The BANDIT Computer Program for the Reduction of Matrix Bandwidth for NASTRAN

By Gordon C. Everstine

Computation and Mathematics Department
Naval Ship Research and Development Center
Bethesda, Maryland 20034

SUMMARY

This paper describes a computer program called BANDIT, which has been developed for use as a preprocessor to NASTRAN to automatically resequence the grid point numbers for reduced matrix bandwidth. The BANDIT program accepts a standard NASTRAN data deck as input, resequences the grid point numbers for reduced bandwidth (if possible), and generates a set of SEQGP cards for insertion into the NASTRAN bulk data deck. BANDIT uses the Cuthill-McKee strategy for resequencing grid points. The program is operational on the CDC 6000 series of computers, the IBM 360, and the UNIVAC 1108.

INTRODUCTION

The structural matrices formed during a NASTRAN analysis are normally both symmetric and sparse. With a suitable choice of the numbers (labels) assigned to the grid points, the matrices are also banded (i.e., the non-zero entries in each matrix are clustered about the main diagonal). Matrix bandwidth is important to the NASTRAN user for both computer storage and running time considerations. Indeed, many of the routines used in NASTRAN for the solution of linear equations and for the extraction of eigenvalues operate most efficiently when the bandwidths of the structural matrices are minimum. In such routines, the number of calculations required (and hence the computer running time) is of order NB^2 for large N and B, where N and B are the matrix order and bandwidth*, respectively.

Thus, it is clearly essential to the NASTRAN user to have matrices with small bandwidth. On the other hand, NASTRAN currently places the burden on

* For a structure with grid points numbered 1 through N, the bandwidth (BW) is defined as the maximum numerical difference between any two connected grid point numbers. A pair of grid points are "connected" if there is an explicit connection between them or if they appear in the same element.

the user to number his structure so as to provide such a bandwidth. In many cases, this is an excessive burden, particularly for large, complex structures or for those emanating from automatic data generators. In any case, the burden is unnecessary since several algorithms have been developed for the express purpose of permuting grid point sequences to reduce matrix bandwidth.

Although there might be merit in including such an algorithm within NASTRAN, the desired effect can be achieved externally by means of the SEQGP bulk data cards. These cards define a look-up table giving the correspondence between the original grid numbers used by a structural analyst in defining his problem and a new set of numbers to be used internally for all calculations.

This paper describes a computer program called BANDIT (ref. 1) developed for use as a preprocessor to NASTRAN to automatically resequence the grid point numbers for reduced matrix bandwidth. BANDIT accepts a standard NASTRAN data deck as input, resequences the grid point numbers for reduced bandwidth (if possible), and generates a set of SEQGP punched cards for insertion into the NASTRAN bulk data deck.

THE RENUMBERING STRATEGY

The heart of any bandwidth-reduction program is its renumbering strategy. The need to resort to "strategies" at all becomes evident when one considers that N grid points can be sequenced in N! distinct ways. Thus, with any strategy, there is no guarantee that an optimum numbering will be achieved at acceptable cost.

The renumbering strategy used in BANDIT is a fast, direct method developed by Cuthill and McKee (ref. 2). Considerable experience gained over the past two years has shown this strategy to perform well consistently for the types of structures of interest to the Navy users of NASTRAN. Consistency is emphasized since, by its very nature, the business of bandwidth reduction is very heuristic. Indeed, no algorithm is best for all structures. Comparisons between the Cuthill-McKee approach and other schemes appear both in reference 2 and a sequel by Cuthill (ref. 3).

Although a complete discussion of the Cuthill-McKee strategy appears in the source paper (ref. 2), the main ideas are presented here for completeness.

For the purposes of this discussion, the degree of a grid point (or node)* is defined as the number of neighboring nodes to which it is connected. A starting node is one given the new label 1. The strategy first involves the selection of one or more possible starting nodes. Although these nodes are normally of low degree, the one eventually chosen to be the starting node

* Throughout this paper, "grid point" and "node" are used interchangeably.

need not be of minimum degree.

After the selection of a starting node, the remaining nodes are relabeled according to the following prescription: The nodes adjacent (i.e., adjoining) to the starting node are labeled in sequence in the order of their increasing degree. These nodes are said to be at Level 1. (The starting node constitutes Level 0.) In general, Level n is the set of all nodes adjacent to Level n-1 nodes and not previously assigned to a level.

Once Level 1 nodes have been sequenced, the Level 2 nodes are numbered as follows: For each node of Level 1 and in sequence, the Level 2 nodes are numbered in the order of their increasing degree. The numbering continues in this fashion, level-by-level, until all nodes have been numbered. In the event of a "tie" among several nodes to receive a given label, the first node to qualify is chosen.

The above procedure is carried out for each possible starting node previously selected. The sequence yielding the lowest bandwidth is finally chosen.

It is apparent that, in the absence of ties for a given label, the relabeling sequence is independent of the original numbering once a starting node has been selected. Thus the original nodal numbering has almost no effect on the final numbering.

This rather simple approach to bandwidth reduction has proved to be surprisingly effective in production use.

PROGRAM CAPABILITIES

BANDIT's primary reason for existence is the generation of the NASTRAN SEQGP data cards to effect low matrix bandwidth. As a by-product, BANDIT can also be used to right-adjust a NASTRAN bulk data deck. In either case, following the execution of BANDIT, the complete right-adjusted data deck is available on disk file. In addition, the user can elect to have punched card output for either the entire deck or the SEQGP cards alone.

The input data deck for BANDIT consists of a standard NASTRAN data deck (ID card through ENDDATA card, inclusive) with the addition of appropriate $ cards (comment cards to NASTRAN) indicating what the user wants BANDIT to do.

Although BANDIT accepts an entire NASTRAN deck as input, resequencing requires only the following NASTRAN cards: BEGIN BULK, ENDDATA, and all "connection" cards. GRID cards are not processed by BANDIT. Input bulk data cards may have either 8- or 16-column field widths. In general, the deck can be unsorted, except that each continuation to a connection card must immediately follow the parent card.

The user is also allowed to specify grid points which he wants ignored during the resequencing. Such points appear last in the new nodal sequence, and hence would normally be assigned by NASTRAN to active columns. This feature lets the user specify points which he feels should be relegated to active columns by NASTRAN. Unfortunately, it is often very difficult to identify such points.

COMPUTER CONFIGURATIONS

The BANDIT program was developed principally for use on the CDC 6000 series of computers and hence has some machine-dependent features. However, for use on other computers, a "machine-independent" version has also been developed. This version is currently operational on the IBM 360 and UNIVAC 1108.

On CDC machines, BANDIT functions as a variable-core program. Hence, it is essentially open-ended with respect to the number of grid points that can be handled. The program will load with less than 50000_8 words of core, in which space typical structures of less than 500 grid points can be handled. For larger structures, more core is usually required, although rarely are more than 60000_8 words needed.

Typical running times on a CDC 6400 computer are of the order of two to three minutes. For example, a 1000-grid-point structure was recently processed in four minutes. As a rough rule-of-thumb, run times increase in proportion to the number of grid points.

DISCUSSION AND EXAMPLES

It would be appropriate to include at least one example of BANDIT's performance. Comparisons among the various algorithms for realistic large structures are rare. Equally rare (at NSRDC, at least) are structures which demonstrate a meaningful "before and after" comparison. The latter is due to the natural tendency of structural analysts to make no serious manual attempt at achieving small bandwidth when aware of the existence of an automatic capability.

The example included here (Figure 1) is a solid structure containing 1686 grid points. It was analyzed on a program other than NASTRAN. Although both manual and automatic renumbering attempts were made, the smallest bandwidth obtained was 202. Subsequent use of the Cuthill-McKee approach further reduced the bandwidth to 151.

Examples such as this tend to gloss over the important role played by the modeler in achieving a small matrix bandwidth. Even with good automatic direct resequencing capabilities available, it still is undesirable for the analyst to disregard bandwidth considerations in his modeling. Some of the complications can be illustrated by two simple examples.

The first example (Figure 2) shows the reduction of bandwidth possible merely by switching from quadrilaterals to triangles. Here the bandwidth has been cut almost in half. This example was reported at the first NASTRAN Users' Colloquium by R.H. Brolliar (ref. 4), who indicated a central processor time saving of 78%.

The second example, obtained from Livesley (ref. 5), shows that bandwidths can sometimes be reduced by the addition of extra nodes. This is illustrated in Figure 3, where the addition of three nodes effected a bandwidth reduction from 4 to 3.

These two examples serve to illustrate that the structural analyst has control over bandwidth considerations that are beyond the scope of automatic bandwidth reduction programs.

REFERENCES

1. Everstine, G.C., The BANDIT Computer Program for the Reduction of Matrix Bandwidth for NASTRAN, NSRDC Report 3827, March 1972.

2. 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.

3. 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.

4. Brolliar, R.H., A NASTRAN Buckling Analysis of a Large Stiffened Cylindrical Shell with a Cutout, NASTRAN: Users' Experiences, Vol. I, NASA TM X-2378, September 1971, pp. 65-84.

5. Livesley, R.K., The Analysis of Large Structural Systems, The Computer Journal, vol. 3, 1960, pp. 34-39.

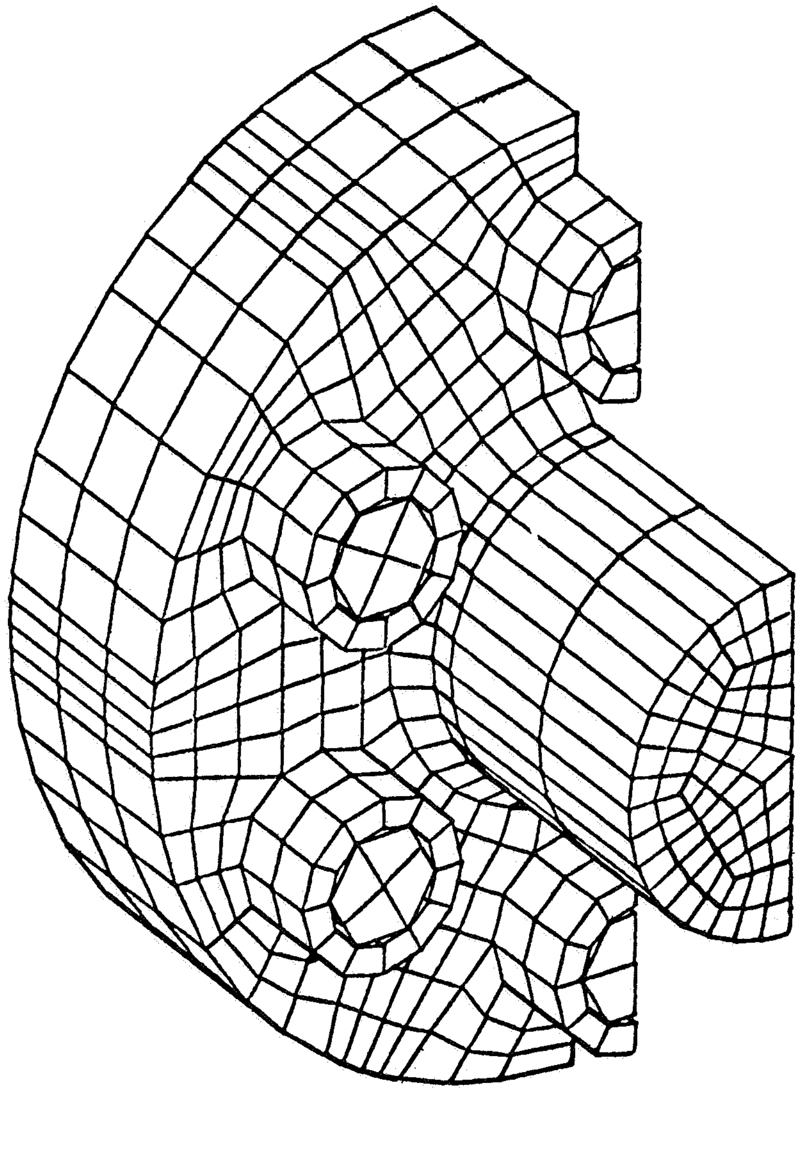
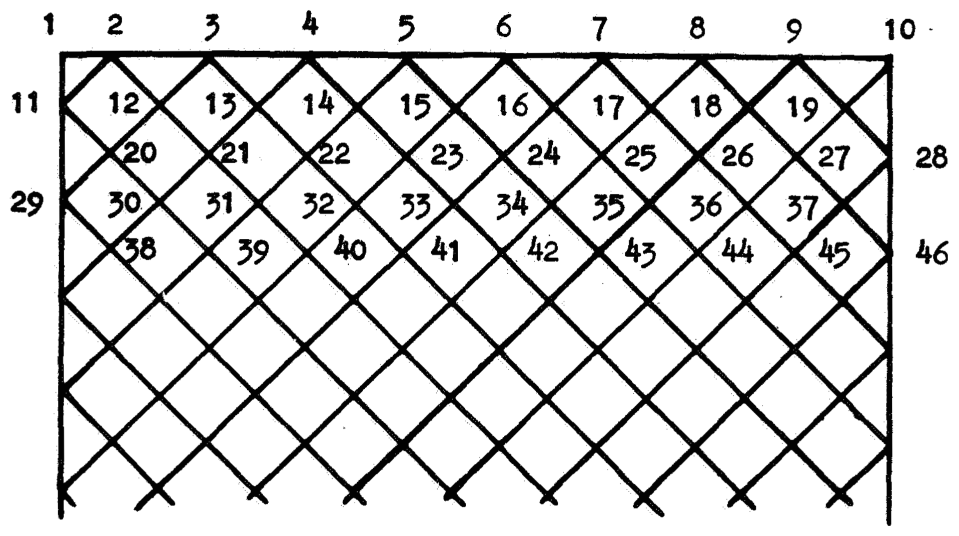
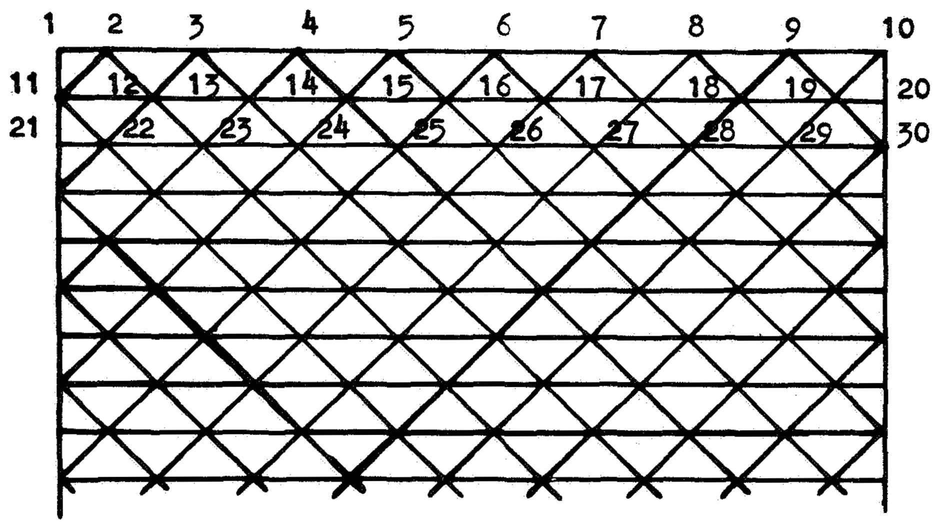


Figure 1. Example with 1686 Nodes.

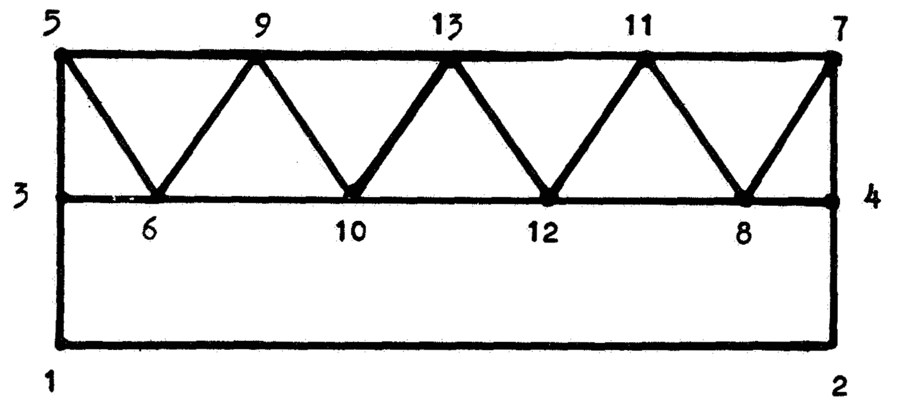


(a) Quadrilateral Modeling (BW = 18).

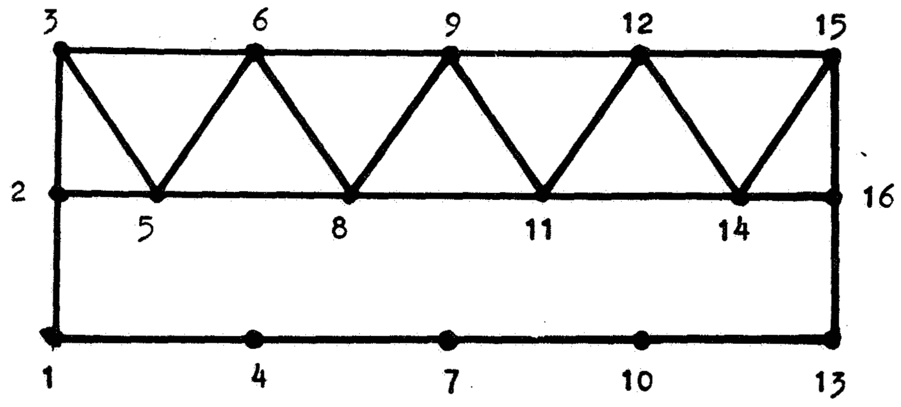


(b) Triangle Modeling (BW = 11).

Figure 2. Example Showing Bandwidth Reduction Possible by Using Triangles Rather than Quadrilaterals.



(a) Without Extra Nodes (N = 13, BW = 4).



(b) With Extra Nodes (N = 16, BW = 3).

Figure 3. Example Showing Bandwidth Reduction Possible by Addition of Extra Nodes.