D8

N78-32474

REDUCTION OF MATRIX WAVEFRONT FOR NASTRAN

Gordon C. Everstine
David W. Taylor Naval Ship Research and Development Center

SUMMARY

The three grid point resequencing algorithms most often run by NASTRAN users are compared for their ability to reduce matrix root-mean-square (rms) wavefront, which is the most critical parameter in determining matrix decomposition time in NASTRAN. The three algorithms are Cuthill-McKee (CM), Gibbs-Poole-Stockmeyer (GPS), and Levy. The first two (CM and GPS) are in the BANDIT program, and the Levy algorithm is in WAVEFRONT. Results are presented for a diversified collection of 30 test problems ranging in size from 59 to 2680 nodes. It is concluded that GPS is exceptionally fast and, for the conditions under which the test was made, the algorithm best able to reduce rms wavefront consistently well.

INTRODUCTION

A central feature of structural analysis with NASTRAN is the factoring (or decomposition) of a matrix into upper and lower triangular forms. NASTRAN's current triangular decomposition algorithm is an active column routine similar to a variable band or wavefront approach. As such, the computer time required to perform a matrix decomposition depends strongly on the sequence assigned to the grid point labels.

For real, symmetric decomposition, for example, the time T required can be estimated from the relation (ref. 1)

$$T = \frac{1}{2} T_m \sum_{i=1}^{N} c_i^2 \qquad (1)$$

where N = matrix order,
c_i = number of active columns in matrix row i, and
T_m = time for multiply-add operation (an experimentally determined machine time constant).

The time T is sequence-dependent since the c_i's are sequence-dependent.

Since c_i is sometimes referred to as the row wavefront for row i, equation (1) can alternatively be written in terms of the root-mean-square (rms) wavefront, W_{rms}:

$$T = \frac{1}{2} T_m N W_{rms}^2 \qquad (2)$$

Because of the large size of Σc_i^2 in equation (1) for many problems, this latter form of the timing equation is often the more convenient one to use in practice. For reference purposes, typical values for the machine constant T_m are listed in Table 1 for several computers.

Core storage requirements for matrix decomposition are also dependent on the nodal sequence, the most critical parameter being the maximum matrix wavefront W_{max}, which is the maximum value of any c_i.

Thus, for most efficient matrix decomposition, the user would like to assign grid point labels so as to minimize both W_{rms} and W_{max}, with the former probably the more important. Unfortunately, it is often difficult for users to know how to sequence the nodes to effect a good numbering, particularly for large complicated meshes or those generated automatically on a computer. As a result, many users turn to NASTRAN preprocessors which automate the labeling process. The two most often run by NASTRAN users are BANDIT (refs. 2-4), which contains the Cuthill-McKee (CM) (ref. 5) and Gibbs-Poole-Stockmeyer (GPS) (refs. 6-7) schemes, and WAVEFRONT (ref. 8), which contains the Levy resequencing algorithm (refs. 9-10). Both preprocessors read NASTRAN data decks as input, resequence the nodes, and generate NASTRAN SEQGP bulk data cards (which tell NASTRAN what the new internal sequence should be).

The questions which then naturally arise are: How do these three resequencing algorithms (CM, GPS, and Levy) compare for their ability to reduce rms wavefront? What are the time and core requirements of the three algorithms?

These questions were addressed recently in another paper (ref. 11), in which the algorithms were also compared for matrix profile reduction. Complete descriptions of the test problems used for the comparison were presented. The purpose of this paper, which is adapted from reference 11, is to summarize for the NASTRAN user community the rms wavefront results obtained.

Subsequent sections of this paper present precise definitions of the relevant terms, a brief description of the three algorithms to be tested, the ground rules of the test, and the test results.

DEFINITIONS

Although the definitions given here are reasonably standard (at least in finite element circles), uniformity of definitions and notation among the various workers in the field does not yet exist.

Given a symmetric square matrix A of order N, we define a "row bandwidth" b_i for row i as the number of columns from the first nonzero in the row to the

diagonal, inclusive. Numerically, b_i exceeds by unity the difference between i and the column index of the first nonzero entry of row i of A. Then the matrix bandwidth B and profile P are defined as

$$B = \max_{i \le N} b_i \tag{3}$$

$$P = \sum_{i=1}^{N} b_i \tag{4}$$

Let c_i denote the number of active columns in row i. By definition, a column j is active in row i if $j \ge i$ and there is a nonzero entry in that column in any row with index $k \le i$. The matrix wavefront W is then defined as

$$W = \max_{i \le N} c_i \tag{5}$$

Sometimes c_i is referred to as the row wavefront for row i. Since the matrix A is symmetric,

$$P = \sum_{i=1}^{N} b_i = \sum_{i=1}^{N} c_i \tag{6}$$

The wavefront W is sometimes called the maximum wavefront W_{max} to distinguish it from the average wavefront W_{avg} and root-mean-square wavefront W_{rms} defined as

$$W_{avg} = \frac{1}{N} \sum_{i=1}^{N} c_i = \frac{P}{N} \tag{7}$$

$$W_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} c_i^2} \tag{8}$$

From these definitions, it follows that, for a given matrix,

$$W_{avg} \le W_{rms} \le W_{max} \le B \le N \tag{9}$$

The first two inequalities would be equalities only for uninteresting special cases such as diagonal matrices.

We define the degree d_i of node i as the number of other nodes to which it is connected; i.e., more precisely, d_i is the number of nonzero off-diagonal terms in row i of the matrix A. (This implies, for example, that all nodes in the same finite element are "connected" to each other.) Hence, the maximum nodal degree M is

$$M = \max_{i \le N} d_i \tag{10}$$

The number of unique edges E is defined as the number of nonzero off-diagonal terms above the diagonal. Hence, for a symmetric matrix,

$$E = \frac{1}{2} \sum_{i=1}^{N} d_i \tag{11}$$

Thus the total number of nonzeros in A is 2E+N, and the density ρ of the matrix A is

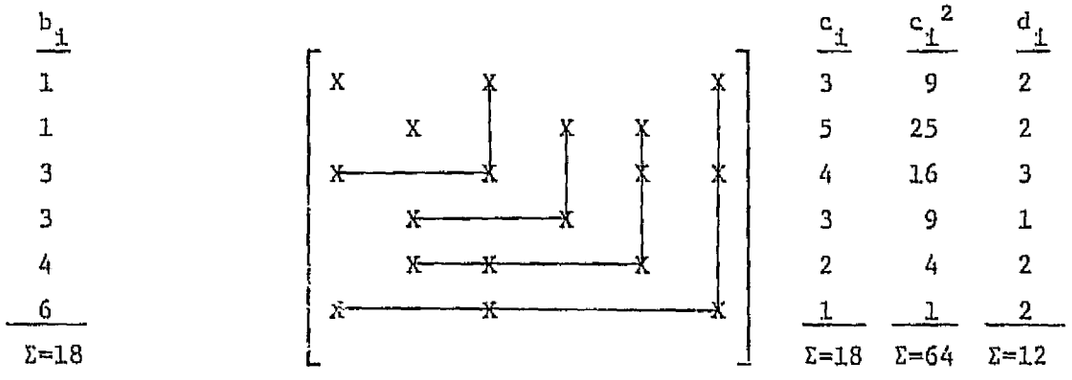
$$\rho = (2E+N)/N^2 \tag{12}$$

Note that, in these definitions, the diagonal entries of the matrix A are included in b_i and c_i (and hence in B, P, W_{max}, W_{avg}, and W_{rms}). These definitions make it easy to convert the various parameters from one convention (including the diagonal) to the other (not including the diagonal).

Also note that, in this context, the order N of the matrix A is sometimes taken to be the same as the number of nodes. In general finite element usage, however, each node (grid point) has several degrees of freedom (DOF), not just one. For structures having, say, six DOF per node, the actual DOF values of B, W_{max}, W_{avg}, or W_{rms} would be (in the absence of constraints) six times their corresponding grid point values.

Example

Definitions (3)-(12) can be illustrated by the following simple example. Consider the matrix shown below, in which nonzeros are indicated by X's.



In each row and column a line is drawn from the first nonzero to the diagonal. Thus b_i is the number of columns traversed by the solid line in row i. Similarly, the number of active columns c_i in row i is the number of vertical lines in row i to the right of and including the diagonal. Thus, from the definitions, B=6, W_{max}=5, P=18, W_{avg}=3.0, W_{rms}=3.3, M=3, E=6, and ρ= 50.0%.

THE RESEQUENCING ALGORITHMS TESTED

The three algorithms tested are Cuthill-McKee (CM) (ref. 5), Gibbs-Poole-Stockmeyer (GPS) (refs. 6-7), and Levy (refs. 9-10). In this section each algorithm is described briefly, with details concerning the specific implementation used. It is recognized that one cannot really evaluate algorithms per se, but only specific implementations of algorithms.

Cuthill-McKee (CM) (ref. 5)

The original version of CM operated generally according to the following procedure: Among the nodes of low degree, select as potential starting nodes those which can root a graph of minimal width. (The term "starting node" refers to a node which is assigned the label 1 in the new sequence.) For each potential starting node, assign the labels 2 through N by numbering those adjacent to new label I (and unnumbered) in order of increasing degree, starting with I=1 and continuing with increasing I until all nodes are sequenced. Of the sequences attempted, select the one having the smallest bandwidth.

The implementation of CM used in these tests is that appearing in the BANDIT computer program, version 8 (refs. 2-4), which contains a version of CM differing from the original algorithm in two ways. First, the new sequence obtained is reversed (by setting I to N+1-I for each I), since it was observed by George (ref. 12) and proved by Liu and Sherman (ref. 13) that such a reversal (which preserves matrix bandwidth) will often reduce the matrix profile and never increase it. Second, of all sequences attempted, the one with the smallest rms wavefront is the one selected. Except for these two changes, the CM computer code is that originally written by Cuthill and McKee.

The data structure originally used by CM required about (M+8)N words of core storage for the problem-dependent arrays, where N is the number of grid points and M is the maximum nodal degree. In the BANDIT implementation of CM, word packing is used to reduce the storage requirements to (M/L+8)N, where L, the packing density, is an integer (between 2 and 6, inclusive) which depends on the problem size and the computer being used. On a CDC 6400, for example, the CP time penalty for packing is about 80 μsec per pack or unpack.

Gibbs-Poole-Stockmeyer (GPS) (refs. 6-7)

The GPS algorithm differs from CM primarily in the selection of starting nodes. In GPS, only one starting node is selected, and it is an endpoint of a pseudo-diameter of the graph associated with the matrix. Thus, the structure need be numbered only once, using a procedure which is similar to the CM numbering algorithm.

The storage requirements of GPS are identical to those of CM, including the use of integer packing in the BANDIT (version 8) implementation, which is the form of GPS used for the testing. The original GPS code was written by the developers (ref. 7).

Levy (refs. 9-10)

Unlike CM and GPS, which were developed to reduce matrix bandwidth and profile, the Levy algorithm was designed specifically to reduce the maximum matrix wavefront, W_{max}. The algorithm operates generally according to the following recursive procedure: Given the first I nodes of a new sequence, the node selected as I+1 is the one for which the increase in the row wavefront between rows I and I+1 will be minimum. Levy calls this a "minimum growth" method.

This procedure is followed for one or more trial starting nodes, and the sequence yielding the smallest wavefront W_{max} is selected. The first sequence attempted uses as the starting node either a user-selected node or a node of minimum degree. The latter option was chosen for these tests since it was felt that, for a production mode program, the user ought to be relieved of the burden of specifying a starting node. The second and succeeding sequences attempted by the Levy algorithm select starting nodes randomly. The number of new sequences to be attempted must be specified by the user. After some preliminary experimentation to estimate the speed of the algorithm, it was decided to request ten sequencing attempts for each test problem. Clearly a different number would yield different results.

The implementation used for the tests was that obtained by the author from Levy in 1973, the only change being that the sequence selected as best is the one yielding the smallest rms wavefront W_{rms}. Since the Levy algorithm aborts any resequencing attempt in progress once it determines that it cannot reduce the previous best W_{max}, the sequence finally selected will be the one among those carried to completion yielding the smallest W_{rms}.

The Levy data structure requires about 6N+10E words of core storage for the problem-dependent arrays, where N is the number of grid points and E is the number of unique edges. The code was not rewritten to use word packing for the tests.

TEST RESULTS AND DISCUSSION

The three grid point resequencing algorithms described in the preceding section were tested on a collection of 30 finite element meshes. These problems were collected over a period of several years from NASTRAN users representing various U.S. Navy, Army, Air Force, and NASA laboratories. Since these meshes are described in detail and plotted elsewhere (ref. 11), that information need not be repeated here. In general, however, the collection is probably large enough and diversified enough to provide a good test of nodal resequencing algorithms.

The nodes for the 30 test problems were resequenced using the three algorithms, the objective being to reduce rms wavefront. All computer runs were made on a CDC 6400 computer under the NOS/BE operating system. The source code

was compiled using the FTN compiler, OPT=1. For reference purposes, a CDC 6400 is about one-third as fast as a CDC 6600.

The results of the tests appear in Table 2. In addition to the rms wavefront obtained by each algorithm, Table 2 also lists, for each algorithm, the CP time expended and the storage requirements for the problem-dependent arrays. In the case of CM and GPS, which use word packing, the worst-case of half-word packing is assumed. The CP times listed do not include basic setup of the arrays.

The first conclusion to be drawn from Table 2 is that, for most problems, all three algorithms achieve about the same reduction in rms wavefront. This is, perhaps, somewhat unexpected since CM and GPS were designed primarily to reduce matrix bandwidth, whereas the Levy scheme was designed to reduce matrix wavefront. For the 30 problems, Levy achieved the best reduction in rms wavefront 13 times, GPS 11 times, and CM 5 times. However, on four occasions (N=503, N=607, N=878, and N=918) Levy did significantly worse than the best achieved; on three occasions (N=209, N=245, and N=1242) GPS did significantly worse; and on two occasions (N=245 and N=592) CM did significantly worse.

The second, and perhaps most striking, conclusion to be drawn from Table 2 is that GPS is exceptionally fast. In all cases, CM is second fastest, the Levy algorithm slowest. The user, of course, has some control over the running time of the Levy program (but not of CM and GPS) through his specification of the number of resequencing attempts.

The third conclusion to be drawn from Table 2 is that the Levy algorithm, as is, requires considerably more array storage than either CM or GPS, which use the same data structure. In fact, for the Levy program, one problem (N=2680) was too big for a CDC 6400 and could not be run. Clearly, the program could be rewritten to use word packing (as CM and GPS do), but this may be a nontrivial task, since the programmer has to decide which arrays to pack to yield the best compromise between storage and CP time. (Word packing, of course, saves core at the expense of CP time.)

Table 2 indicates that Levy's wavefront reduction performance was generally best for the smaller problems and GPS's was generally best for the larger problems. This is probably due to the author's choice of ten sequencing attempts for the Levy algorithm. As the problems get larger, the probability of Levy's selecting a good starting node at random goes down. One can infer that the algorithm's performance would improve if the program were allowed to run longer. However, whether the expenditure of more computer time is justified would be a matter for each user to decide. One issue that enters into such a decision is the number of times a given matrix problem is to be solved. If a given problem is to be solved many times (as, for example, in nonlinear analysis), or if many right-hand sides are involved (as, for example, in time-dependent problems), the time spent in sequencing becomes less important.

One might also infer that the performance of the Levy algorithm would improve if trial starting nodes were selected using a strategy such as that used in CM or GPS, rather than at random. While this may be true sometimes, it

was not true for the test problem on which Levy performed the worst (N=918), because for this problem the first trial starting node selected by Levy (which uses a node of minimum degree for the first attempt) was the same starting node selected by GPS. This same problem (N=918) was also run by Gibbs with his profile algorithm (ref. 14) (which is a hybrid of GPS and King (ref. 15), the latter being similar to Levy) with good results. This would indicate that Gibbs' modification to the King numbering approach (given a starting node) has a significant effect for some problems.

Overall, GPS's combination of speed and consistency probably rate it the best algorithm of the three for rms wavefront reduction. Previous testing (ref. 3) has already shown it to be an excellent algorithm for matrix bandwidth reduction, for which it was designed.

Finally, the three algorithms tested were selected because of their heavy use by NASTRAN users. However, it would be interesting to see how other strategies, including Gibbs-King (ref. 14) and Snay (ref. 16), would perform on the same data. Both give good results for profile reduction and hence would probably also do well in rms wavefront reduction.

REFERENCES

1. The NASTRAN Theoretical Manual. NASA SP-221 (04), Washington, D.C., December 1977.

2. Everstine, G.C.: The BANDIT Computer Program for the Reduction of Matrix Bandwidth for NASTRAN. NSRDC Report 3827, Naval Ship Research and Development Center, Bethesda, MD, March 1972.

3. Everstine, G.C.: Recent Improvements to BANDIT. NASTRAN: Users' Experiences, NASA TM X-3278, Sept. 1975, pp. 511-521.

4. Everstine, G.C.: BANDIT User's Guide. TM-184-77-03, David W. Taylor Naval Ship Research and Development Center, Bethesda, MD, 1977.

5. Cuthill, E.H.; and McKee, J.M.: Reducing the Bandwidth of Sparse Symmetric Matrices. Proc. 24th Nat. Conf., Assn. for Computing Machinery, ACM Pub. P69, New York, 1969, pp. 157-172.

6. Gibbs, N.E.; Poole, W.G., Jr.; and Stockmeyer, P.K.: An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix. SIAM J. Numer. Anal., vol. 13, no. 2, April 1976, pp. 236-250.

7. Crane, H.L., Jr.; Gibbs, N.E.; Poole, W.G., Jr.,; and Stockmeyer, P.K.: Algorithm 508: Matrix Bandwidth and Profile Reduction. ACM Trans. Math. Software, vol. 2, no. 4, Dec. 1976, pp. 375-377.

8. Levy, R.: Structural Stiffness Matrix Wavefront Resequencing Program (WAVEFRONT). JPL Tech. Rep. 32-1526, vol. XIV, 1972, pp. 50-55.

9. Levy R.: Resequencing of the Structural Stiffness Matrix to Improve Computational Efficiency. JPL Quart. Tech. Review, vol. 1, no. 2, July 1971, pp. 61-70.

10. Levy, R.; and Wall, S.: Savings in NASTRAN Decomposition Time by Sequencing to Reduce Active Columns. NASTRAN: Users' Experiences, NASA TM X-2378, Sept. 1971, pp. 627-631.

11. Everstine, G.C.: A Comparison of Three Resequencing Algorithms for the Reduction of Matrix Profile and Wavefront. To appear.

12. George, A.: Computer Implementation of the Finite Element Method. Ph.D. Dissertation, Tech. Rep. STAN-CS-71-208, Computer Sci. Dept., Stanford Univ., Stanford, CA, 1971.

13. Liu, W.-H.; and Sherman, A.H.: Comparative Analysis of the Cuthill-McKee and the Reverse Cuthill-McKee Ordering Algorithms for Sparse Matrices. SIAM J. Numer. Anal., vol. 13, no. 2, April 1976, pp. 198-213.

14. Gibbs, N.E.: Algorithm 509: A Hybrid Profile Reduction Algorithm. ACM Trans. Math. Software, vol. 2, no. 4, Dec. 1976, pp. 378-387.

15. King, I.P.: An Automatic Reordering Scheme for Simultaneous Equations Derived from Network Systems. Intl. J. Numer. Meth. in Engr., vol. 2, no. 4, 1970, pp. 523-533.

16. Snay, R.A.: Reducing the Profile of Sparse Symmetric Matrices. Bulletin Géodésique, vol. 50, no. 4, 1976, pp. 341-352.

TABLE 1.-MULTIPLY-ADD TIME CONSTANTS (T_m)

(Double precision for IBM and Univac, single precision for CDC)

Computer	T_m (microseconds)
CDC 6400	16
6600	4.5
7600	0.6
Cyber 173	8.2
174	8.2
175	1.1
176	0.7
IBM 360/370 - 50	100
65	20
75	12
85	2
91,95	1.7
155	25
165	2
195	0.5
Univac 1108	14
1110	4

Source: NASTRAN level 17 block data deck NTMXBD

TABLE 2 - RMS WAVEFRONT TEST RESULTS

NO. OF GRID POINTS (N)	RMS WAVEFRONT				CP TIME (SEC.)			STORAGE (WORDS)	
	BEFORE	AFTER CM	AFTER GPS	AFTER LEVY	CM	GPS	LEVY	CM&GPS (M/2+8)N	LEVY 6N+10E
59	8.2	* 5.5	6.0	6.1	0.5	0.2	2.7	620	1394
66	11.0	3.2	* 2.9	3.0	0.6	0.2	1.5	693	1566
72	3.5	NI	NI	NI	0.3	0.2	1.2	720	1182
87	29.4	* 8.3	8.9	8.9	1.5	0.4	6.1	1218	2792
152	19.0	10.3	10.6	* 8.6	2.8	0.8	13.4	1944	6072
193	43.8	26.0	27.1	* 24.7	11.9	6.6	36.2	4343	17558
198	30.9	7.3	* 7.1	7.2	2.7	1.6	23.1	2673	7158
209	50.3	19.9	24.5	* 18.4	6.0	1.3	37.6	3344	8926
221	50.4	* 10.2	10.4	13.3	5.7	1.5	38.0	2984	8366
234	9.4	7.3	7.1	* 5.1	1.5	0.9	14.9	2925	4404
245	18.5	17.5	18.4	* 13.5	4.5	1.4	26.4	3430	7550
307	27.4	NI	NI	* 25.7	10.7	1.9	73.7	3684	12922
310	9.9	NI	NI	* 9.7	16.2	2.2	32.0	4030	12550
346	27.1	22.8	24.3	* 21.8	18.0	2.7	61.5	5882	16476
361	15.4	14.3	* 14.2	14.3	11.3	1.8	38.7	4332	15126
419	107.1	22.5	22.2	* 19.8	19.5	2.5	155.1	5866	18234
492	79.5	14.5	13.0	* 10.6	13.3	2.9	145.7	6396	16272
503	78.6	* 33.1	34.6	41.9	43.3	4.2	294.3	10060	30538
512	14.5	12.7	12.5	* 12.4	10.1	4.5	161.0	7680	18022
592	55.2	25.6	* 20.5	21.3	56.3	5.2	133.1	8880	26112
607	55.4	29.2	* 28.9	38.0	37.6	4.0	362.5	8802	26262
758	37.9	15.9	* 12.1	15.2	93.4	6.2	306.7	9854	30728
869	25.0	20.4	20.7	* 19.8	132.2	10.4	450.2	12601	37294
878	31.9	23.7	* 22.9	NI	46.0	12.2	311.2	10975	38118
918	131.1	25.7	* 24.3	51.1	95.2	9.7	745.7	12852	37838
992	302.0	35.9	* 34.7	38.8	141.2	34.8	801.8	16368	84712
1005	137.7	* 43.5	49.3	44.5	252.6	7.0	1010.0	21105	44110
1007	26.9	24.5	* 22.9	NI	42.6	14.6	300.3	12588	43882
1242	105.2	42.9	48.6	* 38.7	124.2	16.9	1270.9	16767	53372
2680	234.4	40.4	* 39.9	#	342.3	23.5	#	45560	127810

* = GREATEST REDUCTION, NI = NO IMPROVEMENT, # = NOT RUN