BIT 29 (1989), 635–657

THE EFFECT OF ORDERING ON PRECONDITIONED CONJUGATE GRADIENTS

IAIN S. DUFF and GÉRARD A. MEURANT

Computer Science and Systems Division, Harwell Laboratory, Oxon OX11 0RA, UK

CEA, Centre d'Etudes de Limeil-Valenton, BP 27, 94190 Villeneuve St Georges, France.

Abstract.

We investigate the effect of the ordering of the unknowns on the convergence of the preconditioned conjugate gradient method. We examine a wide range of ordering methods including nested dissection, minimum degree, and red-black and consider preconditionings without fill-in. We show empirically that there can be a significant difference in the number of iterations required by the conjugate gradient method and suggest reasons for this marked difference in performance.

We also consider the effect of orderings when an incomplete factorization which allows some fill-in is performed. We consider the effect of automatically controlling the sparsity of the incomplete factorization through drop tolerances and level of fill-in.

AMS Classification: 65F10.

Keywords: Sparse matrices, preconditioning, ordering strategies, conjugate gradients.

1. Introduction.

One of the most popular and successfull methods for the solution of positive definite symmetric systems arising from the discretization of partial differential equations is the preconditioned conjugate gradient method. Many kinds of both point and block preconditioners are possible (for example, Concus *et al.* [5]), but those based on the incomplete factorization of the coefficient matrix have been the most useful in practice since they give a good acceleration of the conjugate gradient method while being easy to generate and use. It is particularly simple to write codes for a partial factorization where the factors are constrained to have the same sparsity pattern as the original matrix. In the parlance of Meijerink and van der Vorst [19], this method is called the ICCG(0) method (in later papers they refer to it as ICCG(1,1)).

Received February 1989. Revised April 1989

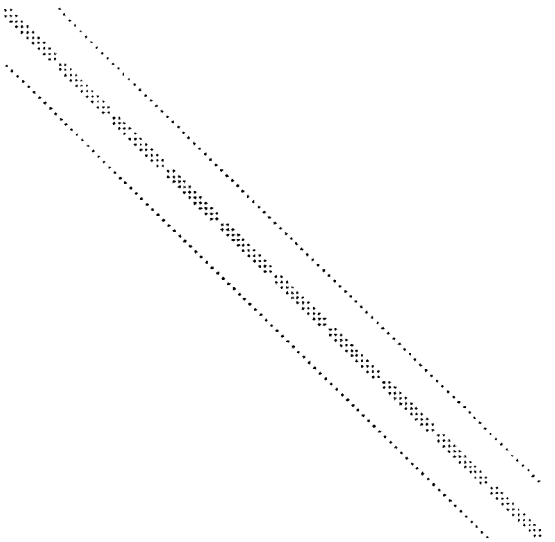

Fig. 1.1. *The five-diagonal matrix.*

In their original work, Meijerink and van der Vorst [19] only consider a factorization where the unknowns are numbered in a rowwise (or columnwise) fashion so that, for the 5-point discretization of Poisson's equation for example, the matrix has the five-diagonal band structure of Figure 1.1. In the study of direct solution methods for such problems (see, for example, [7]), the issue of how the unknowns are ordered is of paramount importance. Some strategies can reduce the work in the factorization of the Figure 1.1 matrix on a $q \times q$ grid by one full power in q from $O(q^4)$ to $O(q^3)$ and the storage from $O(q^3)$ to $O(q^2 \log_2 q)$. However, to our knowledge, the effect of the ordering of the unknowns on the performance of ICCG techniques has not been fully explored although Simon [22] has observed the effect, and the topic is beginning to be discussed more at meetings and conferences. Some ordering strategies have been designed to increase parallelism (for example, O'Leary [20], Schreiber and Tang [21], Adams and Jordan [1], Lichnewsky [18]), and the common strategy of working with a reduced system amounts to a red-black ordering followed by a rowwise scheme (for example, Hageman and Young [16] and Eisenstat *et al.* [10].

In this paper we study the effect of a wide range of orderings on the convergence of the point preconditioned conjugate gradient method on both model problems and more complicated elliptic equations. We show that some orderings which are widely used in the direct solution of equations can cause the conjugate gradient algorithm to take over six times the number of steps as the original ICCG(0) scheme. We also indicate that some other orderings can be competitive with rowwise ordering. This can give insight for finite-element problems for which there is not as straightforward an ordering as the row ordering for finite differences.

In Section 2, we describe in detail some of the orderings which we consider. Many have their origins in direct solution methods or in the oil industry but some we have invented for the purpose of these tests. We discuss the preconditioned conjugate gradient algorithm which we use in Section 3, the test problems in Section 4, and present our results in Section 5. Finally, in Section 6, we give some reasons to explain our findings.

2. Ordering strategies.

In this secton we discuss the orderings which we use in our experiments. Some other orderings were tried but did not add anything to those given here. For each ordering, we show the resulting order of the variables on a 10×10 grid and the pattern of the reordered matrix. In Table 2.1, we list the orderings together with a reference to the application or to the literature.

Table 2.1. *Orderings used in the paper.*

Ordering	Reference/application	Abbrevation used in text
Row ordering	Partial differential equations	row
Cuthill McKee (CM)	Cuthill and McKee [6]	cm
Reverse CM	George [11]	rcm
Block CM	This paper	block
Minimum degree	Tinney and Walker [23]	mind
Red black	Partial differential equations	rb
Alternating diagonal	Oil reservoir modelling (D4 ordering)	altd
Zebra ordering	Relaxation methods – oil reservoir modelling	zebra
Nested dissection	George [12]	nest
One-way dissection	George and Liu [13]	
1-level		diss1
2-level		diss2
Spiral	Duff *et al.* [8]	spiral
4-colour ordering	Adams, LeVeque, and Young [2]	4col
Parallel ordering 1	Van der Vorst (1988)	vdv1
Parallel ordering 2	Van der Vorst (1988)	vdv2
Union Jack	This paper	ujac
Localized row/column	This paper	loc

Some of the orderings in the table can be parameterized in a simple way. For example, we can control the number of levels in one-way dissection or nested dissection; the remaining unordered nodes are then numbered rowwise. Since we feel that many readers will not be familiar with all of these orderings and, additionally since we have not seen them all displayed together, we show in Figures 2.1 to 2.17 the ordering of the variables produced by each ordering where the connectivity on the grid is that of the five-point star.

In most cases, the general form of each ordering is clear from the grid shown. The four-colour ordering in Figure 2.13 labels the grid-points rowwise using four colours and then numbers (again rowwise) each colour in turn. The van der Vorst ordering in Figure 2.14 divides the grid into four and then numbers each quadrant in turn in a rowwise fashion; his second version in Figure 2.15 uses a Cuthill-McKee strategy to number in each quadrant. The Union Jack ordering in Figure 2.16 divides the grid into octants, numbering the cut sets last and the interior of the octants rowwise alternately in forward and backward directions as indicated in the figure. Finally, the Figure 2.17 ordering again divides the grid into quadrants numbering each in a forward and backward row and column ordering as illustrated in the figure.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Fig. 2.1. *Row ordering.*

1	3	6	10	15	21	28	36	45	55
2	5	9	14	20	27	35	44	54	64
4	8	13	19	26	34	43	53	63	72
7	12	18	25	33	42	52	62	71	79
11	17	24	32	41	51	61	70	78	85
16	23	31	40	50	60	69	77	84	90
22	30	39	49	59	68	76	83	89	94
29	38	48	58	67	75	82	88	93	97
37	47	57	66	74	81	87	92	96	99
46	56	65	73	80	86	91	95	98	100

Fig. 2.2. Cuthill McKee ordering.

100	98	95	91	86	80	73	65	56	46
99	96	92	87	81	74	66	57	47	37
97	93	88	82	75	67	58	48	38	29
94	89	83	76	68	59	49	39	30	22
90	84	77	69	60	50	40	31	23	16
85	78	70	61	51	41	32	24	17	11
79	71	62	52	42	33	25	18	12	7
72	63	53	43	34	26	19	13	8	4
64	54	44	35	27	20	14	9	5	2
55	45	36	28	21	15	10	6	3	1

Fig. 2.3. Reverse Cuthill McKee ordering.

1	2	9	10	21	22	37	38	57	58
3	4	11	12	23	24	39	40	59	60
5	6	17	18	33	34	53	54	73	74
7	8	19	20	35	36	55	56	75	76
13	14	29	30	49	50	69	70	85	86
15	16	31	32	51	52	71	72	87	88
25	26	45	46	65	66	81	82	93	94
27	28	47	48	67	68	83	84	95	96
41	42	61	62	77	78	89	90	97	98
43	44	63	64	79	80	91	92	99	100

Fig. 2.4. *Block Cuthill McKee ordering.*

4	5	52	21	20	86	19	45	7	3
6	28	53	51	77	85	59	58	27	8
75	74	44	80	60	43	84	42	62	61
22	50	81	41	82	83	40	76	49	23
18	54	95	68	39	87	69	38	91	17
93	96	37	94	88	36	92	89	35	90
16	57	72	34	99	70	33	79	67	15
46	56	32	73	31	97	78	30	65	55
9	26	64	48	100	29	71	66	25	11
2	10	63	24	14	98	13	47	12	1

Fig. 2.5. *Minimum degree ordering.*

1	51	2	52	3	53	4	54	5	55
56	6	57	7	58	8	59	9	60	10
11	61	12	62	13	63	14	64	15	65
66	16	67	17	68	18	69	19	70	20
21	71	22	72	23	73	24	74	25	75
76	26	77	27	78	28	79	29	80	30
31	81	32	82	33	83	34	84	35	85
86	36	87	37	88	38	89	39	90	40
41	91	42	92	43	93	44	94	45	95
96	46	97	47	98	48	99	49	100	50

Fig. 2.6. *Red-black ordering.*

1	51	2	53	5	57	10	63	17	71
52	3	54	6	58	11	64	18	72	26
4	55	7	59	12	65	19	73	27	81
56	8	60	13	66	20	74	28	82	35
9	61	14	67	21	75	29	83	36	89
62	15	68	22	76	30	84	37	90	42
16	69	23	77	31	85	38	91	43	95
70	24	78	32	86	39	92	44	96	47
25	79	33	87	40	93	45	97	48	99
80	34	88	41	94	46	98	49	100	50

Fig. 2.7. *Alternating diagonal ordering.*

1	2	3	4	5	6	7	8	9	10
51	52	53	54	55	56	57	58	59	60
11	12	13	14	15	16	17	18	19	20
61	62	63	64	65	66	67	68	69	70
21	22	23	24	25	26	27	28	29	30
71	72	73	74	75	76	77	78	79	80
31	32	33	34	35	36	37	38	39	40
81	82	83	84	85	86	87	88	89	90
41	42	43	44	45	46	47	48	49	50
91	92	93	94	95	96	97	98	99	100

Fig. 2.8. *Zebra ordering.*

3	4	13	1	91	21	22	33	17	18
5	6	14	2	92	23	24	34	19	20
11	12	15	10	93	31	32	35	29	30
7	8	16	9	94	25	26	36	27	28
87	88	89	90	95	82	83	84	85	86
39	40	52	37	96	61	62	77	57	58
41	42	53	38	97	63	64	78	59	60
50	51	54	49	98	75	76	79	73	74
43	44	55	47	99	65	66	80	69	70
45	46	56	48	100	67	68	81	71	72

Fig. 2.9. *Nested dissection ordering.*

1	2	3	4	5	91	51	52	53	54
6	7	8	9	10	92	55	56	57	58
11	12	13	14	15	93	59	60	61	62
16	17	18	19	20	94	63	64	65	66
21	22	23	24	25	95	67	68	69	70
26	27	28	29	30	96	71	72	73	74
31	32	33	34	35	97	75	76	77	78
36	37	38	39	40	98	79	80	81	82
41	42	43	44	45	99	83	84	85	86
46	47	48	49	50	100	87	88	89	90

Fig. 2.10. *One way dissection ordering – 1 level.*

1	2	81	21	22	91	41	71	51	52
3	4	82	23	24	92	42	72	53	54
5	6	83	25	26	93	43	73	55	56
7	8	84	27	28	94	44	74	57	58
9	10	85	29	30	95	45	75	59	60
11	12	86	31	32	96	46	76	61	62
13	14	87	33	34	97	47	77	63	64
15	16	88	35	36	98	48	78	65	66
17	18	89	37	38	99	49	79	67	68
19	20	90	39	40	100	50	80	69	70

Fig. 2.11. *One way dissection ordering – 2 level.*

1	2	3	4	5	6	7	8	9	10
36	37	38	39	40	41	42	43	44	11
35	64	65	66	67	68	69	70	45	12
34	63	84	85	86	87	88	71	46	13
33	62	83	96	97	98	89	72	47	14
32	61	82	95	100	99	90	73	48	15
31	60	81	94	93	92	91	74	49	16
30	59	80	79	78	77	76	75	50	17
29	58	57	56	55	54	53	52	51	18
28	27	26	25	24	23	22	21	20	19

Fig. 2.12. *Spiral ordering.*

1	26	51	76	2	27	52	77	3	28
53	78	4	29	54	79	5	30	55	80
6	31	56	81	7	32	57	82	8	33
58	83	9	34	59	84	10	35	60	85
11	36	61	86	12	37	62	87	13	38
63	88	14	39	64	89	15	40	65	90
16	41	66	91	17	42	67	92	18	43
68	93	19	44	69	94	20	45	70	95
21	46	71	96	22	47	72	97	23	48
73	98	24	49	74	99	25	50	75	100

Fig. 2.13. *Four-colour ordering.*

1	2	3	4	5	55	54	53	52	51
6	7	8	9	10	60	59	58	57	56
11	12	13	14	15	65	64	63	62	61
16	17	18	19	20	70	69	68	67	66
21	22	23	24	25	75	74	73	72	71
46	47	48	49	50	100	99	98	97	96
41	42	43	44	45	95	94	93	92	91
36	37	38	39	40	90	89	88	87	86
31	32	33	34	35	85	84	83	82	81
26	27	28	29	30	80	79	78	77	76

Fig. 2.14. *Van der Vorst ordering. Version* 1.

1	3	6	10	15	65	60	56	53	51
2	5	9	14	19	69	64	59	55	52
4	8	13	18	22	72	68	63	58	54
7	12	17	21	24	74	71	67	62	57
11	16	20	23	25	75	73	70	66	61
36	41	45	48	50	100	98	95	91	86
32	37	42	46	49	99	96	92	87	82
29	33	38	43	47	97	93	88	83	79
27	30	34	39	44	94	89	84	80	77
26	28	31	35	40	90	85	81	78	76

Fig. 2.15. *Van der Vorst ordering. Version* 2.

81	1	2	3	100	10	9	8	7	65
64	80	4	5	99	13	12	11	66	22
63	62	79	6	98	15	14	67	20	21
61	60	59	78	97	16	68	17	18	19
90	89	88	87	96	86	85	84	83	82
52	51	50	49	95	77	23	24	25	26
55	54	53	69	94	42	76	27	28	29
57	56	70	48	93	41	40	75	30	31
58	71	46	47	92	39	38	37	74	32
72	43	44	45	91	36	35	34	33	73

Fig. 2.16. *Union Jack ordering.*

1	2	3	4	5	26	35	36	45	46
10	9	8	7	6	27	34	37	44	47
11	12	13	14	15	28	33	38	43	48
20	19	18	17	16	29	32	39	42	49
21	22	23	24	25	30	31	40	41	50
51	60	61	70	71	76	77	78	79	80
52	59	62	69	72	85	84	83	82	81
53	58	63	68	73	86	87	88	89	90
54	57	64	67	74	95	94	93	92	91
55	56	65	66	75	96	97	98	99	100

Fig. 2.17. *Localized row/column ordering.*

Finally, since we will be concerned with the effect of these orderings on the ICCG methods we show, for each ordering, the pattern of the reordered matrix for the five-point formula on a 10×10 grid (matrix order 100) in Figures 2.18 to 2.32.

The orderings in Figures 2.2 and 2.3 are standard profile minimization orderings widely used with variable-band factorizations and Figure 2.4 illustrates a block version of the same. The minimum degree ordering, shown in Figure 2.5, is the most widely used scheme for obtaining low fill-in factorizations of general symmetric matrices. Since, at any stage, there are usually many variables with the same minimum degree, there is normally some strategy (called a tie-breaking strategy) for choosing the next variable among those of minimum degree. Thus the minimum degree ordering is not unique, and different tie-breaking strategies could result in orderings different from that shown. The red-black ordering of Figure 2.6 has been widely used in the oil industry because the resulting matrix (Figure 2.21) can be trivially reduced to give a system of half the order of the original. The alternating diagonal ordering (or D4 ordering) and the zebra (or alternating line) ordering in Figures 2.7 and 2.8 are also widely used in the oil industry (Behie and Forsyth [4], for example) and show good convergence for standard iterative schemes on examples from that application area. The dissection methods in Figures 2.9 to 2.11 are related to the current vogue for domain decomposition (or substructuring) techniques and are now partially motivated by their attractiveness for use on parallel computers. The spiral ordering, illustrated in Figure 2.12, was used by Duff *et al.* [8] to illustrate the (bad) effect of tie-breaking on the minimum degree

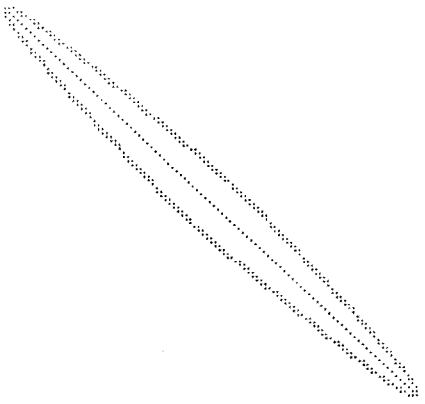

Fig. 2.18. *Cuthill McKee matrix.*

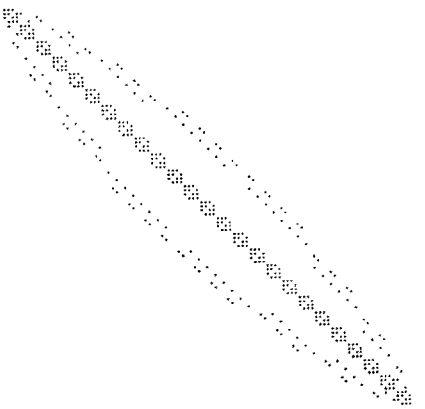

Fig. 2.19. *Block Cuthill McKee matrix.*

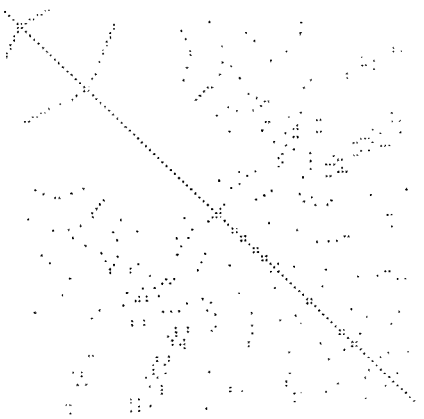

Fig. 2.20. *Minimum degree matrix.*

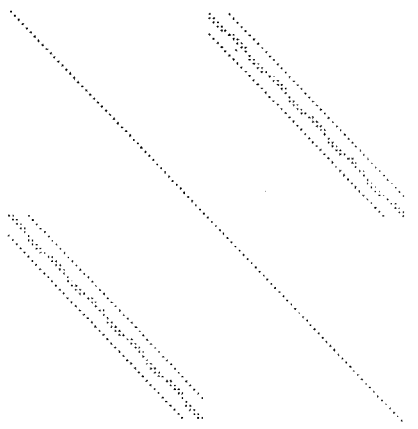

Fig. 2.21. *Red-black matrix.*

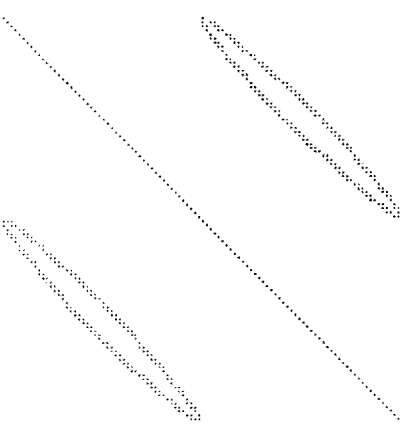

Fig. 2.22. *Alternating diagonal matrix.*

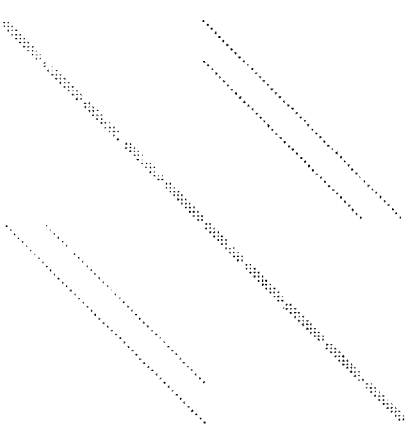

Fig. 2.23. *Zebra matrix.*

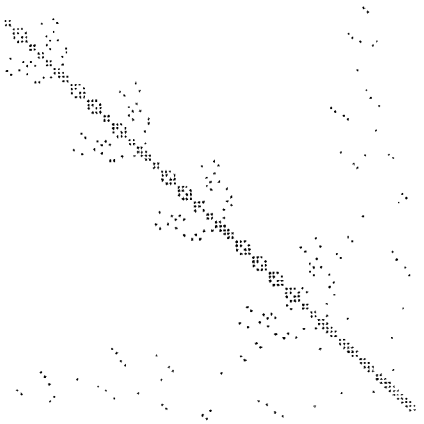

Fig. 2.24. *Nested dissection matrix.*

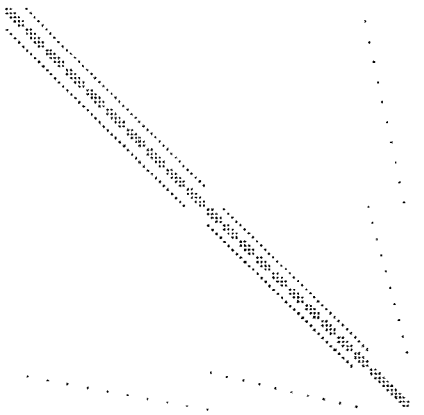

Fig. 2.25. *One way dissection matrix – 1 level.*

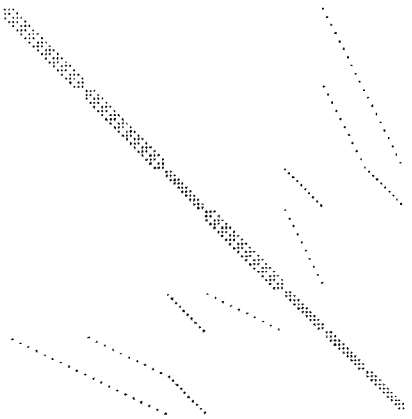

Fig. 2.26. *One way dissection matrix – 2 level.*

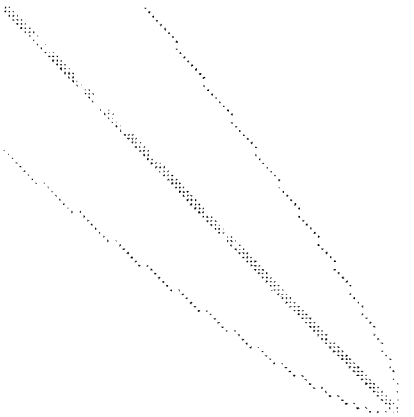

Fig. 2.27. *Spiral matrix.*

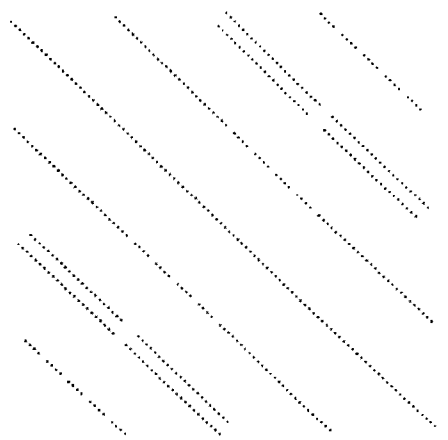

Fig. 2.28. *Four-colour ordering.*

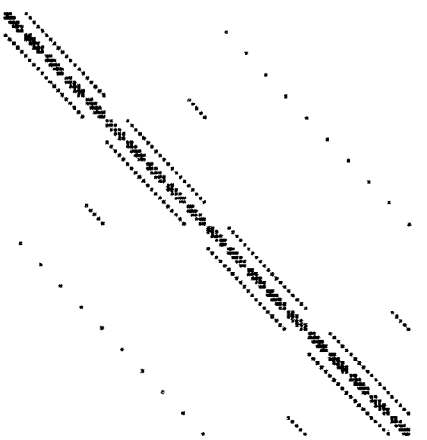

Fig. 2.29. *Van der Vorst ordering. Version* 1.

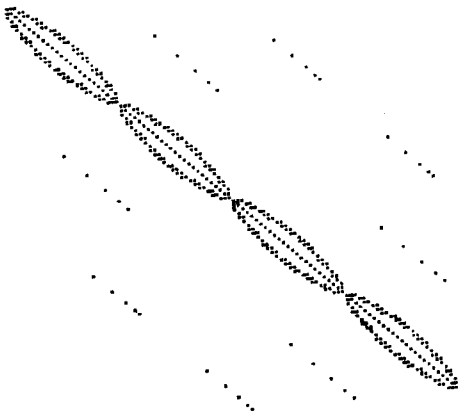

Fig. 2.30. *Van der Vorst ordering. Version* 2.

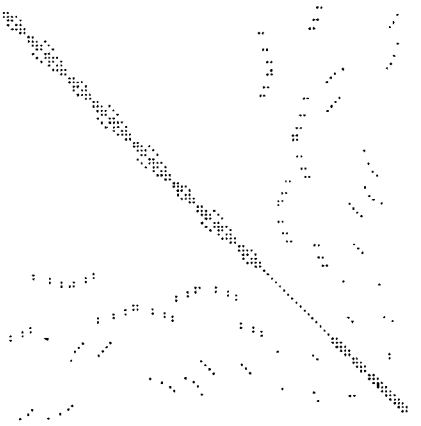

Fig. 2.31. *Union Jack matrix.*

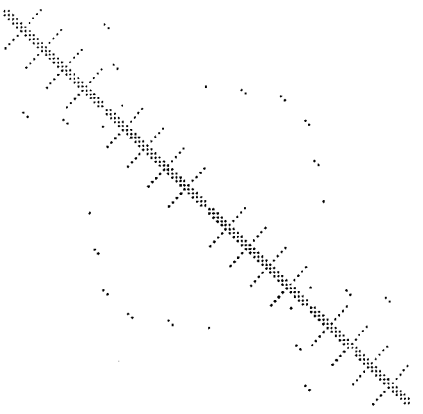

Fig. 2.32. *Localized row/column matrix.*

ordering. In Figures 2.13 to 2.15, we show three orderings designed to enhance potential parallelism; the two by van der Vorst (private communication 1988) retaining a somewhat local flavour and the popular four-colour ordering discussed by Adams, Le Veque, and Young [2]. Finally, we present two orderings where we have tried to minimize bandwidth in a local rather than global sense because our initial experiments led us to believe that this might be beneficial to the performance of our preconditioner.

3. Preconditioned conjugate gradients.

The form of preconditioned conjugate gradients which we use is the two-term algorithm as described, for instance, by Golub and Meurant [14], p. 221. For the

sake of completeness we give this algorithm below. We are assuming the system to be solved has the form $Ax = b$.

Let x^0 be given, $r^0 = b - Ax^0$, and define p^{-1} arbitrarily. For $k = 0, 1, \ldots$ until convergence perform the steps

$$\begin{aligned} Mz^k &= r^k \\ \beta_k &= (Mz^k, z^k)/(Mz^{k-1}, z^{k-1}) \qquad k \geq 1, \beta_0 = 0 \\ p^k &= z^k + \beta_k p^{k-1} \\ \alpha_k &= (Mz^k, z^k)/(Ap^k, p^k) \\ x^{k+1} &= x^k + \alpha_k p^k \\ r^{k+1} &= r^k - \alpha_k Ap^k. \end{aligned}$$

The matrix M is the preconditioning matrix and, for the approaches which we discuss, we can write

$$M = LDL^T$$

where D is a diagonal matrix and L has, for most of our runs, the same sparsity pattern as the lower triangular part of A, so that this factorization is not an exact factorization of A and

$$M = A + R.$$

The matrix R is sometimes referred to as the remainder matrix. We also consider some runs where L is allowed a limited number of nonzeros outside the sparsity pattern of A.

Although, for the regular structures which we consider in this paper, a much more economic route could be employed, we use a modified version of the Yale Sparse Matrix Package (Eisenstat *et al.* [9]) to compute the partial factorization used as M above. We have modified YSMP to prevent any fill-in, thus obtaining an ICCG(0) preconditioning. We also considered using drop tolerances, dropping any entry less than a prescribed value from the partial factors. This particular form of preconditioning is hard to implement with a specialized, simple approach.

4. Test problems.

We ran the preconditioned conjugate gradient method with the various orderings on a number of test problems. We discuss these problems in this section.

Two of our problems were the simple model problem

$$-\Delta u = f$$

on a square grid with Dirichlet boundary conditions using a five-point and a nine-point discretization of the Laplacian.

We wanted to check the effect of discontinuous coefficients, and so we include the problem

$$-\frac{\partial}{\partial x}\left(\lambda_x \frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(\lambda_y \frac{\partial u}{\partial y}\right) + \sigma u = f \qquad \text{in } \Omega =]0,1[\times]0,1[$$

with boundary condition $\dfrac{\partial u}{\partial n} = 0$ on $\partial\Omega$, the boundary of Ω, where n is the outer normal to $\partial\Omega$. The coefficients are defined by $\sigma = 0.01$ and λ_x, λ_y by

$$\begin{aligned}
&\text{In } \Omega_1 =]0,0.5] \times]0,0.5], \lambda_x = 1, \lambda_y = 1\\
&\text{In } \Omega_2 =]0.5,1[\times]0,0.5], \lambda_x = 100, \lambda_y = 1\\
&\text{In } \Omega_3 =]0,0.5] \times]0.5,1[, \lambda_x = 1, \lambda_y = 100\\
&\text{In } \Omega_4 =]0.5,1[\times]0.5,1[, \lambda_x = 100, \lambda_y = 100.
\end{aligned}$$

To model the anisotropy common in problems from the oil industry we use the problem

$$-\frac{\partial}{\partial x}\left(\lambda_x \frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(\lambda_y \frac{\partial u}{\partial y}\right) = f \qquad \text{in } \Omega =]0,1[\times]0,1[$$

with Dirichlet boundary conditions and $\lambda_x = 100$, $\lambda_y = 1$.

In both the discontinuous coefficient and anisotropic problems, we use a five-point discretization scheme. We call our five test problems LAPD5, LAPD9, EIST, and ANIS respectively, the third name being used because the problem was suggested by Stan Eisenstat (private communication 1984).

5. Results.

The computations were done on a CRAY 1-S and a CRAY X-MP using $\|\boldsymbol{r}^k\|_\infty \leq 10^{-6} \|\boldsymbol{r}^0\|_\infty$ as the stopping criterion and choosing the components of the initial vector $\boldsymbol{x}^0$ as random numbers between -1 and 1.

It should be noticed that, in these results, we have only included the number of iterations and not the computing times because, as the number of nonzeros in the incomplete factors is the same for all the orderings, the computing time is almost directly proportional to the number of iterations. In the column headed "Number of modifications" we give the number of modifications (including fill-in) that would occur in the factorization if the particular ordering were used for a complete $\boldsymbol{LU}$ factorization of the sparse matrix. $\boldsymbol{R}$ is the remainder matrix defined in the previous section. In addition to the number of entries in $\boldsymbol{R}$, we show both the (squared) Frobenius norm $\left(\sum_{i,j} |r_{ij}|^2\right)$ and the maximum entry in $\boldsymbol{R}$. We have performed experiments on larger and smaller grids and have chosen to illustrate the results by

runs on a 30 × 30 grid because that is the smallest one that exhibits the features we wish to discuss.

Table 5.1. *Results for LAPD5 on* 30 × 30 *grid.*

Ordering	Number of iterations	Number of modifications	Number of entries in $\boldsymbol{R}$	$\|\boldsymbol{R}\|_F^2$	$\max_{ij} \|r_{ij}\|$
row	23	24389	841	142.5	0.293
cm	23	16675	841	142.5	0.293
rcm	23	16675	841	142.5	0.293
block	23	24151	841	142.5	0.293
mind	39	7971	1582	467.3	0.541
rb	38	12853	1681	525.5	0.500
altd	38	9395	1681	525.5	0.500
zebra	28	31305	1261	180.2	0.268
nest	25	15228	1012	157.1	0.293
diss1	23	23996	871	145.1	0.293
diss2	24	24785	931	150.3	0.293
spiral	23	60173	841	141.0	0.295
4col	33	57253	1471	353.5	0.517
vdv1	20	23080	841	140.7	0.298
vdv2	20	17413	841	140.7	0.298
ujac	28	19218	1000	180.6	0.539
loc	23	24375	855	143.8	0.295

Table 5.2. *Results for LAPD9 on* 30 × 30 *grid.*

Ordering	Number of iterations	Number of modifications	Number of entries in $\boldsymbol{R}$	$\|\boldsymbol{R}\|_F^2$	$\max_{ij} \|r_{ij}\|$
row	15	23548	1624	600.	0.605
cm	16	32074	1735	613.	0.605
rcm	15	30450	1568	576.	0.663
block	16	30828	1792	560.	0.643
mind	24	14280	3093	2369.	2.049
rb	23	211764	2870	2386.	1.387
altd	23	211050	2870	2386.	1.387
zebra	27	30464	3248	2049.	1.160
nest	20	16162	2227	1077.	1.402
diss1	18	23938	1736	678.	1.097
diss2	18	24930	1960	835.	1.097
spiral	15	59278	1568	547.	0.672
4col	27	159118	3458	3039.	1.530
vdv1	15	22591	1621	574.	0.630
vdv2	15	23570	1621	574.	0.630
ujac	20	32531	1945	1077.	1.513
loc	16	23926	1664	641.	1.193

Table 5.3. *Results for EIST on* 30 × 30 *grid.*

Ordering	Number of iterations	Number of modifications	Number of entries in R	$\|R\|_F^2$	$\max \|r_{ij}\|$
row	68	24389	841	$0.73\,10^6$	38.03
cm	68	16675	841	$0.73\,10^6$	38.03
rcm	69	16675	841	$0.69\,10^6$	31.78
block	68	24151	841	$0.73\,10^6$	38.03
mind	108	7971	1582	$0.32\,10^7$	62.34
rb	107	12853	1681	$0.35\,10^7$	58.22
altd	107	9395	1681	$0.35\,10^7$	58.22
zebra	71	31305	1261	$0.88\,10^6$	26.79
nest	83	15228	931	$0.97\,10^6$	49.63
diss1	75	23996	871	$0.81\,10^6$	49.51
diss2	84	24785	931	$0.97\,10^6$	49.63
spiral	69	60173	841	$0.71\,10^6$	38.03
4col	103	57253	1471	$0.22\,10^7$	62.34
vdv1	68	23080	841	$0.69\,10^6$	31.78
vdv2	67	17413	841	$0.69\,10^6$	31.78
ujac	86	19218	1000	$0.11\,10^7$	60.18
loc	70	24375	855	$0.75\,10^6$	49.64

Table 5.4. *Results for ANIS on* 30 × 30 *grid.*

Ordering	Number of iterations	Number of modifications	Number of entries in R	$\|R\|_F^2$	$\max \|r_{ij}\|$
row	9	24389	841	$0.12\,10^4$	0.87
cm	9	16675	841	$0.12\,10^4$	0.87
rcm	9	16675	841	$0.12\,10^4$	0.87
block	9	24151	841	$0.12\,10^4$	0.87
mind	48	7971	1582	$0.18\,10^7$	49.51
rb	47	12853	1681	$0.21\,10^7$	49.51
altd	47	9395	1681	$0.21\,10^7$	49.51
zebra	9	31305	1261	$0.12\,10^4$	0.87
nest	26	15228	1012	$0.43\,10^6$	49.51
diss1	21	23996	871	$0.15\,10^6$	49.51
diss2	26	24785	931	$0.44\,10^6$	49.51
spiral	9	60173	841	$0.11\,10^4$	0.87
4col	34	57253	1471	$0.10\,10^7$	49.50
vdv1	9	23080	841	$1.08\,10^3$	0.86
vdv2	9	17413	841	$1.08\,10^3$	0.86
ujac	34	19218	1000	$0.52\,10^6$	49.51
loc	17	24375	855	$0.35\,10^5$	49.51

To get a feeling for the effect of dropping entries from the factorization, we show in Figures 5.1 to 5.7 the magnitude of the entries in a "full" factorization of the matrix for the model problem with a five-point scheme (LAPD5) using the symbols shown in Table 5.5 to indicate bands of magnitude. It is interesting to see that orderings

with much fill-in, for example the row ordering of Figure 5.1, often have most fill-ins of low magnitude. The opposite is the case with the minimum degree ordering shown in Figure 5.2.

We have also performed runs with a modified incomplete factorization preconditioner where the diagonal matrix $\boldsymbol{D}$ is chosen so that the row sums of the $\boldsymbol{R}$ matrix are all zero (see, for example, Gustafsson [15]). This strategy has been found beneficial on some of the easier problems from the oil industry. Our results on the

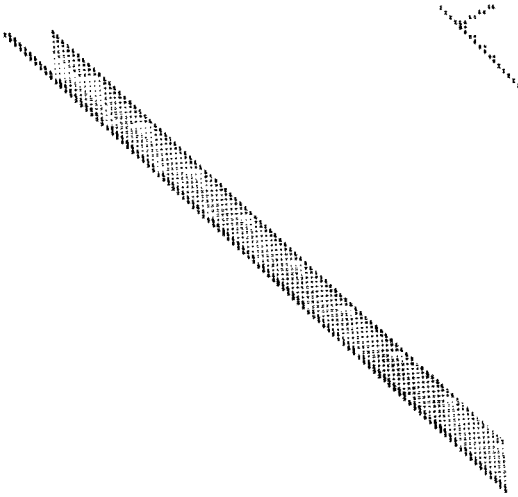

Fig. 5.1. *Row ordered matrix.*

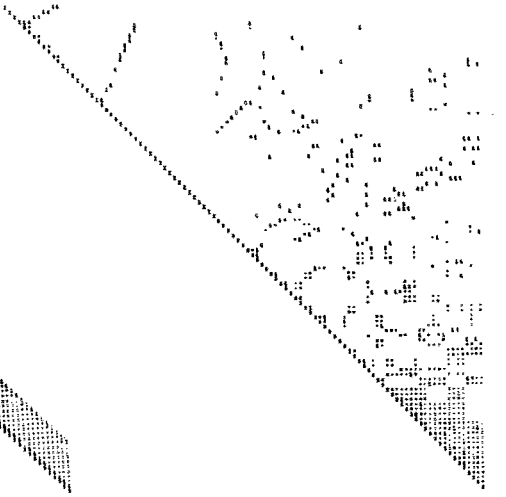

Fig. 5.2. *Minimum degree matrix.*

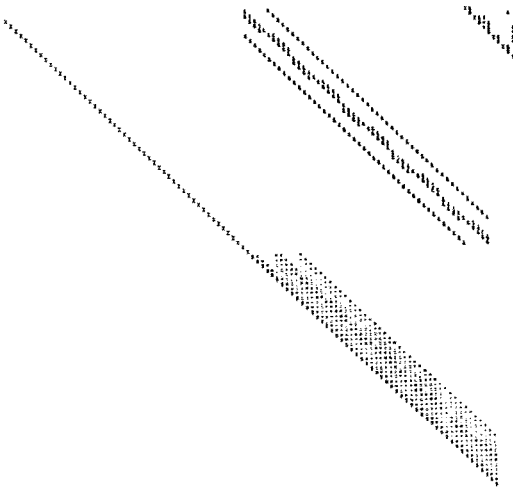

Fig. 5.3. *Red-black matrix.*

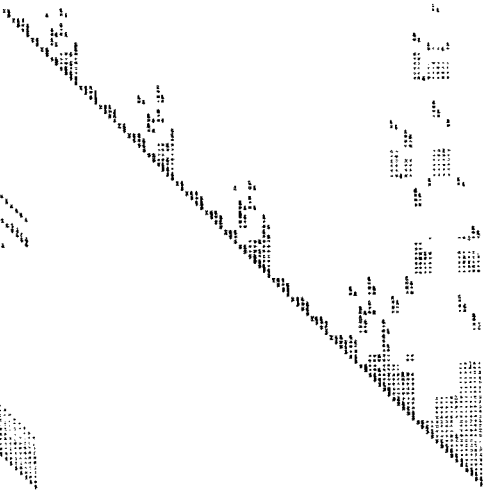

Fig. 5.4. *Nested dissection matrix.*

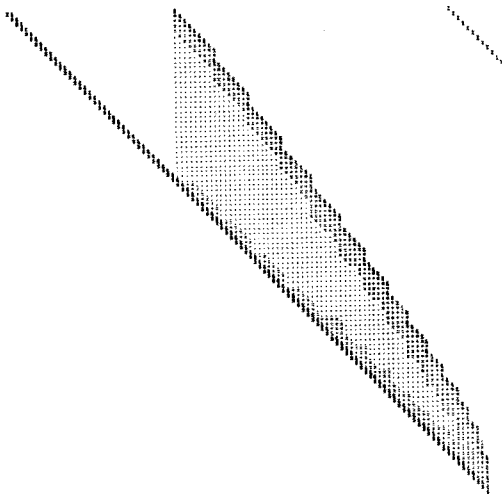

Fig. 5.5. *Spiral matrix.*

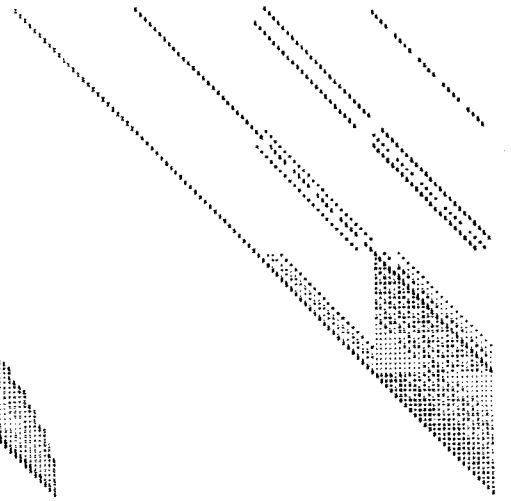

Fig. 5.6. *Four colour matrix.*

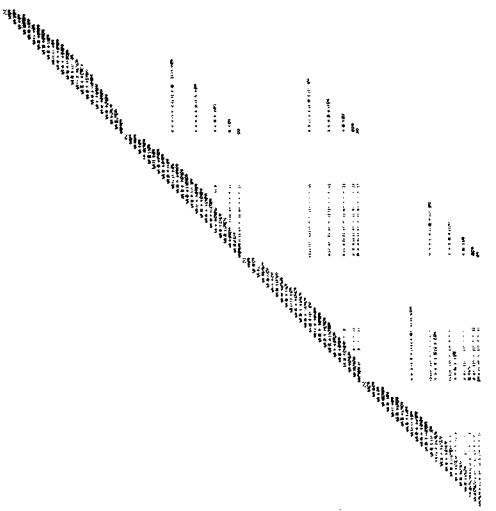

Fig. 5.7. *Van der Vorst Version* 2 *matrix.*

Table 5.5. *Symbols indicating order of magnitude of entries in factors.*

Magnitude of entries	Symbol
> 10	>
$(2, 10)$	%
$(1, 2)$	$
$(0.5, 1)$	&
$(0.1, 0.5)$	*
$(0.01, 0.1)$	+
$(0.001, 0.01)$	=
< 0.001	.

simple LAPD5 model problem are shown in Table 5.6 and indicate that some of the ordering schemes do benefit through the use of the modified preconditioner. However, some of the orderings do not converge if this preconditioner is used, probably because the preconditioning matrix is not positive definite in these cases. Note that the number of nonzeros in $\boldsymbol{R}$ increases when this scheme is used because our original choice for $\boldsymbol{D}$ ensured that the diagonal of $\boldsymbol{R}$ was zero.

Table 5.6. *Results for LAPD5 on* 30×30 *grid using modified ICCG*(0)

Ordering	Number of iterations	Number of modifications	Number of entries in $\boldsymbol{R}$	$\|\boldsymbol{R}\|_F^2$	$\max \|r_{ij}\|$
row	18	24389	1741	1010.	0.979
cm	18	16675	1741	1010.	0.979
rcm	18	16675	1741	1010.	0.979
block	18	24151	1741	1010.	0.979
mind	> 200	7971	2482	3568.	3.000
rb	> 200	12853	2581	4143.	3.000
altd	> 200	9395	2581	4143.	3.000
zebra	59	31305	2161	1151.	1.333
nest	38	15228	1912	1107.	1.666
diss1	22	23996	1771	1037.	1.666
diss2	29	24785	1831	1075.	1.166
spiral	15	60173	1741	898.	1.389
4col	> 200	57253	2371	2327.	2.160
vdv1	16	23080	1741	894.	0.960
vdv2	16	17413	1741	894.	0.960
ujac	81	19218	1900	1352.	2.742
loc	20	24375	1755	1039.	1.445

Our previous results have all used a preconditioning matrix whose pattern is identical to the coefficient matrix. We have also considered the case when some fill-in is allowed, both on the basis of a drop tolerance, where values less than a preset value are dropped from the structure, or on the basis of the level of fill-in. This second criterion is based on the observation that fill-ins that are themselves caused by fill-ins tend to be of lower numerical value, particularly for regular problems from discretizations of partial differential equations. Indeed, this is evident from studying the pattern of the magnitude of the fill-ins in Figures 5.1 to 5.7. We show the results of using a single level of fill-in in Table 5.7 where the number of entries in $\boldsymbol{L}$ should be contrasted with the 2640 entries present in the original lower triangle of $\boldsymbol{A}$. We see from this that, for most orderings, the reduction in the number of iterations does not quite compensate for the extra work for each iteration. The reverse is true of the drop tolerance results in Table 5.8 where the greater reduction in iterations more than compensates the increased storage and work for $\boldsymbol{L}$. For other values of drop tolerance than shown in that table, the reduction in iterations more or less matches the increase in work from a denser $\boldsymbol{L}$. For example, for the *row* ordering the number

Table 5.7. Results for LAPD5 on 30 × 30 grid using single level of fill-in.

Ordering	Number of iterations	Number of modifications	Number of entries in R	$\|R\|_F^2$	$\max \|r_{ij}\|$	Number of entries in L
row	17	24389	1653	2.43	0.087	3481
cm	17	16675	1653	2.43	0.087	3481
rcm	17	16675	1653	2.43	0.087	3481
block	17	24151	1837	2.44	0.088	3481
mind	23	7971	2467	38.81	2.509	4282
rb	16	12853	2016	16.47	0.090	4321
altd	16	9395	2016	16.47	0.090	4321
zebra	21	31305	2465	50.04	0.158	3901
nest	19	15228	2187	35.34	0.173	3652
diss1	18	23996	1738	26.70	0.158	3511
diss2	18	24785	1908	30.65	0.158	3571
spiral	17	60173	1624	23.82	0.186	3481
4col	25	57253	2653	70.94	0.220	4111
vdv1	17	23080	1651	25.20	0.190	3481
vdv2	17	17413	1651	25.20	0.190	3481
ujac	20	19218	2038	28.81	0.172	3640
loc	17	24375	1694	25.24	0.165	3495

Table 5.8. *Results for LAPD5 on 30 × 30 grid using drop tolerance of* 0.05.

Ordering	Number of iterations	Number of modifications	Number of entries in R	$\|R\|_F^2$	$\max \|r_{ij}\|$	Number of entries in L
row	12	24389	1595	4.260	0.039	4293
cm	10	16675	1540	2.846	0.041	4293
rcm	10	16675	1540	2.846	0.041	4293
block	11	24151	1526	3.451	0.045	4474
mind	10	7971	1657	2.285	0.049	5531
rb	8	12853	1484	1.683	0.042	4699
altd	8	9395	1069	1.284	0.048	5037
zebra	10	31305	2636	1.774	0.050	6605
nest	12	15228	2622	3.890	0.049	5574
diss1	12	23996	1752	4.173	0.049	4519
diss2	12	24785	2066	3.997	0.049	4971
spiral	11	60173	1513	3.779	0.047	4267
4col	9	57253	2945	1.664	0.050	5909
vdv1	12	23080	1612	3.862	0.048	4342
vdv2	10	17413	1560	2.732	0.046	4319
ujac	12	19218	2211	3.584	0.049	5229
loc	12	24375	1645	4.233	0.050	4373

of iterations for a tolerance of 0.001 and 0.0001 are 3 and 2 respectively, while the number of entries in L are 15929 and 22720 respectively. Another interesting feature of these preconditioners is that the relative performance of the different ordering schemes has changed, for example *mind*, *rb*, and *altd* do somewhat better when some fill-in is allowed in L. A reason for this is that many of the first level fill-ins for these

orderings are quite large, unlike *row*, for example, where the fill-ins rapidly decrease in value (see Figures 5.1 to 5.7).

6. Remarks and conclusions.

As one can see from the results in Section 5, the number of conjugate gradient iterations is not related to the number of fill-ins we are dropping (as was conjectured by Simon [22]) but is almost directly related to the norm of the residual matrix $\boldsymbol{R}$. We note that $\|\boldsymbol{R}\|$ can be used to estimate the condition numbers of incomplete factorization methods (Axelsson and Eijkhout, [3]). The number of fill-ins is related to the structure of the matrix, that is the structure of the underlying grid and to the ordering scheme we are using, but the incomplete decomposition is dependent on the values of the coefficients. The goal is not the same for direct methods, where the aim is usually to minimize storage and the number of floating-point operations. However, for incomplete elimination where we allow no fill-in, the structure of the approximation and the number of operations will be the same whatever the ordering is. What is interesting is how small $\boldsymbol{R}$ is or how close to the identity matrix $\boldsymbol{M}^{-1}\boldsymbol{A}$ is.

We show orderings like *mind* (minimum degree) with very few fill-ins but with a "large" $\boldsymbol{R}$ matrix, which give a large number of iterations, and orderings like *spiral* with a large number of fill-ins but with a "small" $\boldsymbol{R}$ which give a number of iterations comparable to the one given by the *row* ordering. This conclusion holds over all the problems. This can be seen from the Figures 5.1 to 5.7 showing the values of the fill-ins for the complete factorization.

Since the factor that determines the convergence of conjugate gradients is the spectrum of the iteration matrix $\boldsymbol{M}^{-1}\boldsymbol{A}$, we show an example of this spectrum for three of our orderings in Figures 6.1 to 6.3 below. The superiority of the *spiral* ordering over the *mind* ordering is evident.

Although this is only a sufficient condition, it seems that the orderings which are "local" in the sense that neighbouring nodes in the underlying mesh (or unknowns in the original system) have numbers that are not too far apart, give the best results. This is the case for the *row* ordering, for *cm* (or *rcm*), for *loc*, *vdv1*, *vdv2* and to a lesser extent for *zebra*. An example which proves that this is only a sufficient condition is

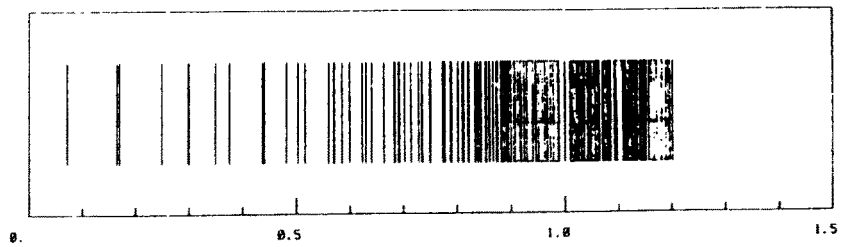

Fig. 6.1. *Spectrum of* $M^{-1}A$ *for row ordering.*

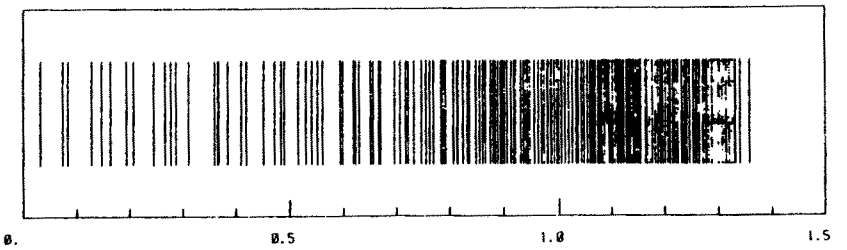

Fig. 6.2. *Spectrum of* $M^{-1}A$ *for mind ordering.*

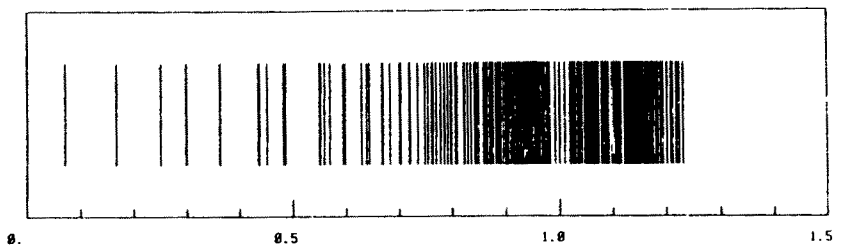

Fig. 6.3. *Spectrum of* $M^{-1}A$ *for spiral ordering.*

the *spiral* ordering which is not local and gives very good results on all of our examples. It remains an interesting (but non-trivial) problem: given a matrix find the "most" local ordering. This amounts to something very close to minimizing the bandwidth of a matrix. Another interpretation, suggested by the referee, is that "decoupling" orderings are bad. It appears also from the numerical results that, the harder is the problem at hand (discontinuous coefficients, anisotropy, etc...), the more important is the ordering for the incomplete decomposition. This may explain why, in previous work (for example Lichnewsky [18]), the influence of the ordering on the convergence of conjugate gradients has been underestimated. It is also interesting to see that, if some fill-in is allowed to the incomplete factorization, then the relative merits of the orderings differ. In particular, the red-black and alternating diagonal orderings are then very competitive.

Finally, although this study uses only finite-difference discretizations, our conclusions are more generally valid. For example, if one wants to use ICCG with finite-element methods, it should be beneficial to order the nodes with the Cuthill-McKee algorithm. Unfortunately, another conclusion of this study is that many of the orderings which are better suited for parallel computations, like the dissection methods or the node colouring ones do not give very good results. For the red-black ordering this is confirmed by the analysis of Kuo and Chan [17]. There seems to be an incompatibility between parallelism and good orderings for ICCG. A solution to this problem might be to use domain decomposition techniques and within each subdomain a "local" ordering of the nodes. Very recent orderings suggested by van der Vorst seem to have exactly this property although on larger grids they show no superiority in iteration count to the best simple orderings.

Acknowledgement.

We would like to thank the referees for their helpful comments. Work on this paper was started while the authors were visiting Stanford University. We are grateful for the support of Professor Gene Golub.

REFERENCES

[1] Adams, L. M. and Jordan, H. F. (1984). *Is SOR color-blind?* SIAM J. Sci. Stat. Comput. **7**, 490–506.
[2] Adams, L. M., LeVeque, R. J., and Young, D. M. (1988). *Analysis of the SOR iteration for the q-point Laplacian.* SIAM J. Numer. Anal. **25**, 1156–1180.
[3] Axelsson, O. and Eijkhout, V. (1988). *Robust vectorizable preconditioners for three-dimensional elliptic difference equations with anisotropy.* In *Algorithms and Applications on Vector and Parallel Computers.* H J J te Riele, Th J Dekker, and H A van der Vorst (editors). North-Holland, Amsterdam, New York, and London.
[4] Behie, A. and Forsyth, P. A. (1984). *Incomplete factorization methods for fully implicit simulation of enhanced oil recovery.* SIAM J. Sci. Stat. Comput. **5**, 543–561.
[5] Concus, P., Golub, G. H., and Meurant, G. (1985). *Block preconditioning for the conjugate gradient method.* SIAM J. Sci. Comput. **6**, 220–252.
[6] Cuthill, E. and McKee, J. (1969). *Reducing the bandwidth of sparse symmetric matrices.* Proceedings 24th National Conference of the Association for Computing Machinery, Brandon Press, New Jersey, 157–172.
[7] Duff, I. S., Erisman, A. M., and Reid, J. K. (1986). *Direct Methods for Sparse Matrices.* Oxford University Press, London.
[8] Duff, I. S., Erisman, A. M., and Reid, J. K. (1976). *On George's nested dissection method.* SIAM J. Numer. Anal. **13**, 686–695.
[9] Eisenstat, S. C., Gursky, M. C., Schultz, M. H., and Sherman, A. H. (1982). *Yale sparse matrix package. I: The symmetric codes.* Int. J. Numer. Meth. Engng. **18**, 1145–1151.
[10] Eisenstat, S. C., Elman, H. C., and Schultz, M. H. (1988). *Block-preconditioned conjugate gradient-like methods for numerical reservoir simulations.* SPE Reservoir Engineering. (Feb. 1988), 307–312.
[11] George, A. (1971). *Computer implementation of the finite-element method.* Report Stan CS-71-208, Ph.D Thesis, Department of Computer Science, Stanford University, Stanford, California.
[12] George, A. (1973). *Nested dissection of a regular finite-element mesh.* SIAM J. Numer. Anal. **10**, 345–363.
[13] George, A. and Liu, J. W. H. (1981). *Computer Solution of Large Sparse Positive-definite Systems.* Prentice-Hall, Englewood Cliffs, New Jersey.
[14] Golub, G. H. and Meurant, G. A. (1983). *Résolution Numérique des Grandes Systèmes Linéaires.* Eyrolles, Paris.
[15] Gustafsson, I. (1979). *Stability and rate of convergence of modified incomplete Cholesky factorization methods.* PhD dissertation, Chalmers University, Göteborg, Sweden.
[16] Hageman, L. A. and Young, D. M. (1981). *Applied Iterative Methods.* Academic Press, New York and London.
[17] Kuo, C. C. J. and Chan, T. F. (1988). *2-color Fourier analysis of iterative algorithms for elliptic problems with red/black ordering.* CAM Report 88-15, Computational and Applied Mathematics, University of California, Los Angeles.
[18] Lichnewsky, A. (1984). *Some vector and parallel implementations for preconditioned conjugate gradient algorithms.* In *High-speed Computation.* NATO ASI Series. Vol. F.7, edited by J. S. Kowalik. Springer-Verlag, Berlin, 343–359.
[19] Meijerink, J. A. and van der Vorst, H. A. (1977). *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix.* Math. Comp. **31**, 148–162.
[20] O'Leary, D. P. (1982). *Solving sparse matrix problems on parallel computers.* Report 1234, Computer Science Center, University of Maryland, Maryland.

[21] Schreiber, R. and Tang, W-P. (1982) *Vectorizing the conjugate gradient method.* Unpublished document. Department of Computer Science, Stanford University, Stanford, California.

[22] Simon, H. D. (1985). *Incomplete LU preconditioners for conjugate-gradient-type iterative methods.* Paper SPE 13533, Proceedings of the 1985 Reservoir Simulation Symposium, Dallas, Feb. 10-13, 1985, 387–396.

[23] Tinney, W. F. and Walker, J. W. (1967). *Direct solutions of sparse network equations by optimally ordered triangular factorization.* Proc. IEEE **55**, 1801–1809.