Direct Methods in Reservoir Simulation

H. S. PRICE
K. H. COATS
MEMBERS SPE-AIME

INTERCOMP RESOURCE DEVELOPMENT AND ENGINEERING, INC.
HOUSTON, TEX.

ABSTRACT

During the past decade, efforts in reservoir modeling have focused on the three areas of capability, efficiency, and reliability. Capability means the ability to handle larger and more complex problems where complexity includes physical phenomena, such as gas percolation and variable PVT properties, and severe heterogeneity due to property variation or geometry, or both. Efficiency is increased by improving model formulations and solution techniques to increase tolerable time-step size and reduce computer time per time step. Reliability refers to ease of use and minimum burden in selecting or experimenting with time-step size, solution technique options, iteration parameters, and closure tolerances.

The single facet of a reservoir simulator that has the greatest combined influence in all three categories is the technique used to solve the large systems of equations arising from the numerical approximation of the nonlinear fluid flow equations. Available techniques include both direct solution and iterative methods such as ADIP,[1,2] SOR,[3] and SIP.[4] Iterative methods are currently used almost to the exclusion of direct solution because of the significantly higher computer storage and time requirements of the latter.

This paper describes some new ordering schemes for Gaussian elimination that reduce computing time and storage requirements by factors as large as 6 and 3, respectively, relative to more standard orderings. Computational work estimates are given for these methods, for the standard Gaussian ordering, and for several iterative methods. These work estimates are checked by comparisons of actual run times using different solution techniques. Numerical examples are given to illustrate the increased efficiency and reliability that can be achieved in many cases through use of the new direct solution methods.

Original manuscript received in Society of Petroleum Engineers office Jan. 15, 1973. Revised manuscript received Nov. 14, 1973. Paper (SPE 4278) was first presented at the SPE-AIME Third Symposium on Numerical Simulation of Reservoir Performance, held in Houston, Jan. 10-12, 1973.

[1]References listed at end of paper.

This paper will be printed in *Transactions* volume 257, which will cover 1974.

INTRODUCTION

It is well known that the way we number or order the unknowns of a sparse system of linear algebraic equations can drastically affect the amount of computation and storage for a direct solution. However, until recently the best ordering scheme that appeared in the literature numbered the points of a three-dimensional grid first along the shortest direction — i.e., the dimension with the fewest number of grid points — then in the next shortest direction, and finally in the longest direction. This ordering, which we shall call the standard ordering for Gaussian elimination is still widely used even though it is substantially slower than many other orderings.

Ogbuobiri *et al.*[5] present a survey of the literature related to ordering schemes that exploit matrix sparsity. These schemes are grouped into the two classes of matrix-banding schemes[6] and optimal or pseudo-optimal schemes.[7] The latter schemes purport to yield generally greater efficiency.[5]

In a recent paper, George[8] has shown that for five-point difference approximations on square $n \times n$ two-dimensional grids, the total work for certain orderings of the grid points is $\sim C_1 n^3$ and the storage is $\sim C_2 n^2 \log n$, compared with n^4 and n^3, respectively, for the standard ordering. Moreover, George has shown that no ordering scheme can require less work than the order of n^3. For the special case of $n = 2^l$ he shows that work $W < 10n^3$ and the storage $S \sim 8ln^2$ for symmetric matrices. For nonsymmetric matrices these results become $W \sim 20n^3$ and $S \sim 16ln^2$, respectively.

In this paper we describe some specific orderings in the matrix-banding class. Analyses of work and storage requirements are given for these orderings as applied to the diffusivity-type pressure equation that arises in reservoir simulation problems. These work and storage requirements are compared with those of the standard Gaussian ordering and of some iterative methods. These comparisons are performed for problems ranging from simple homogeneous squares to practical reservoir problems of typical heterogeneity and irregular geometry.

The work requirements of the orderings presented here are also compared experimentally with those of one of the leading pseudo-optimal schemes. A

theoretical comparison is given with George's "optimal" scheme of order, n^3. Finally, comparisons are made regarding programming ease relative to George's scheme and the pseudo-optimal schemes.

All the comparisons are based on asymmetry of the A matrix since this is the case in general for the pressure equation arising in reservoir simulation problems. If symmetry were present and taken advantage of, the Scheme D4 computing times for large I, J, K would be nearly half those given below.

PROBLEM DEFINITION

In this paper we consider orderings for the set of equations

$$\underline{A}\underline{p} = \underline{b} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

which represents the finite-difference approximation for a single parabolic diffusivity-type equation written for all grid blocks in an $I \times J \times K$ rectangular mesh. The form of the matrix A depends upon the ordering scheme by which the blocks of the mesh are linearly indexed. For example, Fig. 1 illustrates the common row-by-row ordering in the case of an $I \times J$ two-dimensional mesh. We refer to this as the standard Gaussian ordering. Fig. 2 shows that the corresponding matrix $A = \{a_{i,j}\}$ is a sparse band matrix. A is diagonally dominant for any ordering and is in general asymmetric. The incidence matrix[9] M corresponding to A is defined as $m_{i,j} = 1$ if $a_{i,j} \neq 0$ and $m_{i,j} = 0$ if $a_{i,j} = 0$. The incidence matrix M is symmetric for any ordering. Unless otherwise noted, all analyses and results in this paper are based on an asymmetric A matrix and a symmetric incidence matrix M.

Defining w_i as the number of nonzero entries in the ith equation to the right of the diagonal, we have the work W and storage S of Gaussian elimination as

$$W = \sum_{i=1}^{N} \left[(w_i+1)^2 + w_i \right] , \quad \ldots (2)$$

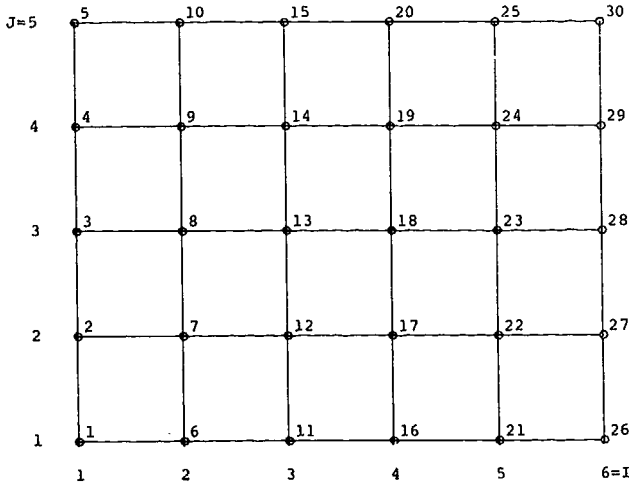

FIG. 1 — STANDARD ROW-BY-ROW GAUSSIAN ORDERING.

$$S = \sum_{i=1}^{N} w_i , \quad \ldots\ldots\ldots\ldots (3)$$

where $N = I \times J \times K$ = total number of unknowns. W is defined as the number of multiplications and divisions necessary to eliminate the matrix to upper triangular form and to perform the back substitution. Storage is required only for nonzero entries to the right of the diagonal. w_i is the number of nonzero entries at the stage in the elimination when $a_{i,i}$ is used as the pivot. Thus in general, w_i is not determinable by inspection of the original form of A.

The problem considered here is that of selecting ordering schemes that reduce work W by reducing $\sum_{1}^{N} w_i^2$ (i.e., sparsity-conserving ordering schemes). The two major classes of sparsity-conserving ordering schemes are matrix-banding schemes and schemes that are not matrix-banding. We will consider examples from both classes; however, our emphasis will be on matrix-banding schemes.

MATRIX-BANDING SCHEMES

Ordering schemes in the matrix-banding class yield A matrices with nonzero entries restricted to relatively narrow bands about either the major (upper left to lower right) or minor (lower left to upper right) diagonals. We present here descriptions and work estimates for several matrix-banding ordering schemes we have used, beginning with the standard Gaussian ordering, which is used throughout as a basis of comparison.

WORK AND STORAGE REQUIREMENTS FOR THE STANDARD ORDERING

In two dimensions, standard Gaussian elimination requires work of

$$W_1 = (IJ-2J+1)\left[(J+1)^2 + J\right] + \frac{J(J-1)(2J-1)}{3} + 3J(J-1) + (J+2)^2 + J - 9 , \quad \ldots (4)$$

where $J \leq I$. For large I and J this is essentially IJ^3.

In two dimensions the storage requirement S is

$$S_1 = \Sigma w_i = IJ^2 . \quad \ldots\ldots\ldots (5)$$

DIAGONAL ORDERING SCHEME D2

Fig. 3 shows the diagonal ordering scheme D2 for the 6 × 5 two-dimensional case. Fig. 4 shows the corresponding matrix A. A general concept behind this scheme and Scheme D4 described below is attributed to M. Silverberg and B. F. Wallenberg by Ogbuobiri *et al.*[5] The band width w_i is seen to grow from 2 to J with increasing i and then shrink again. It is important to select the shorter direction

as the primary direction for this numbering. This ordering scheme can be characterized by groups of equations, each group consisting of all points on one diagonal of the mesh. Thus, we can tabulate

Class	Group Number	Number of Equations	w_i for Each Equation in Group
A	1	1	2
•	2	2	3
•	•	•	•
•	•	•	•
A	J-1	J-1	J
B	J	J	J
•	J+1	J	J
•	•	•	•
•	•	•	•
B	I	J	J
C	I+1	J-1	J-1
•	I+2	J-2	J-2
•	•	•	•
•	•	•	•
C	I+J-1	1	1

Starting with Group I, the last equation of each group has a band width one less than that indicated in the fourth column. This last equation could be included in the subsequent group, but then each group would no longer consist of all points on one diagonal of the mesh. The calculated work requirement for this ordering scheme is insignificantly less than given in Eq. 6 below if this band width discrepancy were considered.

Calculating work, W, and storage, S, from Eqs. 2 and 3, we obtain

$$W_2 = \frac{J^2(J-1)^2}{2} + \frac{4}{3}J(J-1)(2J-1) + 3J(J-1) + J(I-J+1)\left[(J+1)^2 + J\right] \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (6)$$

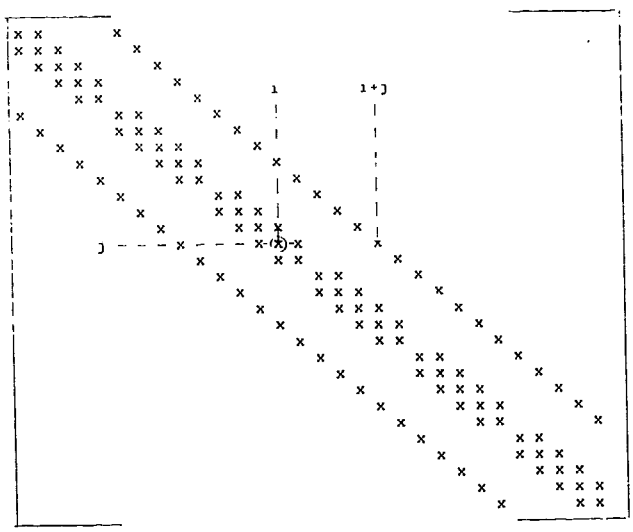

FIG. 2 — MATRIX A CORRESPONDING TO ORDERING OF FIG. 1.

$$S_2 = \sum_{1}^{IJ} w_i = (I-J+1)J^2 + \frac{J(J-1)(2J-1)}{3} + \frac{J(J-1)}{2} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (7)$$

For I and J large, the estimates are roughly

$$W_2 = IJ^3 - \frac{J^4}{2} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (8)$$

$$S_2 = IJ^2 - \frac{J^3}{3} \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (9)$$

Thus, for a square $I = J$, this scheme requires one-half the work and two-thirds the storage of the standard ordering.

The reader should note from Fig. 4 that the matrix A for this method is close-packed in the sense that insignificant inefficiencies occur by performing elimination assuming right-side bands are full to w_i.

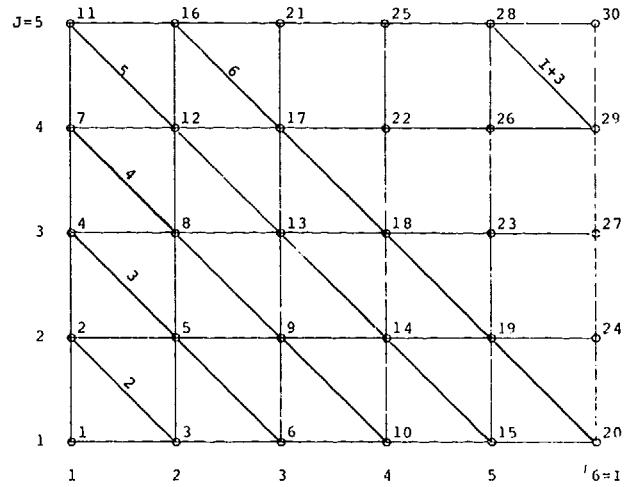

FIG. 3 — ORDERING SCHEME D2.

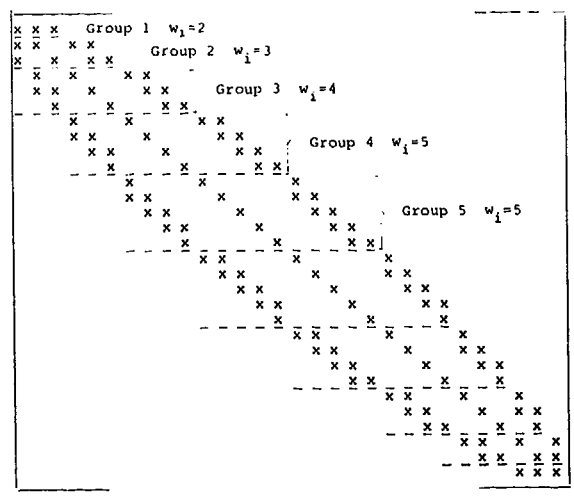

FIG. 4 — MATRIX A FOR SCHEME D2 ORDERING OF FIG. 3.

ALTERNATING POINT ORDERING SCHEME A3

Fig. 5 shows the alternating point ordering scheme for a 6 × 5 two-dimensional region. Fig. 6 shows the corresponding matrix A. Elimination through the first $N/2$ equations creates no additional nonzero entries in the top half of the matrix and zeroes all original entries to the lower left. This elimination process creates additional nonzero entries in the lower half as indicated by the circles of Fig. 6.

The matrix shown in Fig. 6 can be seen to be the normal form of a matrix that is cyclic of Index 2 (see Varga[10]). This is significant because it is easily seen that the elimination of any matrix that can be put into this form has the properties that, for the first $N/2$ points, $w_i < 2r$, where r is the dimensionality of the problem. For the ordering described here it is simple to show that for the second $N/2$ points, w_i follows exactly the form it takes for the standard ordering. Therefore, for large I, and J

$$W_3 = \frac{IJ^3}{2} \quad \ldots\ldots\ldots\ldots\ldots\ldots (10)$$

$$S_3 = \frac{IJ^2}{2} \quad \ldots\ldots\ldots\ldots\ldots\ldots (11)$$

Since it applies to any ordering of the points that puts the matrix into its two-cycle normal form, this one simple idea provides great potential. In fact, the combination of a two-cyclic ordering with the diagonal ordering Scheme D2 above led us to the next method, which represents the most significant improvements we have found to date.

As a final remark we should point out that for any matrix that is cyclic of index $p \geq 2$, any ordering of the points that puts the matrix in its normal form leads to an elimination such that

$$W_3 = \frac{IJ^3}{p} \quad \ldots\ldots\ldots\ldots\ldots\ldots (12)$$

and

$$S_3 = \frac{IJ^2}{p} \quad \ldots\ldots\ldots\ldots\ldots\ldots (13)$$

for two-dimensional problems.

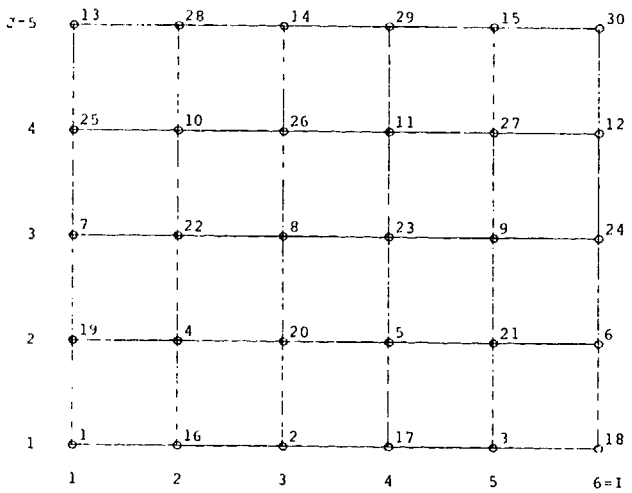

FIG. 5 — CYCLIC 2 ORDERING.

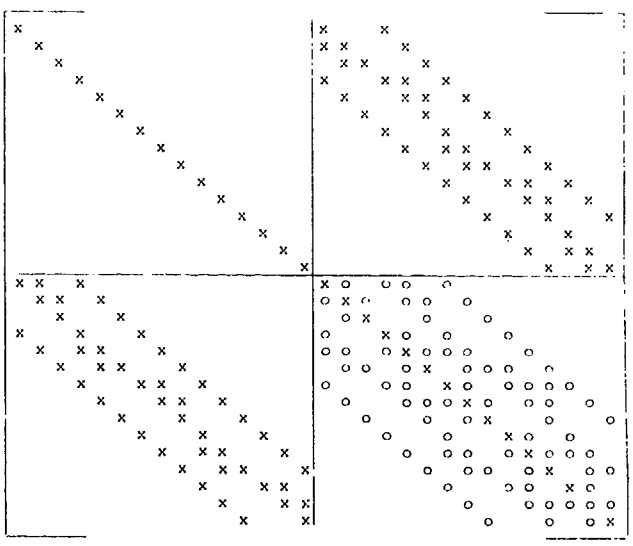

FIG. 6 — NORMAL FORM OF 2-CYCLIC MATRIX CORRESPONDING TO ORDERING OF FIG. 5.

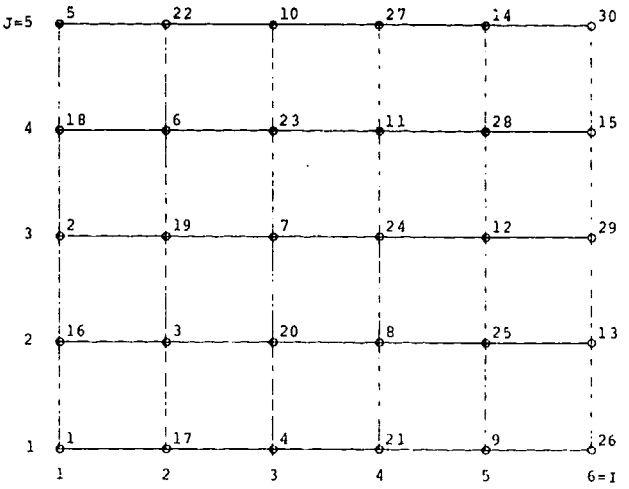

FIG. 7 — ORDERING SCHEME D4.

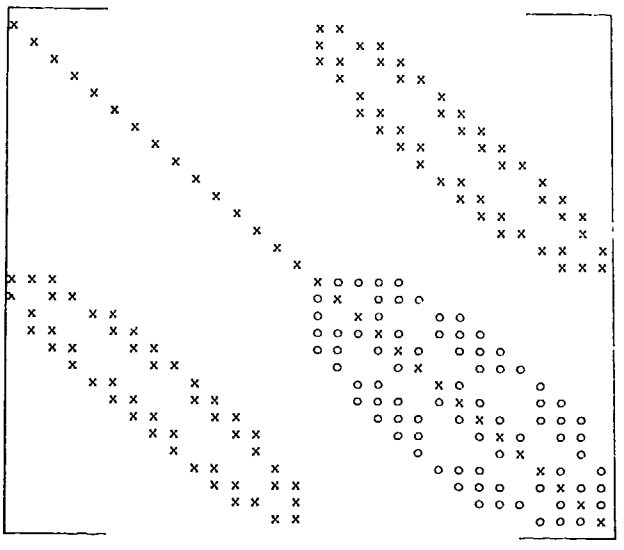

FIG. 8 — MATRIX A CORRESPONDING TO SCHEME D4 ORDERING OF FIG. 7.

ALTERNATING DIAGONAL ORDERING SCHEME D4

Fig. 7 shows the alternating diagonal ordering for the 6 × 5 two-dimensional case. Fig. 8 shows the corresponding matrix A. This ordering scheme partitions the matrix into upper and lower halves. Elimination through the first $N/2$ equations creates additional nonzero entries in the lower half as indicated by the circles of Fig. 8. For even I and J, $I > J$, this altered lower half matrix resulting from the top half elimination can be characterized by

Class	Group Number	Number of Equations	w_i
A	1	2	4
•	2	4	6
•	3	6	8
•	•	•	•
•	•	•	•
•	•	•	•
A	J/2-1	•	•
B	J/2	J	J
•	J/2+1	J	J
•	•	•	•
•	•	•	•
•	•	•	•
B	I/2	J	J
C	I/2+1	J-2	J-2
•	I/2+2	J-4	J-4
•	•	•	•
•	•	•	•
C	I/2+J/2-1	2	2

As indicated by this table, the band width w_i builds stepwise to J, but never exceeds J, and then shrinks again. This variable band width can be seen in Fig. 9, which shows the altered lower half matrix for the case of an 8 × 8 square. Again, the

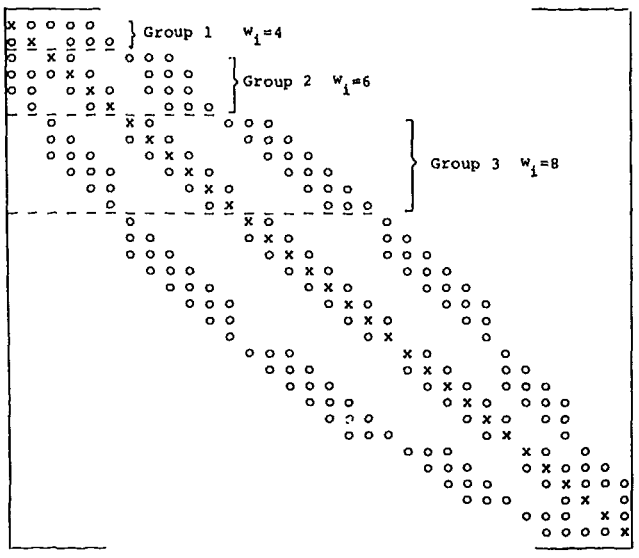

FIG. 9 — ALTERED LOWER HALF MATRIX FOR SCHEME D4, 8 × 8 SQUARE.

circles are nonzero entries introduced by the top half elimination. The obvious sparsity shown in Fig. 9 indicates that some reordering of these $N/2$ equations might reduce further the work of this Scheme D4.

Using Eqs. 2 and 3 we find the work and storage for this Scheme D4 are

$$W_4 = \frac{I}{2} J(J+1)^2 - \frac{J(J+1)^2(J-2)}{2} + \frac{J^2(J-2)(J-4)}{4} + \frac{5}{3} J(J-1)(J-2) + \frac{7}{4} J(J-2) + \frac{IJ}{2} (J+29) \quad (14)$$

$$S_4 = \left(\frac{I}{2} - \frac{J}{2} + 1\right) J^2 + \frac{1}{3} J(J-1)(J-2) + \frac{1}{2} J(J-2) \quad (15)$$

For large I, J these simplify to

$$W_4 = \frac{IJ^3}{2} - \frac{J^4}{4} \quad (16)$$

$$S_4 = \frac{IJ^2}{2} - \frac{J^3}{6} \quad (17)$$

For $n \times n$ squares these become $W_4 = n^4/4$ and $S_4 = n^3/3$. Thus, for large squares this Scheme D4 requires one-fourth the computing time and one-third the storage of standard ordering.

As in the case of diagonal Scheme D2, it is important here to select the direction having the fewer grid points as the primary direction for the numbering. This ensures that the band width, w_i, will not exceed J at any stage in the elimination of the lower half of the matrix.

EXTENSIONS TO THREE DIMENSIONS

The extensions of the standard ordering and the alternating point schemes require only that numbering be performed in the shortest direction first, the next shortest direction second, and the longest direction last. If these rules are observed the maximum band will be JK for $I \geq J \geq K$ and the work and storage estimates are easily shown to be

$$W_1 = IJ^3K^3 \quad (18)$$

$$S_1 = IJ^2K^2 \quad (19)$$

and

$$W_3 = \frac{IJ^3K^3}{2} \quad (20)$$

$$S_3 = \frac{IJ^2K^2}{2} \quad \ldots\ldots\ldots\ldots (21)$$

The diagonal ordering schemes can also be extended to three dimensions. For brevity we will consider here only the alternating diagonal Scheme D4. In three dimensions we number points in order of alternate diagonal planes rather than alternate diagonal lines. Introducing locally the notation of i, j, k as mesh indices ($i = 1, 2, \ldots, I$; $j = 1, 2, \ldots, J$; $k = 1, 2, \ldots, K$), we denote a diagonal plane by Integer m where all grid (mesh) points on Diagonal Plane m obey

$$i+j+k = m, \quad m = 3, 4, \ldots, M \quad \ldots\ldots\ldots\ldots\ldots\ldots (22)$$

and $M = I + J + K$.

If M is even, then the planes should be chosen in the order 3, 5, 7, ..., M − 1, 4, 6, 8, ..., M. If M is odd, then the order should be 3, 5, ..., M, 4, 6, ..., M − 1. The points in the plane should then be numbered in order of decreasing k and for each constant value of k in terms of decreasing j and increasing i where $I \geq J \geq K$. The following table illustrates this for the first few planes.

Plane	Grid Indices i	j	k	Point Number
3	1	1	1	1
5	1	1	3	2
•	1	2	2	3
•	2	1	2	4
•	1	3	1	5
•	2	2	1	6
5	3	1	1	7
7	1	1	5	8
•	1	2	4	9
•	2	1	4	10
•	1	3	3	11
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
7	5	1	1	•

The calculation of the work and storage for this Scheme D4 in three dimensions is considerably more complex than in two dimensions. The reason primarily is that there are so many terms in the expansion. For example, the general expression for work is given by

$$W_4 = \sum_{i=0}^{7} \sum_{\alpha_j+\beta_j+\gamma_j=i} \cdot d_{i,j} I^{\alpha_j} J^{\beta_j} K^{\gamma_j} \alpha_j, \beta_j, \gamma_j \geq 0. \quad (23)$$

where $d_{i,j}$ is a dummy variable, and $\sum_{\alpha_j+\beta_j+\gamma_j=i}$ means we are to include all nonnegative integers that add up to i. A similar expression for the storage is given by

$$S_4 = \sum_{i=0}^{5} \sum_{\alpha_j+\beta_j+\gamma_j=i} e_{i,j} I^{\alpha_j} J^{\beta_j} K^{\gamma_j} \quad \ldots\ldots\ldots\ldots\ldots\ldots (24)$$

Again $e_{i,j}$ is a dummy variable.

Both W_4 and S_4 are calculated by developing expressions for the w_i of Eqs. 2 and 3 and summing over the grid points. We have included in the Appendix the general expressions for the w_i and the summations required; however, for brevity we have chosen to include here the work estimates for only a few special cases.

Case I. Cube

For the cube $I = J = K$, Eqs. 23 and 24 reduce to only eight and six terms, respectively. However, the task of generating the $d_{i,j}$'s and $e_{i,j}$'s is still not trivial because these coefficients differ slightly depending on whether I is odd or even. However, the effects are lower order, so for the cube including only the two highest-order terms we obtain

$$W_4 \approx \frac{23}{90} I^6 + \frac{6}{35} I^7 \quad \ldots\ldots (25)$$

A similar expression for the standard ordering is given by

$$W_1 \approx -\frac{4}{3} I^6 + I^7 \quad \ldots\ldots\ldots (26)$$

Case II. $I - J \geq K$

When $I - J \geq K$, Eq. 23 again simplifies, since it can be seen from the expansions in Appendix A that $\alpha_j \leq 1$ and $\beta_j \leq 4$. This reduces the total number of terms from 120 to 33. Since even this is a large number, we will present only the highest-order terms. For this case we obtain

$$W_4 \approx \frac{IJ^3K^3}{2} - \frac{J^4K^3}{4} - \frac{J^2K^5}{8} + \frac{JK^6}{40} \quad \ldots\ldots\ldots\ldots\ldots\ldots (27)$$

When $K = 1$, Eq. 27 becomes

$$W_4 \approx \frac{IJ^3}{2} - \frac{J^4}{4} - \frac{J^2}{8} + \frac{J}{40} \quad \ldots (28)$$

and neglecting lower-order terms in Eq. 28, this reduces exactly to Eq. 16. The corresponding expression for the standard ordering is

$$W_1 \approx IJ^3K^3 \quad \ldots\ldots\ldots\ldots (29)$$

Case III. $I = J$

Since the general class of cases $I - J < K$ is

still quite complex, we have chosen the special case of $I = J$. If we also let $J = lK$ where the elongation $l > 1$, then the high-order terms of Eq. 23 become

$$W_4 \approx \left(\frac{l^4}{4} - \frac{l^2}{8} + \frac{l}{20} - \frac{1}{280}\right)K^7 \cdot \quad \ldots\ldots\ldots\ldots\ldots\ldots (30)$$

PSEUDO-OPTIMAL ORDERING SCHEMES

Tinney and Walker[7] describe three ordering schemes that aim at optimum conservation of matrix sparsity in Gaussian elimination. For electrical network problems these schemes are considered generally more efficient than matrix-banding schemes. However, we will show here that this is not necessarily true for the matrices arising from finite-difference approximations. The usual convention is to denote these as Scheme 1, Scheme 2, and Scheme 3.

Scheme 1: Number the rows of a matrix in ascending order of the number of off-diagonal nonzero elements. If more than one row has the same number of nondiagonal nonzero terms, select these rows in any order.

Scheme 2: At each stage of an elimination, select that row that has the fewest number of nonzero off-diagonal elements. If more than one row has this minimum number, select any one of them.

Scheme 3: At each stage of an elimination, select that node that minimizes the number of nonzero entries created in the rest of the matrix. If more than one node has this property, pick any one of them.

It should become immediately clear merely from reading the foregoing that Scheme 1 will provide no help for the problem we are considering, and Schemes 2 and 3 require auxiliary programs to simulate an elimination and provide as a result the selected ordering.

We selected Scheme 2 and programmed a simulated elimination because at least one author[5] has called it the best. This program is not particularly complex; however, it can be quite slow because it requires a significant amount of searching. For example, it required 135 seconds of CDC 6600 time to generate the ordering for a two-dimensional 30 × 30 grid. This computer time is roughly proportional to the total number of grid points, so that to generate the ordering for a 4,000-grid block problem would require about 600 seconds of CDC 6600 time.

While this is a substantial effort, it need be done only once for any given problem and could ultimately result in large savings. We have run a number of two- and three-dimensional cases using this Scheme 2, and Table 1 compares the results with both the standard ordering and the matrix-banding Scheme D4 presented above. Table 1 indicates that substantial savings are possible for two-dimensional squares. However, the same does not seem to be true for elongated rectangles and the three-dimensional cases considered.

Although these results are in no way conclusive, they do indicate that schemes like these offer real potential and deserve further study.

Because these schemes are not matrix-banding, they present some significant programming difficulties. However, as mentioned above, they still can offer some real advantages. The $0(n^3)$ ordering of George[8] is not matrix-banding, but it clearly offers significant advantages for sufficiently large n. That is, George's ordering would be faster than the alternate diagonal ordering D4 when $n > 80$.

TABLE 1 — FOR COMPARISON OF WORK ESTIMATES OF PSEUDO-OPTIMAL AND MATRIX-BANDING ORDERINGS

Grid	Scheme 2	Scheme D4	Standard
8 × 8	2524	2860	4908
10 × 10	5214	5404	11596
12 × 12	9488	10270	23508
20 × 20	52080	59450	172996
30 × 30	213572	261250	854196
20 × 10	14394	13304	37796
30 × 10	23064	21204	24696
4 × 4 × 4	5950	5658	14490
5 × 5 × 5	23891	21915	66307
8 × 8 × 8	594260	467172	1868314
10 × 10 × 4	224762	186926	601702

COMPARISON OF SCHEME D4 WITH THE STANDARD ORDERING

For two dimensions, Eqs. 4 and 16 can be combined to give the ratio W_4/W_1 for $I \times J$ mesh. For large I, J this ratio is approximately

$$\frac{W_4}{W_1} = \frac{\frac{IJ^3}{2} - \frac{J^4}{4}}{IJ^3} \quad \ldots\ldots\ldots (31)$$

Defining elongation l by $I = lJ$ where $l > 1$, we have

$$\frac{1}{2} > \frac{W_4}{W_1} = \frac{2l-1}{4l} > \frac{1}{4} \quad \ldots\ldots (32)$$

Thus for large I and J, Scheme D4 is twice as fast as standard Gaussian elimination for highly elongated rectangles and is nearly four times faster for $I = 1$.

Following are some ratios W_4/W_1 computed from Eqs. 4 and 16 retaining lower-order terms:

I	J	W_4/W_1
10	10	0.492
14	14	0.401
20	20	0.344
30	30	0.306
70	70	0.271
24	16	0.436
60	40	0.364
20	10	0.555
40	20	0.441
40	8	0.649
80	16	0.515

The programming for matrix-banding schemes can

become somewhat involved and the amount of unnecessary logic or other inefficiencies will depend upon the ingenuity and effort expended in programming. We therefore experimentally checked work comparisons based on the above equations for W, using a FORTRAN program written to solve the diffusivity equation with the methods described here. The FORTRAN program was executed to solve the diffusivity equation for 14×14 and 20×20 squares using Scheme D4 and using standard ordering. Clock calls were used in the program to determine the time spent solely in the eliminations. The resulting experimental W_4/W_1 ratios were 0.384 and 0.330 compared with the above values of 0.401 and 0.344. This indicates that savings calculated from the work estimates, W, are attainable in practice.

THREE-DIMENSIONAL COMPARISONS

For three-dimensional comparisons we will again limit ourselves to the following three cases:

I. $I = J = K$

II. $I = J \geq K$

III. $I = J$

Case I. $I = J = K$

If we combine Eqs. 25 and 26, we obtain

$$\frac{W_4}{W_1} \approx .171 + .582\, I^{-1} \quad \ldots\ldots (33)$$

which approaches 0.171 as I becomes large. Fig. 10 presents plots of W_4/W_1 vs I for a number of cases, and the curve labeled 1:1 shows that Eq. 33 is an excellent approximation. For example, when I is 8 and 18, Eq. 33 gives values for W_4/W_1 of 0.244 and 0.203, respectively, which are extremely close to the values plotted.

Case II. $I = J \geq K$

Combining Eqs. 27 and 29 results in

$$\frac{W_4}{W_1} \approx \frac{1}{2} - \frac{1}{4}\frac{J}{I} - \frac{1}{8}\frac{K^2}{IJ} + \frac{K^3}{40\, IJ^2} \quad \ldots\ldots (34)$$

If we now let

$$J = l_1 K$$

and

$$I = l_2 K,$$

where $l_2 - l_1 \geq 1$, Eq. 34 becomes

$$\frac{W_4}{W_1} \approx \frac{1}{2} - \frac{1}{4}\frac{l_1}{l_2} - \frac{1}{8 l_1 l_2} + \frac{1}{40 l_2 l_1^2} \quad \ldots\ldots (35)$$

For the ratios used to generate the curves in Fig. 10, we obtain the asymptotic estimates:

l_1	l_2	W_4/W_1
2	6	0.408
2	4	0.361
4	6	0.328
2	3	0.314
3	4	0.304

Case III. $I = J$

If we let $I = J$ and assume that $J = lK$, then dividing Eq. 30 by Eq. 29 gives

$$\frac{W_4}{W_1} = \frac{1}{4} - \frac{1}{8l^2} + \frac{1}{20l^3} - \frac{1}{280l^4} \quad \ldots\ldots (36)$$

therefore,

$$\frac{6}{35} \leq \frac{W_4}{W_1} \leq \frac{1}{4} \text{ for all } l \geq 1 .$$

From Fig. 10 it can be seen that for $l = 5$ the asymptotic limit is about 0.25 as expected, and these limits decrease as $l \to 1$.

COMPARISON OF SCHEME D4 WITH ITERATIVE METHODS

We define nominal band width as the product of the numbers of grid blocks in the two shorter directions. That is, nominal band width is J for two-dimensional problems where $I \geq J$ and is $J \times K$ for three-dimensional problems where $I \geq J \geq K$. Critical nominal band width is defined as the value of nominal band width at which work of an iterative method equals work of a direct method. A common rule of thumb gives 15 as critical nominal band width relative to the standard ordering scheme. Since few, if any, practical three-dimensional problems have nominal band widths less than 15, this rule of thumb indicates that iterative methods are more efficient for virtually all three-dimensional problems.

The work of an iterative method is

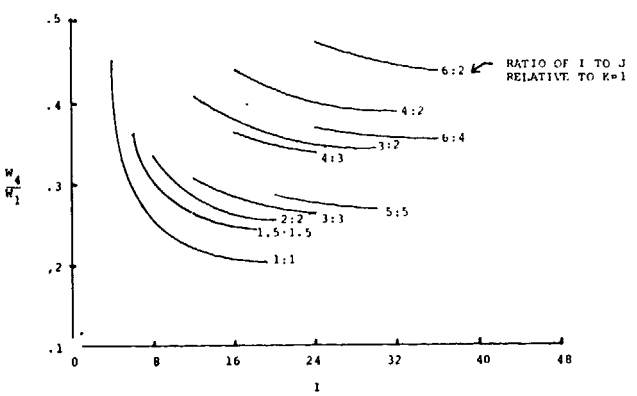

FIG. 10 — RATIO (f) OF SCHEME D4 WORK TO STANDARD ORDERING WORK IN THREE DIMENSIONS.

$$W_{it} = cN_i IJK \quad \ldots\ldots\ldots\ldots \quad (37)$$

where c = number of multiplications and divisions per iteration per grid point,

N_i = number of iterations.

As seen above there is no simple one-term expression for the work of the alternating diagonal ordering Scheme D4; however, for comparison one can assume that for all direct schemes

$$W \sim fI(JK)^3 \quad \ldots\ldots\ldots\ldots \quad (38)$$

For the standard ordering, f equals 1, and for the alternating diagonal ordering schemes we can show from Eqs. 25, 27, and 30 that f lies between 0.5 and 0.17. Moreover, for most practical three-dimensional reservoir problems it can be seen from Fig. 10 that a reasonable value of f would be 0.3. It is also clear from Eq. 16 that 0.3 is not a bad value even for two-dimensional reservoir problems. Therefore, for the remainder of this section we will assume that for Scheme D4 f in Eq. 38 will be 0.3.

Now combining Eqs. 37 and 38, we obtain

$$\frac{W}{W_{it}} = \frac{f(JK)^2}{cN_i} = \frac{fw^2}{cN_i} \quad \ldots\ldots\ldots\ldots \quad (39)$$

where w denotes nominal band width. Critical nominal band width (w_c) is obtained by setting W/W_{it} to 1 in Eq. 39 and solving for w. This defines

$$w_c \equiv \sqrt{cN_i/f} \quad \ldots\ldots\ldots\ldots \quad (40)$$

A value of 15 for w_c is obtained by setting f to 1.0 (for the standard ordering direct scheme), c to 19 (for ADI in two dimensions), and assuming 12 ADI iterations.

Since the use of Scheme D4 gives f values between 0.171 and 0.5 Eq. 40 shows that Scheme D4 increases the critical nominal band width by 40 percent to 140 percent.

Values of c for SIP, LSOR and ADI are as follows:

	Value of c	
	Two Dimensions	Three Dimensions
SIP	24	37
ADI	19	28
LSOR	9	11

These c values include the work of re-forming the residuals between iterations.

Now using the approximate value of 0.3 for f in Eq. 39 the value of N_i for which W_4 equals W_{it} is given by

$$N_i = \frac{.3(JK)^2}{c} \quad \ldots\ldots\ldots\ldots \quad (41)$$

Using c values from above gives the following table:

	Number of Iterations for Same Work as Scheme D4	
	Two Dimensions	Three Dimensions
SIP	$0.0125\,J^2$	$0.0081\,(JK)^2$
ADI	$0.0158\,J^2$	$0.0107\,(JK)^2$
LSOR	$0.0333\,J^2$	$0.0272\,(JK)^2$

If fewer than the above iterations are required on a problem, then the iterative method will be less expensive than the direct Scheme D4. This table indicates that direct Scheme D4 would be less expensive than ADI on a three-dimensional problem with nominal band width (JK) of 50 if ADI required more than 25 iterations.

An alternative form of rough comparison can be obtained by assuming $N_i = 40$ for LSOR, $f = 0.3$, and calculating w_c from Eq. 40. This gives a critical nominal band width of 34 in two dimensions and 38 in three dimensions.

We have found in extensive use of the direct Scheme D4 on a wide variety of actual reservoir problems that savings can be obtained relative to iterative methods on three-dimensional problems with nominal band widths as large as 80. The reason for the discrepancy between this and the value of 38 indicated above is that the above comparisons assume a full two- or three-dimensional mesh. In general, as a result of reservoir geometry, there are a number of missing grid blocks. These missing blocks will at best reduce the expense of iterative methods only slightly and may easily increase the number of iterations required. However, the missing blocks will reduce the Scheme D4 expense significantly since the band widths, w_i, will decrease considerably.

General comparisons of direct and iterative techniques are difficult to obtain because the work of an iterative method depends upon the number of iterations, which in turn depends upon (1) closure tolerances, (2) heterogeneity of the particular problem used in the comparison, and (3) time-step size. In general, larger time steps increase the number of iterations required through reducing diagonal dominance of the A matrix. Because of this complexity we have not attempted a complete or exhaustive comparison of direct and iterative techniques. We have, however, included three example problems as an illustrative comparison for some reasonably typical reservoir problems.

The reservoir dimensions for Example Problem 1 are 3,600 × 2,400 × 120 ft. An 18 × 12 × 6 grid is used so that each grid block has dimensions 200 × 200 × 20 ft. Fig. 11 shows the uniform (throughout z-direction) areal geometry and the uniform (throughout y-direction) vertical geometry. Horizontal and vertical permeabilities are 100 and 10 md, respectively, and porosity is 20 percent. The single-phase fluid has unit viscosity (1 cp) and a constant compressibility of 0.00001 1/psi. Fluid is injected at $i = 1$, $j = 7$, $k = 3$ at a constant 1,000 cu ft/D and produced at the same rate from $i = 18$, $j = 7$, $k = 3$. Computations using different methods were

performed for one 30-day time step.

Only 759 of the total 1,298 grid blocks are active. The nominal band width for this problem is 12 × 6, or 72. If all blocks were active, the maximum w_i of Scheme D4 would also be 72. Because of the geometry the actual maximum w_i for Scheme D4 was only 53, and only 3 of 759 equations had this band.

Fig. 10 gives an f of 0.36 for an 18 × 12 × 6 full mesh. Eq. 39 then gives

$$\frac{W_4}{W_{it}} = \frac{.36(72)^2}{cN_i}, \quad \ldots\ldots\ldots\ldots (42)$$

where for LSOR, ADI, and SIP becomes

$$\frac{W_4}{W_{LSOR}} = \frac{170}{N_i} \quad \ldots\ldots\ldots\ldots (43a)$$

$$\frac{W_4}{W_{ADI}} = \frac{66.8}{N_i} \quad \ldots\ldots\ldots\ldots (43b)$$

$$\frac{W_4}{W_{SIP}} = \frac{50.5}{N_i} \quad \ldots\ldots\ldots\ldots (43c)$$

For LSOR, three closure criteria were used as follows:

$$C_1 = \left| \sum_1^N R_i/q \right| < \delta_1$$

$$C_2 = \sum_1^N \left| R_i/q \right| < \delta_2$$

$$C_3 = \left| p_i^{(k)} - p_i^* \right|_{\max i} < \delta_3,$$

where R_i is residual $\sum_{j=1}^{N} a_{i,j}p_j - b_i$, summation $\sum_1^N$ is performed over all active grid blocks, q is total reservoir production rate, $p_i^{(k)}$ is the kth iterate, p_i^* is the exact solution, and the term max implies maximum value over all active grid points. Values of 0.01 were used for all of δ_1, δ_2, δ_3.

The value of C_3 was estimated as $|p_i^{(k+1)} - p_i^{(k)}|_{\max i}$ divided by $2 - \omega$ where ω is the LSOR acceleration parameter. We used only Criteria 1 and 2 for SIP and ADI. Criterion 1 is simply an incremental material-balance error.

After several trials a set of ADI parameters was selected as 0.002, 0.004, 0.008, ..., 0.128. LSOR used an automatically determined optimal parameter ω of 1.9061.

We obtained poor results with SIP on this problem. After 40 iterations $\Sigma R/1{,}000$ and $\Sigma|R|/1{,}000$ were – 0.102 and 0.251, respectively, and falling very slowly. Many sets of parameters were tried with little improvement.

Table 2 shows that the direct scheme is actually faster than any iterative scheme in spite of the indication from Eqs. 43 that it should be slower. The reason for this is the relatively greater efficiency effected in direct Scheme D4 as opposed to the iterative methods by the missing grid blocks.

To obtain an LSOR time equal to that of the direct solution, we would have had to terminate LSOR after 63 iterations. At that point LSOR had an incremental material-balance error (i.e., $\Sigma R/1{,}000$) of 44.8 percent.

The reservoir dimensions for Example Problem 2 are 7,500 × 7,500 × 100 ft. The 15 × 15 × 4 grid yields a grid block dimension of 500 × 500 × 25. Other data are k_H = 100 md; k_V = 20 md; ϕ = 0.2; and depth to top of grid block (1, 1, 1) = 3,400 ft. Sine of the constant dip angles in the x and y directions is 0.1. Initial pressure is 1,600 psia (bubble point) at the gas/oil contact (100 percent oil point) at 3,600 ft. A water/oil contact is at 4,400 ft.

Four production wells were located at

I	J	k
7	7	1 – 4
7	3	1 – 4
3	7	1 – 4

and production rates of 250 STB/D were assigned to each well.

Fig. 12 shows the irregular areal geometry. Of the total 900 grid blocks, only 720 are active.

Initial average (volume weighted) pressure is

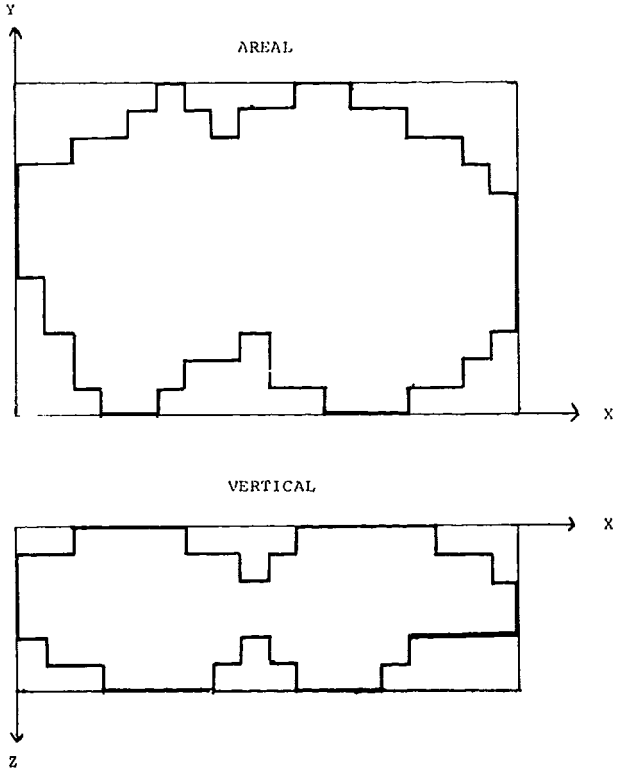

FIG. 11 — PROBLEM 1 GEOMETRY.

TABLE 2 — COMPARISON OF SCHEMES

	P_{avg}	Pressure		CDC 6600 Computing Time (seconds)	Number of Iterations	Closure Criteria	
		Injection Well	Production Well			C_1	C_2
ADI	-0.042	33.2996	-42.6484	15.555	56	0.00171	0.234
LSOR	0.0058	33.3367	-42.6258	13.55	107	-0.00994	0.00994
SIP	2.53	34.8629	-38.9796	9.43	40	-0.102	0.251
Direct Scheme 4	0.	33.3284	-42.6292	7.97		0.	0.

1785.3 psi, initial oil in place is 92.244×10^6 STB.

We ran this problem to 180 days using LSOR and again using Scheme D4 direct solution and wrote a restart record at that time. We then ran two 730-day runs to 910 days total using five 120-day time steps and one 130-day step, and using LSOR and Scheme D4 direct.

Clock calls were inserted in the program to print out the computer time spent solely in the solution methods. LSOR required an average of 109 iterations per time step using an optimum parameter of 1.8463. Closure tolerance was a maximum pressure error of 0.05 psi. Two direct solutions per time step were performed to allow updating of nonlinearities in the flow equations. The total Scheme D4 computing time for the six time steps was 82 percent of that for LSOR.

The two runs yielded virtually identical answers, with pressures printed to one decimal place all agreeing within 0.1 psi.

From Fig. 10, I for a $15 \times 15 \times 4$ problem is about 0.29. Eq. 39 then gives

$$\frac{W_4}{W_{LSOR}} = \frac{95 \times 2}{111} = 1.71 .$$

The actual ratio was 0.82. Again, this significantly lower ratio in favor of Scheme D4 is largely due to the missing grid blocks, which significantly reduce band widths, w_i.

Example Problem 3 is an actual reservoir having significant heterogeneity and irregular geometry, as shown in Fig. 13. We had difficulty selecting iteration parameters using SIP and found LSOR to be more efficient. The three-dimensional $21 \times 20 \times 3$ grid has a nominal band width of 60. Because of the missing grid blocks, the maximum w_i using Scheme D4 is only 44. Only 635 of the total 1,260 grid blocks are active.

We report here results from simulation of the first year of production, during which time the reservoir remained undersaturated with no free gas. Only one production well was active, and its rate varied from 56 to 80 STB/D. Closure tolerance for LSOR was a maximum pressure error of 0.25 psi. Table 3 summarizes LSOR performance for the run. This table shows the tendency for number of iterations, N_i, to increase as time-step size increases.

The same run using Scheme D4 gave pressure changes and distributions over the 365 days, which differed by less than 0.3 psia from the answers obtained using LSOR. The computing time spent in solution of the pressure equation was 33 percent less using Scheme D4 as opposed to LSOR. This 33 percent saving is the average saving over all time steps. The saving of Scheme D4 is greater when computed on the basis of the 91.25-day time steps alone since LSOR requires more iterations on the larger time steps.

Since direct solution was performed twice per time step, Eq. 39 gives

$$\frac{W_4}{W_{LSOR}} = \frac{0.33(60)^2}{11N_i} \times 2 = \frac{190}{N_i} = 2.47 ,$$

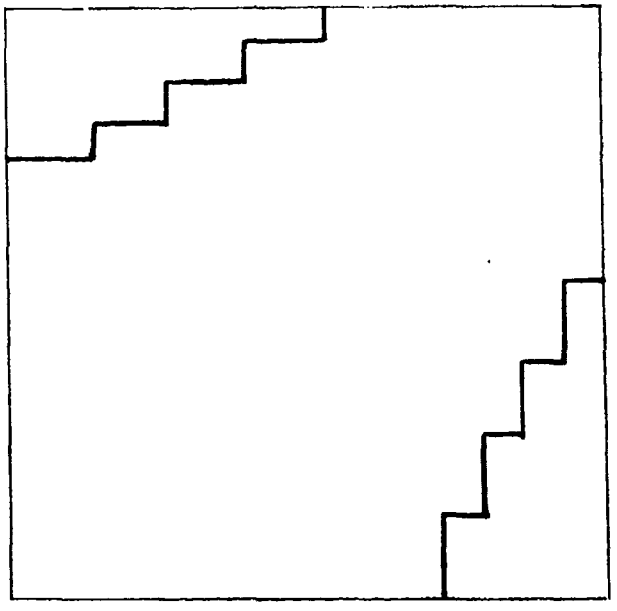

FIG. 12 — PROBLEM 2 AREAL GEOMETRY.

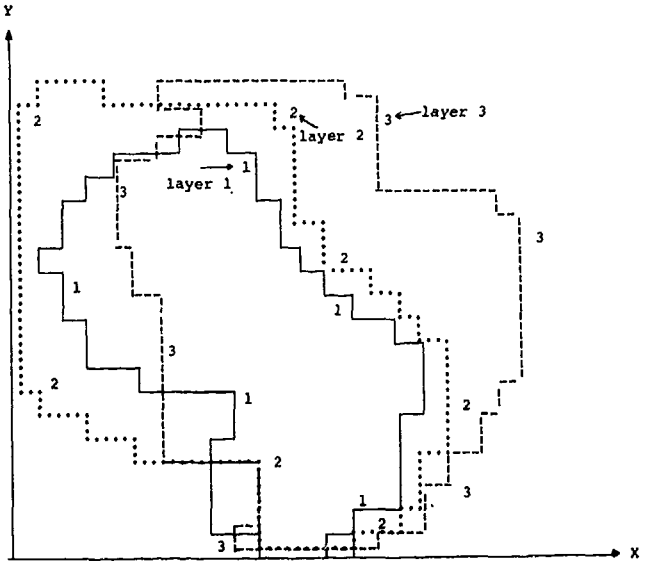

FIG. 13 — PROBLEM 3 GEOMETRY.

using the average number of LSOR iterations per time step of 77. The value of 0.33 for f was estimated from Fig. 10 for a problem having a ratio of about 7:7 relative to $K = 1$. As a result of the relatively large percentage of inactive blocks, the actual ratio was 0.67, which is only about one-fourth the work ratio of 2.47.

Since most reservoir problems involve many inactive blocks, this reduced actual work ratio must be considered seriously before selecting a solution technique on the basis of any rules of thumb.

CONCLUSIONS

This paper has described and analyzed schemes for the direct solutions of the diffusivity-type difference equations in two and three dimensions. The most efficient of several matrix-banding schemes described is shown to reduce computing expense by a factor as large as 5.8 relative to the standard ordering. In fact, because all direct methods are significantly improved when there are ineffective grid blocks in the system, this method should be considered for almost all practical reservoir simulation problems. Some of the other conclusions drawn from these results are as follows:

1. The most efficient of the matrix-banding schemes described here is shown to be more efficient than a leading pseudo-optimal scheme for rectangles and three-dimensional grids.
2. This same matrix-banding Scheme D4 is shown to be very competitive with the leading iterative techniques. It is shown to be faster than iterative schemes for full systems (no ineffective grid blocks) with nominal band widths up to 38.
3. Three typical three-dimensional reservoir problems were presented, and Scheme D4 was faster than the iterative methods for all three cases. One of the problems had a nominal band width as large as 72.
4. The results presented using the pseudo-optimal orderings were inconclusive. However, it appears that with more work in this area these schemes could extend the application of direct methods to reservoir problems with even larger nominal band widths than those considered here.

NOMENCLATURE

$\underline{b}$ = column vector $\{b_i\}$, $i = 1, N$, where b_i is the right-hand side of the pressure equation written for grid point i

I = number of grid points in the x direction of a mesh

J = number of grid points in the y direction of a mesh

K = number of grid points in the z direction of a mesh

n = number of grid points along each side of a square or cubic grid

$\underline{P}$ = column vector $\{p_i\}$, $i = 1, N$ where p_i is pressure at Grid Point i

S = computer storage requirement, number of words

w = nominal band width — minimum of the three products, IJ, IK, JK

w_c = critical nominal band width — value of nominal band width at which work of an iterative method equals work of a direct method

w_i = band width — number of nonzero entries to the right of the diagonal in the ith equation of the set of equations $A\underline{p} = \underline{b}$

W = work of a computational scheme, defined as the number of multiplications and divisions necessary to solve Eq. 1

W_{it} = work W for an iterative method

REFERENCES

1. Peaceman, D. W., and Rachford, H. H., Jr.: "The Numerical Solution of Parabolic and Elliptic Differential Equations," *J. Soc. of Ind. Appl. Math.* (1955) Vol. 3, 28.
2. Douglas, Jim, Jr., and Rachford, H. H., Jr.: "On the Numerical Solution of Heat Conduction Problems in Two or Three Space Variables," *Trans. Amer. Math. Soc.* (1956) Vol. 82, 421-439.
3. Young, David M.: "Iterative Methods for Solving Partial Differential Equations of Elliptic Types," *Trans. Amer. Math. Soc.* (1954) Vol. 76, 92-111.
4. Weinstein, H. G., Stone, H. L., and Kwan, T. V.: "Iterative Procedure for Solutions of Systems of Parabolic and Elliptic Equations in Three Dimensions," *Ind. and Eng. Chem. Fundamentals* (May 1969) Vol. 8, No. 2, 281-287.
5. Ogbuobiri, E. C., Tinney, W. F., and Walker, J. W.: "Sparsity-Directed Decomposition for Gaussian Elimination on Matrices," *IEEE Trans. on Power Apparatus and Systems* (Jan. 1970) Vol. 89, No. 1, 141-150.

TABLE 3 — LSOR PERFORMANCE, EXAMPLE PROBLEM 3

Time-Step Number	Time-Step Size (days)	Time (days)	Optimum ω for LSOR	Number of LSOR Iterations	Maximum Change in Grid-Point Pressure Over Step
1	10	10	1.8357	49	-92.9
2	23	33	1.8825	73	-61.2
3	23	56	1.8825	63	-32.5
4	23	79	1.8830	58	-18.3
5	12.3	91.3	1.8494	40	- 6.9
6	91.25	182.5	1.9359	101	-19.9
7	91.25	273.7	1.9359	123	-68.6
8	91.25	365	1.9358	105	- 6.4

6. Sato, N., and Tinney, W. F.: "Techniques for Exploiting the Sparsity of the Network Admittance Matrix," *IEEE Trans. Power Apparatus and Systems* (Dec. 1963) Vol. 82, 944-950.

7. Tinney, W. F., and Walker, J. W.: "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization," *Proc.*, IEEE (Nov. 1967) Vol. 55, 1801-1809.

8. George, Alan: "Nested Dissection of a Regular Finite Element Mesh," *J. Numer. Anal.*, SIAM (April 1973) Vol. 10, No. 2, 345-363.

9. Todd, John: *Survey of Numerical Analysis*, McGraw-Hill Book Co., Inc., New York (1962).

10. Varga, R. S.: *Matrix Iterative Analysis*, Prentice-Hall, Inc., Englewood, Cliffs, N. J.

APPENDIX

As indicated in the text, if the band width for the mth equation is defined to be w_m, then the work for a direct scheme is given by Eq. 2 as

$$W = \sum_{m=1}^{N} [(w_m+1)^2 + w_m], \quad \ldots \quad \text{(A-1)}$$

where $N = I \times J \times K$ equals the total number of unknowns. Now let $M = I + J + K$, where I, J, and K are the number of grid points in the x, y, and z direction, respectively, and for simplicity assume that I, J, and K are even.

If we now let indices i, j, and k refer to a grid point in our three-dimensional grid, then the point (i, j, k) is on the plane P_{m-2}, where $m = i + j + k$. Moreover there are M–2 planes in all. Since M is even, half these planes have m odd and the band for any point on these planes satisfies $w_m \leq 6$. Also there are $N/2$ grid points on the odd-numbered planes, so the work for these points is easily seen to be less than $55\,IJK/2$.

For the even-numbered planes (m even) the sums are more complex and we need to renumber the even planes as follows:

$P_{m-4} = 1$
$P_4 = 2$
$P_{m-6} = 3$
$P_6 = 4$, etc.

Continue in this fashion until all the planes except P_{m-2} have been numbered. Then number P_{m-2} last. With this ordering we shall now define the band widths for each point on each plane and provide the indicated summations. The reader can verify that the indicated summations are correct. For the remainder of this Appendix we shall use the letters α, β, γ as indices of summation.

SET 1

This set contains points on the first K–2 planes. We can show that

$$w_{\alpha,\beta,\gamma} = \left(\frac{\gamma(\gamma+1)}{2} + 2\beta + 3\right)^2, \quad \ldots \quad \text{(A-2)}$$

then the work for these points is given by

$$W = \sum_{\gamma=1}^{K-2} \sum_{\beta=1}^{\gamma} \sum_{\alpha=1}^{\beta} [(w_{\alpha,\beta,\gamma}+1)^2 + w_{\alpha,\beta,\gamma}], \quad \ldots \quad \text{(A-3)}$$

A total of $2K + (K^2/4) - 1$ points have been omitted from this plane, but will be included in the final sum.

SET 2

This set includes the next J–K planes taken from the renumbered set. For these planes we have

$$w_{\alpha,\beta,\gamma} = \frac{(K+\gamma-2)(K+\gamma-1)}{2} - \frac{(\gamma-2)(\gamma-1)}{2} + 2\beta, \quad \ldots \quad \text{(A-4)}$$

and the work is given by

$$W = \sum_{\gamma=1}^{J-K} \sum_{\beta=1}^{K} \sum_{\alpha=1}^{\gamma+\beta-2} [(w_{\alpha,\beta,\gamma}+1)^2 + w_{\alpha,\beta,\gamma}], \quad \ldots \quad \text{(A-5)}$$

Some points left out of this summation will be picked up later.

SET 3

This set includes the next r planes from the renumbered set, where r is the smaller of K and $I-J$.

For this case,

$$w_{\alpha,\beta,\gamma} = \frac{J(J-1)}{2} - \frac{(J-K-2+\gamma)(J-K+\gamma-1)}{2} + (\gamma-1)J, \quad \ldots \quad \text{(A-6)}$$

Then we have

$$W = \left\{ \sum_{\gamma=1}^{r} \sum_{\beta=1}^{K-\gamma} \sum_{\alpha=1}^{J-K-2+\beta+\gamma} \left([w_{\alpha,\beta,\gamma} + 2\beta + 1]^2 + w_{\alpha,\beta,\gamma+2\beta} \right) \right. + \left. \sum_{\alpha=1}^{J-1} \left([w_{\alpha,\beta,\gamma+2K-2\gamma+2}]^2 \right. \right.$$

$$+ w_{\alpha,\beta,\gamma+2K-2\gamma+1}\Big)$$

$$+ \sum_{\beta=K-\gamma+2}^{K} \sum_{\alpha=1}^{J} \Big([w_{\alpha,\beta,\gamma+2K-2\gamma+2}]^2$$

$$+ w_{\alpha,\beta,\gamma+2K-2\gamma+1} \Big) \Big] \cdot \quad \ldots (A\text{-}7)$$

SET 4

If $K < I - J$, then w is constant at JK and the work for the remaining planes becomes

$$W = \frac{(I-J-K)}{2} [(JK+1)^2 + JK] JK \cdot \quad \ldots (A\text{-}8)$$

If $I - J < K$, we have

$$w_{\alpha,\beta,\gamma} = J(J+1)$$

$$- \frac{(I-K-2+\gamma)(I-K-1+\gamma)}{2} + (\gamma-1)J$$

$$- \frac{(J-\gamma+1)(J-\gamma+2)}{2} + 2\beta \cdot \quad \ldots (A\text{-}9)$$

$$w_{\beta,\gamma} = w_{\alpha,\beta,\gamma}$$

$$+ 2K-2I+2J-2\gamma+2-2\beta \cdot \quad (A\text{-}10)$$

and

$$w_{\gamma} = w_{\beta,\gamma} + 2\beta - 1, \quad \ldots (A\text{-}11)$$

then

$$W = \sum_{\gamma=1}^{\frac{K+J-I}{2}} \Bigg\{ \sum_{\beta=1}^{K-1+J-\gamma} \sum_{\alpha=1}^{I-K-2+\beta+\gamma}$$

$$\cdot [(w_{\alpha,\beta,\gamma}+1)^2 + w_{\alpha,\beta,\gamma}]$$

$$+ [(w_{\gamma}+1)^2 + w_{\gamma}] \big(2(J-1)$$

$$+ (I-J-1)J\big) + \sum_{\beta=1}^{\gamma-1} [(w_{\beta,\gamma}+1)^2$$

$$+ w_{\beta,\gamma}] \cdot \quad \ldots (A\text{-}12)$$

SET 5

This set consists of all the points that were missed previously and it can easily be shown that for these points,

$$W \leq \sum_{\gamma=1}^{JK} \gamma^2+\gamma-1 \cdot \quad \ldots (A\text{-}13)$$

***