CoNF-9104189--1

Two Variants of Minimum Discarded Fill Ordering*

P. A. Forsyth
and
W. P. Tang
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

E. F. D'Azevedo
Oak Ridge National Laboratory
Mathematical Sciences Section
P.O. Box 2009, Building 9207A
Oak Ridge, Tennessee 37831-8083

CONF-9104189--1

DE91 007852

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

*This work was supported by the Natural Sciences and Engineering Research Council of Canada, by the Information Technology Research Centre, which is funded by the Province of Ontario, and by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy under contract DE-AC05-840R21400 with Martin Marietta Energy Systems, Inc. through an appointment to the U.S. Department of Energy Postgraduate Research Program administered by Oak Ridge Associated Universities.

TWO VARIANTS OF MINIMUM DISCARDED FILL ORDERING [*]

E. F. D'AZEVEDO [†], P. A. FORSYTH[‡] AND WEI-PAI TANG [‡]

Abstract. It is well known that the ordering of the unknowns can have a significant effect on the convergence of Preconditioned Conjugate Gradient (PCG) methods. There has been considerable experimental work on the effects of ordering for regular finite difference problems. In many cases, good results have been obtained with preconditioners based on diagonal, spiral or natural row orderings. However, for finite element problems having unstructured grids or grids generated by a local refinement approach, it is difficult to define many of the orderings used for more regular problems. A recently proposed Minimum Discarded Fill (MDF) ordering technique is effective in finding high quality Incomplete LU (ILU) preconditioners, especially for problems arising from unstructured finite element grids. Testing indicates this algorithm can identify a rather complicated physical structure in an anisotropic problem and orders the unknowns in the "preferred" direction. The MDF technique may be viewed as the numerical analogue of the minimum deficiency algorithm in sparse matrix technology. At any stage of the partial elimination, the MDF technique chooses the next pivot node so as to minimize the amount of discarded fill, and hence the MDF ordering is expensive to compute for high-level ILU preconditioners.

In this work, two efficient variants of the MDF technique are explored to produce cost-effective high-order ILU preconditioners. The Threshold MDF orderings combine MDF ideas with drop tolerance techniques to identify the sparsity pattern in the ILU preconditioners. These techniques identify an ordering that encourages fast decay of the entries in the ILU factorization. The Minimum Update Matrix (MUM) ordering technique is a simplification of the MDF ordering and is closely related to the minimum degree algorithm. The MUM ordering is especially effective for large problems arising from Navier-Stokes problems. Some interesting pictures of the orderings are presented using a visualization tool developed at the University of Waterloo.

1. Introduction. There has been much interest in the effects of ordering on ILU preconditioners [3], [6], [8], [9], [14]. In this work, two less expensive variants of the Minimum Discarded Fill ordering (MDF) [3] for use with drop tolerance incomplete factorization techniques are explored to produce cost-effective high order ILU preconditioners.

Munksgaard [13] and Zlatev [22] have considered the technique of generating ILU preconditioners by discarding "small" fill entries that are less than a given tolerance ε. The approach works well for some problems but is sensitive to the initial ordering of the matrix for strongly anisotropic problems. To see this, consider the following anisotropic problem

$$KU_{xx} + U_{yy} = f(x,y)\ , \qquad (x,y) \in (0,1) \times (0,1) \tag{1}$$

with Neuman boundary conditions, $K = 100$ and discretized on a 30×30 regular grid using the five point molecule with h as the grid size. The right hand side $f(x,y)$ was defined as

$$f(x,y) = \begin{cases} 1 & \text{if } (x,y) = (h,h), \\ -1 & \text{if } (x,y) = (1-h, 1-h), \\ 0 & \text{elsewhere} \end{cases} .$$

[*] This work was supported by the Natural Sciences and Engineering Research Council of Canada; by the Information Technology Research Centre, which is funded by the Province of Ontario; and by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc., through an appointment to the U.S. Department of Energy Postgraduate Research Program administered by Oak Ridge Associated Universities.

[†] Mathematical Sciences Section, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831 (efdazedo@msr.epm.ornl.gov).

[‡] Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1.

TABLE 1

Factorization and solution times for anisotropic problem with drop tolerance ILU preconditioning.

Drop tolerance preconditioning, $\varepsilon = 0.001$				
	Fact.	Fill	Iteration	
Ordering	time	in L	$\tau = 10^{-6}$	$\tau = 10^{-12}$
x–y	0.15	10242	0.19(7)	0.27(23)
y–x	0.04	2705	0.11(13)	0.15(18)

The resulting matrix from (1) was scaled to unit diagonal and solved with a preconditioned conjugate gradient (PCG) method using an ILU preconditioner obtained from a drop tolerance strategy with $\varepsilon = 0.001$. The zero initial guess was used and the convergence tolerance was a specific reduction τ in the l_2 norm of the residual. Table 1 shows the effects of the natural x–y ordering and natural y–x ordering on the (incomplete) factorization time, number of off-diagonals in the ILU L factor, solution time and iteration counts (in brackets).

Close examination of the entries in the ILU factor L reveals a slow decay if the natural x–y ordering was used (nodes in the x-direction numbered first). If the natural y–x ordering was used, an effective preconditioner can be constructed using the drop tolerance technique.

In this work, our experiments show drop tolerance incomplete factorization with the threshold MDF strategy is effective in identifying anisotropy in complicated physical structures and orders the unknowns in a "preferred" direction. This threshold MDF technique may be viewed as a numerical analogue to the minimum deficiency algorithm [18]. A rough estimate of the time complexity of threshold MDF algorithm is $O(nd^3)$ where d is the average number of nonzero entries in each row of the fill matrix $L + U$ and n is the size of the original matrix. The threshold MDF technique is most practical when similar matrix problems need to be solved repeatedly, such as solving the Jacobian matrices in Newton iterations [2], [5].

There are matrices arising from three-dimensional finite element analysis that have a large connectivity initially, making the threshold MDF ordering too costly to compute. For such problems, we propose a more efficient *minimum update matrix* (MUM) ordering that is analogous to the minimum degree ordering [11]. Only the norm of the updating matrix (without consideration of actual fill-in entries) is examined in the ordering process.

The organization of this paper is as follows. In §2, we introduce our notation for describing the drop tolerance incomplete factorization. In §3, we provide additional insight into why the x–y ordering performs poorly when solving (1). A description of the threshold MDF algorithm is contained in §4. A similar presentation for the MUM algorithm is given in §5. We describe our test problems in §6, and results are presented in §7.

2. Drop tolerance incomplete factorization. The LDU factorization of an $n \times n$ matrix A can be described by the following equations:

$$A = A_0 = \begin{bmatrix} d_1 & \beta_1^t \\ \alpha_1 & B_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \alpha_1 & I_{n-1} \end{bmatrix} \begin{bmatrix} d_1 & 0 \\ 0 & A_1 \end{bmatrix} \begin{bmatrix} 1 & \beta_1^t \\ 0 & I_{n-1} \end{bmatrix}$$

where

$$A_1 = B_1 - U_1 \ , \qquad U_1 = \alpha_1 \beta_1^t / d_1 \ .$$

At the k^{th} step,

$$A_{k-1} = \begin{bmatrix} d_k & \beta_k^t \\ \alpha_k & B_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \alpha_k & I_{n-k} \end{bmatrix} \begin{bmatrix} d_k & 0 \\ 0 & A_k \end{bmatrix} \begin{bmatrix} 1 & \beta_k^t \\ 0 & I_{n-k} \end{bmatrix}$$

where

$$A_k = B_k - U_k \ , \qquad U_k = \alpha_k \beta_k^t / d_k \ .$$

Here I_{n-k} denotes the $(n-k) \times (n-k)$ identity, d_k a scalar, α_k and β_k are column vectors. The matrix A_k is the $(n-k) \times (n-k)$ submatrix that remains to be factored.

In the drop tolerance incomplete factorization of matrix A, "small" entries in the factor are discarded to prevent excessive fill and computation. Let F_k contain the dropped entries. Then the incomplete factorization proceeds with the perturbed matrix

$$\tilde{A}_k = A_k - F_k = B_k - U_k - F_k \ . \tag{2}$$

In this work, we have used the criterion

$$|u_{ij}^{(k)}| \le \varepsilon \min(R_i, R_j) \ , \qquad R_i = \|a_{i*}\|_\infty = \max_{m=1,\cdots,n} |a_{im}|$$

for discarding small entries in matrix U_k.

3. Anisotropic problem. With Theorem 3.1 (see [4] and [12] for proof), we shall see why the natural x–y ordering performed poorly with the anisotropic problem (1).

If A is symmetric, the fill entries in factor L can be conveniently described through a graph model [15], [18]. Let the elimination sequence be $v_1, \cdots, v_n$ and $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$ be the graph of $A_k = \left[a_{ij}^{(k)}\right]$,

$$\mathcal{V}_k = \{v_{k+1}, \cdots, v_n\} \ , \qquad \mathcal{E}_k = \left\{(v_i, v_j) \mid a_{ij}^{(k)} \ne 0\right\} \ .$$

It can be shown [10] that, there is a nonzero entry l_{ij} if and only if there exists a path $(v_i, v_{i_1}, \cdots, v_{i_m}, v_j)$ in the graph of A where

$$v_{i_1}, \cdots, v_{i_m} \in \{v_1, \cdots, v_{j-1}\} \ .$$

The size of l_{ij} is related to the size of entries on this path.

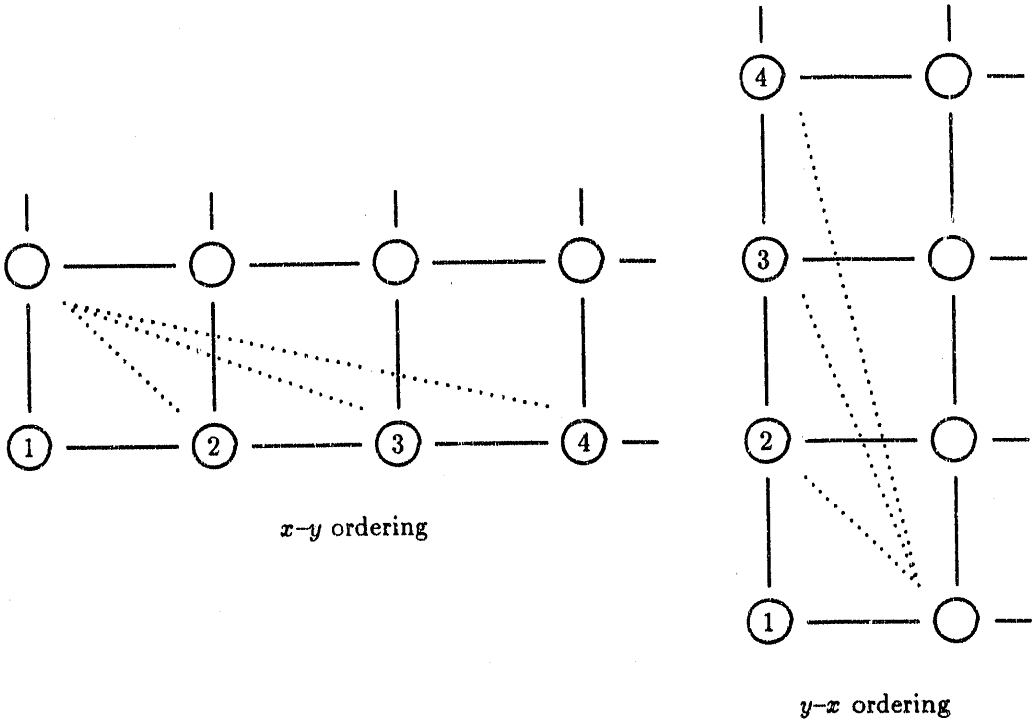
THEOREM 3.1. *Let A be an M-matrix and let $(v_i, v_{i_1}, \cdots, v_{i_m}, v_j)$ be a path in the graph of A where*

$$v_{i_1}, \cdots, v_{i_m} \in \{v_1, \cdots, v_{j-1}\} \ ,$$

then for $i > j$

$$|l_{ij}| \ge \frac{|a_{ii_1} a_{i_1 i_2} \cdots a_{i_m j}|}{d_{i_1} d_{i_2} \cdots d_{i_m} d_j} \ , \qquad d_k = a_{kk} \ .$$

For the anisotropic problem (1), the resulting matrix is a symmetric M-matrix. All edges aligned along the x-axis (y-axis) have values $O(K/(K+1))$ ($O(1/K)$). If the natural x–y row order is used, all new fill entries will have the "x-orientation" (see Fig. 1). From the lower bound in Theorem 3.1, the new fill entries will have a slow decay rate. Conversely, if the natural y–x ordering is used, all fill entries will have the "y-orientation". The resulting ILU decomposition will be much sparser.



x–y ordering

y–x ordering

FIG. 1. *Orientation of new fill entries in x–y and y–x orderings. New fill entries indicated by $\cdots$.*

4. Threshold MDF. The minimum discarded fill strategy is based on the observation that a small discarded fill matrix F_k in (2)

$$\tilde{A}_k = A_k - F_k = B_k - U_k - F_k$$

would produce a more "authentic" factorization for matrix A. We associate the Frobenius norm of discarded fill matrix F_k to the discard value for node v_k as

$$discard(v_k) = \|F_k\|_F = \left(\sum_{i \geq 1} \sum_{j \geq 1} F_{ij}^2 \right)^{1/2} ,$$

where

$$F_{ij} = \begin{cases} u_{ij} & \text{if } |u_{ij}| \leq \varepsilon \min(R_i, R_j), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The basic idea of the threshold MDF scheme is to eliminate the node with the minimum discard value at each stage of the incomplete factorization process. Note that at each step, if v_k is chosen to be eliminated, only discard values for neighbors of v_k need to be updated [18].

The following is a description of the threshold MDF algorithm:

```
$A_0 := A$
for each node $v_i$
    Compute the discard value $discard(v_i)$
end
for $k = 1 \ldots n-1$
    Choose $v_j$ with minimum discard value $discard(v_j)$ in matrix $A_{k-1}$ (break ties arbitrarily).
    Update the decomposition,
        $U_k := \alpha_k \beta_k^t / d_k$
        $A_k := B_k - U_k - F_k$ ,    where $P_k A_{k-1} P_k^t = \begin{bmatrix} d_k & \beta_k^t \\ \alpha_k & B_k \end{bmatrix}$ ,
     $F_k$ defined by (3) and $P_k$ permutes row $v_j$ to first position.
    Update the discard values of uneliminated neighbors of $v_j$.
end
```

5. MUM ordering. The most costly part of computing a threshold MDF ordering is the traversal of adjacency lists in determining which u_{ij}'s are discarded while updating the discard values. If the average node degree is high, the threshold MDF ordering is quite expensive. We propose a simpler scheme, the *minimum update matrix* (MUM) ordering that is motivated by the minimum degree ordering [11]. The computation of the discard value is

$$discard(v_k) = \|\tilde{U}_k\|_F \ , \qquad \tilde{u}_{ij} = \begin{cases} u_{ij} & \text{if } i \neq j, \\ 0 & \text{otherwise} \end{cases}$$

where the matrix $\tilde{U}_k$ contains the off-diagonal entries of U_k.

Note that threshold MDF ordering has discard value computations intimately coupled to the incomplete factorization strategy. The discard values are the norm of the *actual* discarded fill matrices. The MUM ordering uses a simpler (less precise) measure for the discard value and is less coupled to the factorization strategy used.

For anisotropic problems with small computation molecules the MUM ordering may fail to identify the "preferred" direction. Examination of the ordering process for the anisotropic problem (1) shows both the x–y and y–x orientations have updating matrices with the same norm. Without considering the actual discarded fill entries, the "preferred" direction cannot be identified. However, we find the MUM ordering works quite well if the average node degree is high (a large computation molecule) and the decay rate in LU decomposition is less sensitive to the anisotropy in the problem.

6. Test problems. The following problems are derived from solving convection-diffusion equations and linearized Navier-Stokes equations.

Problem 1 (STONE). This is Stone's third problem [19]. The equation

$$\frac{\partial}{\partial x}\left(K_x \frac{\partial P}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial P}{\partial y}\right) = -q \tag{4}$$

was discretized on the unit square with a finite difference technique (see Fig. 2). A 33×33 grid was used, and a harmonic average was used for discontinuities in K_x and K_y [1]. Denoting x_i and y_i for $i = 1, \cdots, 33$ to be the grid nodes on the x-axis and

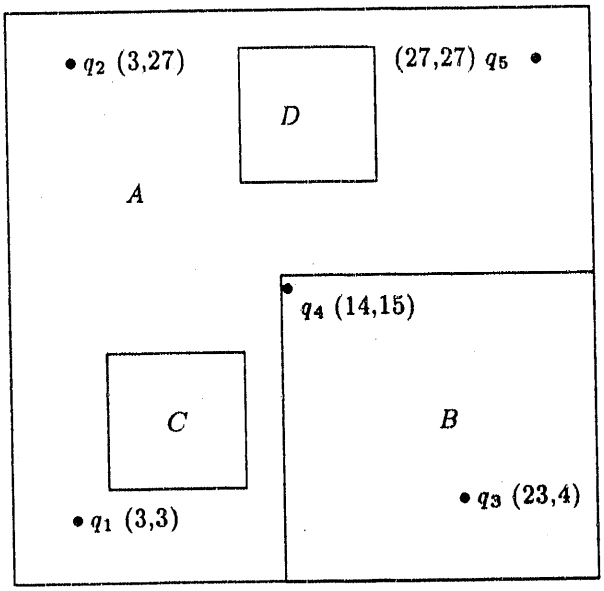


FIG. 2. *Stone's third problem.*

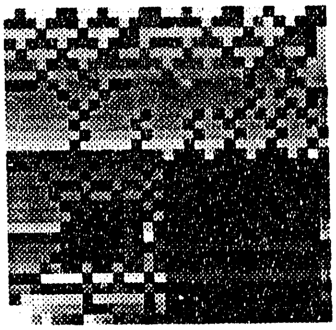


FIG. 3. *Threshold MDF ordering with* ϵ = 0.001 *for Stone's third problem. Lightest, first; darkest, last.*

y-axis respectively, the values of K_x, K_y and q are:

$$(K_x, K_y) = \begin{cases} (1,100) & \text{if } (x,y) \in B, \quad 14 \le x \le 30,\ 0 \le y \le 16 \\ (100,1) & \text{if } (x,y) \in C, \quad 5 \le x \le 12,\ 5 \le y \le 12 \\ (0,0) & \text{if } (x,y) \in D, \quad 12 \le x \le 19,\ 21 \le y \le 28 \\ (1,1) & \text{if } (x,y) \in A, \quad \text{elsewhere} \end{cases},$$

$$q_1(3,3) = 1.0, \quad q_2(3,27) = 0.5, \quad q_3(23,4) = 0.6,$$
$$q_4(14,15) = -1.83, \quad q_5(27,27) = -0.27\ .$$

Figure 3, which was produced by the visualization tool Matview [20], shows the threshold MDF ordering.

Problem 2 (ANISO). This problem has the same equation as in (4) except the

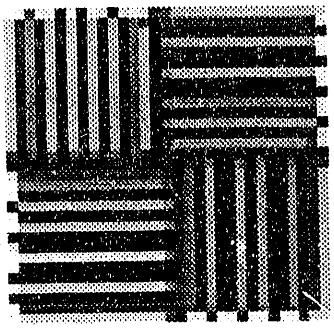


FIG. 4. *Threshold MDF ordering with $\varepsilon = 0.001$ for anisotropic problem (ANISO). Lightest, first; darkest, last.*

value distributions of K_x and K_y are different:

$$(K_x, K_y) = \begin{cases} (1,100) & \text{if } 0 \leq x \leq 1/2, \quad 0 \leq y \leq 1/2 \\ (100,1) & \text{if } 1/2 \leq x \leq 1, \quad 0 \leq y \leq 1/2 \\ (1,100) & \text{if } 1/2 \leq x \leq 1, \quad 1/2 \leq y \leq 1 \\ (100,1) & \text{if } 0 \leq x \leq 1/2, \quad 1/2 \leq y \leq 1 \end{cases},$$

A 30×30 grid was used. Figure 4 shows the threshold MDF ordering for this problem.

Provlems 3,4 (LNS511, LNSP511). LNS511 and LNSP511 are unsymmetric matrices of order 511, arising in the solution of linearized Navier-Stokes equations using velocity-pressure formulation. They were extracted from the Harwell-Boeing collection of test matrices [7]. LNS511 corresponds to an ordering by variable type with velocity variables, preceding temperature, preceding pressure variables. The discretization scheme resulted in zero diagonals in rows corresponding to the pressure equations. LNSP511 is LNS511 permuted so that all variables at the same grid point are grouped together.

Problem 5 (NS2D). This problem was derived from a finite volume discretization of the incompressible Navier-Stokes equations [16]. A 40×40 grid was used to model the backward step problem, with a Reynold's number $R_e = 1000$. The matrix was generated from the Jacobian produced for the second Newton iteration.

Problem 6 (NS3D). This problem was derived from a small three dimensional finite element discretization of the incompressible Navier-Stokes equations [17]. Use of a finite element method in three dimensions resulted in a very large computational molecule. The test matrix was generated from a Jacobian produced near the start of a pseudo-time solution of the steady-state problem.

7. Numerical Results. The computations to solve the test problems were carried out on an IBM RISC/6000 workstation. The initial vector $\mathbf{x}^0$ was chosen to be the zero vector and the convergence criteria were

$$||\mathbf{r}^k||_2 \leq \tau ||\mathbf{r}^0||_2 , \qquad \tau = 10^{-6}, 10^{-12},$$

where $\mathbf{r}^k$ was the residual vector after the k^{th} iteration. CGSTAB [21] acceleration was used for the unsymmetric Navier-Stokes problems. The best value for the drop tolerance threshold ε is obviously problem dependent, but we have found $10^{-2} \leq \varepsilon \leq 10^{-4}$ to be quite effective.

Tables 2–7 show the ordering times, amount of fill, factorization times (in seconds), the acceleration times and the number of iterations (in brackets). ORG(1d-3),

TABLE 2
Results for Stone's third problem.

STONE, order 961, 4681 entries						
Ordering & Threshold	Ordering Time	Nonzeros in L	Nonzeros in U	Fact. Time	Iteration	
					$\tau = 10^{-6}$	$\tau = 10^{-12}$
ORG(1d-2)	N/A	3417	3417	0.06	0.28(31)	0.35(40)
ORG(1d-3)	N/A	6695	6695	0.09	0.14(13)	0.20(19)
ORG(1d-4)	N/A	11262	11262	0.16	0.07(5)	0.12(9)
MDF(1d-2)	0.67	4090	4090	0.06	0.14(14)	0.19(20)
MDF(1d-3)	2.04	6666	6666	0.09	0.07(7)	0.12(11)
MDF(1d-4)	5.06	9094	9094	0.13	0.06(5)	0.09(7)

TABLE 3
Results for ANISO.

ANISO, order 900, 4681 entries						
Ordering & Threshold	Ordering Time	Nonzeros in L	Nonzeros in U	Fact. Time	Iteration	
					$\tau = 10^{-6}$	$\tau = 10^{-12}$
ORG(1d-2)	N/A	1967	1967	0.04	0.34(40)	0.61(73)
ORG(1d-3)	N/A	6234	6234	0.10	0.18(17)	0.37(36)
ORG(1d-4)	N/A	10955	10955	0.16	0.15(12)	0.33(27)
MDF(1d-2)	0.30	2697	2697	0.05	0.36(40)	0.64(72)
MDF(1d-3)	0.78	4872	4872	0.07	0.09(9)	0.20(20)
MDF(1d-4)	1.21	5864	5864	0.08	0.05(5)	0.13(13)

MUM(1d-3), and MDF(1d-3) denote the original ordering, the MUM ordering and threshold MDF ordering respectively with drop tolerance factorization where $\varepsilon = 0.001$.

Without consideration of actual fill entries, the MUM ordering failed to detect the "preferred" direction for STONE and ANISO. Results of MUM are similar to those of the original ordering and are omitted. Threshold MDF showed improvement in the amount of fill and faster convergence over the original ordering (Tables 2, 3).

Since ties were broken arbitrarily in MUM and threshold MDF, results in LNS511 and LNSP511 were similar but not identical. Tables 4 and 5 indicate threshold MDF ordering yielded better preconditioners but were more costly to produce than MUM ordering.

For problems with large computation molecules, MUM is much less costly than threshold MDF and still produce substantial improvements (Tables 6, 7). Ordering times for threshold MDF for problems NS2D were over 600 seconds and are not presented.

To prevent excessive fill and computation for the three-dimensional problem NS3D, only ILU (level 0) factorization with no new fill was used. The original ordering did not converge, while MUM ordering (surprisingly) performed even better than MDF ordering (Table 7).

To summarize, our numerical tests show the drop tolerance incomplete factorization with threshold MDF ordering to be effective for problems with small computation molecules, while MUM ordering works well for problems with large molecules.

TABLE 4
Results for LNS511.

LNS511, order 511, 2796 entries						
Ordering & Threshold	Ordering Time	Nonzeros in L	Nonzeros in U	Fact. Time	Iteration	
					$\tau = 10^{-6}$	$\tau = 10^{-12}$
ORG(1d-2)	N/A	12116	6878	0.17	1.03(74)	1.21(88)
ORG(1d-3)	N/A	15589	9073	0.27	0.41(24)	0.47(28)
ORG(1d-4)	N/A	19748	13179	0.48	0.33(16)	0.41(20)
MUM(1d-2)	0.69	3944	4245	0.08	0.28(31)	0.36(40)
MUM(1d-3)	1.05	5493	5060	0.10	0.24(24)	0.27(27)
MUM(1d-4)	1.89	7255	7133	0.17	0.16(14)	0.22(19)
MDF(1d-2)	5.48	4616	4467	0.08	0.19(18)	0.21(23)
MDF(1d-3)	8.44	5121	5456	0.10	0.09(9)	0.11(11)
MDF(1d-4)	8.02	5224	5260	0.09	0.03(3)	0.06(5)

TABLE 5
Results for LNSP511.

LNSP511, order 511, 2796 entries						
Ordering & Threshold	Ordering Time	Nonzeros in L	Nonzeros in U	Fact. Time	Iteration	
					$\tau = 10^{-6}$	$\tau = 10^{-12}$
ORG(1d-2)	N/A	5039	4855	0.08	0.54(55)	0.77(80)
ORG(1d-3)	N/A	5690	5791	0.08	0.25(22)	0.31(30)
ORG(1d-4)	N/A	6071	6507	0.10	0.10(9)	0.12(12)
MUM(1d-2)	0.76	4191	4414	0.07	0.23(23)	0.26(29)
MUM(1d-3)	1.10	5709	5139	0.10	0.17(17)	0.23(23)
MUM(1d-4)	2.24	7800	7736	0.19	0.15(12)	0.19(15)
MDF(1d-2)	5.73	4737	4567	0.09	0.10(10)	0.19(20)
MDF(1d-3)	6.49	5013	5017	0.09	0.10(10)	0.13(13)
MDF(1d-4)	8.08	5220	5257	0.10	0.03(3)	0.05(5)

TABLE 6
Results for NS2D.

NS2D, order 2369, 20619 entries						
Ordering & Threshold	Ordering Time	Nonzeros in L	Nonzeros in U	Fact. Time	Iteration	
					$\tau = 10^{-6}$	$\tau = 10^{-12}$
ORG(1d-2)	N/A	45588	45584	1.17	5.19(75)	7.27(105)
ORG(1d-3)	N/A	54418	54388	1.53	3.64(47)	5.17(67)
ORG(1d-4)	N/A	54947	54650	1.54	3.42(44)	4.38(57)
MUM(1d-2)	9.37	42894	51293	0.93	0.93(13)	1.26(18)
MUM(1d-3)	111.36	153461	140878	7.48	1.13(7)	1.61(10)

TABLE 7
Results for NS3D.

NS3D, order 684, 52930 entries						
Ordering & Threshold	Ordering Time	Nonzeros in L	Nonzeros in U	Fact. Time	Iteration	
					$\tau = 10^{-6}$	$\tau = 10^{-12}$
ORG	N/A	26123	26123	2.00	diverge	diverge
MUM	9.50	26123	26123	2.26	4.59(89)	6.24(121)
MDF	134.53	26123	26123	2.21	9.02(175)	12.65(245)

REFERENCES

[1] A. BEHIE AND P. A. FORSYTH, *Incomplete factorization methods for fully implicit simulation of enhanced oil recovery*, SIAM J. Sci. Statist. Comp., 5 (1984), pp. 543–561.

[2] P. CHIN, E. F. D'AZEVEDO, P. A. FORSYTH, AND W.-P. TANG, *Iterative methods for solution of full Newton Jacobians in incompressible viscous flow*, Tech. Report CS-91, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1991.

[3] E. F. D'AZEVEDO, P. A. FORSYTH, AND W.-P. TANG, *Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems*, in Copper Mountain Conference on Iterative Method, Copper Mountain, Colorado, 1990.

[4] ———, *Towards a cost-effective high order ILU preconditioner*, Tech. Report CS-90, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1990.

[5] ———, *An automatic ordering method for incomplete factorization iterative solvers*, in Proceedings of the 1991 Reservoir Simulation Symposium, Anaheim, 1991. Paper SPE 21226.

[6] S. DOI AND A. LICHNEWSKY, *A graph-theory approach for analysing the effects of ordering on ILU preconditioning*. (submitted to SIAM J. Sci. Statist. Comp.), 1990.

[7] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (89), pp. 1–14.

[8] I. S. DUFF AND G. A. MEURANT, *The effect of ordering on preconditioned conjugate gradients*, BIT, (1989), pp. 635–657.

[9] V. EIJKHOUT, *Analysis of parallel incomplete point factorizations*, Tech. Report CSRD Report No. 1045, Center for Supercomputing Research and Development, University of Illinois, Urbana, Illinois, 1990.

[10] A. GEORGE AND J. W. H. LIU, *Computer solution of large sparse positive-definite systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

[11] ———, *The evolution of the minimum degree ordering algorithm*, SIAM Rev., 31 (1989), pp. 1–19.

[12] J. A. MEIJERINK AND H. A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a M-matrix*, Math. Comp., 31 (1977), pp. 148–162.

[13] N. MUNKSGAARD, *Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients*, ACM Trans. Math. Software, 6 (1980), pp. 206–219.

[14] J. M. ORTEGA, *Orderings for conjugate gradient preconditionings*, Tech. Report TR-90-24, Department of Computer Science, University of Virginia, Charlottesville, Virginia, 1990.

[15] S. V. PARTER, *The use of linear graphs in Gaussian elimination*, SIAM Rev., 3 (1961), pp. 364–369.

[16] S. V. PATANKAR, *Numerical heat tranfer and fluid flow*, Hemisphere, New York, 1980.

[17] O. PIRONNEAU, *Finite element methods for fluids*, John Wiley & Sons, New York, 1989.

[18] D. ROSE, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, in Graph Theory and Computing, R. C. Read, ed., Academic Press, 1972, pp. 183–217.

[19] H. L. STONE, *Iterative solution of implicit approximations of multidimensional partial differential equations*, SIAM J. Num. Anal., 5 (1968), pp. 530–558.

[20] G. TOWNSEND AND W.-P. TANG, *Matview 1.1 — A two dimensional matrix visualization tool*, Tech. Report CS-89-36, Department of Computer Science, University of Waterloo, Waterloo, Ontario, 1989.

[21] H. A. VAN DER VORST AND P. SONNEVELD, *CGSTAB: A more smoothly converging variant of CG-S* , Tech. Report 90-50, Delft University of Technology, Delft, the Netherlands, 1990.

[22] Z. ZLATEV, *Use of iterative refinement in the solution of sparse linear systems*, SIAM J. Num. Anal., 19 (1982), pp. 381–399.