Reprinted from "Grid-Point Sequencing," Part 4, Section 3.2 by G.C. Everstine in Finite Element Handbook, edited by H. Kardestuncer and D.H. Norrie, McGraw-Hill Book Company, pp. 4.177-4.183 (1987).

3.2 GRID-POINT SEQUENCING

A key numerical problem which arises throughout finite-element analysis (whether linear or nonlinear, static or dynamic) is that of the solution of large sets of linear algebraic equations such as, in matrix form,

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.6}$$

where the vector $\mathbf{b}$ and the square matrix $\mathbf{A}$ are known, and the unknown vector $\mathbf{x}$ is sought. In linear static analysis, for example, $\mathbf{A}$ is the system stiffness matrix (or its equivalent for nonstructural applications). For dynamic analysis, $\mathbf{A}$ is some linear combination of the system stiffness, mass, and damping matrices.

In finite-element applications, $\mathbf{A}$ contains mostly zeros (and is thus said to be *sparse* or *sparsely populated*) since the procedure under which finite-element matrices are assembled dictates that the off-diagonal matrix terms coupling any two degrees of freedom to each other are zero unless those degrees of freedom are common to the same finite element [2]. It also follows that the locations of the nonzero elements of the matrix **A** depend solely on the ordering of the unknowns. In finite-element applications, the ordering of the unknowns corresponds to the selection of grid-point (or joint) labels for the mesh points. It is thus possible to choose an ordering for sparse matrices so that the nonzeros are located to allow subsequent matrix operations such as equation solving or eigenvalue extraction. A good ordering is essential to the finite-element user since virtually all finite-element computer programs contain equation solution and eigenvalue routines which have been written expressly to operate efficiently on matrices possessing small bandwidth, profile, or wavefront. (All these terms will be defined precisely in the next section; in general, a banded matrix has all its nonzero entries clustered about the main diagonal.)

Efficiency in equation solving is obtained by avoiding arithmetic operations (multiplications and additions) on matrix terms that are known in advance to be zero. The computer execution time for most equation solvers and triangular factorization routines is proportional to the order N of the matrix. More important, however, is that the execution time is also proportional to the *square* of some other matrix characteristic such as bandwidth B, profile P, or root-mean-square wavefront W_{rms}, depending on the specific type of equation solver. For a given finite-element model of a structure, N is fixed, but B, P, and W_{rms} depend on the ordering of the grid points. Clearly, in this case, it is desirable to make these quantities as small as possible.

First we will define the various terms encountered in connection with grid-point sequencing in finite-element analysis. Next, we will describe briefly the relationship between finite-element meshes and the matrices which result. Finally, we will discuss algorithms which have been developed to automate the grid-point sequencing problem.

3.2A 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 *semibandwidth* B and *profile* P are defined as

$$B = \max_{i \leq N} b_i \tag{3.7}$$

$$P = \sum_{i=1}^{N} b_i \tag{3.8}$$

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

$$W = \max_{i \leq N} c_i \tag{3.9}$$

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

$$P = \sum_{i=1}^{N} b_i = \sum_{i=1}^{N} c_i \tag{3.10}$$

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

$$W_{av} = \frac{1}{N} \sum_{i=1}^{N} c_i = \frac{P}{N} \tag{3.11}$$

$$W_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} c_i^2} \tag{3.12}$$

Thus, for symmetric matrices the average wavefront and average semibandwidth are equal. From these definitions, it follows that for a given matrix

$$W_{av} \leq W_{rms} \leq W_{max} \leq B \leq N \tag{3.13}$$

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

The *degree* d_i of grid point i is defined as the number of other grid points (or nodes) to which it is "connected." In the finite-element context, two grid points are said to be *connected* if they are common to the same finite element. The rationale behind this definition is that two grid points common to the same finite element are generally coupled to each other by a nonzero off-diagonal term in the stiffness matrix. For a finite-element mesh having only one degree of freedom (DOF) per mesh point, d_i is also the number of nonzero off-diagonal terms in row i of matrix **A**. The *maximum nodal degree* M is

$$M = \max_{i \leq N} d_i \tag{3.14}$$

The *number of unique edges* E in a finite-element mesh is defined as the total number of unique connections in the mesh. Thus, for a mesh having one DOF per grid point, E is equal to the number of nonzero off-diagonal matrix terms above the diagonal. Hence, for a symmetric matrix,

$$E = \sum_{i=1}^{N} \frac{d_i}{2} \tag{3.15}$$

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

$$\rho = \frac{2E + N}{N^2} \tag{3.16}$$

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_{av}, 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, not just one. For meshes having, say, six degrees of freedom per node, the actual values of B, W_{max}, W_{av}, or W_{rms} would be, in the absence of constraints, six times their corresponding values computed for one degree of freedom per point.

These definitions can be illustrated by the following simple example. Figure 3.22 shows a matrix of order 6 in which X denotes a nonzero entry. In each row and column a line is

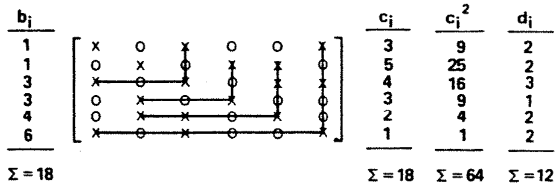


FIG. 3.22 Example illustrating definitions of matrix bandwidth, profile, and wavefront.

drawn from the first nonzero to the diagonal. Thus b_i is the number of columns traversed by the solid line in row i to the left of and including the diagonal. 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, $B = 6$, $W_{max} = 5$, $P = 18$, $W_{av} = 3.0$, $W_{rms} = 3.3$, $M = 3$, $E = 6$, and $\rho = 50.0\%$. For large matrices arising out of realistic applications, the matrix density is usually below 5% [31].

3.2B The Relationship between Meshes and Matrices

Consider the one-dimensional, six-DOF system of six scalar springs shown in Fig. 3.23. For the grid-point numbering (labeling) indicated in that figure, the 6×6 system stiffness matrix looks like

$$\mathbf{K} = \begin{bmatrix} X & X & & X & & \\ X & X & X & & & \\ & X & X & X & & \\ X & & X & X & X & \\ & & & X & X & X \\ & & & & X & X \end{bmatrix} \tag{3.17}$$

where X indicates the location of a nonzero entry. From Eq. (3.17) and the definition of Eq. (3.7), the matrix bandwidth B is 4. For consecutively numbered structures, the bandwidth can also be obtained directly from the structure (Fig. 3.23) by adding unity to the maximum

FIG. 3.23 Six-degree-of-freedom spring system.

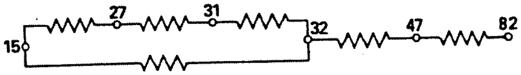


FIG. 3.24 Six-degree-of-freedom spring system with nonconsecutive grid labels.

numerical difference between connected node numbers (where node 1 is connected to node 4). The same bandwidth is also obtained for the numbering of Fig. 3.24, since the ordering of the mesh points is the same. From the point of view of the matrix connectivity, there is no difference between the structures in Figs. 3.23 and 3.24. Some finite-element programs allow the user to specify nonconsecutive grid-point numbers as in Fig. 3.24, rather than requiring consecutive numbering. The matrix bandwidth cannot be computed directly from the mesh by looking at the maximum numerical difference between connected node numbers when the nodes are not numbered consecutively. Instead, Fig. 3.24 must first be simplified to Fig. 3.23.

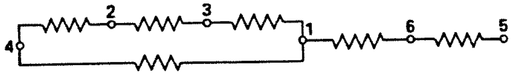


FIG. 3.25 Six-degree-of-freedom system with poor mesh labeling.

To illustrate the difference that sequencing makes, consider the numbering of Fig. 3.25. Here, the bandwidth is 6, so that the ordering of Fig. 3.23 is to be preferred over that of Fig. 3.25. However, a still better sequence (i.e., one with a smaller bandwidth) is shown in Fig. 3.26, where $B = 3$.

FIG. 3.26 Six-degree-of-freedom system with good mesh labeling.

The same concepts can also be applied to two- and three-dimensional meshes. For example, consider the plate in Fig. 3.27 modeled with a 2×4 array of quadrilateral elements. Since all nodes common to the same finite element are "connected" to each other, the grid-point bandwidth is 7. A better sequence (i.e., one with a smaller bandwidth) would number first across the "short" direction (in the sense of number of nodes rather than actual distance), as in Fig. 3.28. With this sequence, the grid-point bandwidth is now 5.

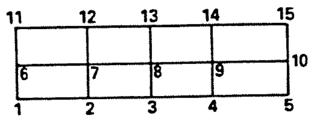


FIG. 3.27 A 2D mesh.

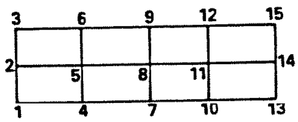


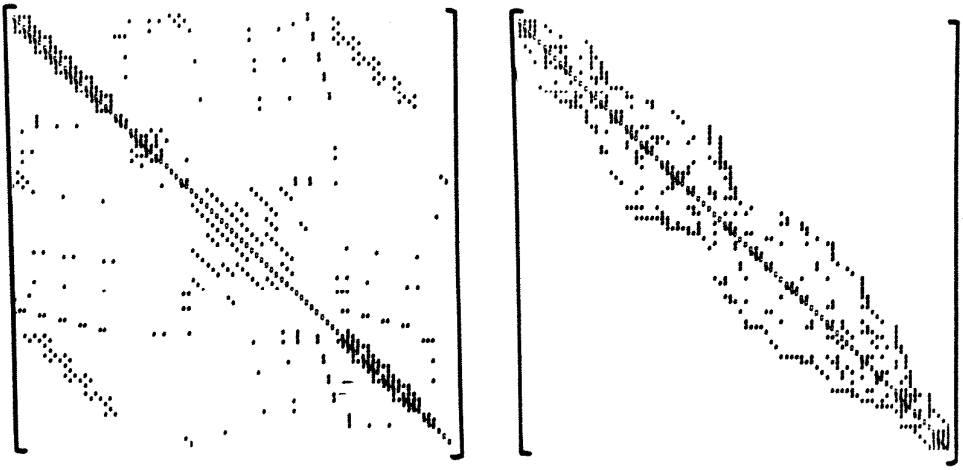
FIG. 3.28 2D mesh with good grid-point labeling.

In general, the plates of Figs. 3.27 and 3.28 have more than one DOF per node. Thus, although it generally suffices to consider only grid-point bandwidth when choosing an ordering, the actual bandwidth which the equation solver encounters would be much larger. For example, structures having six DOF per node and a grid-point bandwidth of B would have a DOF bandwidth of $6B$.

Although the above discussion was written from the point of view of matrix bandwidth, similar comments could be made from the point of view of matrix profile or wavefront. Some finite-element codes, for instance, have matrix decomposition routines which operate fastest on those matrices having the smallest rms wavefront.

3.2c Automatic Resequencing Algorithms

So far, we have defined the relevant terms and indicated how a finite-element mesh might be resequenced to reduce the resulting matrix bandwidth. In general, however, it is very difficult for the finite-element practitioner to know how to sequence a given mesh to effect a good numbering, particularly for large, complicated meshes or those generated automatically on a computer. Even if a good sequence is known, it may be tedious to implement for large meshes. To overcome this problem, a large number of algorithms have been developed to automate the assignment of grid-point labels, given the connectivity of the mesh. Since it is clearly impractical to check each of the $N!$ possible sequences associated with a given matrix **A** of order N, each algorithm uses some presumably rational strategy to arrive quickly at a good grid-point sequence.



(a) (b)

FIG. 3.29 Locations of nonzero terms in a stiffness matrix ($N = 87$) before and after automatic grid-point reordering. (*a*) Before: $B = 64$, $W_{max} = 43$, $P = 2336$, $W_{av} = 26.9$, $W_{rms} = 29.4$. (*b*) After: $B = 18$, $W_{max} = 13$, $P = 685$, $W_{av} = 7.9$, $W_{rms} = 8.3$; CYBER 74 computer time required = 0.6 second.

No attempt will be made here to review the various resequencing algorithms. An extensive bibliography of available algorithms is included in Ref. [31].

One excellent resequencing algorithm which is widely used was developed by Gibbs, Poole, and Stockmeyer [32]. This algorithm, known as the GPS algorithm, is both exceptionally fast and consistently reliable for the reduction of matrix bandwidth, profile, and wavefront. Lewis [33] has recently developed some improvements to the GPS scheme.

The GPS strategy can be summarized in general terms. First, it selects a single starting node (the node assigned the label 1 in the new sequence) to be an endpoint of a pseudodiameter of the finite-element mesh. The *diameter* of a mesh is the shortest path (in terms of number of edges) connecting any two grid points of maximal distance apart. A *pseudodiameter* locates points at nearly maximal distance apart. Second, the mesh is numbered using a procedure which is similar to the Cuthill-McKee (CM) numbering algorithm [34], in which the new labels 2 through N are assigned by numbering the unnumbered nodes connected to new label I in order of increasing degree, starting with $I = 1$ and continuing with increasing I until all nodes are sequenced. This basic ordering strategy generally works well and has been used in several algorithms.

An example of a finite-element stiffness matrix before and after resequencing by an automatic program is shown in Fig. 3.29 for a small structure having 87 grid points.

3.5 REFERENCES

2. Zienkiewicz, O. C., *The Finite Element Method*, 3d ed., McGraw-Hill, New York, 1977.

31. Everstine, G. C., "A Comparison of Three Resequencing Algorithms for the Reduction of Matrix Profile and Wavefront," *Int. J. Numer. Meth. Eng.*, **14**:837–853 (1979).
32. Gibbs, N. E., W. G. Poole, Jr., and P. K. Stockmeyer, "An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix," *SIAM J. Numer. Anal.*, **13**:236–250 (1976).
33. Lewis, J. G., "Implementation of the Gibbs-Poole-Stockmeyer and Gibbs-King Algorithms," *ACM Trans. Math. Software* **8**:180–189 (1982).
34. Cuthill, E., and J. M. McKee, "Reducing the Bandwidth of Sparse Symmetric Matrices," *Proc. 24th Nat. Conf. Assoc. Comput. Mach.*, ACM Pub. P69, New York, 1969, pp. 157–172.