RECENT IMPROVEMENTS TO BANDIT

Gordon C. Everstine

Naval Ship Research and Development Center
Bethesda, Maryland 20084

SUMMARY

The NASTRAN preprocessor BANDIT, which improves NASTRAN's computer efficiency by resequencing grid point labels for reduced matrix bandwidth, has been improved by the addition of (1) the Gibbs-Poole-Stockmeyer (GPS) algorithm, and (2) the user option to reduce matrix profile rather than matrix bandwidth. After describing these program additions, this paper shows that, compared to the Cuthill-McKee (CM) algorithm on which BANDIT was originally based, GPS is faster and achieves similar results. For completeness, BANDIT's current capabilities and options are summarized.

BACKGROUND

The NASTRAN structural analysis computer program (ref. 1, 2), as a finite element program, assembles matrices which are normally both symmetric and sparsely-populated. The locations of the nonzero terms in the matrices are determined solely by the choice of numbers (labels) assigned to the grid points. Like most finite element codes, NASTRAN's computer running time can be reduced if the labels can be chosen in such a way that the nonzero terms cluster tightly about the main diagonal. The NASTRAN user has complete control over that clustering by his choice of grid point labels and his optional use of SEQGP bulk data cards, which effect an internal grid point resequencing for calculation purposes.

Soon after NASTRAN became available some five years ago, it was apparent that the program user could benefit from an automatic capability to perform the resequencing and generate the SEQGP cards. Indeed, for large complex structures or those generated automatically, the job of determining a good grid point sequence manually was, at best, tedious and often very difficult.

To fill the need for an automatic capability, several NASTRAN preprocessor computer programs were developed: BANDIT (refs. 3, 4), WAVEFRONT (refs. 5-7), and BANDAID (ref. 8). (For a general survey of NASTRAN preprocessors and postprocessors, see reference 9.) Both BANDIT and BANDAID are intended to reduce matrix bandwidth, while WAVEFRONT is intended to reduce matrix wavefront. (These terms are defined in the next section.) Of these preprocessors, BANDIT and WAVEFRONT appear to be the most popular. BANDIT was originally based on the Cuthill-McKee resequencing algorithm (ref. 10). WAVEFRONT and BANDAID are based on strategies developed by their authors, Levy (ref. 5) and Cook (ref. 8), respectively. These algorithms and others have been reviewed and compared by Cuthill (ref. 11).

Recently, a new bandwidth and profile reducing algorithm was developed by Gibbs, Poole, and Stockmeyer (GPS) (ref. 12) of The College of William and Mary. Since their testing of it showed it to be both effective and efficient (ref. 13), we have incorporated it in the BANDIT program to supplement the Cuthill-McKee (CM) strategy already there. (Actually, BANDIT uses the so-called reverse Cuthill-McKee algorithm since it was observed by George (ref. 14) and later proved by Liu and Sherman (ref. 15) that reversing the sequence generated by CM can never increase the profile and frequently reduces it. Such a reversal has no effect on the matrix bandwidth.) In general, GPS executes faster than CM and achieves comparable results. Unfortunately, for a given structure, it is not possible to predict a priori which strategy will yield the smaller matrix bandwidth or profile. However, since excessive resequencing time has never been considered to be a problem, BANDIT's current default mode of operation is to apply both CM and GPS to the structure in order to get the better of the two results.

DEFINITIONS

For the purposes of this discussion, some useful terms will be defined which generally follow the material given in Cuthill's survey (ref. 11).

Given a symmetric matrix A of order N, we define a "row bandwidth" b_i for row i to be the number of columns separating the first nonzero in the row from the diagonal. Alternatively, b_i is 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

$$B = \max_{i \leq N} b_i \tag{1}$$

$$P = \sum_{i=1}^{N} b_i \tag{2}$$

Let w_i denote the number of active columns in row i. A column j is active in row i if $j>i$ and there is a nonzero entry in that column in any row with index $k \leq i$. Thus, a given column is activated at the first nonzero encountered (reading from top to bottom) and remains active until the diagonal is reached. The matrix wavefront W is then defined as

$$W = \max_{i \leq N} w_i \tag{3}$$

Since the matrix A is symmetric,

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

Now, for row i, let $\underline{b}_i$ denote the columnar distance between the diagonal and the last active column in row i. Then

$$B = \max_{i \le N} b_i = \max_{i \le N} \underline{b}_i \tag{5}$$

Since, by definition,

$$\underline{b}_i \geq w_i \tag{6}$$

for each i, it follows that

$$B = \max_{i \le N} \underline{b}_i \geq \max_{i \le N} w_i = W \tag{7}$$

and

$$S = \sum_{i=1}^{N} \underline{b}_i \geq \sum_{i=1}^{N} w_i = P \tag{8}$$

Hence, as a consequence of these definitions, the matrix wavefront W for a given matrix is less than the matrix bandwidth B, and the matrix profile P is equal to both the sum of the "row bandwidths" and the sum of the "row wavefronts."

These definitions are generally modified slightly by preprocessors such as BANDIT. Since NASTRAN requires all external resequencing via SEQGP cards to be performed at the grid point level rather than the degree of freedom (DOF) level, BANDIT treats each grid point as if it had only one DOF. In general, a NASTRAN grid point can have as many as six DOF's. Thus, to convert BANDIT's values of bandwidth and profile to meaningful approximate values for NASTRAN's structural matrices, one must multiply by the average number of DOF's per grid point.

A NEW RESEQUENCING STRATEGY

The principal recent improvement to BANDIT is the installation of the new bandwidth and profile reducing algorithm developed by Gibbs, Poole, and Stockmeyer (GPS) (refs. 12, 16) of The College of William and Mary. Rather than describe how GPS works, we shall instead demonstrate its performance on a set of test problems. The test problems used here constitute the current extent of a growing collection of diversified NASTRAN data decks to be used for the testing of resequencing and equation solving algorithms. It is expected that a complete description of the set, including plots of each structure, will eventually be published.

The results of the resequencing tests are shown in Table 1. In that table, the following definitions apply:

N = number of grid points (nodes)

M = maximum nodal degree (i.e., the maximum number of nodes connected to any node)

B = matrix bandwidth (in terms of grid points rather than DOF)

P = matrix profile (in terms of grid points)

T = time, CDC 6400 CP seconds

Orig. = an original value (before resequencing) of B or P

CM = Cuthill-McKee strategy

GPS = Gibbs-Poole-Stockmeyer algorithm

Decomp. = matrix decomposition

For each of 20 structures, ranging in size up to 2680 grid points, the grid point labels were resequenced using both CM and GPS. Before and after results for both bandwidth (B) and profile (P) are shown. Since the test criterion was to reduce B rather than P, the P results are less significant. With CM, a user choice of profile reduction rather than bandwidth reduction will generally give different results for both P and B. All tests were run on a CDC 6400 computer with the SCOPE 3.4.2 operating system. Central processor (CP) times are given for both CM and GPS.

Since some of the structures are clearly very large, rough estimates of the NASTRAN real, symmetric, single-precision decomposition times on a CDC 6400 are given in the last column of Table 1. These values were computed using the following formula extracted from the NASTRAN subroutine RSPSDC:

$$T = T_B(n-2b/3)b^2/2 + T_P(n-b/2)b \tag{9}$$

For decomposition times in Table 1, it is assumed that (1) there are no active columns (in the NASTRAN sense), (2) no "spill" occurs, and (3) the structure has six DOF's per node. Hence n=6N and b=6B, where the bandwidth B used is the minimum of that obtained by CM and GPS. The constants T_B and T_P are computer-dependent time constants equal, respectively, to 15 μsec and 140 μsec for the CDC 6400.

TABLE 1 - RESEQUENCING TEST RESULTS

Case	N	M	B (Orig.)	B (CM)	B (GPS)	P* (Orig.)	P* (CM)	P* (GPS)	T (CM)	T (GPS)	T (Decomp.)
1	59	5	25	8	8	405	256	283	1.25	0.283	7.78
2	66	5	44	3	3	574	157	127	1.46	0.378	1.91
3	72	4	12	7	6	172	284	267	0.903	0.270	6.05
4	87	12	63	17	19	2249	598	642	2.65	0.419	42.2
5	162	8	156	17	13	2644	1443	1500	4.80	0.903	52.2
6	193	29	62	42	42	7760	4671	4820	19.8	8.95	508.
7	209	16	184	33	42	9503	3742	4540	12.3	1.97	362.
8	307	40	63	40	44	7825	8645	9370	17.1	2.29	784.
9#	310	10	28	14	14	2696	2725	2726	29.9	3.57	117.
10#	346	18	318	43	46	8708	7180	7650	28.7	3.49	1021.
11	361	8	50	15	14	5084	4714	4699	21.8	2.76	137.
12	503	24	452	53	54	35914	15457	15571	80.4	6.44	2255.
13#	512	14	73	28	29	6018	4838	4669	25.3	10.2	697.
14	592	14	259	40	36	28805	14171	10725	83.7	6.86	1297.
15	758	10	200	26	25	23113	10644	7465	142.	8.62	845.
16	869	9	586	38	39	18987	14335	14587	183.	13.3	2136.
17#	918	12	839	46	49	108355	21479	20369	168.	16.1	3249.
18	992	17	513	52	35	262306	33992	33076	216.	47.8	2094.
19	1242	11	936	84	99	110188	50151	54496	221.	27.5	14065.
20	2680	18	2499	68	68	587863	102534	101451	602.	38.6	20643.

* The test criterion was to reduce bandwidth, not profile.

# Denotes cases also presented in references 12 and 13.

Several conclusions can be drawn from the table:

1. CM and GPS generally obtain comparable bandwidth results, although occasionally one does significantly better than the other.

2. GPS is faster than CM.

3. Both CM and GPS are generally fast compared to estimated decomposition times. In the absence of resequencing, the decomposition times would usually be much larger.

Conclusions 1 and 3 indicate that the user would, in general, benefit from having both CM and GPS attempt to resequence his structure. Thus, the default mode of operation in BANDIT uses both and delivers to the user SEQGP cards for the better result.

REDUCTION OF MATRIX PROFILE

The second recent improvement to BANDIT is that the user now has the option of selecting matrix profile reduction rather than matrix bandwidth reduction. This option was installed primarily to facilitate testing with NASTRAN Level 15.9 to determine whether profile reduction has any advantages over band reduction. At this writing the question is still open. However, based on the close relationship between a matrix's bandwidth and its profile, it seems unlikely that major advantages will result. Indeed, in a larger sense, equation solvers which exploit matrix bandwidth, profile, or wavefront can all be classified under the general category of "envelope methods" (ref. 15), which ignore only those zeros in a matrix outside a particular region of the matrix. Distinct from the envelope methods are the general sparse methods, which ignore all the zeros in a matrix.

CURRENT BANDIT USAGE

This section summarizes briefly how a NASTRAN user runs BANDIT and what BANDIT's list of options are. It is assumed here that the prospective BANDIT user has already compiled the program and has it in executable form.

Versions of BANDIT exist for all computers on which NASTRAN runs: CDC 6000, IBM 360/370, UNIVAC 1100, and Honeywell 6000 (ref. 17).

Input to BANDIT generally consists of a standard NASTRAN data deck (ID through ENDDATA) plus one or more special $ cards (which are comments to NASTRAN) for supplying various instructions to BANDIT. The minimum BANDIT data deck consists of $ option cards, BEGIN BULK, element connection cards, and ENDDATA. BANDIT does not use GRID cards.

Output from BANDIT consists generally of printed output, punched output, and a file (FORTRAN logical unit 8) containing the complete input deck plus any SEQGP cards generated. This file, which is created automatically, is rewound before BANDIT execution terminates so that it is ready to be used as input to NASTRAN.

The current version of BANDIT, designated Version 5.1 and dated 04/28/75, contains in its element library all NASTRAN elements in Level 15.5 plus some additional elements appearing in several non-standard versions of NASTRAN. Multipoint constraint (MPC) cards are also recognized and accounted for if the user so elects.

Instructions from the user to BANDIT are passed via $ cards having the general format

$KEYWORD1   KEYWORD2

where the $ must appear in card column 1, and the first letter of KEYWORD1 must appear in column 2. Otherwise, the format of such cards is free field: keywords, which can contain no embedded blanks, must be separated by one or more blanks, and at least two letters of each keyword are required for recognition by BANDIT. Since the $ cards are interpreted by NASTRAN as comments, they can be left in the deck during a NASTRAN run.

The complete list of current $ cards is summarized in Table 2. Such cards can appear in any order but must be placed somewhere ahead of BEGIN BULK. The cards defined under Part B are specialized cards created for particular users with special needs. For most $ cards, a default is defined and denoted in Table 2 by underlining. The default applies whenever the $ card is omitted from the deck.

For example, referring to Table 2, if resequencing is to be performed, the user inserts the card

$SEQUENCE YES

into the deck anywhere before the BEGIN BULK card. In most cases, this is the only $ card added to the deck.

Although many of the cards listed in Table 2 are probably self-explanatory, several require additional explanation. The $GRID card is used to declare an upper bound (preferably least upper bound) on the number of grid points. The inclusion of this card is sometimes necessary (and never hurts) if BANDIT's default allocation of "open core" to various tables is inadequate. Generally, the default is such that the maximum nodal degree is limited to about 19. (The degree of a node is the number of other nodes connected to it.) Thus, for example, a $GRID card is required whenever solid elements are present.

Sometimes, in order to induce active columns in NASTRAN, the user would like BANDIT to ignore connections to selected grid points. Such

TABLE 2 - SUMMARY OF BANDIT $ CARDS

A. For General Use

$SEQUENCE (NO, YES)	Is resequencing to be performed?
$PUNCH (NONE, SEQGP, ALL)	What should be punched?
$CRITERION (BAND, PROFILE)	What should be reduced?
$METHOD (CM, GPS, BOTH)	By what method?
$MPC (NO, YES)	Take MPC's into account?
$PRINT (MIN, MAX)	What printed output?
$GRID N	Upper bound on number of grids.
$IGNORE G1,G2,···	Grid points to ignore.

B. For Particular Users

$NASTRAN (NO, YES)	NASTRAN to follow BANDIT?
$INSERT	Location of cards to insert.
$INSERT N	Number and location of cards to insert.
$LINES N	Number of lines per page.
$PLUS +	User-defined plus sign.
$CONNECTION (NO, YES)	Punch connection table?
$START G1,G2,···	User-supplied CM starting nodes.
$DEGREE N	Ignore nodes of degree exceeding N.

points are listed on the $IGNORE card and are resequenced last.

The $MPC card is used to tell BANDIT to modify the matrix connectivity according to the multipoint constraints (MPC's) in the deck. If this option is invoked, all MPC's present are included, regardless of any set ID's. The presence of MPC's creates a dilemma from the resequencing point-of-view, since resequencing is always performed at the grid point level, whereas MPC's are always applied at the DOF level. BANDIT treats MPC's by first generating additional connections between each independent point in the constraint relation and every other point to which the dependent point was previously connected. Second, each dependent point is eliminated from the connection table. Thus, if most or all of the DOF's for the dependent points appear in MPC relations (as, for example, with rigid links), MPC's should be taken into account. This guideline is based on experience with NASTRAN Level 15.5 and will probably have to be modified with Level 15.9 and subsequent versions, since a new equation solver has been developed for them (ref. 18).

The $NASTRAN card was created for IBM users wanting to create a single BANDIT-NASTRAN cataloged procedure in which the user could execute either BANDIT or NASTRAN (or both) and be able to control the choice with $ cards. The YES choice results in a FORTRAN STOP 5 at successful termination, thus supplying a testable condition code to the cataloged procedure.

CONCLUDING REMARKS

From the test results presented, it is clear that the addition of the new resequencing strategy by Gibbs, Poole, and Stockmeyer greatly enhances BANDIT's capabilities. The addition of the user option to reduce matrix profile rather than matrix bandwidth is a useful addition, but testing with NASTRAN Level 15.9 will be required to determine the extent of its usefulness. From the NASTRAN user's point of view, the relevant question is: For Levels 15.9 and 16, how should the grid point labels be sequenced? When these versions become available, this question will hopefully be answered by testing with band, profile, and wavefront reducers.

ACKNOWLEDGMENTS

The author is indebted to Professors N.E. Gibbs, W.G. Poole, Jr., and P.K. Stockmeyer of The College of William and Mary for supplying a copy of their source code, and to Mr. Curtis Glenn II of NSRDC for installing it into BANDIT and collecting and executing the test problems.

REFERENCES

1. MacNeal, R.H., ed.: The NASTRAN Theoretical Manual, NASA SP-221(01), Washington, D.C., 1962.

2. Butler, T.G., and Michel, D.: "NASTRAN: A Summary of the Functions and Capabilities of the NASA Structural Analysis Computer System," NASA SP-260, Washington, D.C., 1971.

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

4. Everstine, G.C.: "The BANDIT Computer Program for the Reduction of Matrix Bandwidth for NASTRAN," NASTRAN: Users' Experiences, NASA TM X-2637, 1972, pp. 407-414.

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

6. Levy, R.: "Savings in NASTRAN Decomposition Time by Sequencing to Reduce Active Columns," NASTRAN: Users' Experiences, NASA TM X-2378, 1971, pp. 627-631.

7. Levy, R.: "Structural Stiffness Matrix Wavefront Resequencing Program (WAVEFRONT)," JPL Technical Report 32-1526, vol. XIV, 1972, pp. 50-55.

8. Cook, W.L.: "Automated Input Preparation for NASTRAN," Goddard Space Flight Center Report X-321-69-237, April 1969.

9. Everstine, G.C., and McKee, J.M.: "A Survey of Pre- and Postprocessors for NASTRAN," in Structural Mechanics Computer Programs: Surveys, Assessments, and Availability, edited by W. Pilkey, K. Saczalski, and H. Schaeffer, The University Press of Virginia, Charlottesville, 1974, pp. 825-847; also, NSRDC Report 4391, June 1974.

10. Cuthill, E.H., and McKee, J.M.: "Reducing the Bandwidth of Sparse Symmetric Matrices," Proceedings of the 24th National Conference ACM 1969, pp. 157-172.

11. Cuthill, E.H.: "Several Strategies for Reducing the Bandwidth of Matrices," Sparse Matrices and Their Applications, edited by D.J. Rose and R.A. Willoughby, Plenum Press, New York, 1972, pp. 157-166.

12. Gibbs, N.E., Poole, W.G., Jr., and Stockmeyer, P.K.: "An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix," Institute for Computer Applications in Science and Engineering (ICASE) Report, Hampton, Virginia, July 1974.

13. Gibbs, N.E., Poole, W.G., Jr., and Stockmeyer, P.K.: "A Comparison of Several Bandwidth and Profile Reduction Algorithms," Institute for Computer Applications in Science and Engineering (ICASE) Report, Hampton, Virginia, November 1974.

14. George, J.A.: "Computer Implementation of the Finite Element Method," Stanford University Computer Science Department Technical Report STAN-CS-71-208, Stanford, California, 1971.

15. Liu, W.H., and Sherman, A.H.: "Comparative Analysis of the Cuthill-McKee and the Reverse Cuthill-McKee Ordering Algorithms for Sparse Matrices," Department of Computer Science Research Report #28, Yale University, New Haven, Connecticut, 1974.

16. Crane, H.L., Jr., Gibbs, N.E., Poole, W.G., Jr., and Stockmeyer, P.K.: "Matrix Bandwidth and Profile Reduction," Institute for Computer Applications in Science and Engineering (ICASE) Report 75-9, Hampton, Virginia, April 1975.

17. Golden, M.E.: "Conversion of NASTRAN to the Honeywell H6000," Proceedings of the Fifth Navy-NASTRAN Colloquium, Naval Ship Research and Development Center, CMD-32-74, September 1974, p. 23; also, Defense Documentation Center (DDC) Report No. ADA004604.

18. McCormick, C.W.: "Review of NASTRAN Development Relative to Efficiency of Execution," NASTRAN: Users' Experiences, NASA TM X-2893, 1973, pp. 7-28.