordering is, modulo the tie-breaking within rows, the result. The eigenvector exhibits the same anisotropy as the matrix. We observe this in other problems as well, such as the ANISO problem below.

At this point we assert, admittedly on less than rigorous grounds, that the spectral ordering algorithm is a reasonable embodyment of the ordering heuristics, and that these heuristics are a reasonable approach to ordering two-dimensional problems for ILU factorization.

6. Experimental results for 2D problems. For comparison to the spectrally based orderings, we will also compute the Reverse Cuthill-McKee (RCM) (George and Liu 1981), Minimum Update Matrix (MUM), and Minimum Discarded Fill (MDF) orderings (D'Azevedo et al. 1992a, D'Azevedo et al. 1992b, D'Azevedo et al. 1992c). RCM has been shown to be a good bandwidth reducing ordering (Duff and Meurant 1989, Dutto 1992), and works only on the sparsity pattern of the matrix. MUM and MDF are sensitive to the matrix coefficients, and have been shown to produce highly effective orderings in a number of situations (although MUM is better for problems with larger computational molecules) (Clift and Forsyth 1994, Chin, D'Azevedo, Forsyth and Tang 1992, D'Azevedo et al. 1992b, Notay 1993). The MUM and MDF orderings are, relative to RCM, quite expensive to compute.

Table 9.1 shows the effect of the various orderings on a set of 2D problems over a regular grid. Appendix A gives the details of how each problem is specified. Each problem has a different layout, or degree of anisotropy. The problems were solved using the conjugate gradient method, with a level 1 ILU preconditioner (denoted ILU(1)), i.e. the first level of fill was kept. ILU(0) was found to be relatively insensitive to ordering, and since the heuristic only really addresses the first level of fill, higher levels would be expected, and were found, to benefit less from this ordering strategy. (In addition, higher levels of fill are not practical for many real applications). The solutions were converged to a residual of 10^{-12} relative to the residual of the initial zero vector guess.

The work required is measured as the number of iterations required during the solution phase, multiplied by the total number of non-zero entries in the matrix and the preconditioner. We justify this work measurement by the fact that the matrix-vector multiply and preconditioner-vector solve are the two variable components in the cost of the PCG algorithm, and the most expensive operations. The computation of the preconditioner

is generally a small cost relative the solution iteration. The first column (RCM ordering) of Table 9.1 shows the absolute measure of work, and the remaining columns are shown as fractions of the work required to solve the problem relative to RCM ordering.

6.1. Spectral ordering. Spectral ordering performed generally as expected. For the isotropic problem, LAPD5, bandwidth reduction is the only heuristic applicable. The BIG1DIR problem, with its uniform anisotropy, was ordered quite well. The ANISO problem, with abutting anisotropic regions of differing directions, was also ordered properly. With one exception, the spectral ordering algorithm performed well whenever the MDF algorithm did well.

The STONE and STONEROT90 problems have exactly the same layout, except for a rotation of 90 degrees. Figures 9.4 and 9.5 show the eigenvectors of the two problems. Note that in both, the region of $K_x = K_y = 0$ appears as a flat spot on the surface plot. The STONEROT90 case eigenvector is such that this appears where it will be ordered roughly between the nodes surrounding it, whereas in STONE the region will not be ordered so as to keep it with its neighboring nodes. This shows a weakness of the spectral ordering algorithm in dealing with such blocks.

The LONGTHIN problem has a direction of anisotropy that would cause the weak-direction following heuristic to order in the direction that maximizes the bandwidth. Figure 9.6 shows the sparsity pattern for spectral ordering, indicating that the bandwidth reduction tendencies of the ordering dominate the other heuristic.

7. Three dimensional grid ordering heuristic. In three dimensions the ordering heuristics becomes less clear cut. As with many other topics related to topology (such as triangulations, hierarchichal bases, etc.), the weak connection heuristic for the plane graph can not be generalized to the three-dimensional grid. The bandwidth reduction hypothesis still holds, but following the weak connections in the graph can fail to improve the ILU factorization.

Consider the incomplete factorization process for equation 3.1 in three dimensions, discretized over a square grid with constant coefficients k_x, k_y and k_z. During the elimination process, we will follow the x direction first, then the y direction. Figure 9.7 shows this schematically, with the shaded nodes being eliminated in the order shown by the arrow. By theorem 2.1 the magnitude of the three first level fill elements created when node 1 is eliminated (shown in dashed lines) will be

$$\text{Level 1 Fill} \begin{cases} k_z k_x \\ k_z k_y \\ k_y k_x \end{cases} .$$

The second level of fill generated when node 2, and, later, the first node in the y direction, node 4, are eliminated will have four elements with

magnitudes of order

$$\text{Level 2 Fill} \begin{cases} k_z k_x k_x \\ k_y k_x k_x \\ k_z k_y k_y \\ 2(k_z k_y k_x) \end{cases}.$$

For an ILU(1) factorization, these second level fill elements are discarded. Figure 9.2 shows the magnitude of the sum of the fill discarded given different combinations of weighting coefficients. Note that when there are two weak, and one strong direction, there is an ordering that discards substantially less fill ($\approx 10^3$ versus $\approx 10^6$ in magnitude). If there are two strong directions, then there is no ordering that discards substantially less fill than the others.

This is borne out experimentally. Table 9.3 shows the number of PCG iterations required to solve three problems with homogeneous anisotropy tested over the six natural orderings (the fill patterns were identical, hence measuring using solver work as defined in section 6 was not required). The problem with one strong, and two weak connections shows that an ordering with the two weak connections first produces a good preconditioner, and thus fewer iterations were required to get a solution. The problem with two strong weights had no good ordering at all. The problem with a weak, medium, and strong and weight showed that although ordering the directions from those of weakest to strongest connections produced a good ordering.

Also of note is the measure of the two-sum of the inverse-weighted Laplacian, shown in table 9.4 in entries corresponding to table 9.3. A low two-sum was consistently associated with a good ordering, when one existed. Good orderings did not necessarily have a low two-sum.

8. Experimental results for 3D problems. As in section 6, Table 9.5 shows the effect of the various orderings on a set of 3D problems over a regular grid. These problems were solved using the same method and tolerances as the 2D problems. The table is also displayed in terms of solver work, with RCM given as an absolute value and other orderings shown as fractions of that work.

Spectral ordering did not perform as well as hoped. The BIG1DIR3F case looks good because the RCM algorithm accidentally created a very poor ordering (RCM is coefficient insensitive). Only in the weak, weak, strong case (BIG1DIR3G) were the uniform anisotropy cases effectively ordered. All other results were poor, indicating that the heuristic embodied in spectral ordering has a limited application.

An interesting case in point is the ANISO3D problem. Figure 9.8 visualizes three planes through the eigenvector values field. Smaller icons represent lower values. Note that the division between the eight quadrants of the problem, particularly between the front and back (as shown) are

quite visibly expressed in the eigenvector. However, Figure 9.9 shows that the sharp distinctions are lost in the ordering, (as visualized by the same technique). We note that opposite corners on the back plane would be ordered together, which runs counter to the heuristics. This failure to capture distinct regions suggests why the spectral ordering technique fared poorly, even though the anisotropic blocks have one strong, and two weak coefficient directions.

It is interesting to note that MDF, which uses the more aggressive heuristic of evaluating discarded fill directly to determine successive nodes in an ordering, was able to produce more consistent results. The BIG1DIR3H problem, which has two strong and one weak direction was slightly improved over RCM ordering.

9. Conclusions. In this work a more rigorous basis for matrix ordering heuristics was set in the context of two-dimensional, anisotropic diffusion problems discretized over a regular grid. The minimization of the two-sum of the weighted pseudo-Laplacian of the matrix has been shown to be a good expression of the weak connection following heuristic, and the bandwidth reduction heuristic. Using a well-known spectral algorithm to minimize this two-sum, the resulting orderings have been experimentally tested for their effectiveness for both two and three dimensional problems.

This study shows that the weak connection following heuristic is most effective over the problems from which it was derived: two dimensional, regular grid problems with fairly simple patterns of anisotropy. We have demonstrated that there are three dimensional cases where following the weak connections in the grid must fail to improve the ILU factorization. These cases, where more than one of the grid directions has strong connectivity, appear to be reasonably ordering insensitive. Two dimensional problems, with minimal connectivity (i.e. the 5 point finite difference stencil) seem to be the most sensitive to the ordering used for ILU factorizations. MDF ordering produced improvements in cases demonstrated to be insensitive to the direction heuristic, indicating that more aggressive ordering techniques can still produce savings.

In (Clift and Tang 1995) an inexpensive approximation to the weak connection following ordering heuristic was developed and tested. However, that study did not reveal or predict the reasons why the methods succeeded or failed. Rapid methods for ILU ordering still remain to be developed for anisotropic convection-diffusion problems where convection dominates. This is the object of a current research project.

Acknowledgements. The authors would like to thank Dr. Alex Pothen for his insights into direct matrix solver orderings using spectral methods, and his suggestion that we investigate spectral methods for incomplete LU decompositions, which lead to this study.

REFERENCES

Barnard, S. T., Pothen, A. and Simon, H. D.: 1993, A spectral algorithm for envelope reduction of sparse matrices, *Tech. Rep. CS-93-49*, University of Waterloo.

Chan, P. K., Schlag, M. D. F. and Zien, J. Y.: 1994, Spectral k-way ratio-cut partitioning and clustering. Computer Engineering, U. California, Santa Cruz.

Chin, P., D'Azevedo, E. F., Forsyth, P. A. and Tang, W.-P.: 1992, Preconditioned conjugate gradient methods for the incompressible Navier-Stokes equations, *International Journal for Numerical Methods in Fluids* **15**, 273–295.

Clift, S. S. and Forsyth, P. A.: 1994, Lienar and non-linear iterative methods for the incompressible Navier-Stokes equations, *International Journal for Numerical Methods in Fluids* **18**, 229–256.

Clift, S. S. and Tang, W.-P.: 1995, Weighted graph based ordering techniques for preconditioned conjugate gradient methods, *BIT: Nordisk Tijdskrift for Informationbehandling* **35**, 30–47.

D'Azevedo, E. F., Forsyth, P. A. and Tang, W. P.: 1992a, Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems, *SIAM J. Matrix Anal. Appl.* **13**(3), 944–961.

D'Azevedo, E. F., Forsyth, P. A. and Tang, W.-P.: 1992b, Towards a cost-effective ILU preconditioner with high level fill, *BIT* **32**, 442–463.

D'Azevedo, E. F., Forsyth, P. A. and Tang, W.-P.: 1992c, Two variants of minimum discarded fill ordering, *in* R. Beauwens and P. de Groen (eds), *Proc. IMACS Intern. Symp. on Iterative Methods in Linear Algebra*, North-Holland, pp. 603–612.

Doi, S.: 1991, A Gustafsson-type modification for parallel ordered incomplete factorizations, *Technical report*, C&C Information Technology Research Labs, NEC Corp.,.

Duff, I. S. and Meurant, G. A.: 1989, The effect of ordering on preconditioned conjugate gradients, *BIT: Nordisk Tijdskrift for Informationbehandling* **29**, 635–657.

Dutto, L. C.: 1992, The effect of ordering on preconditioned GMRES algorithm, for solving the compressible Navier-Stokes equations, *Research report*, Université de Montréal, CRM. to appear in International Journal for Numerical Methods in Engineering.

Engel, G. and Schneider, H.: 1977, The Hadamard-Fischer inequality for a class of matrices defined by eigenvalue monotonicity, *Linear Multilinear Algebra* **4**, 155–176.

Fiedler, M.: 1975, A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory, *Czech. Math. Journal* **25**(100), 619–633.

George, A. and Liu, J. W. H.: 1981, *Computer Solutions of large sparse positive-definite systems,*, Prentice-Hall, Englewood Cliffs, New Jersey.

George, A. and Pothen, A.: 1994, Analysis of a spectral envelope reduction algorithm via quadratic assignment problems. Draft Preprint.

Hall, K. M.: 1970, An r-dimensional quadratic placement algorithm, *Management Science — Theory Series* **17**(3), 219–229.

Hendrickson, B. and Leland, R.: 1993, A multilevel algorithm for partitioning graphs, *Technical Report SAND93-1301*, Sandia National Laboratories.

Juvan, M. and Mohar, B.: 1992, Optimal linear labelings and eigenvalues of graphs, *Discrete Applied Mathematics* **36**, 153–168.

Juvan, M. and Mohar, B.: 1993, Laplace eigenvalues and bandwidth-type invariants of graphs, *Journal of Graph Theory* **17**(3), 393–407.

Langtangen, H. P.: 1989, Conjugate gradient methods and ILU preconditioning of non-symmetric matrix systems with arbitrary sparsity patterns, *International Journal for Numerical Methods in Fluids* **9**, 213–233.

Margenov, S. D. and Vassilevski, P. S.: 1994, Algebraic multilevel preconditioning of anisotropic elliptic problems, *SIAM J. Sci. Comput.* **15**(5), 0126–1037.

Maybee, J., Olesky, D., van den Driessche and Wiener, G.: 1989, Matrices, digraphs and determinants, *SIAM J. Matrix Anal. Appl.* **10**, 500–519.

Mohar, B. and Poljak, S.: 1994, Eigenvalues in combinatorial optimization. Preprint.
Notay, Y.: 1993, Ordering methods for approximate factorization preconditioning, *Technical report*, Service de Métrologie Nucléaire, Université Libre de Bruxelles.
Parter, S. V.: 1961, The use of linear graphs in gaussian elimination, *SIAM Review* **3**, 364–369.
Peyton, B. W., Pothen, A. and Yuan, X.: 1992, Partitioning a chordal graph into transitive subgraphs for parallel sparse triangular solution, *Technical Report CS-92-55*, University of Waterloo. Submitted to Linear Algebra and its Applications.
Pothen, A., Simon, H. D. and Wang, L.: 1992, Spectral nested dissection, *Technical Report 92-01*, Computer Science Dept., Pennsylvania State University.
Raw, M.: 1994, Coupled algebraic multigrid for the solution of the discretized 3D Navier-Stokes equations, *in* J. J. Gottlieb and C. R. Ethier (eds), *Proceedings of CFD 94*, CFD Society of Canada, pp. 335–344.
Rose, D.: 1972, A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations, *in* R. C. Read (ed.), *Graph Theory and Computing*, Academic Press, pp. 183–217.
Stevenson, R.: 1994, Modified ILU as a smoother, *Numerische Mathematik* **67**, 295–309.
Wallis, J. R.: 1983, Incomplete gaussian elimination as a preconditioning for generalized conjugate gradient acceleration, *Proceedings of the 1983 SPE Symposium on Reservoir Simulation in San Francisco.* SPE 12265.

A. Problem Definitions. Tables A.1 and A.2 define the problems used in this work by listing the grid size, block sizes and associated coefficient values, and the "background" coefficient values that cover the rest of the field. Source terms not given since these do not substantially affect the ordering processes used in this work.

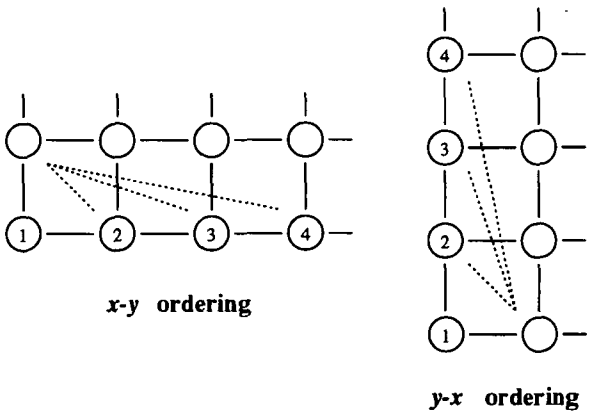

FIG. 9.1. *The fill resulting from elimination with natural $x-y$ and $y-x$ ordering.*

1. COMPUTE B_l.
2. COMPUTE THE SECOND LOWEST EIGENVALUE λ_2 OF B_l AND THE ASSOCIATED EIGENVECTOR E_2.
3. CONSTRUCT POINTER ARRAY `order` SUCH THAT $\texttt{order}(i) = i$ FOR ALL $i = 1 \ldots n$.
4. SORT THE `order` ARRAY SUCH THAT $\{E_2(\texttt{order}(i)) \mid i = 1 \ldots n\}$ IS IN ASCENDING ORDER.

FIG. 9.2. *Spectral ordering algorithm.*

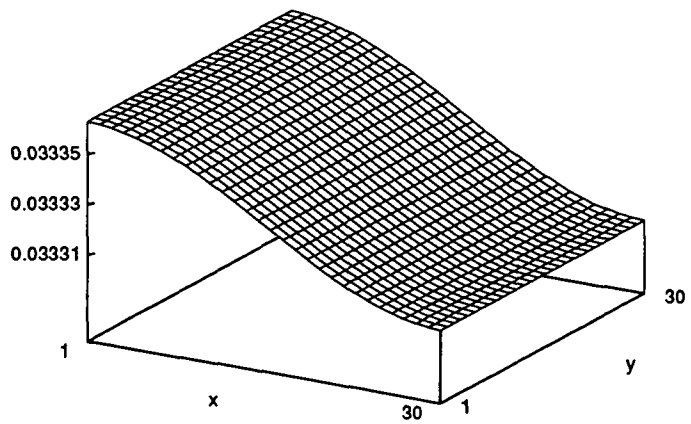

FIG. 9.3. *Second weighted pseudo-Laplacian eigenvector for a simple anisotropic problem.*

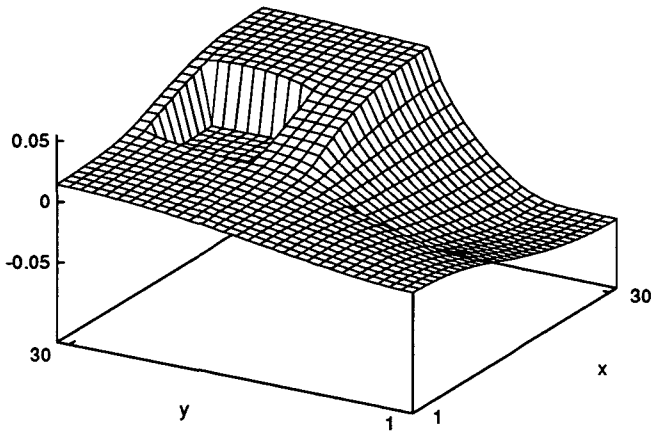

FIG. 9.4. *Second weighted pseudo-Laplacian eigenvector for the STONE problem.*

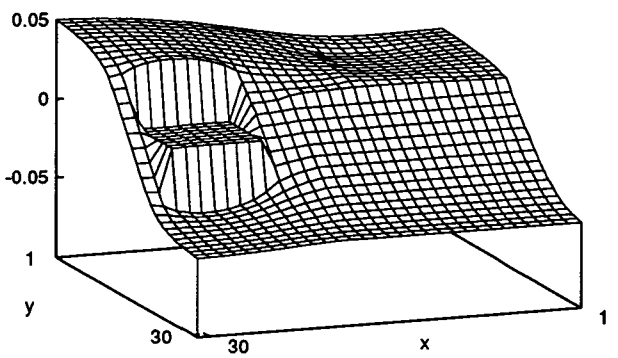

FIG. 9.5. *Second weighted pseudo-Laplacian eigenvector for the STONEROT90 problem.*

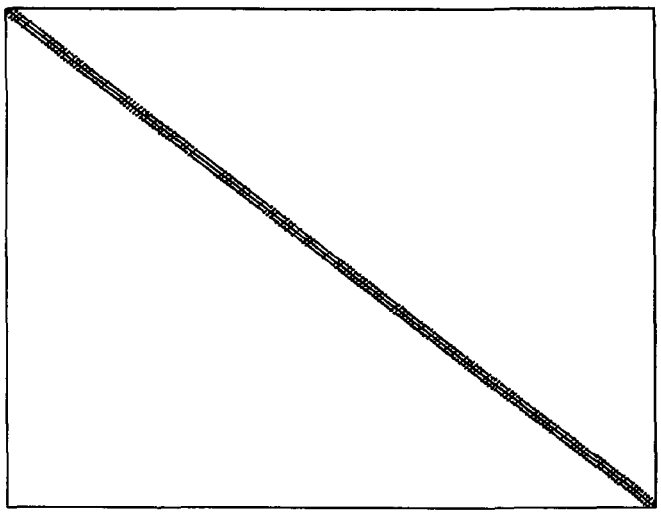

FIG. 9.6. *Sparsity pattern for the LONGTHIN with spectral ordering*

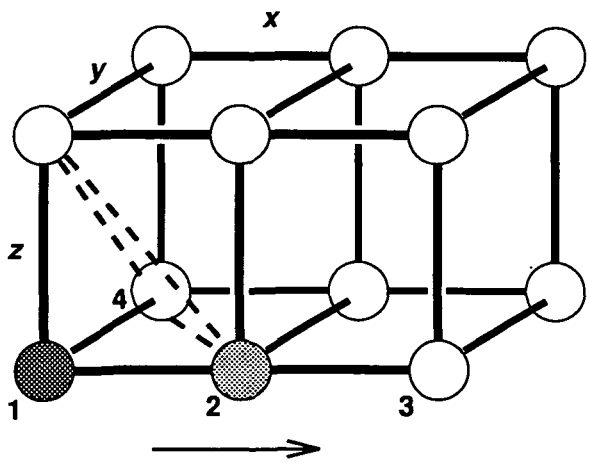

FIG. 9.7. *Elimination of a 3D grid in the x direction first. Dashed lines indicate the first level of fill generated by the elimination of the node 1.*

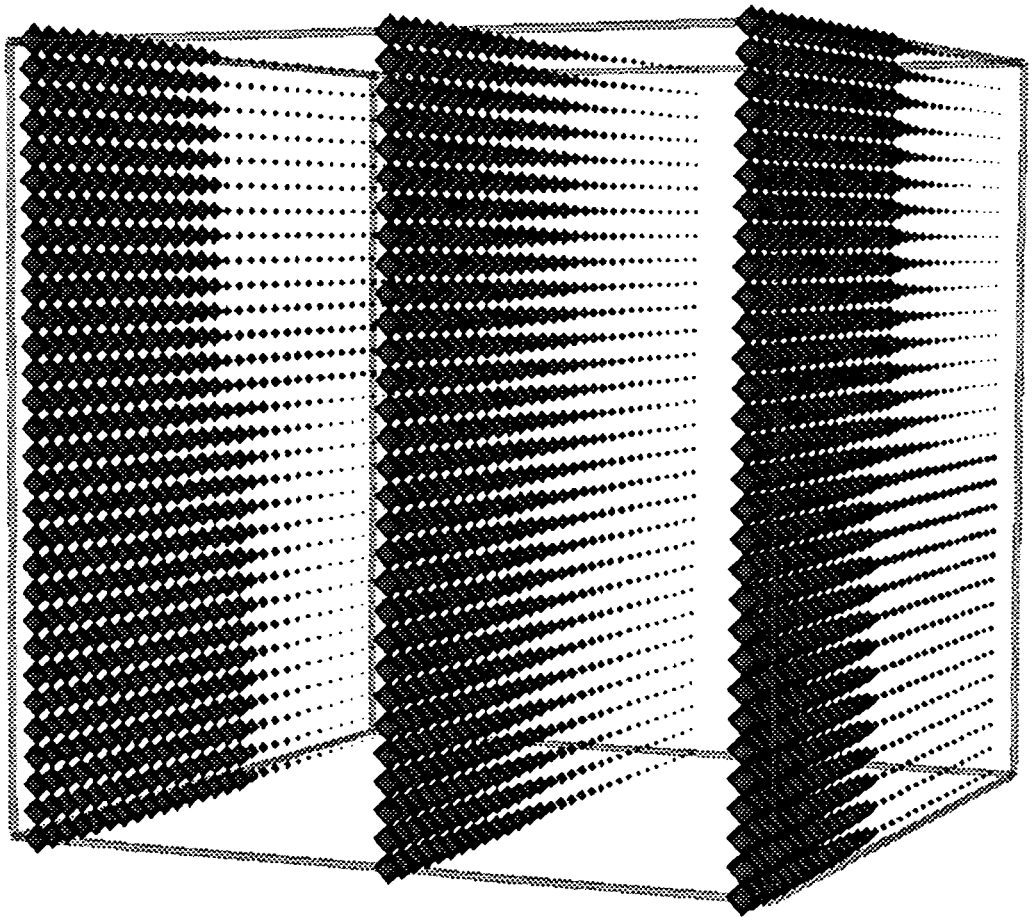

FIG. 9.8. *Eigenvalue field for ANISO3D. Smaller icons denote smaller values for the eigenvector at that point.*

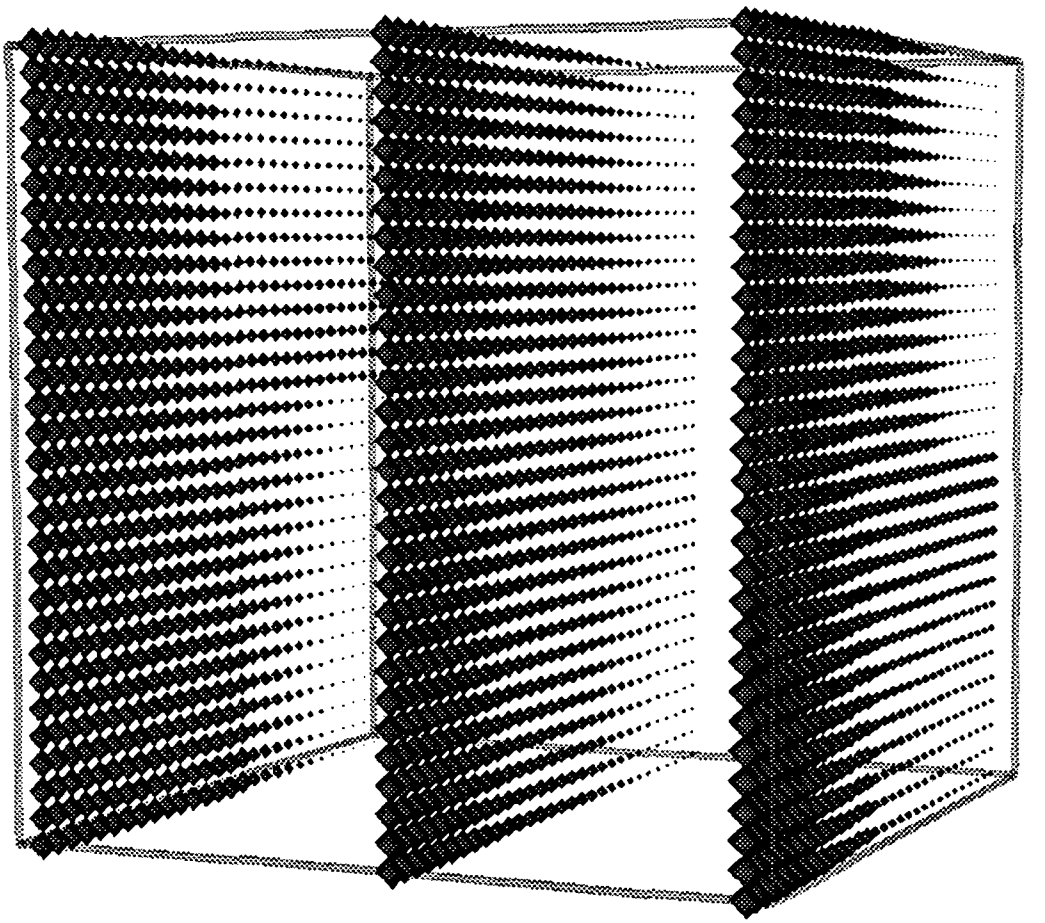

FIG. 9.9. *Ordering vector for ANISO3D. Smaller icons denote earlier elements in the ordering.*

	Ordering Method			
Problem	RCM	MDF	MUM	Spectral
ANISO	3.97e+05	0.70	1.26	0.63
BIG1DIR	3.86e+05	0.41	0.95	0.47
ANISOCENT	9.36e+05	0.95	1.06	1.00
EXTREMEANI	9.74e+05	0.69	1.19	0.81
LAPD5	3.76e+05	0.93	1.00	1.03
LONGTHIN	2.27e+04	1.12	1.00	1.03
STONE	4.24e+05	0.85	1.14	1.33
STONEROT90	4.24e+05	0.85	1.16	0.90
VDVORST	4.13e+05	0.33	1.00	0.53

TABLE 9.1

Work required to perform 2D regular grid problem iteration at ILU(1). The RCM column shows actual work required. The other columns show work as a fraction relative to the RCM column.

Element Weights in Each Ordering Direction			
First	Second	Third	$\sum$(discarded fill elements)
1	1	1000	$\approx 4 \times 10^3$
1	1000	1	$\approx 1 \times 10^6$
1000	1	1	$\approx 2 \times 10^6$
1	1000	1000	$\approx 2 \times 10^6$
1000	1	1000	$\approx 3 \times 10^6$
1000	1000	1	$\approx 4 \times 10^6$

TABLE 9.2

Sum of the magnitude of fill discarded in an ILU(1) factorization of a set of regular grid problems with the given weights in each ordering direction. Grid size 30 x 30 x 30

K_x, K_y, K_z	Ordering directions (first, second, third)					
Weights	x,y,z	x,z,y	y,x,z	y,z,x	z,x,y	z,y,x
(1,100,1000)	145	260	145	146	260	260
(1000,1,1)	139	139	120	60	121	60
(1000,1000,1)	301	302	298	299	302	299

TABLE 9.3

Solver iterations required when natural orderings are applied to uniformly anisotropic 3D problems. Grid size 30 x 30 x 30, *solved with ILU(1) preconditioner.*

K_x, K_y, K_z Weights	Ordering directions (first, second, third)					
	x, y, z	x, z, y	y, x, z	y, z, x	z, x, y	z, y, x
(1,100,1000)	6542	20565	9448	205626	21675	205627
(1000,1,1)	205740	205740	205626	9451	205626	9451
(1000,1000,1)	205626	9448	205626	9448	6510	6510

TABLE 9.4
Inverse-weight Laplacian two-sum for natural orderings applied to uniformly anisotropic 3D problems. Grid size 30 × 30 × 30.

Problem	Ordering Method			
	RCM	MDF	MUM	Spectral
ANISO3D	4.82e+07	0.82	1.26	1.54
ANISO3E	5.19e+07	0.89	0.98	1.02
ANISO3F	5.70e+07	0.87	1.48	1.59
BIG1DIR3D	7.52e+07	0.97	1.73	1.02
BIG1DIR3E	7.52e+07	0.97	1.73	1.42
BIG1DIR3F	1.35e+08	0.55	0.96	0.57
BIG1DIR3G	7.21e+07	0.53	0.98	0.44
BIG1DIR3H	1.56e+08	0.88	0.98	1.00
LAP7D	3.53e+07	1.01	0.99	1.00
STONE3D	6.42e+07	0.85	0.94	0.91
STONE3E	6.70e+07	0.87	0.98	1.02
STONE3F	4.58e+07	0.90	1.20	1.05

TABLE 9.5
Work required to perform the solution iteration on 3D regular grid problems at ILU(1). The RCM column shows actual work required. The other columns show work as a fraction relative to the RCM column.

Problem	Grid	Block Range	K_x	K_y
ANISO	30 × 30	(1, 1)→(15,15)	100	1
		(16,16)→(30,30)	100	1
		Background	1	100
BIG1DIR	30 × 30	Everywhere	1000	1
ANISOCENT	40 × 40	(11,11)→(20,20)	1	100
		(11,21)→(21,30)	100	1
		Background	1	1
EXTREMEANI	40 × 40	(10,31)→(40,40)	1	1000
		(10,11)→(40,30)	1000	1
		Background	2	1
LAPD5	30 × 30	Everywhere	1	1
LONGTHIN	200 × 10	Everywhere	1000	1
STONE	31 × 31	(15, 1)→(31,17)	1	100
		(6, 6)→(13,13)	100	1
		(13,22)→(20,29)	0	0
		Background	1	1
STONEROT90	31 × 31	(1,15)→(17,31)	1	100
		(6, 6)→(13,13)	100	1
		(22,13)→(29,20)	0	0
		Background	1	1
VDVORST	41 × 41	(11,30)→(11,30)	100	0.1
		Background	1	0.0001

TABLE A.1
Definitions for 2D regular grid problems used in this work

Problem	Grid	Block Range	K_x	K_y	K_z
ANISO3D	30 × 30 × 30	(1, 1, 1)→(15,15,15)	100	1	1
		(16,16, 1)→(30,30,16)	100	1	1
		(16, 1,16)→(30,15,30)	1	100	1
		(1,16,16)→(15,30,30)	1	100	1
		Background	1	1	100
ANISO3E	30 × 30 × 30	(0, 0, 0)→(15,15,15)	1	100	100
		(16,16, 0)→(30,30,16)	1	100	100
		(16, 1,16)→(30,15,30)	100	1	100
		(1,16,16)→(15,30,30)	100	1	100
		Background	100	100	1
ANISO3F	30 × 30 × 30	(0, 0, 0)→(15,15,15)	1	100	1000
		(16,16, 0)→(30,30,16)	1	100	1000
		(16, 1,16)→(30,15,30)	1000	100	1
		(1,16,16)→(15,30,30)	1000	100	1
		Background	100	1000	1
BIG1DIR3D	30 × 30 × 30	Everywhere	1	100	1000
BIG1DIR3E	30 × 30 × 30	Everywhere	100	1	1000
BIG1DIR3F	30 × 30 × 30	Everywhere	1000	100	1
BIG1DIR3G	30 × 30 × 30	Everywhere	1000	1	1
BIG1DIR3H	30 × 30 × 30	Everywhere	1000	1000	1
LAP7D	30 × 30 × 30	Everywhere	1	1	1
STONE3D	31 × 31 × 31	(15, 1, 1)→(31,17,17)	1	100	100
		(6, 6, 6)→(13,13,13)	100	1	1
		(13,22,22)→(20,29,29)	0	0	0
		Background	1	1	1
STONE3E	31 × 31 × 31	(15, 1, 1)→(31,17,17)	1	100	100
		(6, 6, 6)→(13,13,13)	100	100	1
		(13,22,22)→(20,29,29)	0	0	0
		Background	1	1	1
STONE3E	31 × 31 × 31	(15, 1, 1)→(31,17,17)	1	1	100
		(6, 6, 6)→(13,13,13)	100	1	1
		(13,22,22)→(20,29,29)	0	0	0
		Background	1	1	1

TABLE A.2
Definitions for 3D regular grid problems used in this work

RIACS
Mail Stop T041-5
NASA Ames Research Center
Moffett Field, CA 94035