PARALLEL ITERATIVE SOLUTION OF SPARSE LINEAR SYSTEMS USING ORDERINGS FROM GRAPH COLORING HEURISTICS

Mark T. Jones and Paul E. Plassmann
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439-4801

MCS-P198-1290

December 1990

Abstract. The efficiency of a parallel implementation of the conjugate gradient method preconditioned by an incomplete Cholesky factorization can vary dramatically depending on the column ordering chosen. One method to minimize the number of major parallel steps is to choose an ordering based on a coloring of the symmetric graph representing the nonzero adjacency structure of the matrix. In this paper, we compare the performance of the preconditioned conjugate gradient method using these coloring orderings with a number of standard orderings on matrices arising from applications in structural engineering. Because optimal colorings for these systems may not be *a priori* known, we employ several graph coloring heuristics to obtain consistent colorings. Based on lower bounds obtained from the local structure of these systems, we find that the colorings determined by these heuristics are nearly optimal. For these problems, we find that the increase in parallelism afforded by the coloring-based orderings more than offsets any increase in the number of iterations required for the convergence of the conjugate gradient algorithm.

1. Introduction. The preconditioned conjugate gradient method [10] is one of the most successful iterative methods for solving large, sparse, symmetric, positive-definite linear systems. A preconditioner that has been shown to be very effective over a wide variety of problems is the incomplete Cholesky factorization [12]. Recently, several authors [4, 5, 15, 17] have examined the effect of matrix orderings based on multicolorings on the convergence properties of iterative methods. However, this work has considered only problems generated from regular grids, for which an optimal coloring is *a priori* known. These problems generate M-matrices [14] that are not representative of general systems of equations for which the straightforward incomplete Cholesky factorization may not exist.

In this paper, we consider sparse linear systems that arise from applications in structural engineering as well as the standard grid problems. For many of these problems, optimal multicolorings are not known. In general, the determination of an optimal coloring is an NP-hard problem [6]. Thus, we have explored the use of graph coloring heuristics to obtain the desired orderings. Our experimental results show that the combination of incomplete factorization and coloring heuristics results in a parallel preconditioner that is applicable to symmetric, positive-definite matrices arising in practical applications. We also compare the effectiveness of the coloring heuristics to some standard orderings: minimum degree, reverse Cuthill-McKee, and nested dissection.

The parallelism inherent in computing and applying the preconditioner is limited

by the solution of the triangular systems generated by the incomplete Cholesky factors [17]. It was first noted by Schreiber and Tang [16] that if the nonzero structure of the triangular factors is identical to that of the original matrix, then the minimum number of major parallel steps possible in the solution of the triangular system is given by the chromatic number of the symmetric adjacency graph representing those nonzeros. Thus, given the nonzero structure of a matrix A, an approach to generating greater parallelism is to compute a permutation matrix, P, based on a coloring of the symmetric graph $G(A)$. The incomplete Cholesky factor $\tilde{L}$ of the permuted matrix PAP^T is computed, instead of the factor based on the original matrix A.

In this permutation, vertices of the same color are grouped and ordered sequentially. As a consequence, during the triangular system solves, the unknowns corresponding to these vertices can be solved for in parallel, after the updates from previous color groups have been performed. The result of Schreiber and Tang states that the minimum number of inherently sequential computational steps required to solve one of the triangular systems, $\tilde{L}u = v$ or $\tilde{L}^T w = u$, is given by the minimum possible number of colors, or chromatic number, of the graph.

This reordering of the matrix is reminiscent of the reorderings done to minimize the fill in a direct factorization. However, in the incomplete factorization, fill that corresponds to initial zeros of the matrix is ignored. Instead, the permutation is chosen to minimize the number of communication steps inherent in the solution of the triangular systems generated by the incomplete factorization.

The organization of this paper is as follows. In Section 2 we briefly present some important issues in computing graph coloring and review two coloring heuristics that we have employed in our experiments. Here we also introduce the concept of an r-element, which can be used to generate an effective lower bound for the chromatic number of the graphs arising from our testbed applications. In Section 3 we introduce a suite of test problems and present experimental results. Finally, in Section 4 we summarize our research and suggest areas for future investigation.

2. Coloring Heuristics. Given the nonzero structure of an $n \times n$ symmetric matrix A, one can associate the symmetric graph $G(A) = (V, E)$ with the matrix, where the vertex set is given by $V = \{1, \ldots, n\}$ and the edge set is given by $E = \{ (i,j) \mid A_{ij} \neq 0, \text{ and } i \neq j\}$. We say that the function $\sigma : V \rightarrow \{1, \ldots, s\}$ is an s-coloring of $G(A)$, if $\sigma(i) \neq \sigma(j)$ for all edges (i,j) in E. The minimum possible value for s is known as the *chromatic number* of $G(A)$, which we denote as $\chi(G)$.

The question as to whether a general graph $G(A)$ is s-colorable is NP-complete [6]. It is known that unless $P = NP$, there does not exist a polynomial approximation scheme for solving the graph coloring problem [6]. In fact, the best polynomial time heuristic known [11] can theoretically guarantee a coloring of only size $c\,(n/\log n)\,\chi(G)$, where c is some constant.

It is therefore rather surprising that a coloring heuristic should perform well at all in practice. However, for the problems considered in this paper, we find that the heuristics obtain colorings only slightly worse than a lower bound determined from the local structure of the graphs considered. To obtain a lower bound, we employ the following well-known result which bounds the chromatic number by the size of any

complete subgraph in G.

Given a subset V' of the vertices V, the *induced subgraph* $G' = (V', E')$ of G contains the edges in the set $E' = \{ (i,j) \mid (i,j) \in E, \text{ and } i,j \in V' \}$. A complete subgraph of size r, which we call an *r-clique*, is a subset V' of V, with $|V'| = r$, for which every possible edge exists in the induced subgraph. Since the r vertices in an r-clique must be assigned different colors, we have the following lemma.

LEMMA 1. *If G contains an r-clique, then $\chi(G) \geq r$.*

The matrices we have considered for our experiments, with the exception of the standard grid problems, are all from structural engineering applications. We say that the finite-element model contains an *r-element* if there is an element in the model that contains r independent variables, which are represented in the model as being directly interacting. Thus, in the resulting linear system, we have that there is an r-clique associated with this r-element. By the above lemma, this clique corresponds to a lower bound for the chromatic number of the graph $G(A)$.

THEOREM 2. *If the finite element model contains an r-element, then $\chi(G) \geq r$.* The advantage of this observation is that it usually straightforward to determine the maximum sized r-element in the finite element model.

It is known that an optimal coloring can be obtained via a greedy heuristic if the vertices are visited in the correct order [1]. The basic structure of the greedy heuristic is shown below.

GREEDY HEURISTIC. *Compute a vertex ordering $\{v_1, \ldots, v_n\}$ for V. For $i = 1, \ldots, n$, set $\sigma(v_i)$ equal to the smallest available consistent color.*

The only aspect of the heuristic that must be specified is the method for obtaining the initial vertex ordering. In work by other authors, several strategies for obtaining this vertex ordering have been proposed. We have used two vertex ordering strategies.

The first of these heuristics is the *smallest-last ordering* (SLO) suggested by Matula, Marble, and Isaacson [13]. The second of these heuristics is the *incidence degree ordering* (IDO) suggested by Brélaz [2] and modified by Coleman and Moré in their work [3] on using coloring heuristics to obtain consistent partitions for use in Jacobian estimation.

These vertex orderings are defined as follows. Suppose that vertices $v_{i+1}, \ldots, v_n$ have been chosen. In the SLO strategy, vertex v_i is the vertex that has minimum degree in the induced subgraph of G with vertices $V \setminus \{v_{i+1}, \ldots, v_n\}$. With the IDO strategy, suppose that vertices $v_1, \ldots, v_{i-1}$ have been chosen. Vertex v_i is chosen to be a vertex whose degree is a maximum in the induced subgraph of G with vertices $\{v_1, \ldots, v_{i-1}\}$.

It is well known that the maximum degree of graph determines an upper bound for the chromatic number [1]. Let $\Delta(G) = \max_{v \in V} deg(v)$, where $deg(v)$ is the degree of vertex v in G. The upper bound is given by $\chi(G) \leq \Delta(G) + 1$. Note that the greedy heuristics will always satisfy this bound. Also, the graphs that arise from the applications we consider have bounded degree, independent of the size of system. Thus, the colorings determined by the heuristics will also be bounded.

Both these heuristics have the additional desirable property that they can be implemented to run in a time proportional to $\sum_{v \in V} deg(v)$, or the number of nonzeros in the matrix. This initial computational cost is modest compared to the time to

construct the incomplete factors and repeatedly solve the resulting triangular systems.

3. Experimental Results. In this section, we present experimental results that demonstrate the effect of matrix orderings derived from the coloring heuristics on the convergence of the conjugate gradient algorithm. The majority of the matrices in the test set are difficult problems that arise from finite element models from structural engineering applications.

A finite element model is constructed by piecing together many elements to approximate a structure. Each element typically contains $k \geq 2$ nodes, each node typically having $1 \leq d \leq 6$ degrees of freedom, resulting in kd unknowns per element. Adjacent elements share nodes, reducing the total number of unknowns. Also, the number of unknowns may be reduced by several other factors, including the application of constraints on the structure.

The subgraph containing these unknowns is usually completely connected and thus comprises an r-element, with $r = kd$. Many different element types, of course, can be included in a model. The coloring heuristics can be expected to perform well on these matrices because of the local nature of the models and the bound of kd on the maximum clique size.

For the sake of comparison, matrices arising from the five-point and nine-point finite-difference discretizations on a 30×30 grid were included in the test suite. It is well known that matrices arising from these stencils can be colored using two and four colors, respectively.

The CLAM package [9] was the environment used to run the experiments. In Table 1, the complete suite of test problems is described.[1] The diagonal of each matrix was scaled to be the identity matrix. For every problem, the right-hand side was the vector of ones, scaled to have a 2-norm of one. The initial guess for the conjugate gradient algorithm was the zero vector. Solutions were sought to a relative accuracy of 0.001, where the relative accuracy at step k is defined as $\|r_k\|_2/\|r_0\|_2$, where r_k is the residual at step k. When the incomplete Cholesky factorization failed, 0.01 was added to the diagonal until the factorization succeeded.

The first experiment shows the performance of the coloring heuristics on LAP5 and LAP9 for which an optimal coloring is known. The results in Table 2 show that both algorithms produce optimal or slightly suboptimal colorings but the IDO heuristic is slightly superior.

In the second set of experiments, the performance of the coloring heuristics is compared with three other ordering algorithms: minimum degree, reverse Cuthill-McKee (RCM), and nested dissection. A good description of these other algorithms can be found in [7]. In Appendix A, figures showing the orderings produced by the various heuristics for the PANEL problem are given. Points of interest are the number of iterations needed to solve the linear systems and the depth of the dependency graph for incomplete factorization and forward/backward substitution. Results are given in

[1] Because of the application of constraints and the elimination of superfluous degrees of freedom, the maximum clique size for several of the problems is less than the maximum number of degrees of freedom per node times the number of nodes per element. In addition, the PLANE problem was too large for our system to determine the condition number.

TABLE 1
The suite of test problems.

Name	Size	$\kappa(A)$	Description
LAP5	900	389	5-pt finite difference discretization on a 30x30 grid
LAP9	900	260	9-pt finite difference discretization on a 30x30 grid
CUBE3	180	3725	finite element model of a cube with 3x3x3 8-node elements with 3 degrees of freedom per node
CUBE4	363	1.1E4	finite element model of a cube with 4x4x4 8-node elements with 3 degrees of freedom per node
CUBE5	636	1.4E4	finite element model of a cube with 5x5x5 8-node elements with 3 degrees of freedom per node
CUBE6	1017	1.6E4	finite element model of a cube with 6x6x6 8-node elements with 3 degrees of freedom per node
CUBE7	1524	2.9E4	finite element model of a cube with 7x7x7 8-node elements with 3 degrees of freedom per node
CYL7	216	3.9E4	finite element model of a circular cylindrical shell with 36 4-node elements with *up to* 6 dof per node
CYL11	510	7.5E4	finite element model of a circular cylindrical shell with 100 4-node elements with *up to* 6 dof per node
PANEL	477	2.1E4	finite element model of a blade-stiffened panel with discontinuous stiffener. The model uses 86 4-node elements with *up to* 6 dof per node
PLT4	327	1.1E7	finite element model of a plate with 64 4-node elements with *up to* 6 dof per node
PLT9	1295	4.2E7	finite element model of a plate with 64 9-node elements with *up to* 6 dof per node
PLANE	2141	*	finite element model of a airplane with a mixture of 2-dimensional element types

TABLE 2
The performance of the coloring heuristics.

Heuristic	Problem	Number of colors
Known Coloring	LAP5	2
SLO	LAP5	4
IDO	LAP5	2
Known Coloring	LAP9	4
SLO	LAP9	7
IDO	LAP9	5

Table 3, where $\Delta(G)$ is the maximum degree of a node in $G(A)$ and $cl(G)$ is the largest apparent clique,[2] or r-element, used in constructing $G(A)$. For each problem, the numbers in the first row are the number of iterations required for convergence, and the numbers is the second row give the number of levels in the dependence graph. These results show that the coloring heuristics produce orderings that are very close to the largest clique size in the graphs of the matrices. The amount of parallelism available from the colorings is much greater than that produced by the other orderings; more than enough to offset the larger number of iterations required.

An examination of the CUBE problems reveals how the coloring heuristics perform as the size of the problem increases. The IDO heuristic did very well; the number of colors increased only slightly as the size of the problem was increased by a large factor. The effect of using elements with more nodes can be seen in the two plate problems, PLT4 and PLT9, where the number of colors increased by approximately a factor of two as the number of nodes per element went from 4 to 9.

The reverse Cuthill-McKee (RCM) ordering seems to be the best ordering to choose to minimize the number of iterations. However, the triangular system solves on parallel machines are extremely communication intensive. As a rough measure of the communication complexity of the different orderings we can use the product of the number of iterations and the dependency graph depth. In Figure 1 we show the ratio of this complexity measure for the RCM ordering when compared to the IDO ordering. The results for the PLANE problem show a substantial improvement when using the IDO ordering. We note that the results for the CUBE and CYL problems indicate that there is a improvement in this ratio as the size of the problems increases. Even though the maximum clique size increases for the PLT problem when the number of node is increased from 4 to 9, we see an improvement in this ratio.

When the number of colors does not allow for enough parallelism, the number of colors can be reduced by setting selected off-diagonals to zero. For example, if A is a 3×3 matrix in which node 1 is colored red and nodes 2 and 3 are colored black, then by setting $a_{1,2}$, $a_{1,3}$, $a_{2,1}$, and $a_{3,1}$ to 0, all three nodes can be colored red. Of course, removing these off-diagonals will increase the number of iterations required for convergence. To observe the effect of removing these nonzeroes on the number of iterations, we reran the problems used in experiment 2, this time using the coloring heuristics followed by a coloring reduction. The coloring reduction compacted the nodes in a color into the next color if the number of nodes in the color was less than a specified minimum value. The first column in Table 4 gives the minimum number of elements allowed in a color. The results show that a significant gain in parallelism can be realized without paying an excessive price in terms of an increase in the number of iterations.

4. Conclusions. Our results have shown that for these systems the coloring heuristics have close to optimal colorings for the graphs based on these matrices. In

[2] The determination of the largest clique in a general graph is an NP-hard problem. We have used a heuristic to find a *large* clique, which is reported as $cl(G)$. Therefore, there may exist a larger clique than this number.

TABLE 3
Comparison of orderings.

Name	$\Delta(G)$	$cl(G)$	Nested Dissection	Reverse Cuthill-McKee	Minimum Degree	SLO	IDO
PLANE	68	20	1251	1362	1082	1636	1585
			(78)	(542)	(294)	(23)	(21)
CUBE3	81	24	37	34	33	59	69
			(68)	(111)	(81)	(26)	(25)
CUBE4	81	24	84	48	81	104	90
			(108)	(148)	(93)	(26)	(26)
CUBE5	81	24	100	48	102	130	128
			(123)	(207)	(171)	(44)	(27)
CUBE6	81	24	163	67	128	194	178
			(164)	(224)	(202)	(31)	(29)
CUBE7	81	24	175	80	185	215	223
			(176)	(269)	(263)	(42)	(29)
CYL7	54	18	80	101	92	86	107
			(47)	(79)	(54)	(20)	(19)
CYL11	45	16	108	148	195	199	177
			(75)	(138)	(90)	(20)	(16)
PANEL	42	12	73	70	82	84	100
			(47)	(103)	(38)	(15)	(14)
PLT4	43	16	556	567	520	411	470
			(81)	(115)	(68)	(20)	(18)
PLT9	124	35	2013	2015	1744	2542	2568
			(143)	(406)	(142)	(36)	(35)

TABLE 4
Effect of coloring reduction.

Name	Minimum Color Size	SLO # of Its.	SLO # of Levels	IDO # of Its.	IDO # of Levels
PLANE	100	1784	14	1725	13
CUBE3	10	77	12	81	13
CUBE4	20	119	13	115	13
CUBE5	30	160	15	145	13
CUBE6	40	269	18	191	18
CUBE7	50	294	22	213	24
CYL7	20	107	8	111	7
CYL11	40	252	9	189	8
PANEL	30	85	12	104	10
PLT4	20	421	12	554	11
PLT9	100	3759	11	3633	11

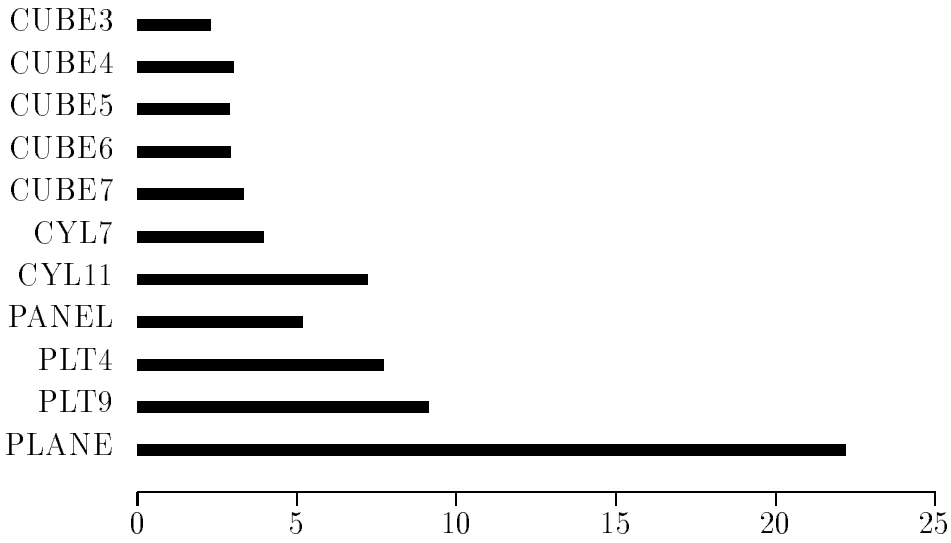

FIG. 1. *The improvement in communication complexity afforded by the IDO ordering when compared to the RCM ordering.*

addition, this work has shown that the increase in parallelism afforded by this reordering more than offsets any of the increases seen in the number of iterations required for convergence over other commonly used ordering heuristics. It was also observed that the benefit reaped from reducing the number of colors by ignoring specific off-diagonals is large enough to justify the increased number of iterations.

A significant pitfall for the straightforward incomplete factorization algorithm is that it may fail to produce a positive-definite factorization, even though the matrix is positive definite. Because positive definiteness is required for the conjugate gradient method [8], some mechanism must be included to deal with the detection of indefiniteness during the incomplete factorization process. For the results presented in this paper, we have used the technique of adding an increasing multiple of the diagonal until the matrix can be successfully factored. However, we note that an improvement of these methods for forcing a positive definite factorization while still maintaining a good preconditioner is an important area for future research.

Acknowledgement. This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under contract W-31-109-Eng-38. The authors would like to thank Chris Bischof for his comments on a preliminary version of this paper.

REFERENCES

[1] B. BOLLOBÁS, *Graph Theory*, Springer-Verlag, 1979.

[2] D. BRÉLAZ, *New methods to color the vertices of a graph*, Comm. ACM, 22 (1979), pp. 251–256.

[3] T. F. COLEMAN AND J. J. MORÉ, *Estimation of sparse Jacobian matrices and graph coloring problems*, SIAM Journal on Numerical Analysis, 20 (1983), pp. 187–209.

[4] I. S. DUFF AND G. A. MEURANT, *The effect of ordering on preconditioned conjugate gradients*, BIT, 29 (1989), pp. 635–657.

[5] H. C. Elman and E. Agrón, *Ordering techniques for the preconditioned conjugate gradient method on parallel computers*, Computer Physics Communications, 53 (1989), pp. 253–269.
[6] M. R. Garey and D. S. Johnson, *Computers and Intractability*, W.H. Freeman, 1979.
[7] A. George and J. W.-H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, 1981.
[8] G. H. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, 1983.
[9] W. D. Gropp, D. E. Foulser, and S. Chang, *CLAM User's Guide*, Scientific Computing Associates, Inc., 1989.
[10] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 409–436.
[11] D. S. Johnson, *Worst case behavior of graph coloring algorithms*, in Proceedings 5th Southeastern Conference on Combinatorics, Graph Theory, and Computing, Utilitas Mathematica Publishing, 1974, pp. 513–527.
[12] T. A. Manteuffel, *An incomplete factorization technique for positive definite linear systems*, Mathematics of Computation, 34 (1980), pp. 473–497.
[13] D. Matula, G. Marble, and J. Isaacson, *Graph coloring algorithms*, in Graph Theory and Computing, R. Read, ed., Academic Press, 1972, pp. 104–122.
[14] J. Meijerink and H. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Mathematics of Computation, 31 (1977), pp. 148–162.
[15] J. M. Ortega, *Orderings for conjugate gradient preconditionings*, Tech. Rep., University of Virginia, 1990.
[16] R. Schreiber and W.-P. Tang, *Vectorizing the conjugate gradient method*, Unpublished manuscript, Department of Computer Science, Stanford University, 1982.
[17] H. A. van der Vorst, *High performance preconditioning*, SIAM Journal on Scientific and Statistical Computing, 10 (1989), pp. 1174–1185.

Appendix A.

Fig. 2. *Ordering produced by SLO for the PANEL problem.*

FIG. 3. *Ordering produced by IDO for the PANEL problem.*

FIG. 4. *Ordering produced by minimum degree for the PANEL problem.*

FIG. 5. *Ordering produced by Reverse Cuthill-McKee for the PANEL problem.*

FIG. 6. *Ordering produced by nested dissection for the PANEL problem.*