4
Computer-Oriented Direct Stiffness Method

4.1 Introduction. In the preceding chapter the stiffness method was developed initially by superposition of actions for the free displacement coordinates (Sec. 3.2). Then the method was formalized and extended in Sec. 3.6 using the compatibility matrix $\mathbf{C}_{MJ}$ and the virtual work concept. With this second approach the complete joint stiffness matrix $\mathbf{S}_J$ (for both free and restrained displacements) was assembled by the triple matrix multiplication given as Eq. (3-32). While the formal version is enlightening and well organized, it involves a large and sparse compatibility matrix containing terms that are not easy to evaluate correctly. Neither the generation of this matrix nor the multiplication process for assembly would be suitable for computer programming. A better methodology consists of drawing ideas from both approaches and adding a few computer-oriented techniques to evolve what is known as the *direct stiffness method.*

The primary objective of this chapter is to further develop the stiffness method into a form that may be readily programmed on a digital computer. In fact, the procedures established in this chapter have their counterparts in the form of flow charts in Chapter 5. In Sec. 4.2 the direct stiffness method is described, and an outline is presented for the purpose of organizing the calculations into a sequence that is conducive to programming. Since member stiffnesses play an essential role in the analyses of all types of framed structures, this topic is treated next in Sec. 4.3. Then other aspects of the direct stiffness method common to all types of framed structures are discussed in Secs. 4.4 through 4.7. These matters are demonstrated with a familiar example that has been solved previously. The remaining sections of the chapter deal with applications to various types of framed structures. For simplicity, only prismatic members and the effects of applied loads are considered in this chapter. Methods for handling nonprismatic members, temperature changes, prestrains, support displacements, and other effects are described in Chapter 6.

4.2 Direct Stiffness Method. The key to simplifying the assembly process for the joint stiffness matrix $\mathbf{S}_J$ is to use member stiffness matrices for actions and displacements at both ends of each member. If the member displacements are referenced to structural (global) coordinates, they coincide with joint displacements. In that case all of the geometric complications must be handled locally, and the transfer of member information to structural arrays is straightforward. That is, the stiffness matrix and the equivalent load vector can be assembled by direct addition instead of by matrix multiplication.

Thus, the assembly of the joint stiffness matrix, assuming *m* members, may be stated as

$$\mathbf{S}_{\mathrm{J}} = \sum_{i=1}^{m} \mathbf{S}_{\mathrm{MS}\,i} \tag{4-1}$$

In this expression the symbol $\mathbf{S}_{\mathrm{MS}\,i}$ represents the *i*-th member stiffness matrix with end-actions and displacements (for both ends) taken in the directions of structural axes. To be conformable for matrix addition, all such member stiffness matrices should be expanded to the same size as $\mathbf{S}_{\mathrm{J}}$ by augmenting them with rows and columns of zeros. However, this operation can be avoided in computer programming by merely placing terms from $\mathbf{S}_{\mathrm{MS}\,i}$ into $\mathbf{S}_{\mathrm{J}}$ where they belong.

Similarly, an equivalent joint load vector $\mathbf{A}_{\mathrm{E}}$ can be constructed from member contributions, as follows:

$$\mathbf{A}_{\mathrm{E}} = -\sum_{i=1}^{m} \mathbf{A}_{\mathrm{MS}\,i} \tag{4-2}$$

In this instance $\mathbf{A}_{\mathrm{MS}\,i}$ is a vector of fixed-end actions in the directions of the structural axes at both ends of member *i*. As before, it is theoretically necessary to augment $\mathbf{A}_{\mathrm{MS}\,i}$ with zeros to make it conformable for matrix addition, but this step can be avoided in programming. The equivalent joint loads in Eq. (4-2) will be added to actual loads applied at the joints to form the total (or combined) load vector.

In order to separate terms pertaining to the free displacements of the structure from those for support restraints, it is necessary to rearrange and partition the stiffness and load matrices. This rearrangement of terms may be accomplished after the assembly is completed. However, in a computer program the rearrangement is more easily done when the information is transferred from small member arrays to large structural arrays.*

After the stiffness and load matrices have been rearranged, the solution proceeds in the manner already established in Sec. 3.6. Thus, the basic solution for free joint displacements (see Eq. 3-35), due to loads only, becomes

$$\mathbf{D}_{\mathrm{F}} = \mathbf{S}_{\mathrm{FF}}^{-1}\mathbf{A}_{\mathrm{FC}} \tag{4-3}$$

in which $\mathbf{A}_{\mathrm{FC}}$ is a vector of combined joint loads (actual and equivalent) corresponding to $\mathbf{D}_{\mathrm{F}}$. While it is symbolically convenient to imply inversion of the stiffness matrix $\mathbf{S}_{\mathrm{FF}}$ in Eq. (4-3), it is not efficient to actually calculate

*While it is possible to solve the equations of equilibrium "in-place" without rearrangement (see Appendix E), it is computationally more efficient to separate the independent equations from the dependent equations.

$\mathbf{S}_{\mathrm{FF}}^{-1}$ in a computer program. Instead, the coefficient matrix $\mathbf{S}_{\mathrm{FF}}$ should be factored and the solution for $\mathbf{D}_{\mathrm{F}}$ obtained in forward and backward sweeps. (The topics of factorization and solution are covered in Appendix D.)

Other items of interest consist of support reactions and member end-actions. Due to the effects of loads only, Eq. (3-36) gives the reactions

$$\mathbf{A}_{\mathrm{R}} = -\mathbf{A}_{\mathrm{RC}} + \mathbf{S}_{\mathrm{RF}}\mathbf{D}_{\mathrm{F}} \tag{4-4}$$

In addition, Eq. (3-37) for the member end-actions may be rewritten so that it pertains to only one member at a time, as follows:

$$\mathbf{A}_{\mathrm{M}i} = \mathbf{A}_{\mathrm{ML}i} + \mathbf{S}_{\mathrm{M}i}\mathbf{D}_{\mathrm{M}i} \tag{4-5}$$

All of the matrices in this expression are referenced to member axes. The action vector $\mathbf{A}_{\mathrm{ML}i}$ contains fixed-end actions (in the directions of member axes) due to loads applied on the member itself. In addition, the member stiffness matrix $\mathbf{S}_{\mathrm{M}i}$ has terms (in member directions) for both ends of the member. Consequently, the vector $\mathbf{D}_{\mathrm{M}i}$ must consist of displacements (in member directions) at both ends of the member. These displacements can be obtained by expressing the joint displacements at the j and k ends as components in the member directions.

A computer program for the analysis of a structure by the direct stiffness method divides conveniently into several phases. These phases are not the same as those in Chapter 3 for hand calculations. One difference is that when using a computer it is desirable to work in the early phases of the program with only the data pertaining to the structure. This part of the program includes the formation of the stiffness matrix, which is a property of the structure. Subsequently, the load data is manipulated, after which the final results of the analysis are computed. This sequence is particularly efficient if more than one load system is being considered, because the initial phases of the calculations need not be repeated. The steps to be considered in the subsequent discussions are the following:

(1) Identification of Structural Data. Information pertaining to the structure itself must be assembled and recorded. This information includes the number of members, the number of joints, the number of degrees of freedom, and the elastic properties of the material. The locations of the joints of the structure are specified by means of geometric coordinates. In addition, the section properties of each member in the structure must be given. Finally, the conditions of restraint at the supports of the structure must be identified. In computer programming, all such information is coded in some convenient way, as will be shown subsequently in this chapter and also in Chapter 5.

(2) Construction of Stiffness Matrix. The stiffness matrix is an inherent property of the structure and is based upon the structural data only. In computer programming it is convenient to obtain the joint stiffness matrix by summing contributions from individual member stiffness matrices (a

discussion of complete member stiffnesses is given in Sec. 4.3). The joint stiffness matrix to be considered is related to all possible joint displacements, including support displacements, as was discussed in Sec. 3.6. This array shall be called the *over-all joint stiffness matrix*. Its formation and rearrangement are described in Secs. 4.4 and 4.6, respectively.

(3) Identification of Load Data. All loads acting on the structure must be specified in a manner suitable for computer programming. Both joint loads and member loads must be given. The former may be handled directly, but the latter are handled indirectly by supplying as data the fixed-end actions caused by the loads on the members.

(4) Construction of Load Vector. The fixed-end actions due to loads on members may be converted to *equivalent joint loads*, as described previously in Sec. 1.12. These equivalent joint loads may then be added to the actual joint loads to produce a problem in which the structure is imagined to be loaded at the joints only. Formation and rearrangement of the load vector are described in Secs. 4.5 and 4.6, respectively.

(5) Calculation of Results. In the final phase of the analysis all of the joint displacements, reactions, and member end-actions are computed. The calculation of member end-actions proceeds member by member (see Eq. 4-5) instead of for the structure as a whole. Such calculations require the use of complete member stiffness matrices for member directions, a subject that is covered in the next section. A simple calculation of this type is demonstrated for illustrative purposes in Sec. 4.7.

There are many possible variations in organizing the stiffness method for computer programming. The phases of analysis listed above constitute an orderly approach having certain essential features that are advantageous when dealing with large, complicated frameworks. These steps will be discussed and illustrated by means of a familiar example in the following sections.

4.3 Complete Member Stiffness Matrices.

In order to construct the joint stiffness matrix with the summation process indicated by Eq. (4-1), it is necessary to generate complete member stiffness matrices $\mathbf{S}_{\mathrm{MS}i}$ for structure-oriented axes. In addition, member stiffness matrices $\mathbf{S}_{\mathrm{M}i}$ for member-oriented axes are required for the purpose of calculating member end-actions with Eq. (4-5). It is always possible to obtain the member stiffnesses with respect to the member axes, as is done in this section, and then to transform these stiffnesses to the structural axes. The procedures for accomplishing this transformation are described later in the chapter for each type of structure.

In the special case of a continuous beam, the member axes are inherently parallel to those for the structure as a whole; so no transformation is necessary. Thus,

$$(\mathbf{S}_{\mathrm{M}\,i})_{\mathrm{beam}} = (\mathbf{S}_{\mathrm{MS}\,i})_{\mathrm{beam}} \qquad \text{(a)}$$