

Chapter 2

Optimal Force Method: Analysis of Skeletal Structures

2.1 Introduction

This chapter starts with presenting simple and general methods for calculating the degree of static indeterminacy of different types of skeletal structures, such as rigid-jointed planar and space frames, pin-jointed planar trusses and ball-jointed space trusses.

Then the progress made in the force method of structural analysis in recent years is presented, and the state of art is summarized. Efficient methods are developed leading to highly sparse flexibility matrices. The methods are mainly developed for frame structures, however, extensions are made to general skeletal structures.

The force method of structural analysis, in which the member forces are used as unknowns, is appealing to engineers, since the properties of members of a structure most often depend on the member forces rather than joint displacements. This method was used extensively until 1960. After this, the advent of the digital computer and the amenability of the displacement method for computation attracted most researchers. As a result, the force method and some of the advantages it offers in optimisation and non-linear analysis, have been neglected.

Six different approaches are adopted for the force method of structural analysis, which will be classified as follows:

1. Topological force methods,
2. Combinatorial force methods,
3. Algebraic force methods,
4. Mixed algebraic-combinatorial force methods,
5. Integrated force method, and
6. Metaheuristic based methods.

Topological methods have been developed by Henderson [1], Maunder [2] and Kaveh [3]. Combinatorial force method is mainly developed by Kaveh [3] using different graph theoretical algorithms. Algebraic topology is employed extensively in the work of Langefors [4]. Algebraic methods have been developed by Denke [5],

Robinson [6], Topçu [7], and Kaneko et al. [8], and mixed algebraic-topological methods have been used by Gilbert et al. [9], Coleman and Pothen [10]. The integrated force method has been developed by Patnaik [11]. Meta-heuristic based methods are also developed for the formation of null basis in the work of Kaveh and Jahamshahi [12] and Kaveh and Daei [13].

2.2 Static Indeterminacy of Structures

Skeletal structures are the most common type of structures encountered in civil engineering practice. These structures sustain the applied loads mainly by virtue of their topology, i.e. the way members are connected to each other (connectivity). Therefore, topology plays a vital role in their design. The first step in design of such structures is to provide sufficient rigidity and make it reliable, but this depends in part on the degrees of static indeterminacy of the structures. One way to calculate the degree of static indeterminacy is to use classical formulae such as those given in Timoshenko and Young [14]; however, the application of these usually provides only a small part of the necessary topological properties. The methods presented in this chapter provide powerful means for understanding the distribution of the indeterminacy within a structure. The concepts presented are efficient in both the optimal force method of structural analysis, as will be discussed in the second part of this chapter.

In the analysis of skeletal structures, three different properties are encountered, which are classified as topological, geometrical and material. Separate study of these properties results in a considerable simplification in understanding the structural behaviour leading to methods for efficient analysis. This chapter is confined to a study of those topological properties of skeletal structures needed to study force and displacement methods. The number of equations to be solved in the two methods may differ widely for the same structure. This number depends on the size of the flexibility and the stiffness matrices. The orders of this matrix are the same as the degree of static indeterminacy and the degree of kinematic indeterminacy of a structure, respectively. Obviously, the method that leads to the required results with the least amount of computational time and storage should be used for the analysis of a given structure. Thus, the comparison of the degree of static indeterminacy and the degree of kinematic indeterminacy may be the main criterion for selecting the method of analysis.

The degree of kinematic indeterminacy of a structure, also known as its total number of degrees of freedom, can easily be obtained by summing up the degrees of freedom of its nodes. A node of planar and space trusses has two and three degrees of freedom, respectively. For planar and space frames, these numbers are 3 and 6, respectively. Single-layer grids have also three degrees of freedom for each node.

For determining the degree of static indeterminacy of structures, numerous formulae depending on the kinds of members or types of joints have been given, e.g. Ref. [14]. The use of these classical formulae, in general, requires counting the

number of members and joints, which becomes a tedious process for multi-member and/or complex pattern structures; moreover, the count provides no additional information about connectivity.

Henderson and Bickley [1] related the degree of static indeterminacy of a rigid-jointed frame to the first Betti number of its graph model S . Generalising the Betti's number to a linear function and using an expansion process, Kaveh [15] developed a general method for determining the degree of static indeterminacy and degree of kinematic indeterminacy of different types of skeletal structures. Special methods have also been developed to transform the topological properties of space structures to those of their planar drawings, in order to simplify the calculation of their degrees of static indeterminacy, Ref. [16].

It should be noted that various methods for determining the degree of static indeterminacy of structures are a by-product of the general methods developed by Kaveh [15]. The method of expansion and its control at each step, using the intersection theorem presented in this chapter, provides a powerful tool for further studies in the field of structural analysis.

2.2.1 Mathematical Model of a Skeletal Structure

The mathematical model of a structure is considered to be a finite, connected graph S . There is a one-to-one correspondence between the elements of the structure and the members (edges) of S . There is also a one-to-one correspondence between the joints of the structure and the nodes of S , except for the support joints of some models.

For frame structures, shown in Fig. 2.1(a1) and (a2), two graph models can be considered. For the first model, all the support joints are identified as a single node called a *ground node*, as shown in Fig. 2.1(b1) and (b2). For the second model, all the joints are connected by an artificial arbitrary spanning tree, termed *ground tree*, Fig. 2.1(c1) and (c2).

Truss structures shown in Fig. 2.2(a1) and (a2) are assumed to be supported in a statically determinate fashion (Fig. 2.2(b1) and (b2)), and the effect of additional supports can easily be included in calculating the degree of static indeterminacy (DSI) of the corresponding structures. Alternatively artificial members can be added as shown in Fig. 2.2(c1) and (c2) to model the components of the corresponding supports. For a fixed support, two members and three members are considered for planar and space trusses, respectively, and one member is used for representing a roller.

The skeletal structures are considered to be in perfect condition; i.e. planar and space trusses have pin and ball joints only. Obviously, the effect of extra constraints or releases can be taken into account in determining their degrees of static indeterminacy and also in their analysis, Mauch and Fenves [17].

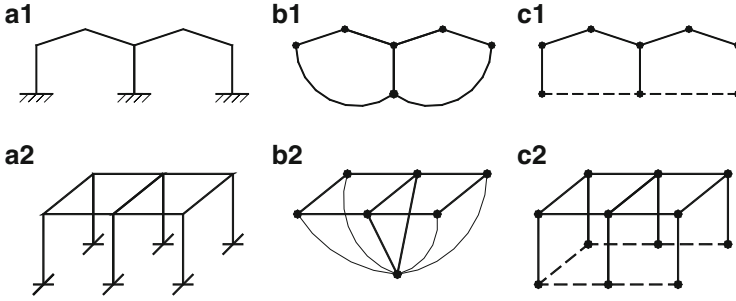


Fig. 2.1 Frame structures and their mathematical models. (a1) A plane frame. (b1) First model with a ground node. (c1) Second model with a ground tree. (a2) A space frame. (b2) First model with a ground node. (c2) Second model with a ground tree

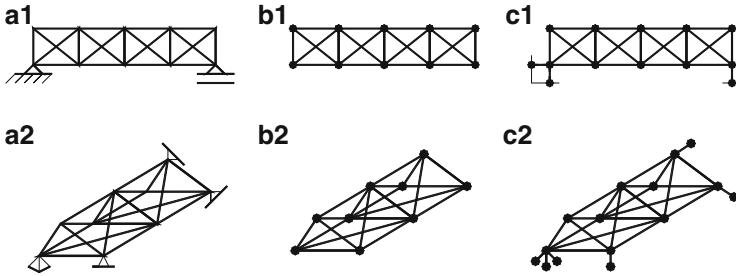


Fig. 2.2 Trusses and their graph models. (a1) A plane truss. (b1) First model without added members. (c1) Second model with replaced members. (a2) A space truss. (b2) First model without added members. (c2) Second model with replaced members

2.2.2 Expansion Process for Determining the Degree of Static Indeterminacy

The *degree of kinematic indeterminacy* of a structure is the number of independent displacement components (translations and rotations) required for describing a general state of deformation of the structure. The degree of kinematic indeterminacy is also referred to as the *total degrees of freedom* of the structure. On the other hand, the *degree of static indeterminacy* (redundancy) of a structure is the number of independent force components (forces and moments) required for describing a general equilibrium state of the structure. The DSI of a structure can be obtained by subtracting the number of independent equilibrium equations from the number of its unknown forces.

2.2.2.1 Classical Formulae

Formulae for calculating the DSI of various skeletal structures can be found in textbooks on structural mechanics, e.g. the DSI of a planar truss, denoted by $\gamma(S)$, can be calculated from,

$$\gamma(S) = M(S) - 2N(S) + 3, \quad (2.1)$$

where S is supported in a statically determinate fashion (internal indeterminacy). For extra supports (external indeterminacy), $\gamma(S)$ should be further increased by the number of additional unknown reactions.

A similar formula holds for space trusses:

$$\gamma(S) = M(S) - 3N(S) + 6. \quad (2.2)$$

For planar and space frames, the classical formulae is given as,

$$\gamma(S) = \alpha[M(S) - N(S) + 1], \quad (2.3)$$

where all supports are modelled as a datum (ground) node, and $\alpha = 3$ or 6 for planar and space frames, respectively.

All these formulae require counting a great number of members and nodes, which makes their application impractical for multi-member and complex pattern structures. These numbers provide only a limited amount of information about the connectivity properties of structures. In order to obtain additional information, the methods developed in the following sections will be utilised:

2.2.2.2 A Unifying Function

All the existing formulae for determining DSI have a common property, namely their linearity with respect to $M(S)$ and $N(S)$. Therefore, a general unifying function can be defined as,

$$\gamma(S) = aM(S) + bN(S) + c\gamma_0(S), \quad (2.4)$$

where $M(S)$, $N(S)$ and $\gamma_0(S)$ are the numbers of members, nodes and components of S , respectively. The coefficients a , b and c are integer numbers depending on both the type of the corresponding structure and the property which the function is expected to represent. For example, $\gamma(S)$ with appropriate values for a , b and c may describe the DSI of certain types of skeletal structures, Table 2.1. For $a = 1$, $b = -1$ and $c = 1$, $\gamma(S)$ becomes the first Betti number $b_1(S)$ of S , as described in Sect. 1.5.1.

Table 2.1 Coefficients of $\gamma(S)$ for different types of structures

Type of structure	a	b	c
Plane truss	+1	-2	+3
Space truss	+1	-3	+6
Plane frame	+3	-3	+3
Space frame	+6	-6	+6

2.2.2.3 An Expansion Process

An expansion process, in its simplest form, has been used by Müller-Breslau [18] for re-forming structural models, such as simple planar and space trusses. In his expansion process, the properties of typical subgraphs, selected in each step to be joined to the previously expanded subgraph, guarantee the determinacy of the simple truss. These subgraphs consist of two and three concurrent bars for planar and space trusses, respectively.

The idea can be extended to other types of structure, and more general subgraphs can be considered for addition at each step of the expansion process. A cycle, a planar subgraph, and a subgraph with prescribed connectivity properties are examples of these, which will be employed in this book. For example, the planar truss of Fig. 2.3a can be formed in four steps, joining a substructure S_i with $\gamma(S_i) = 1$ as shown in Fig. 2.3b, sequentially, as illustrated in Fig. 2.3c.

2.2.2.4 An Intersection Theorem

In a general expansion process, a subgraph S_i may be joined to another subgraph S_j in an arbitrary manner. For example, $\gamma(S_i)$ or $\gamma(S_j)$ may have any arbitrary value and the union $S_i \cup S_j$ may be a connected or a disjoint subgraph. The intersection $S_i \cap S_j$ may also be connected or disjoint. It is important to find the properties of $S_1 \cup S_2$ having the properties of S_1 , S_2 and $S_1 \cap S_2$. The following theorem provides a correct calculation of the properties of $S_i \cup S_j$. In order to have the formula in its general form, q subgraphs are considered in place of two subgraphs.

Theorem (Kaveh [15]). Let S be the union of q subgraphs $S_1, S_2, S_3, \dots, S_q$ with the following functions being defined:

$$\begin{aligned}
 \gamma(S) &= aM(S) + bN(S) + c\gamma_0(S), \\
 \gamma(S_i) &= aM(S_i) + bN(S_i) + c\gamma_0(S_i) \quad i = 1, 2, \dots, q, \\
 \gamma(A_i) &= aM(A_i) + bN(A_i) + c\gamma_0(A_i) \quad i = 2, 3, \dots, q,
 \end{aligned}$$

where $A_i = S^{i-1} \cap S_i$ and $S^i = S_1 \cup S_2 \cup \dots \cup S_i$. Then:

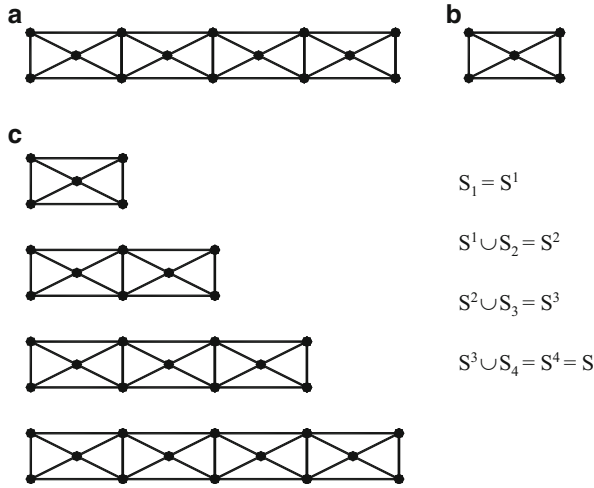


Fig. 2.3 Process for the formation of a planar truss. (a) A planar truss. (b) Selected unit. (c) The process of expansion as $S_1 = S^1 \rightarrow S^2 \rightarrow S^3 \rightarrow S^4 = S$

$$[\gamma(S) - c\gamma_0(S)] = \sum_{i=1}^q [\gamma(S_i) - c\gamma_0(S_i)] - \sum_{i=2}^q [\gamma(A_i) - c\gamma_0(A_i)] \quad (2.5)$$

For proof, the interested reader may refer to Kaveh [19].

Special Case. If S and each of its subgraphs considered for expansion (S_i for $i = 1, \dots, q$) are non-disjoint (connected), then Eq. 2.5 can be simplified as:

$$\gamma(S) = \sum_{i=1}^q \gamma(S_i) - \sum_{i=2}^q \bar{\gamma}(A_i), \quad (2.6)$$

where $\bar{\gamma}(A_i) = aM(A_i) + bN(A_i) + c$.

For calculating the DSI of a multi-member structure, one normally selects a repeated unit of the structure and joins these units sequentially in a connected form. Therefore, Eq. 2.6 can be applied in place of Eq. 2.5 to obtain the overall property of the structure.

2.2.2.5 A Method for Determining the DSI of Structures

Let S be the union of its repeated and/or simple pattern subgraphs S_i ($i = 1, \dots, q$). Calculate the DSI of each subgraph, using the appropriate coefficients from Table 2.1. Now perform the union–intersection method with the following steps:

Step 1: Join S_1 to S_2 to form $S^2 = S_1 \cup S_2$, and calculate the DSI of their intersection $A_2 = S_1 \cap S_2$. The value of $\gamma(S^2)$ can be found using Eq. 2.5 or Eq. 2.6, as appropriate.

Step 2: Join S_3 to S^2 to obtain $S^3 = S^2 \cup S_3$, and determine the DSI of $A_3 = S^2 \cap S_3$. Similarly to Step 1, calculate $\gamma(S^3)$.

Step k: Subsequently join S_{k+1} to S^k , calculating the DSI of $A_{k+1} = S^k \cap S_{k+1}$ and evaluating the magnitude of $\gamma(S^{k+1})$.

Repeat Step k until the entire structural model $S = \bigcup_{i=1}^q S_i$ has been reformed and its DSI determined.

In the above expansion process, the value of q depends on the properties of the substructures (subgraphs) which are considered for reforming S. These subgraphs have either simple patterns for which $\gamma(S_i)$ can easily be calculated, or the DSIs of which are already known.

In the process of expansion, if an intersection A_i itself has a complex pattern, further refinement is also possible; i.e. the intersection can be considered as the union of simpler subgraphs.

2.3 Formulation of the Force Method

In this section, a matrix formulation using the basic tools of structural analysis—equilibrium, compatibility and load–displacement relationships—is described. The notations are chosen from those most commonly utilized in structural mechanics.

2.3.1 Equilibrium Equations

Consider a structure S with M members and N nodes, which is $\gamma(S)$ times statically indeterminate. Select $\gamma(S)$ independent unknown forces as redundants. These unknown forces can be selected from external reactions and/or internal forces of the structure. Denote these redundants by:

$$\mathbf{q} = \left\{ q_1, q_2, \dots, q_{\gamma(S)} \right\}^t. \quad (2.7)$$

Remove the constraints corresponding to redundants, in order to obtain the corresponding statically determinate structure, known as the *basic (released or primary) structure* of S. Obviously, a basic structure should be rigid. Consider the joint loads as,

$$\mathbf{p} = \{p_1, p_2, \dots, p_n\}^t, \quad (2.8)$$

where n is the number of components for applied nodal loads.

Now the stress resultant distribution \mathbf{r} , due to the given load \mathbf{p} , for a linear analysis by the force method can be written as,

$$\mathbf{r} = \mathbf{B}_0 \mathbf{p} + \mathbf{B}_1 \mathbf{q}, \quad (2.9)$$

where \mathbf{B}_0 and \mathbf{B}_1 are rectangular matrices each having m rows, and n and $\gamma(S)$ columns, respectively, m being the number of independent components for member forces. $\mathbf{B}_0 \mathbf{p}$ is known as a *particular solution*, which satisfies equilibrium with the imposed load, and $\mathbf{B}_1 \mathbf{q}$ is a *complementary solution*, formed from a maximal set of independent self-equilibrating stress systems (S.E.Ss), known as a *statical basis*.

Example 1. Consider a planar truss, as shown in Fig. 2.4a, which is two times statically indeterminate. EA is taken to be the same for all the members.

One member force and one component of a reaction may be taken as redundants. Alternatively, two member forces can also be selected as unknowns, as shown in Fig. 2.4b. Selecting the latter choice, the corresponding \mathbf{B}_0 and \mathbf{B}_1 matrices can now be obtained by applying unit values of p_i ($i = 1, 2$) and q_j ($j = 1, 2$), respectively:

$$\mathbf{B}_0^t = \begin{bmatrix} -1 & 0 & 0 & 0 & \sqrt{2} & 0 & -1 & 0 & 0 & 0 \\ -2 & -1 & +1 & 0 & \sqrt{2} & 0 & -1 & \sqrt{2} & 0 & -1 \end{bmatrix},$$

and

$$\mathbf{B}_1^t = \begin{bmatrix} -1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 & +1 & +1 & -1/\sqrt{2} & 0 & 0 & 0 \\ 0 & -1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 & 0 & -1/\sqrt{2} & +1 & +1 & -1/\sqrt{2} \end{bmatrix}.$$

The columns of \mathbf{B}_1 (rows of \mathbf{B}_1^t) form a statical basis of S . The underlying subgraph of a typical self-equilibrating stress system (for $q_2 = 1$) is shown in bold lines, Fig. 2.4b.

Example 2. Consider a portal frame shown in Fig. 2.5a, which is three times statically indeterminate.

This structure is made statically determinate by an imaginary cut at the middle of its beam. The unit value of external load p_1 and each of the bi-actions q_i ($i = 1, 2, 3$) lead to the formation of \mathbf{B}_0 and \mathbf{B}_1 matrices, in which the two end bending moments (M_i, M_j) of a member are taken as its member forces. Using the sign convention introduced in Chap. 1, \mathbf{B}_0 and \mathbf{B}_1 matrices are formed as:

$$\mathbf{B}_0^t = [+4 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0],$$

Fig. 2.4 A statically indeterminate planar truss. (a) A planar truss. (b) The selected unknown forces

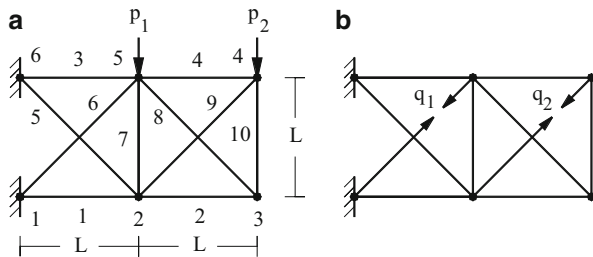
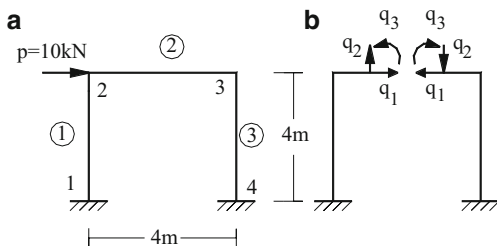


Fig. 2.5 A statically indeterminate frame. (a) A portal frame S. (b) The basic structure of S



and

$$\mathbf{B}_1^t = \begin{bmatrix} +4 & 0 & 0 & 0 & 0 & -4 \\ -2 & +2 & -2 & -2 & +2 & -2 \\ -1 & +1 & -1 & +1 & -1 & +1 \end{bmatrix}.$$

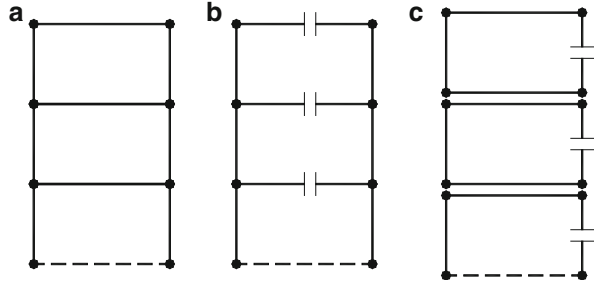
The columns of \mathbf{B}_1 form a statical basis of S, and the underlying subgraph of each self-equilibrating stress system is a cycle, as illustrated in bold lines, Fig. 2.5b. Notice that three self-equilibrating stress systems can be formed on each cycle of a planar frame.

In both of the above examples, particular and complementary solutions are obtained from the same basic structure. However, this is not a necessary requirement, as imagined by some authors. In fact a particular solution is any solution satisfying equilibrium with the applied loads, and a complementary solution is any maximal set of independent self-equilibrating systems. The latter is a basis of a vector space over the field of real numbers, known as a *complementary solution space*, Henderson and Maunder [20].

Using the same basic structure is equivalent to searching for a cycle basis of a graph, but restricting the search to fundamental cycles only, which is convenient but not efficient when the structure is complex or cycle bases with specific properties are needed.

As an example, consider a three-storey frame as shown in Fig. 2.6a. A cut system as shown in Fig. 2.6b corresponds to a statical basis, containing three self-equilibrating stress systems formed on each element of the cycle basis shown in Fig. 2.6b. However, the same particular solution can be employed with a statical basis formed on the cycles of the basis shown in Fig. 2.6c.

Fig. 2.6 A three-storey frame with different cut systems



A basic structure need not be selected as a determinate one. For a redundant basic structure, one may obtain the necessary data either by analysing it first for the loads \mathbf{p} and bi-actions $q_i = 1 (i = 1, 2, \dots, \gamma(S))$, or by using existing information.

2.3.2 Member Flexibility Matrices

In the force method of analysis, the determination of the member flexibility matrix is an important step. A number of alternative methods are available for the formation of displacement-force relationships describing the flexibility properties of the members. Four such approaches are:

1. Inversion of the force-displacement relationship;
2. Unit load method;
3. Castigliano's theorem;
4. Solution of differential equations for member displacements.

In the following, the unit load method is briefly described for the formation of the flexibility matrices:

Consider a general element with n member forces,

$$\mathbf{r}_m^t = \{r_1, r_2, \dots, r_n\}, \quad (2.10)$$

and member displacements:

$$\mathbf{u}_m^t = \{u_1, u_2, \dots, u_n\}. \quad (2.11)$$

A typical component of the displacement u_i can be found using the unit load method as:

$$u_i = \iiint_V \bar{\sigma}_i^t \epsilon dV, \quad (2.12)$$

where $\bar{\sigma}_i$ represents the matrix of statically equivalent stresses due to a unit load in the direction of r_i , and ϵ is the exact strain matrix due to all applied forces \mathbf{r}_m .

The unit loads can be used in turn for all the points where member force are applied, and therefore,

$$\mathbf{u}_m = \iiint_V \bar{\boldsymbol{\sigma}}^t \boldsymbol{\varepsilon} dV, \quad (2.13)$$

where,

$$\bar{\boldsymbol{\sigma}} = \{\bar{\boldsymbol{\sigma}}_1 \bar{\boldsymbol{\sigma}}_2 \dots \bar{\boldsymbol{\sigma}}_n\}^t. \quad (2.14)$$

For a linear system,

$$\boldsymbol{\sigma} = \mathbf{c} \mathbf{r}_m, \quad (2.15)$$

where \mathbf{c} is the stress distribution due to unit forces \mathbf{r}_m .

The stress-strain relationship can be written as:

$$\boldsymbol{\varepsilon} = \boldsymbol{\phi} \boldsymbol{\sigma} = \boldsymbol{\phi} \mathbf{c} \mathbf{r}_m. \quad (2.16)$$

Substituting in Eq. 2.13 leads to,

$$\mathbf{u}_m = \iiint_V \bar{\boldsymbol{\sigma}}^t \boldsymbol{\phi} \mathbf{c} dV \mathbf{r}_m \quad (2.17)$$

or,

$$\mathbf{u}_m = \mathbf{f}_m \mathbf{r}_m, \quad (2.18)$$

where,

$$\mathbf{f}_m = \iiint_V \bar{\boldsymbol{\sigma}}^t \boldsymbol{\phi} \mathbf{c} dV, \quad (2.19)$$

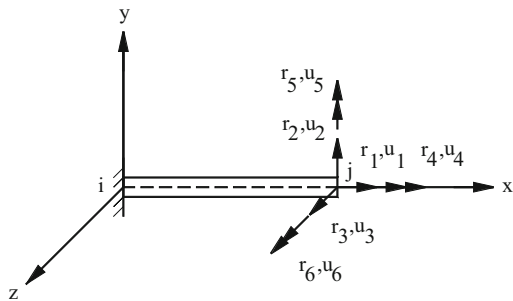
represents the element flexibility matrix.

The evaluation of $\bar{\boldsymbol{\sigma}}$ representing the exact stress distribution due to the forces \mathbf{r}_m , may not be possible, and hence an approximate relationship should be used. Usually the matrix \mathbf{c} is selected such that it will satisfy at least the equations of equilibrium. Denoting this approximate matrix by $\bar{\mathbf{c}}$, and using $\bar{\boldsymbol{\sigma}} = \bar{\mathbf{c}}$:

$$\mathbf{f}_m = \iiint_V \bar{\mathbf{c}}^t \boldsymbol{\phi} \bar{\mathbf{c}} dV. \quad (2.20)$$

This equation will be used for the derivation of the flexibility matrices of some finite elements in the proceeding sections.

Fig. 2.7 A beam element and selected independent member forces



For a bar element of a space truss, however, the flexibility matrix can easily be obtained using Hooke's law as already discussed in Chap. 1. For a beam element ij of a space frame, y and z axes are taken as the principal axes of the beams cross sections, Fig. 2.7. The forces of end j are selected as a set of independent member forces, and the element is considered to be supported at point i . The axial, torsional, and flexural behaviour in respective planes are uncoupled, and therefore, one needs only to consider the flexibility relationships for four separate members:

1. An axial force member (along x axis);
2. A pure torsional member (about x axis);
3. A beam bent about y axis;
4. A beam bent about z axis.

Direct adaptation of the flexibility relationships derived in Chap. 1, gives the following 6×6 flexibility matrix,

$$\mathbf{f}_m = \begin{bmatrix} \frac{L}{EA} & & & & & \\ 0 & \frac{L^3}{3EI_z} & & & & \\ 0 & 0 & \frac{L^3}{3EI_y} & & & \\ 0 & 0 & 0 & \frac{L}{GJ} & & \\ 0 & 0 & -\frac{L^2}{2EI_y} & 0 & \frac{L}{EI_y} & \\ 0 & \frac{L^2}{2EI_z} & 0 & 0 & 0 & \frac{L}{EI_z} \end{bmatrix}, \quad (2.21)$$

where G is the shear modulus, I_y and I_z are the moments of inertia with respect to y and z axes, respectively. J is the Saint-Venant torsion constant of the cross section.

2.3.3 *Explicit Method for Imposing Compatibility*

The compatibility equations in the actual structure will now be derived. Using the displacement-load relationship for each member, and collecting them in the diagonal of the unassembled flexibility matrix \mathbf{F}_m , one can write member distortions as:

$$\mathbf{u} = \mathbf{F}_m \mathbf{r} = \mathbf{F}_m \mathbf{B}_0 \mathbf{p} + \mathbf{F}_m \mathbf{B}_1 \mathbf{q}. \quad (2.22)$$

In matrix form:

$$[\mathbf{u}] = [\mathbf{F}_m] [\mathbf{B}_0 \ \mathbf{B}_1] \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}. \quad (2.23)$$

From the contragradient principle of Chap. 1,

$$[\mathbf{v}] = \begin{bmatrix} \mathbf{B}_0^t \\ \mathbf{B}_1^t \end{bmatrix} [\mathbf{u}]. \quad (2.24)$$

Combining Eqs. 2.23 and 2.24 results in,

$$\begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_c \end{bmatrix} = \begin{bmatrix} \mathbf{B}_0^t \\ \mathbf{B}_1^t \end{bmatrix} [\mathbf{F}_m] [\mathbf{B}_0 \ \mathbf{B}_1] \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}, \quad (2.25)$$

in which \mathbf{v}_0 contains the displacements corresponding to the force components of \mathbf{p} , and \mathbf{v}_c denotes the relative displacements of the cuts for the basic structure. Performing the multiplication,

$$\begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_c \end{bmatrix} = \begin{bmatrix} \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}_0 & \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}_1 \\ \mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_0 & \mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}. \quad (2.26)$$

Defining:

$$\begin{aligned} \mathbf{D}_{00} &= \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}_0, & \mathbf{D}_{10} &= \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}_1, \\ \mathbf{D}_{01} &= \mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_0, & \mathbf{D}_{11} &= \mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_1, \end{aligned} \quad (2.27)$$

the expansion of Eq. 2.14 leads to:

$$\mathbf{v}_0 = \mathbf{D}_{00} \mathbf{p} + \mathbf{D}_{01} \mathbf{q}, \quad (2.28)$$

and

$$\mathbf{v}_c = \mathbf{D}_{10} \mathbf{p} + \mathbf{D}_{11} \mathbf{q}. \quad (2.29)$$

Consider now the compatibility conditions as:

$$\mathbf{v}_c = 0. \quad (2.30)$$

Equation 2.30 together with Eq. 2.29 leads to:

$$\mathbf{q} = -\mathbf{D}_{11}^{-1}\mathbf{D}_{10}\mathbf{p} = \mathbf{F}\mathbf{p}. \quad (2.31)$$

Substituting in Eq. 2.22 yields,

$$\mathbf{v}_0 = [\mathbf{D}_{00} - \mathbf{D}_{01}\mathbf{D}_{11}^{-1}\mathbf{D}_{10}]\mathbf{p}, \quad (2.32)$$

and the stress resultant in a structure can be obtained as:

$$\mathbf{r} = [\mathbf{B}_0 - \mathbf{B}_1\mathbf{D}_{11}^{-1}\mathbf{D}_{10}]\mathbf{p}. \quad (2.33)$$

2.3.4 Implicit Approach for Imposing Compatibility

A direct application of the work principle of Chap. 1, can also be used to impose the compatibility conditions in an implicit form as follows:

Since the structure is considered to be linearly elastic, a linear relation exists between the unknown forces \mathbf{q} and the applied forces \mathbf{p} ; i.e.

$$\mathbf{q} = \mathbf{Q}\mathbf{p}, \quad (2.34)$$

where \mathbf{Q} is a transformation matrix which is still unknown.

Equation 2.9 can now be written as:

$$\mathbf{r} = \mathbf{B}_0\mathbf{p} + \mathbf{B}_1\mathbf{Q}\mathbf{p} = (\mathbf{B}_0 + \mathbf{B}_1\mathbf{Q})\mathbf{p} = \mathbf{B}\mathbf{p}. \quad (2.35)$$

Using the work theorem:

$$\mathbf{P}^t\mathbf{v} = \mathbf{r}^t\mathbf{u} = \mathbf{p}^t\mathbf{B}^t\mathbf{u}. \quad (2.36)$$

Now a set of suitable internal forces, \mathbf{r}^* , is considered which is statically equivalent to the external loads. From work principle:

$$\mathbf{p}^t\mathbf{v} = \mathbf{r}^{*t}\mathbf{u}, \quad (2.37)$$

or

$$\mathbf{p}^t\mathbf{v} = \mathbf{p}^t\mathbf{B}_0^t\mathbf{u}. \quad (2.38)$$

Comparison of the above two equations leads to:

$$\mathbf{p}^t \mathbf{B}^t \mathbf{u} = \mathbf{p}^t \mathbf{B}_0^t \mathbf{u}. \quad (2.39)$$

Substituting $\mathbf{u} = \mathbf{F}_m \mathbf{B} \mathbf{p}$ in the above equation:

$$\mathbf{p}^t \mathbf{B}^t \mathbf{F}_m \mathbf{B} \mathbf{p} = \mathbf{p}^t \mathbf{B}_0^t \mathbf{F}_m \mathbf{B} \mathbf{p}. \quad (2.40)$$

This holds for any \mathbf{p} , and therefore:

$$\mathbf{B}^t \mathbf{F}_m \mathbf{B} = \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}. \quad (2.41)$$

From Eq. 2.35 by transposition,

$$\mathbf{B}^t = \mathbf{B}_0^t + \mathbf{Q}^t \mathbf{B}_1^t, \quad (2.42)$$

therefore,

$$(\mathbf{B}_0^t + \mathbf{Q}^t \mathbf{B}_1^t) \mathbf{F}_m \mathbf{B} = \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}, \quad (2.43)$$

or

$$\mathbf{Q}^t \mathbf{B}_1^t \mathbf{F}_m (\mathbf{B}_0 + \mathbf{B}_1 \mathbf{Q}) = \mathbf{0}, \quad (2.44)$$

or

$$\mathbf{Q}^t (\mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_0 + \mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_1 \mathbf{Q}) = \mathbf{0}. \quad (2.45)$$

Using the notation introduced in Eq. 2.15 leads to,

$$\mathbf{Q}^t (\mathbf{D}_{10} + \mathbf{D}_{11} \mathbf{Q}) = \mathbf{0}, \quad (2.46)$$

or

$$\mathbf{D}_{10} + \mathbf{D}_{11} \mathbf{Q} = \mathbf{0}. \quad (2.47)$$

Therefore,

$$\mathbf{Q} = -\mathbf{D}_{11}^{-1} \mathbf{D}_{10}, \quad (2.48)$$

and

$$\mathbf{q} = -\mathbf{D}_{11}^{-1} \mathbf{D}_{10} \mathbf{p}, \quad (2.49)$$

and Eq. 2.20 is obtained as in the previous approach.

2.3.5 Structural Flexibility Matrices

The overall flexibility matrix of a structure can be expressed as:

$$\mathbf{v} = \mathbf{F}\mathbf{p}. \quad (2.50)$$

Pre-multiplying the above equation by \mathbf{p}^t , we have:

$$\mathbf{p}^t \mathbf{F} \mathbf{p} = \mathbf{p}^t \mathbf{B}_0^t \mathbf{F}_m \mathbf{B} \mathbf{p}. \quad (2.51)$$

Since \mathbf{p} is arbitrary,

$$\mathbf{F} = \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}, \quad (2.52)$$

or

$$\mathbf{F} = \mathbf{B}_0^t \mathbf{F}_m (\mathbf{B}_0 + \mathbf{B}_1 \mathbf{Q}), \quad (2.53)$$

or

$$\mathbf{F} = \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}_0 - \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}_1 \mathbf{D}_{11}^{-1} \mathbf{D}_{10}. \quad (2.54)$$

Since \mathbf{F}_m is symmetric, it follows that:

$$\mathbf{D}_{10}^t = \mathbf{B}_0^t \mathbf{F}_m \mathbf{B}_1 = \mathbf{B}_0^t \mathbf{F}_m^t \mathbf{B}_1. \quad (2.55)$$

Therefore, the *overall flexibility matrix* (known also as influence matrix) of the structure is obtained as,

$$\mathbf{F} = \mathbf{D}_{00} - \mathbf{D}_{10}^t \mathbf{D}_{11}^{-1} \mathbf{D}_{10}, \quad (2.56)$$

and $\mathbf{D}_{11} = \mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_1 = \mathbf{G}$ is also referred to as the *flexibility matrix* of the structure. In this book, properties of \mathbf{G} will be studied, since its pattern is the most important factor in optimal analysis of the structure by the force method.

Equation 2.34 can now be used to calculate the nodal displacements.

2.3.6 Computational Procedure

The sequence of computational steps for the force method can be summarized as:

1. Construct \mathbf{B}_0 and obtain \mathbf{B}_0^t .
2. Construct \mathbf{B}_1 and obtain \mathbf{B}_1^t .
3. Form unassembled flexibility matrix \mathbf{F}_m .

4. Form $\mathbf{F}_m \mathbf{B}_0$ followed by $\mathbf{F}_m \mathbf{B}_1$.
5. Calculate \mathbf{D}_{00} , \mathbf{D}_{10}^t , \mathbf{D}_{10} and \mathbf{D}_{11} , sequentially.
6. Compute $-\mathbf{D}_{11}^{-1}$.
7. Calculate $\mathbf{Q} = -\mathbf{D}_{11}^{-1} \mathbf{D}_{10}$.
8. Form $\mathbf{B}_1 \mathbf{Q}$ and find $\mathbf{B} = \mathbf{B}_0 + \mathbf{B}_1 \mathbf{Q}$.
9. Form $\mathbf{D}_{10}^t \mathbf{Q}$ and find $\mathbf{D}_{00} + \mathbf{D}_{10}^t \mathbf{Q}$.
10. Compute the internal forces as $\mathbf{r} = \mathbf{Bp}$.
11. Compute nodal displacements as $\mathbf{v}_0 = \mathbf{Fp}$.

Example 3. In this example, the complete analysis of the truss of Example 1 will be given.

\mathbf{B}_0 and \mathbf{B}_1 matrices are already formed in Example 1 of Sect. 2.2.1. The unassembled flexibility matrix can be constructed as:

$$\mathbf{F}_m = \frac{L}{EA} \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & \sqrt{2} & & & & \\ & & & & & \sqrt{2} & & & \\ & & & & & & 1 & & \\ & & \mathbf{0} & & & & & \sqrt{2} & \\ & & & & & & & & \sqrt{2} \\ & & & & & & & & & 1 \end{bmatrix}.$$

Using the above matrix and the matrices from Example 1, leads to:

$$\mathbf{D}_{11} = \frac{L}{EA} \begin{bmatrix} 2\sqrt{2} + 3/2 & 1/2 \\ 1/2 & 2\sqrt{2} + 2 \end{bmatrix},$$

and

$$\mathbf{D}_{10} = \frac{L}{EA} \begin{bmatrix} 2 + 2/\sqrt{2} & 2 + 2/\sqrt{2} \\ 1/\sqrt{2} & 2 + 3/\sqrt{2} \end{bmatrix}.$$

Substituting in Eq. 2.25, results in:

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = - \begin{bmatrix} 2\sqrt{2} + 3/2 & 1/2 \\ 1/2 & 2\sqrt{2} + 2 \end{bmatrix}^{-1} \begin{bmatrix} 2 + 2/\sqrt{2} & 2 + 2/\sqrt{2} \\ 1/\sqrt{2} & 2 + 3/\sqrt{2} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}.$$

Taking $p_1 = p_2 = P$ for simplicity, and solving the above equations gives:

$$q_1 = -1.43P \text{ and } q_2 = -1.17P.$$

Equation 2.3 is then used to calculate the member forces as:

$$\mathbf{r} = \{r_1 \ r_2 \ r_3 \ r_4 \ r_5 \ r_6 \ r_7 \ r_8 \ r_9 \ r_{10}\}^t \\ = \{-1.95P \ -0.17P \ 2.05P \ 0.83P \ 1.36P \ -1.44P \ -0.12P \ 0.24P \ -1.17P \ -0.17P\}^t.$$

Nodal displacements can be found using Eq. 2.25.

Example 4. In this example, the complete analysis of the frame in Example 2 is given.

\mathbf{B}_0 and \mathbf{B}_1 matrices are already formed in Example 2 of Sect. 2.2.1. The unassembled flexibility matrix of the structure, using the sign convention introduced in Chap. 1, is formed as:

$$\mathbf{F}_m = \frac{L}{6EI} \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & & & \\ & & 2 & -1 & \\ & & -1 & 2 & \\ & & & & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}.$$

Substituting in Eq. 2.21 leads to:

$$\mathbf{D}_{11} = \frac{L}{6EI} \begin{bmatrix} 64 & 0 & -24 \\ 0 & 56 & 0 \\ -24 & 0 & 18 \end{bmatrix},$$

and

$$\mathbf{D}_{10} = \frac{L}{6EI} \begin{bmatrix} 32 \\ -24 \\ -12 \end{bmatrix}.$$

The inverse of \mathbf{D}_{11} is computed as,

$$\mathbf{D}_{11}^{-1} = \frac{6EI}{L} \begin{bmatrix} 18/576 & 0 & 3/72 \\ 0 & 576 & 0 \\ 3/72 & 0 & 1/9 \end{bmatrix},$$

and \mathbf{Q} can be obtained as:

$$\mathbf{Q} = -\mathbf{D}_{11}^{-1} \mathbf{D}_{10} = \begin{bmatrix} -1/2 \\ +3/7 \\ 0 \end{bmatrix}.$$

Matrix \mathbf{B} is now computed as,

$$\mathbf{B} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} +4 & -2 & -1 \\ 0 & +2 & +1 \\ 0 & -2 & -1 \\ 0 & -2 & +1 \\ 0 & +2 & -1 \\ -4 & -2 & +1 \end{bmatrix} \begin{bmatrix} -1/2 \\ +3/7 \\ 0 \end{bmatrix},$$

and finally by using Eq. 2.23 the member forces are obtained as:

$$\mathbf{r} = \{ +11.43 \quad +8.57 \quad -8.57 \quad -8.57 \quad +8.57 \quad +11.43 \}^t.$$

General Loading. When members are loaded in a general form, then it must be replaced by an equivalent loading. Such a loading can be found as the superposition of two cases; case 1 consists of the given loading but the ends of the member are fixed. The fixed end forces (actions), denoted by FEA, can be found using tables from books on strength of materials. Case 2 is the given structure subjected to the reverse of the fixed end actions only. Obviously, the sum of the loads and reactions of case 1 and case 2 will be the same effect as that of the given loading. This superposition process is illustrated in the following example:

Example 5. A two-span beam is considered as shown in Fig. 2.8a. The fixed end actions are provided in b, and the equivalent forces are illustrated in Fig. 2.8c. The structure is twice indeterminate, and the primary structure is obtained by introducing two hinges as shown in d. The applied nodal forces and redundants are depicted in Fig. 2.8e, f, respectively.

\mathbf{B}_0 and \mathbf{B}_1 matrices are formed as,

$$\mathbf{B}_0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & +1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & +1 \end{bmatrix} \quad \text{and} \quad \mathbf{B}_1 = \begin{bmatrix} -1 & 0 \\ 0 & +1 \\ 0 & -1 \\ 0 & 0 \end{bmatrix},$$

and the unassembled flexibility matrix of the structure is constructed as:

$$\mathbf{F}_m = \frac{L}{6EI} \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & & \\ & & 2 & -1 \\ & & -1 & 2 \end{bmatrix}.$$

Substituting in Eq. 2.27 leads:

$$\mathbf{D}_{11} = \frac{L}{6EI} \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix},$$

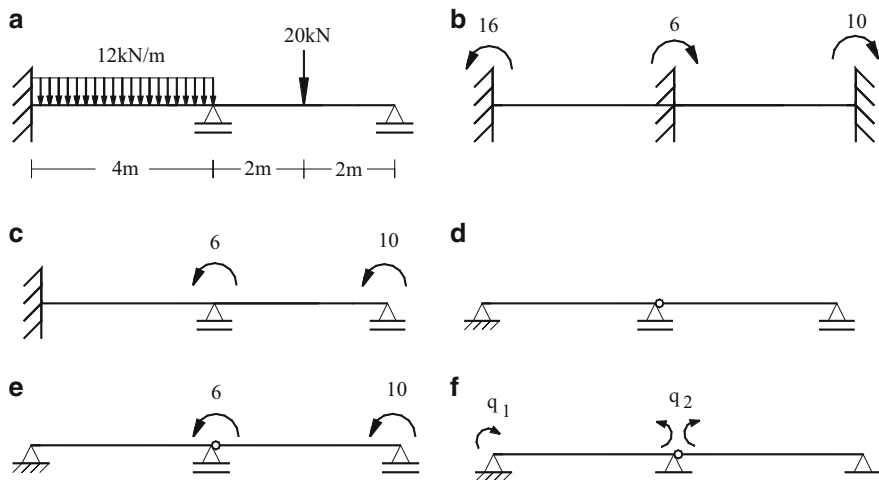


Fig. 2.8 A two-span beam with general loading. (a) A two-span beam. (b) Fixed end actions. (c) The equivalent loading. (d) The selected primary structure. (e) Applied force on primary structure. (f) Redundants on primary structure

and

$$\mathbf{D}_{10} = \frac{L}{6EI} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

The inverse of \mathbf{D}_{11} is computed as,

$$\mathbf{D}_{11}^{-1} = -\frac{1}{7} \times \frac{6EI}{L} \begin{bmatrix} 4 & -1 \\ -1 & 2 \end{bmatrix},$$

and \mathbf{Q} can be obtained as:

$$\mathbf{Q} = -\mathbf{D}_{11}^{-1} \mathbf{D}_{10} = -\frac{1}{7} \begin{bmatrix} 4 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix} = -\frac{1}{7} \begin{bmatrix} 7 & 2 & -1 \\ 0 & 3 & 2 \end{bmatrix}.$$

Now \mathbf{r} is computed as,

$$\mathbf{r}' = \left(\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 2/7 & -1/7 \\ 0 & -3/7 & -2/7 \\ 0 & 3/7 & 2/7 \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} 0 \\ 6 \\ 10 \end{bmatrix} = \begin{bmatrix} 0.285 \\ 0.572 \\ 5.428 \\ 10.00 \end{bmatrix},$$

adding the fixed end reaction, the final member forces are obtained as:

$$\mathbf{r} = \{ 16.285 \quad -15.428 \quad 15.428 \quad 0.000 \}^t.$$

2.3.7 Optimal Force Method

For an efficient force method, the matrix \mathbf{G} should be:

- (a) Sparse;
- (b) Well conditioned;
- (c) Properly structured, i.e. narrowly banded.

In order to provide the properties (a) and (b) for \mathbf{G} , the structure of \mathbf{B}_1 should be carefully designed, since the pattern of \mathbf{F}_m for a given discretization is unchanged; i.e. a suitable statical basis should be selected. This problem is treated in different forms by various methods. In the following, graph theoretical methods are described for the formation of appropriate statical bases of different types of skeletal structures. The property (c) above has a totally combinatorial nature and is studied in Chaps. 5 and 6.

Pattern Equivalence. Matrix \mathbf{B}_1 containing a statical basis, in partitioned form, is pattern equivalent to \mathbf{C}^t , where \mathbf{C} is the cycle-member incidence matrix. Similarly, $\mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_1$ is pattern equivalent to $\mathbf{C} \mathbf{C}^t$ or $\mathbf{C} \mathbf{C}^t$. This correspondence transforms some structural problems associated with the characterization of $\mathbf{G} = \mathbf{B}_1^t \mathbf{F}_m \mathbf{B}_1$ into combinatorial problems of dealing with $\mathbf{C} \mathbf{C}^t$.

As an example, if a sparse matrix \mathbf{G} is required, this can be achieved by increasing the sparsity of $\mathbf{C} \mathbf{C}^t$. Similarly for a banded \mathbf{G} , instead of ordering the elements of a statical basis (self-equilibrating stress systems), one can order the corresponding cycles. This transformation has many advantages, such as:

1. The dimension of $\mathbf{C} \mathbf{C}^t$ is often smaller than that of \mathbf{G} . For example, for a space frame the dimension of $\mathbf{C} \mathbf{C}^t$ is six-fold and for a planar frame three-fold smaller than that of \mathbf{G} . Therefore, the optimisation process becomes much simpler when combinatorial properties are used.
2. The entries of \mathbf{C} and $\mathbf{C} \mathbf{C}^t$ are elements of \mathbb{Z}_2 and therefore easier to operate on, compared to \mathbf{B}_1 and \mathbf{G} which have real numbers as their entries.
3. The advances made in combinatorial mathematics and graph theory become directly applicable to structural problems.
4. A correspondence between algebraic and graph theoretical methods becomes established.

2.4 Force Method for the Analysis of Frame Structures

In this section, frame structures are considered in their perfect conditions; i.e. the joints of a frame are assumed to be rigid, and connected to each other by elastic members and supported by a rigid foundation.

For this type of skeletal structure, a statical basis can be generated on a cycle basis of its graph model. The function representing the degree of static

indeterminacy, $\gamma(S)$, of a rigid-jointed structure is directly related to the first Betti number $b_1(S)$ of its graph model,

$$\gamma(S) = \alpha b_1(S) = \alpha[M(S) - N(S) + b_0(S)], \quad (2.57)$$

where $\alpha = 3$ or 6 depending on whether the structure is either a planar or a space frame.

For a frame structure, matrix \mathbf{B}_0 can easily be generated using a shortest route tree of its model, and \mathbf{B}_1 can be formed by constructing 3 or 6 self-equilibrating stress systems on each element of a cycle basis of S .

In order to obtain a flexibility matrix of maximal sparsity, special cycle bases should be selected as defined in the next section. Methods for the formation of a cycle basis can be divided into two groups, namely

(a) Topological methods, (b) graph theoretical approaches.

Topological methods useful for the formation of cycle bases by hand, were developed by Henderson and Maunder [20] and a complete description of these methods is presented in Kaveh [3]. Graph-theoretical methods suitable for computer applications were developed by Kaveh [21].

2.4.1 Minimal and Optimal Cycle Bases

A matrix is called *sparse* if many of its entries are zero. The interest in sparsity arises because its exploitation can lead to enormous computational saving, and because many large matrices that occur in the analysis of practical structures, can be made sparse if they are not already so. A matrix can therefore be considered sparse, if there is an advantage in exploiting its zero entries.

The *sparsity coefficient* χ of a matrix is defined to be its number of non-zero entries. A cycle basis $C = \{C_1, C_2, C_3, \dots, C_{b_1(S)}\}$ is called *minimal*, if it corresponds to a minimum value of:

$$L(C) = \sum_{i=1}^{b_1(S)} L(C_i). \quad (2.58)$$

Obviously, $\chi(C) = L(C)$ and a minimal cycle basis can be defined as a basis which corresponds to minimum $\chi(C)$. A cycle basis for which $L(C)$ is near minimum is called a *subminimal* cycle basis of S .

A cycle basis corresponding to maximal sparsity of the \mathbf{CC}^t is called an *optimal* cycle basis of S . If $\chi(\mathbf{CC}^t)$ does not differ considerably from its minimum value, then the corresponding basis is termed *suboptimal*.

The matrix intersection coefficient $\sigma_i(C)$ of row i of cycle member incidence matrix \mathbf{C} is the number of row j such that:

- (a) $j \in \{i + 1, i + 2, \dots, b_1(S)\}$,
- (b) $C_i \cap C_j \neq \emptyset$, i.e. there is at least one k such that the column k of both cycles C_i and C_j (rows i and j) contain non-zero entries.

Now it can be shown that:

$$\chi(\mathbf{D}) = b_1(S) + 2 \sum_{i=1}^{b_1(S)-1} \sigma_i(\mathbf{C}). \quad (2.59)$$

This relationship shows the correspondence of a cycle member incidence matrix \mathbf{C} and that of its cycle basis adjacency matrix. In order to minimize $\chi(\mathbf{C}\mathbf{C}^t)$, the value of $\sum_{i=1}^{b_1(S)-1} \sigma_i(\mathbf{C})$ should be minimized, since $b_1(S)$ is a constant for a given structure S , i.e. γ -cycles with a minimum number of overlaps should be selected.

In the force method, an optimal cycle basis is needed corresponding to the maximum sparsity of $\mathbf{C}\mathbf{C}^t$ matrix. However, because of the complexity of this problem, most of the research has been concentrated on minimal cycle basis selection, except those of Ref. [22], which minimize the overlaps of the cycles rather than only their length.

2.4.2 Selection of Minimal and Subminimal Cycle Bases

Cycle bases of graphs have many applications in various fields of engineering. The amount of work in these applications depends on the cycle basis chosen. A basis with shorter cycles reduces the time and storage required for some applications; i.e. it is ideal to select a minimal cycle basis, and for some other applications minimal overlaps of cycles are needed; i.e. optimal cycle bases are preferred. In this section, the formation of minimal and subminimal cycle bases is first discussed. Then the possibility of selecting optimal and suboptimal cycle bases is investigated.

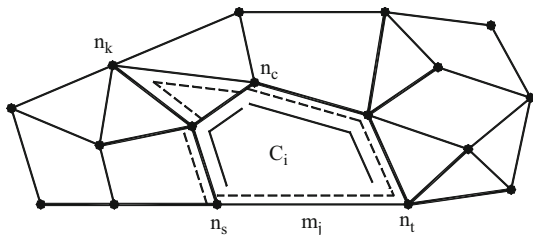
Minimal cycle bases were considered first by Stepanec [23] and improved by Zykov [24]. Many practical algorithms for selecting subminimal cycle bases have been developed by Kaveh [15].

In this section, the merits of the algorithms developed by different authors are discussed; a method is given for selection of minimal cycle bases, and efficient approaches are presented for the generation of subminimal cycle bases.

Formation of a Minimal Cycle on a Member. A minimal length cycle C_i on a member m_j , called its *generator*, can be formed by using the shortest route tree algorithm as follows:

Start the formation of two SRTs rooted at the two end nodes n_s and n_t of m_j , and terminate the process as soon as the SRTs intersect each other (not through m_j itself) at say n_c . The shortest paths between n_s and n_c , and n_t and n_c , together with m_j , form

Fig. 2.9 A minimal cycle on a member



a minimal cycle C_i on m_j . Using this algorithm, cycles of prescribed lengths can also be generated.

As an example, C_i is a minimal cycle on m_j in Fig. 2.9. The SRTs are shown in bold lines. The generation of SRTs is terminated as soon as n_c has been found.

A minimal cycle on a member m_j passing through a specified node n_k can similarly be generated. An SRT rooted at n_k is formed and as soon as it hits the end nodes of m_j , the shortest paths are found by backtracking between n_k and n_s , and n_k and n_t . These paths together with m_j form the required cycle. As an example, a minimal cycle on m_j containing n_k , is illustrated by dashed lines in Fig. 2.9.

Different Cycle Sets for Selecting a Cycle Basis. It is obvious that a general cycle can be decomposed into its simple cycles. Therefore, it is natural to confine the considered set to only simple cycles of S . Even such a cycle set, which forms a subspace of the cycle space of the graph, has many elements and is therefore uneconomical for practical purposes.

In order to overcome the above difficulty, Kaveh [15] used an expansion process, selecting the smallest admissible (independent with additional restriction) cycles, one at a time, until $b_1(S)$ cycles forming a basis had been obtained. In this approach, a very limited number of cycles were checked for being an element of a basis. As an example, the expansion process for selecting a cycle basis of S is illustrated in Fig. 2.10.

Hubicka and Sysl  [25] employed a similar approach, without the restriction of selecting one cycle at each step of expansion. In their method, when a cycle has been added to the previously selected cycles, increasing the first Betti number of the expanded part by “ p ”, then p created cycles have been formed. As an example, in this method, Steps 4 and 5 will be combined into a single step, and addition of cycle 5 will require immediate formation of the cycle 4. The above method is modified, and an efficient algorithm is developed for the formation of cycle bases by Kaveh and Roosta [26],

Finally, Horton [27] proved that the elements of a minimal cycle basis lie in between a cycle set consisting of the minimal cycles on each member of S which passes through each node of S , i.e. each member is taken in turn and all cycles of minimal length on such a member passing through all the nodes of S are generated. Obviously, $M(S) \times M(S)$ such cycles will be generated.

Independence Control. Each cycle of a graph can be considered as a column vector of its cycle-member incidence matrix. An algebraic method such as the

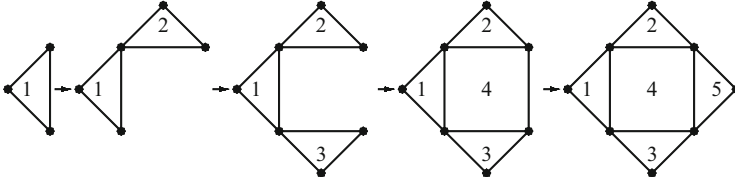


Fig. 2.10 A graph S and selected cycles

Gaussian elimination may then be used for checking the independence of a cycle with respect to the previously selected sub-basis. However, although this method is general and reduces the order dependency of the cycle selection algorithms, like many other algebraic approaches its application requires a considerable amount of storage space.

The most natural graph theoretical approach is to employ a spanning tree of S, and form its fundamental cycles. This method is very simple; however, in general its use leads to long cycles. The method can be improved by allowing the inclusion of each used chord in the branch set of the selected tree. Further reduction in length may be achieved by generating an SRT from a centre node of a graph, and the use of its chords in ascending order of distance from the centre node, Kaveh [21].

A third method, which is also graph-theoretical, consists of using admissible cycles. Consider the following expansion process, with S being a 2-connected graph,

$$C_1 = C^1 \rightarrow C^2 \rightarrow C^3 \rightarrow \dots \rightarrow C^{b_1(S)} = S,$$

where $C^k = \bigcup_{i=1}^k C_i$. A cycle C_{k+1} is called an *admissible* cycle, if for $C^{k+1} = C^k \cup C_{k+1}$:

$$b_1(C^{k+1}) = b_1(C^k \cup C_{k+1}) = b_1(C^k) + 1. \quad (2.60)$$

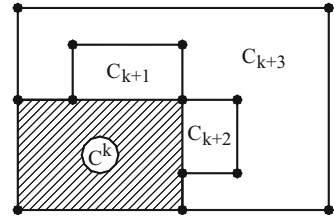
It can easily be proved that, the above admissibility condition is satisfied if any of the following conditions hold:

1. $A_{k+1} = C^k \cap C_{k+1} = \emptyset$, where \emptyset is an empty intersection;
2. $\bar{b}_1(A_{k+1}) = r - s$, where r and s are the numbers of components of C^{k+1} and C^k , respectively;
3. $\bar{b}_1(A_{k+1}) = 0$ when C^k and C^{k+1} are connected ($r = s$).

In the above relations, $\bar{b}_1(A_i) = \bar{M}_i - \bar{N}_i + 1$, where \bar{M}_i and \bar{N}_i are the numbers of members and nodes of A_i , respectively.

As an example, the sequence of cycle selection in Fig. 2.11 will be as specified by their numbers.

Fig. 2.11 A cycle and its bounded cycles



A different approach suggested by Hubicka and SyslØ, in which,

$$b_1(C^{k+1}) = b_1(C^k) + p, \quad (2.61)$$

is considered to be permissible. However, a completion is performed for $p > 1$. As an example, when C_3 is added to C^k , its first Betti number is increased by 3 and therefore, cycles C_1 and C_2 must also be selected at that stage, before further expansion.

Having discussed the mathematical concepts involved in a cycle basis selection, three different algorithms are now described.

Algorithm 1 (Kaveh [15])

- Step 1: Select a pseudo-centre node of maximal degree O . Such a node can be selected manually or automatically using the graph or algebraic graph theoretical methods discussed in Chap. 5.
- Step 2: Generate an SRT rooted at O , form the set of its chords and order them according to their distance from O .
- Step 3: Form one minimal cycle on each chord in turn, starting with the chord nearest to the root node. A corresponding simple path is chosen which contains members of the tree and the previously used chords, hence providing the admissibility of the selected cycle.

This method selects subminimal cycle bases, using the chords of an SRT. The nodes and members of the tree and consequently the cycles are partially ordered according to their distance from O . This is the combinatorial version of the Turn Back method to be discussed in the section on algebraic force methods.

Algorithm 2 (Kaveh [15])

- Step 1: Select a centre or pseudo-centre node of maximal degree O .
- Step 2: Use any member incident with O as the generator of the first minimal cycle. Take any member not used in C_1 and incident with O , and generate on it the second minimal cycle. Continue this process until all the members incident with O are used as the members of the selected cycles. The cycles selected so far are admissible, since the intersection of each cycle with the previously selected cycles is a simple path (or a single node) resulting in an increase of the first Betti number by unity for each cycle.
- Step 3: Choose a starting node O' , adjacent to O , which has the highest degree. Repeat a step similar to Step 2, testing each selected cycle for admissibility.

If the cycle formed on a generator m_k fails the test, then examine the other minimal cycles on m_k if any such cycle exists. If no admissible minimal cycle can be found on m_k , then,

Form admissible minimal cycles on the other members incident with O' . If m_k does not belong to one of these subsequent cycles, then:

Search for an admissible minimal cycle on m_k , since the formation of cycles on other previous members may now have altered the admissibility of this cycle. If no such cycle can be found, leave m_k unused. In this step more than one member may be left unused.

Step 4: Repeat Step 3 using as starting nodes a node adjacent to O and/or O' , having the highest degree. Continue the formation of cycles until all the nodes of S have been tested for cycle selection. If all the members have not been used, select the shortest admissible cycle available for an unused member as generator. Then test the minimal cycles on the other unused members, in case the formation of the longer cycle has altered the admissibility. Each time a minimal cycle is found to be admissible, add to C^i and test all the minimal cycles on the other unused members again. Repeat this process, forming other shortest admissible cycles on unused members as generators, until S is re-formed and a subminimal cycle basis has been obtained.

Both of the above two algorithms are order-dependent, and various starting nodes may alter the result. The following algorithm is more flexible and less order-dependent, and in general leads to the formation of shorter cycle bases.

Algorithm 3 (Kaveh [21])

Step 1: Generate as many admissible cycles of length 3 as possible. Denote the union of the selected cycles by C^n .

Step 2: Select an admissible cycle of length 4 on an unused member. Once such a cycle C_{n+1} is found, check the other unused members for possible admissible cycles of length 3. Again select an admissible cycle of length 4 followed by the formation of possible 3-sided cycles. This process is repeated until no admissible cycles of length 3 and 4 can be formed. Denote the generated cycles by C^m .

Step 3: Select an admissible cycle of length 5 on an unused member. Then check the unused members for the formation of 3-sided admissible cycles. Repeat Step 2 until no cycle of length 3 or 4 can be generated. Repeat Step 3 until no cycle of length 3, 4 or 5 can be found.

Step 4: Repeat similar steps to Step 3, considering higher-length cycles, until $b_1(S)$ admissible cycles forming a subminimal cycle basis are generated.

Remark. The cycle basis C formed by Algorithms 1–3 can further be improved by exchanging the elements of the selected basis. In each step of this process, a shortest cycle C'_i independent of the cycles of $C \setminus C_i$ is replaced by C_i if $L(C'_i) < L(C_i)$. This process is repeated for $i = 1, 2, \dots, b_1(S)$.

This additional operation increases the computational time and storage, and its use is recommended only when the formation of minimal cycle basis is required.

Algorithm 4 (Horton [27])

- Step 1: Find a minimum path $P(n_i, n_j)$ between each pair of nodes n_i and n_j .
 Step 2: For each node n_k and member $m_l = (n_i, n_j)$, generate the cycle having m_l and n_k as $P(n_k, n_i) + P(n_k, n_j) + (n_i, n_j)$ and calculate its length. Degenerate cases in which $P(n_k, n_i)$ and $P(n_k, n_j)$ have nodes other than n_k in common, can be omitted.
 Step 3: Order the cycles by their weight (or length).
 Step 4: Use the Greedy Algorithm, to find a minimal cycle basis from this set of cycles. This algorithm is given in Kaveh [15, 20].

A simplified version of the above Algorithm can be designed as follows:

- Step 1: Form a spanning tree of S rooted from an arbitrary node, and select its chords.
 Step 2: Take the first chord and form $N(S) - 2$ minimal cycles, each being formed on the specified chord containing a node of S (except the two end nodes of this chord).
 Step 3: Repeat Step 2 for the other chords, in turn, until $[M(S) - N(S) + 1] \times [N(S) - 2]$ cycles are generated. Repeated and degenerate cycles should be discarded.
 Step 4: Order the cycles in ascending magnitude of their lengths.
 Step 5: Using the above set of cycles, employ the Greedy Algorithm to form a minimal cycle basis of S .

The main contribution of Horton's Algorithm is the limit imposed on the elements of the cycle-set used in the Greedy Algorithm. The use of matroids and the Greedy Algorithm, has been suggested by Kaveh [15], and they have been employed by Lawler [28] and Kolasinska [29].

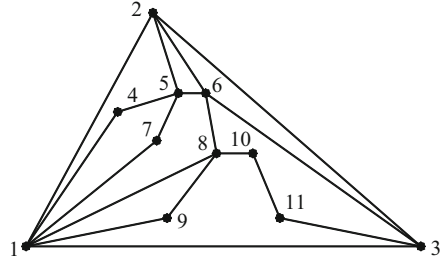
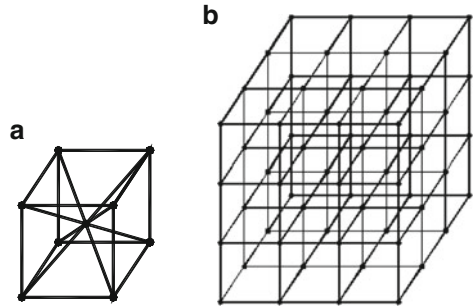
2.4.3 Examples

Example 1. Consider a planar graph S , as shown in Fig. 2.12, for which $b_1(S) = 18 - 11 + 1 = 8$. Using Algorithm 3, the selected basis consists of four cycles of length 3, three cycles of length 4 and one cycle of length 5, as follows:

$$C_1 = (1, 2, 3), C_2 = (1, 8, 9), C_3 = (2, 6, 3), C_4 = (2, 5, 6), C_5 = (1, 4, 5, 2), \\ C_6 = (1, 7, 5, 2), C_7 = (8, 6, 2, 1), C_8 = (10, 8, 6, 3, 11)$$

The total length of the selected basis is $L(C) = 29$, which is a counter example for minimality of a mesh basis, since, for any such basis of S , $L(C) > 29$.

Example 2. In this example, S is the model of a space frame, considered as $S = \bigcup_{i=1}^{27} S_i$, where a typical S_i is depicted in Fig. 2.13a. For S_i there are 12 members joining eight corner nodes, and a central node joined to these corner nodes.

Fig. 2.12 A planar graph S **Fig. 2.13** A space frame S . (a) A typical S_i ($i = 1, \dots, 27$). (b) S with some omitted members

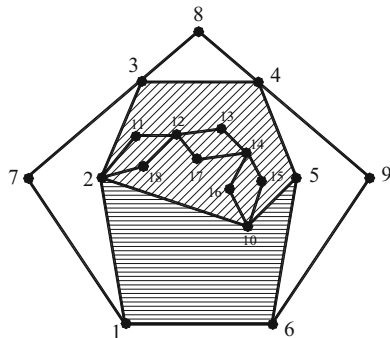
The model S is shown in Fig. 2.13b, in which some of the members are omitted for clarity in the diagram. For this graph, $b_1(S) = 270$.

The selected cycle basis using any of the algorithms consists of 270 cycles of length 3, forming a minimal cycle basis of S . For Algorithm 3, the use of different starting nodes leads to a minimal cycle basis, showing the capability of this method.

Example 3. S is a planar graph with $b_1(S) = 9$, as shown in Fig. 2.14. The application of Algorithm 3 results in the formation of a cycle of length 3 followed by the selection of five cycles of length 4. Then member $\{1, 6\}$ is used as the generator of a six-sided cycle $C_7 = (1, 2, 3, 4, 5, 6, 1)$. Member $\{2, 10\}$ is then employed to form a seven-sided cycle $C_8 = (2, 11, 12, 13, 14, 15, 10, 2)$, followed by the selection of a five-sided cycle $C_9 = (10, 5, 4, 3, 2, 10)$. The selected cycle basis has a total length of $L(C) = 41$, and is not a minimal cycle basis. A shorter cycle basis can be found by Algorithm 4 consisting of one three-sided and five four-sided cycles, together with the following cycles,

$$C_7 = (1, 2, 10, 5, 6, 1), C_8 = (2, 3, 4, 5, 10, 2) \text{ and} \\ C_9 = (2, 11, 12, 13, 14, 15, 10, 2),$$

forming a basis with the total length of 40. However, the computation time and storage for Algorithm 3 is far less than that of Algorithm 4, as compared in Ref. [30].

Fig. 2.14 A planar graph S 

2.4.4 Optimal and Suboptimal Cycle Bases

In what follows, a direct method and an indirect approach, which often lead to the formation of optimal cycle bases, are presented. Much work is needed before the selection of an optimal cycle basis of a graph becomes feasible.

2.4.4.1 Suboptimal Cycle Bases; A Direct Approach

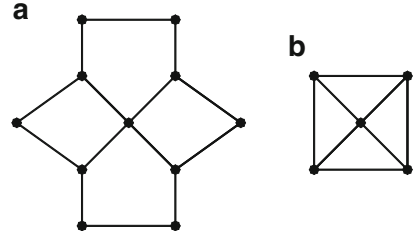
Definition 1. An *elementary contraction* of a graph S is obtained by replacing a path containing all nodes of degree 2 with a new member. A graph S contracted to a graph S' is obtained by a sequence of elementary contractions. Since in each elementary contraction k nodes and k members are reduced, the first Betti number does not change in a contraction, i.e. $b_1(S) = b_1(S')$. The graph S is called homeomorphic to S' , Fig. 2.15.

This operation is performed in order to reduce the size of the graph and also because the number of members in an intersection of two cycles is unimportant; a single member is enough to render $C_i \cap C_j$ nonempty, and hence to produce a non-zero entry in CC^t .

Definition 2. Consider a member m_i of a graph S . On this member, p minimal cycles of length q can be generated. P is called the *incidence number* and q is defined as the *cycle length number* of m_i . In fact, p and q are measures assigned to a member to indicate its potential as a member in the elements of a cycle basis. In the process of expansion for cycle selection, an artificial increase in p results in the exclusion of this element from a minimal cycle, keeping the number of overlaps as small as possible.

Space graphs need special treatment. For these graphs, when a member has $p = 1$, then the next shortest length cycles with $q' = q + 1$ (1 being the next smallest possible integer) are also considered. Denoting the number of such cycles by p' , the incidence number and cycle length number for this type of member are taken as,

Fig. 2.15 S and its contracted graph S' . (a) S. (b) S'



$$I_{jk} = p' + 1 \quad \text{and} \quad I_{jk}^c = (q + p'q') / (1 + p'), \quad (2.62)$$

respectively. The end nodes of the considered member are j and k .

Definition 3. The *weight* of a cycle is defined as the sum of the incidence numbers of its members.

Algorithm A

Step 1: Contract S into S' , and calculate the incidence number (IN) and cycle length number (CLN) of all its members.

Step 2: Start with a member of the least CLN and generate a minimal weight cycle on this member. For members with equal CLNs, the one with the smallest IN should be selected. A member with these two properties will be referred to as “a member of the least CLN with the smallest IN”.

Step 3: On the next unused member of the least CLN with the smallest IN, generate an admissible minimal weight cycle. In the case when a cycle of minimal weight is rejected due to inadmissibility, the next unused member should be considered. This process is continued as far as the generation of admissible minimal weight cycles is possible. After a member has been used as many times as its IN, before each extra usage, increase the IN of such a member by unity.

Step 4: On an unused member of the least CLN, generate one admissible cycle of the smallest weight. This cycle is not a minimal weight cycle, otherwise it would have been selected at Step 3. Such a cycle is called a *subminimal weight cycle*. Again, update the incidence numbers for each extra usage. Now repeat Step 3, since the formation of the new subminimal weight cycle may have altered the admissibility condition of the other cycles, and selection of further minimal weight cycles may now have become possible.

Step 5: Repeat Step 4, selecting admissible minimal and subminimal weight cycles, until $b_1(S')$ of these cycles are generated.

Step 6: A reverse process to that of the contraction of Step 1, transforms the selected cycles of S' into those of S .

This algorithm leads to the formation of a suboptimal cycle basis, and for many models encountered in practice, the selected bases have been optimal.

2.4.4.2 Suboptimal Cycle Bases; an Indirect Approach

Definition 1. The *weight* of a member in the following algorithm is taken as the sum of the degrees of its end nodes.

Algorithm B

Step 1: Order the members of S in ascending order of weight. In all the subsequent steps use this ordered member set.

Step 2: Generate as many admissible cycles of length α as possible, where α is the length of the shortest cycle of S . Denote the union of the selected cycles by C^m . When α is not specified, use the value $\alpha = 3$.

Step 3: Select an admissible cycle of length $\alpha+1$ on an unused member (use the ordered member set). Once such a cycle C_{m+1} is found, control the other unused members for possible admissible cycles of length α . Again select an admissible cycle of length $\alpha+1$ followed by the formation of possible α -sided cycles. This process is repeated until no admissible cycles of length α and $\alpha+1$ can be found. Denote the generated cycles by C^n .

Step 4: Select an admissible cycle C_{n+1} of length $\alpha+2$ on an unused member. Then check the unused members for the formation of α -sided cycles. Repeat Step 2 until no cycle of length α or $\alpha+1$ can be generated. Repeat Step 3 until no cycles of length α , $\alpha+1$ or $\alpha+2$ can be found.

Step 5: Take an unused member and generate an admissible cycle of minimal length on this member. Repeat Steps 1, 2 and 3.

Step 6: Repeat steps similar to that of Step 4 until $b_1(S)$ admissible cycles, forming a suboptimal cycle basis, are generated.

Using the ordered member set affects the selection process in two ways:

1. Generators are selected in ascending weight order, hence increasing the possibility of forming cycles from the dense part of the graph. This increases the chance of cycles with smaller overlaps being selected.
2. From cycles of equal length formed on a generator, the one with smallest total weight (sum of the weights of the members of a cycle) is selected.

The cycle bases generated by this algorithm are suboptimal; however, the results are inferior to those of the direct method A.

Remark. Once a cycle basis C is formed by Algorithm A or Algorithm B, it can further be improved by exchanging the elements of C . In each step of this process, a cycle C_k is controlled for the possibility of being exchanged by ring sum of C_k and a combination of the cycles of $C \setminus C_k$, in order to reduce the overlap of the cycles. The process is repeated until no improvement can be achieved. This additional operation increases the computational time and storage, and should only be used when the corresponding effort is justifiable, e.g. this may be the case when a non-linear analysis or a design optimisation is performed using a fixed cycle basis.

2.4.5 Examples

In this section, examples of planar and space frames are studied. The cycle bases selected by Algorithms A and B are compared with those developed for generating minimal cycle bases (Algorithms 1–4). Simple examples are chosen, in order to illustrate clearly the process of the methods presented. The models, however, can be extended to those containing a greater number of members and nodes of high degree, to show the considerable improvements to the sparsity of matrix \mathbf{CC}^t .

Example 1. Consider a space frame as shown in Fig. 2.16a with the corresponding graph model S as illustrated in Fig. 2.16b. For this graph $b_1(S) = 12$, and therefore 12 independent cycles should be selected as a basis. Algorithm B selects a minimal cycle basis containing the following cycles,

$$\begin{aligned} C_1 &= (1, 2, 3), C_2 = (1, 2, 5), C_3 = (1, 3, 4), C_4 = (1, 5, 4), C_5 = (2, 3, 6, 7), \\ C_6 &= (3, 4, 7, 8), C_7 = (4, 5, 8, 9), C_8 = (6, 7, 8, 9), C_9 = (7, 8, 11, 12), \\ C_{10} &= (6, 7, 10, 11), C_{12} = (9, 8, 12, 13), C_{12} = (10, 11, 12, 13) \end{aligned}$$

which corresponds to:

$$\chi(\mathbf{C}) = 4 \times 3 + 8 \times 4 = 44,$$

and

$$\chi(\mathbf{CC}^t) = 12 + 2 \times 23 = 58.$$

Using Algorithm A leads to the formation of a similar basis, with the difference that $C'_8 = (6, 9, 10, 11)$ is generated in place of $C_8 = (6, 7, 8, 9)$, corresponding to:

$$\begin{aligned} \chi(\mathbf{C}') &= 4 \times 3 + 8 \times 4 = 44, \\ \chi(\mathbf{C}'\mathbf{C}'^t) &= 12 + 2 \times 20 = 52. \end{aligned}$$

The CLNs and Ins of the members used in this algorithm are illustrated in Fig. 2.16b.

Example 2. In this example, S is a space structure with $b_1(S) = 33$, as shown in Fig. 2.17a. Both Algorithms 3 and A select 33 cycles of length 4, i.e. a minimal cycle basis with $\chi(\mathbf{C}) = 4 \times 33 = 132$ is obtained.

The basis selected by Algorithm 3 contains (in the worst case) all four-sided cycles of S except those which are shaded in Fig. 2.17a, with $\chi(\mathbf{CC}^t) = 233$.

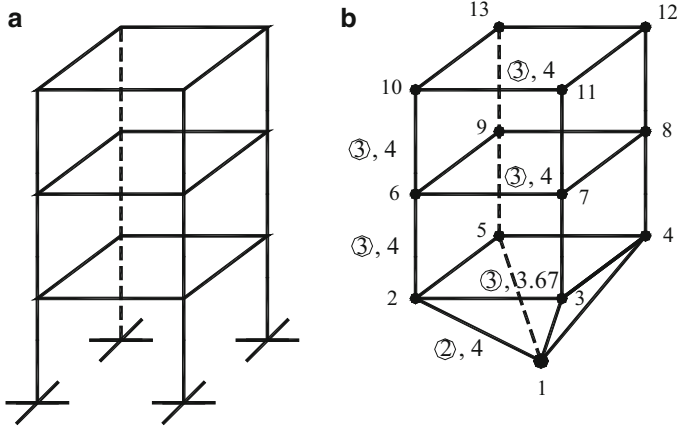


Fig. 2.16 A space frame, and CLNs and Ins of its members. (a) A space structure. (b) The graph model S of the structure

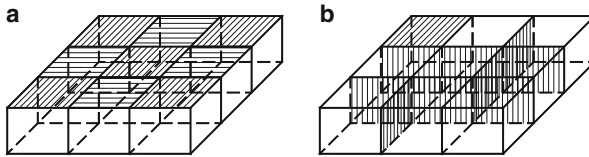


Fig. 2.17 Minimal and suboptimal cycle bases of S. (a) A minimal cycle basis. (b) A suboptimal cycle basis

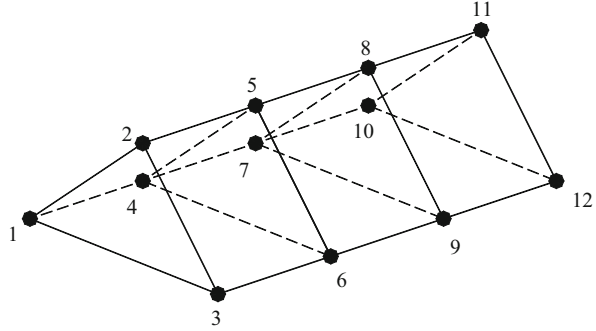
Algorithm A selects all three-sided cycles of S except those shaded in Fig. 2.17b, with $\chi(CC^t) = 190$. It will be noticed that, for structures containing nodes of higher degrees, considerable improvement is obtained by the use of Algorithm A.

Example 3. Consider a space frame as shown in Fig. 2.18, for which $b_1(S) = 10$. The minimal cycle basis selected by Algorithm 3 consists of the following cycles,

$$\begin{aligned} C_1 &= (1, 2, 3), C_2 = (4, 5, 6), C_3 = (7, 8, 9), C_4 = (10, 11, 12), \\ C_5 &= (1, 2, 5, 4), C_6 = (2, 3, 6, 5), C_7 = (4, 5, 8, 7), C_8 = (5, 6, 9, 8), \\ C_9 &= (7, 8, 11, 10), C_{10} = (8, 9, 12, 11), \end{aligned}$$

corresponding to $\chi(C) = 4 \times 3 + 6 \times 4 = 36$ and $\chi(CC^t) = 10 + 2 [0 + 0 + 0 + 2 + 3 + 3 + 4 + 3 + 4] = 10 + 2 \times 19 = 48$.

However, the following non-minimal cycle basis has a higher $\chi(C)$, and leads to a more sparse CC^t matrix. The selected cycles are as follows,

Fig. 2.18 A space frame S

$$\begin{aligned}
 C_1 &= (1, 2, 3), C_2 = (1, 2, 5, 4), C_3 = (2, 3, 6, 5), C_4 = (1, 3, 6, 4), \\
 C_5 &= (4, 5, 8, 7), C_6 = (5, 6, 9, 8), C_7 = (4, 6, 9, 7), C_8 = (7, 8, 11, 10), \\
 C_9 &= (8, 9, 12, 11), C_{10} = (10, 11, 12),
 \end{aligned}$$

for which $\chi(C') = 2 \times 3 + 8 \times 4 = 38$ corresponding to $\chi(C'C'^t) = 10 + 2 [1 + 2 + 3 + 1 + 2 + 3 + 1 + 2 + 2] = 10 + 2 \times 16 = 42$.

Therefore, the idea of having an optimal cycle basis in between minimal cycle bases is incorrect.

Example 4. Consider the skeleton of a structure S, comprising of six flipped flags, as shown in Fig. 2.19a, for which $b_1(S) = 6$. After contraction, S' is obtained as illustrated in Fig. 2.19b. Obviously, this is a planar graph. The CLNs for the members are 3 and IN for member (1, 2) is 6 and for the remaining members it is equal to 1., Algorithm 3 selects a minimal cycle basis for S' , consists of six 3-sided cycles, corresponding to:

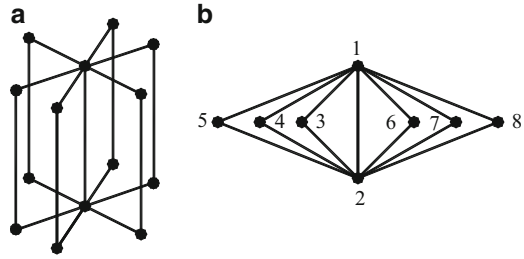
$$\chi(C) = 6 \times 3 = 18 \text{ and } \chi(CC^t) = 6 + 2[0 + 1 + 2 + 3 + 4 + 5] = 6 + 2 \times 15 = 36$$

However, the following non-minimal cycle basis has a higher $\chi(S')$, and leads to a lower sparsity, $\chi(C'C'^t)$:

$$\begin{aligned}
 C_1 &= (1, 3, 2, 4), C_2 = (1, 4, 2, 5), C_3 = (1, 2, 3), C_4 = (1, 2, 6), \\
 C_5 &= (1, 6, 2, 7), C_6 = (1, 7, 2, 8).
 \end{aligned}$$

For this basis, $\chi(C') = 4 \times 4 + 2 \times 3 = 22$, corresponding to $\chi(C'C'^t) = 6 + 2 [0 + 1 + 1 + 1 + 1 + 1 + 1] = 6 + 2 \times 5 = 16$. After the back transformation from S' to S, we have $\chi(C) = 4 \times 6 + 2 \times 4 = 32$, corresponding to $\chi(CC') = 6 + 2 [0 + 1 + 1 + 1 + 1 + 1 + 1] = 16$.

Fig. 2.19 A flipped flag before and after contraction.
(a) S. (b) S'

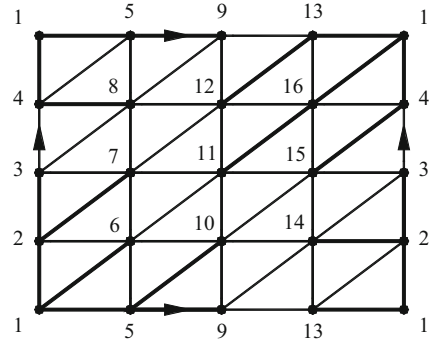


2.4.6 An Improved Turn Back Method for the Formation of Cycle Bases

In this section, the combinatorial Turn-back method of Kaveh [15] is improved to obtain shorter cycle bases. This method covers all the counter examples, known for the minimality of the selected cycle bases.

- Step 1: Generate an SRT rooted from an arbitrary node O. Identify its chords, and order them according to their distance numbers from O.
- Step 2: Select the shortest length cycle of the graph on a chord and add this chord (generator) to the tree members. Repeat this process to all the chords, forming cycles of the least length containing the tree members and the previously used chords only. The selected cycles are all admissible, i.e. the addition of each cycle increases the first Betti number of the expanded part of the graph by unity. Store these cycles in C.
- Step 3: Form all the new cycles of the same length on the remaining chords, allowing the use of more than one unused chords in their formation.
- Step 4: Control the cycles formed in Step 3 to find only one cycle having a generator, which is in none of the other connected cycles formed in Step 3. When such a chord is found, add the corresponding cycle to C and include its generator in the tree members. Repeat this control until no such a cycle can be found.
- Step 5: Select a cycle of the next higher length in the graph containing only one chord. Add the selected cycle to C and its generator to the tree members.
- Step 6: Control the cycles formed in Step 3 to find a cycle containing only one unused chord. Add such a cycle to C and add its chord to the tree members. Repeat this control until no cycle of this property can be found.
- Step 7: Repeat Step 4.
- Step 8: Repeat Steps 5 and 6 and continue this repetition with the same length until no cycle in Step 5 can be found.
- Step 9: Repeat Steps 3 to 8, until $b_1(S)$ cycles forming a cycle basis is included in C.

Fig. 2.20 Graph S and the selected SRT



2.4.7 Examples

Example 1. A graph is considered in the form of the 1-skeleton of a torus-type structure, Fig. 2.20. An SRT is selected, as shown in bold lines. The cycles selected in Step 2 are given in the following:

$$C = \{(1, 2, 6), (1, 4, 5), (1, 5, 6), (1, 2, 13), (1, 4, 16), (1, 13, 16), (2, 3, 7), (2, 6, 7), (2, 3, 14), (2, 13, 14), (4, 5, 8), (4, 15, 16), (5, 6, 10), (5, 9, 10), (5, 8, 9), (12, 13, 16), (11, 12, 16), (11, 15, 16)\}.$$

The execution of Step 3 results in the following cycles:

$$(3, 7, 8), (3, 4, 8), (7, 11, 12), (7, 8, 12), (8, 9, 12), (9, 13, 14), (9, 10, 14), (10, 14, 15), (10, 11, 15), (9, 12, 13), (3, 14, 15), (3, 4, 15).$$

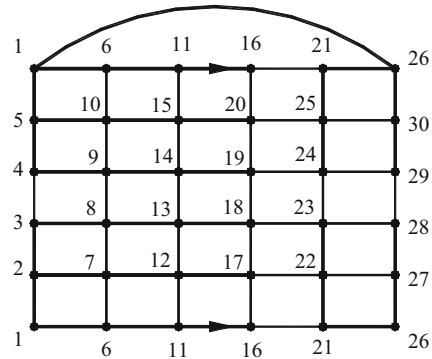
Twelve cycles are generated, increasing the first Betti number by twelve. The control of Step 4, leads to generators $\{10, 11\}$ and $\{7, 11\}$ corresponding to the cycles $(10, 11, 15)$ and $(7, 11, 12)$, respectively. Thus no cycle is selected.

In Step 5, a cycle of length 4 containing an unused chord is formed. On $\{3, 4\}$, cycle $(1, 2, 3, 4)$ is generated and added to C . Then in Step 6, the following cycles are added to C :

$$(3, 4, 8) \text{ for } \{3, 8\}, (3, 7, 8) \text{ for } \{7, 8\}, (3, 4, 15) \text{ for } \{3, 15\}, (3, 14, 15) \text{ for } \{14, 15\}.$$

In Step 7 no cycle is found, but in Step 8, the execution of Step 5 leads to cycle $(1, 5, 9, 13)$ on $\{9, 13\}$, and Step 6 leads to the following cycles completing C , and forming a minimal cycle basis of S :

Fig. 2.21 A space graph and the selected SRT



(9, 12, 13) for $\{9, 12\}$, (9, 13, 14) for $\{9, 14\}$, (8, 9, 12) for $\{8, 12\}$, (7, 8, 12) for $\{7, 12\}$, (7, 11, 12) for $\{7, 11\}$, (9, 10, 14) for $\{10, 14\}$, (10, 14, 15) for $\{10, 15\}$, and (10, 11, 15) for $\{10, 11\}$.

Example 2. A space graph is considered as illustrated in Fig. 2.21. An SRT is selected as shown in bold lines. The application of Step 2, leads to the following cycle set:

$$C = \{(1, 2, 6, 7), (1, 5, 6, 10), (2, 3, 7, 8), (4, 5, 9, 10), (6, 7, 11, 12), (6, 10, 11, 15), \\ (7, 8, 12, 13), (9, 10, 14, 15), (11, 12, 16, 17), (11, 15, 16, 20), (12, 13, 17, 18), \\ (14, 15, 19, 20), (21, 22, 26, 27), (21, 25, 26, 30), (22, 23, 27, 28), (24, 25, 29, 30)\}.$$

In Step 3, the following cycles are generated:

$$(3, 4, 8, 9), (8, 9, 13, 14), (13, 14, 18, 19), (16, 17, 21, 22), (17, 18, 22, 23),$$
$$(18, 19, 22, 23), (18, 19, 23, 24), (19, 20, 24, 25), (16, 20, 21, 25), (23, 24, 28, 29).$$

These cycles contain 11 unused chords. The control of Step 4 shows that $\{3, 4\}$ and $\{28, 29\}$ are included in one cycle, and therefore all the chords remain unused. In the next step, a cycle of length 5 including an unused chord is generated and added to C. Only with chord $\{3, 4\}$, the 5-sided cycle $(1, 2, 3, 4, 5)$ is generated, and in Step 6 the following three-sided cycles are selected:

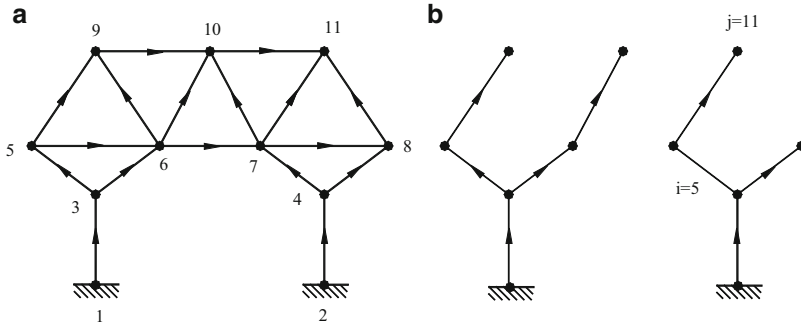


Fig. 2.22 S and two of its SR subtrees

$$(3, 4, 8, 9), (8, 9, 13, 14), \text{ and } (13, 14, 18, 19).$$

Step 7 is carried out and cycle (23, 24, 28, 29) on {28, 29} is found, repetition of this control leads to cycle (18, 19, 23, 24) on {23, 24}. In the next step, no cycle is selected. The execution of Steps 3 and 4 in Step 9 results in no cycle.

The execution of Step 5 in Step 9, forms cycle (1, 6, 11, 16, 21, 26) on chord {16, 21}, and the execution of Step 6 leads to the following cycles,

$$(16, 20, 21, 25) \text{ for } \{20, 25\}, (19, 20, 24, 25) \text{ for } \{19, 24\}, (16, 17, 21, 22) \text{ for } \{17, 22\}, \text{ and } (17, 18, 22, 23) \text{ for } \{18, 23\}.$$

The selected cycles form a minimal cycle basis.

2.4.8 Formation of B_0 and B_1 Matrices

In order to generate the elements of a B_0 matrix, a basic structure of S should be selected. For this purpose a spanning forest consisting of NG(S) SRTs is used, where NG(S) is the number of ground (support) nodes of S. As an example, for S shown in Fig. 2.22a, two SR subtrees are generated, Fig. 2.22b.

The orientation assigned to each member of S is from the lower numbered node to its higher numbered end. For each SR subtree, the orientation is given in the direction of its growth from its support node.

MATRIX B_0 : This is a $6M(S) \times 6NL(S)$ matrix, where M(S) and NL(S) are the numbers of members and loaded nodes of S, respectively. If all the free nodes are loaded, then

$$NL(S) = N(S) - NG(S),$$

where NG(S) is the number of support nodes.

For a member, the internal forces are represented by the components at the lower numbered end. Obviously the components at the other end can be obtained by considering the equilibrium of the member.

The coefficients of \mathbf{B}_0 can be obtained by considering the transformation of each joint load to the ground node of the corresponding subtree. $[\mathbf{B}_0]_{ij}$ for member i and node j is given by a 6×6 submatrix as,

$$[\mathbf{B}_0]_{ij} = \alpha_{ij} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -\Delta z & \Delta y & 1 & 0 & 0 \\ \Delta z & 0 & -\Delta x & 0 & 1 & 0 \\ -\Delta y & \Delta x & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.63)$$

in which Δx , Δy and Δz are the differences of the coordinates of node j with respect to the lower numbered end of member i , in the selected global coordinate system, and α_{ij} is the orientation coefficient defined as:

$$\alpha_{ij} = \begin{cases} +1 & \text{if member is positively oriented in the tree containing } j, \\ -1 & \text{if member is negatively oriented in the tree containing } j, \\ 0 & \text{if member is not in the tree containing node } j. \end{cases}$$

The \mathbf{B}_0 matrix can be obtained by assembling the $[\mathbf{B}_0]_{ij}$ submatrices as shown schematically in the following:

$$\mathbf{B}_0 = \begin{bmatrix} & & \\ & & j \\ i & [B_0]_{ij} & \\ & & \end{bmatrix}_{6M(S) \times 6NL(S)} \quad (2.64)$$

MATRIX \mathbf{B}_j : This is a $6m(S) \times 6b_j(S)$ matrix, which can be formed using the elements of a selected cycle basis. For a space structure, six self-equilibrating stress systems can be formed on each cycle. Consider C_j and take a member of this cycle as its generator. Cut the generator in the neighbourhood of its beginning node and apply six bi-actions as illustrated in Fig. 2.23.

The internal forces under the application of each bi-action are a self-equilibrating stress system. As for the matrix \mathbf{B}_0 , a submatrix $[\mathbf{B}_1]_{ij}$ of \mathbf{B}_1 is a 6×6 submatrix, the columns of which show the internal forces at the lower numbered end of member i under the application of six bi-actions at the cut of the generator j .

Fig. 2.24 A four by four planar frame S

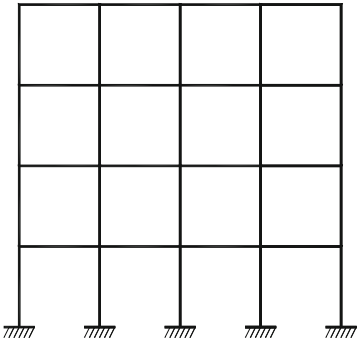


Fig. 2.25 Patterns of \mathbf{B}_1 and $\mathbf{B}_1^t \mathbf{B}_1$ matrices for S. (a) Pattern of \mathbf{B}_1 . (b) Pattern of $\mathbf{B}_1^t \mathbf{B}_1$

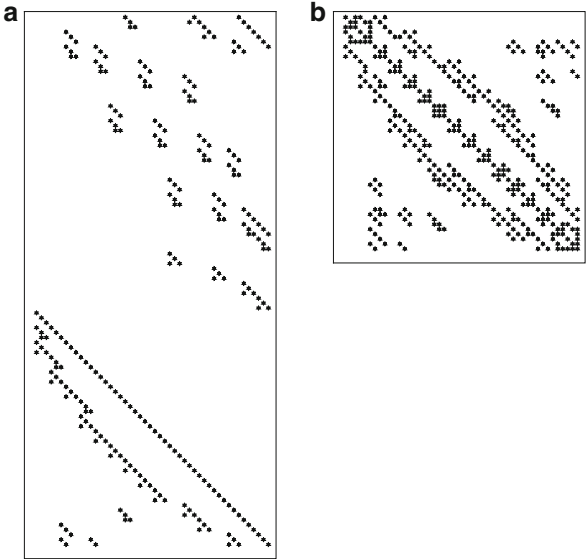


Fig. 2.26 A simple space frame S

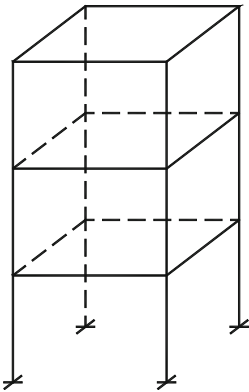
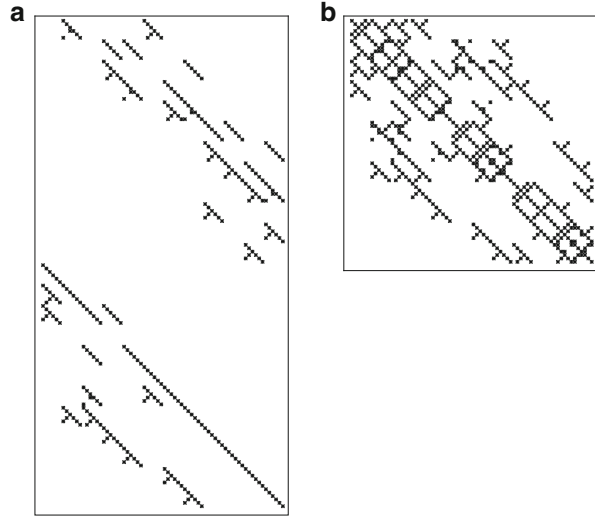


Fig. 2.27 Patterns of \mathbf{B}_1 and $\mathbf{B}_1^t \mathbf{B}_1$ matrices for S. (a) Pattern of \mathbf{B}_1 . (b) Pattern of $\mathbf{B}_1^t \mathbf{B}_1$



Example 1. A four by four planar frame is considered as shown in Fig. 2.24.

The patterns of \mathbf{B}_1 and $\mathbf{B}_1^t \mathbf{B}_1$ formed on the elements of the cycle basis selected by any of the methods of the previous section are depicted in Fig. 2.25, corresponding to $\chi(\mathbf{B}_1) = 241$ and $\chi(\mathbf{B}_1^t \mathbf{B}_1) = 388$.

Example 2. A one-bay three-storey frame is considered as shown in Fig. 2.26.

The patterns of \mathbf{B}_1 and $\mathbf{B}_1^t \mathbf{B}_1$ matrices formed on the elements of the cycle basis selected by any of the graph theoretical algorithms of the previous Section are shown in Fig. 2.27, corresponding to $\chi(\mathbf{B}_1) = 310$ and $\chi(\mathbf{B}_1^t \mathbf{B}_1) = 562$.

Once \mathbf{B}_0 and \mathbf{B}_1 are computed, the remaining steps of the analysis are the same as those presented in Sect. 2.3.6. The interested reader may also refer to standard textbooks such as those of McGuire and Gallagher [31], Przemieniecki [32], or Pestel and Leckie [33] for further information.

2.5 Generalized Cycle Bases of a Graph

In this section, S is considered to be a connected graph. For $\gamma(S) = aM(S) + bN(S) + c\gamma_0(S)$, the coefficients b and c are assumed to be integer multiples of the coefficient $a > 0$. Only those coefficients given in Table 2.1 are of interest.

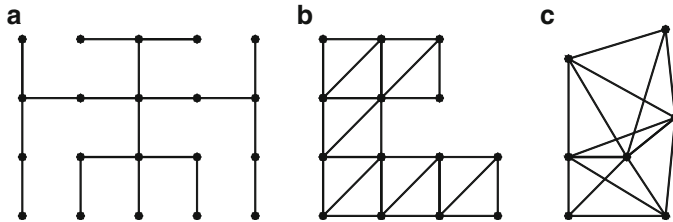


Fig. 2.28 Examples of γ -trees (a) $\gamma(S) = 3M - 3N + 3$. (b) $\gamma(S) = M - 2N + 3$. (c) $\gamma(S) = M - 3N + 6$

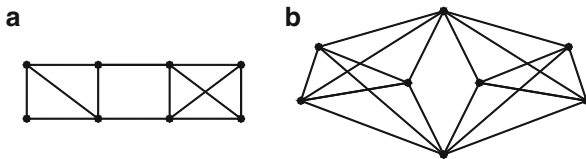


Fig. 2.29 Structures satisfying $\gamma(T) = 0$ which are not rigid. (a) $\gamma(S) = M - 2N + 3$. (b) $\gamma(S) = M - 3N + 6$

2.5.1 Definitions

Definition 1. A subgraph S_i is called an *elementary subgraph* if it does not contain a subgraph $S'_i \subseteq S_i$ with $\gamma(S'_i) > 0$. A connected rigid subgraph T of S containing all the nodes of S is called a γ -tree if $\gamma(T) = 0$. For $\gamma(S_i) = b_1(S_i)$, a γ -tree becomes a tree in graph theory.

Obviously a structure whose model is a γ -tree is statically determinate when $\gamma(S)$ describes the degree of static indeterminacy of the structure. The ensuing stress resultants can uniquely be determined everywhere in the structure by equilibrium only. Examples of γ -trees are shown in Fig. 2.28.

Notice that $\gamma(T) = 0$ does not guarantee the rigidity of a γ -tree. For example, the graphs models depicted in Fig. 2.29 both satisfy $\gamma(T) = 0$; however, neither is rigid.

Definition 2. A member of $S - T$ is called a γ -chord of T . The collection of all γ -chords of a γ -tree is called the γ -cotree of S .

Definition 3. A *removable subgraph* S_j of a graph S_i , is the elementary subgraph for which $\gamma(S_i - S_j) = \gamma(S_i)$, i.e. the removal of S_j from S_i does not alter its DSI. A γ -tree of S containing two chosen nodes, which has no removable subgraph is called a γ -path between these two nodes.

As an example, the graphs shown in Fig. 2.30 are γ -paths between the specified nodes n_s and n_t .

Definition 4. A connected rigid subgraph of S with $\gamma(C_k) = a$, which has no removable subgraph is termed a γ -cycle of S . The total number of members of

Definition 5. Let m_i be a γ -chord of T . Then $T \cup m_i$ contains a γ -cycle C_i which is defined as a *fundamental γ -cycle* of S with respect to T . Using the Intersection Theorem of Sect. 2.2.2, it can easily be shown that,

$$\gamma(T \cup m_i) = 0 + (a + 2b + c) - (2b + c) = a,$$

indicating the existence of a γ -cycle in $T \cup m_i$. For a rigid T , the corresponding fundamental γ -cycle is also rigid, since the addition of an extra member between the existing nodes of a graph cannot destroy the rigidity. A fundamental γ -cycle can be obtained by omitting all the removable subgraphs of $T \cup m_i$.

Definition 6. A maximal set of independent γ -cycles of S is defined as a *generalized cycle basis* (GCB) of S . A maximal set of independent fundamental γ -cycles is termed a *fundamental generalized cycle basis* of S . The dimension of such a basis is given by $\eta(S) = \gamma(S)/a$.

As an example, a generalized cycle basis of a planar truss is illustrated in Fig. 2.32.

Definition 7. A *generalized cycle basis-member incidence matrix* \mathbf{C} is an $\eta(S) \times M$ matrix with entries $-1, 0$ and $+1$, where $c_{ij} = 1$ (or -1) if γ -cycle C_i contains positively (or negatively) oriented member m_j , and $c_{ij} = 0$ otherwise. The *generalized cycle adjacency matrix* is defined as \mathbf{D} which is an $\eta(S) \times \eta(S)$ matrix when undirected γ -cycles are considered; then the negative entries of \mathbf{C} become positive.

2.5.2 Minimal and Optimal Generalized Cycle Bases

A generalized cycle basis $\mathbf{C} = \{C_1, C_2, \dots, C_{\eta(S)}\}$ is called *minimal* if it corresponds to a minimum value of:

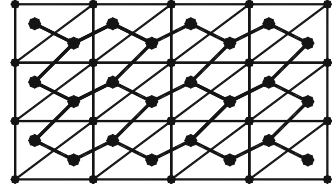
$$L(\mathbf{C}) = \sum_{i=1}^{\eta(S)} L(C_i). \quad (2.67)$$

Obviously, $\chi(\mathbf{C}) = L(\mathbf{C})$ and a minimal GCB can be defined as a basis which corresponds to minimum $\chi(\mathbf{C})$. A GCB for which $L(\mathbf{C})$ is near minimum is called a *subminimal* GCB of S .

A GCB corresponding to maximal sparsity of the GCB adjacency matrix is called an *optimal* generalized cycle basis of S . If $\chi(\mathbf{C}\mathbf{C}^t)$ does not differ considerably from its minimum value, then the corresponding basis is termed *suboptimal*.

The matrix intersection coefficient $\sigma_i(\mathbf{C})$ of row i of GCB incidence matrix \mathbf{C} is the number of row j such that:

Fig. 2.33 A planar truss S and its associate graph $A(S)$



- (a) $j \in \{i + 1, i + 2, \dots, \eta(S)\}$,
- (b) $C_i \cap C_j \neq \emptyset$, i.e. there is at least one k such that the column k of both γ -cycles C_i and C_j (rows i and j) contain non-zero entries.

Now it can be shown that:

$$\chi(CC^t) = \eta(S) + 2 \sum_{i=1}^{\eta(S)-1} \sigma_j(C). \quad (2.68)$$

This relationship shows the correspondence of a GCB incidence matrix C and that of its GCB adjacency matrix. In order to minimize $\chi(CC^t)$, the value of $\sum_{i=1}^{\eta(S)-1} \sigma_j(C)$ should be minimized, since $\eta(S)$ is a constant for a given structure S , i.e. γ -cycles with a minimum number of overlaps should be selected.

2.6 Force Method for the Analysis of Pin-Jointed Planar Trusses

The methods described in Sect. 2.5 are applicable to the selection of generalized cycle bases for different types of skeletal structures. However, the use of these algorithms for trusses engenders some problems, which are discussed in Ref. [34]. In this section, two methods are developed for selecting suitable GCBs for planar trusses. In both methods, special graphs are constructed for the original graph model S of a truss, containing all the connectivity properties required for selecting a suboptimal GCB of S .

2.6.1 Associate Graphs for Selection of a Suboptimal GCB

Let S be the model of a planar truss with triangulated panels, as shown in Fig. 2.33. The associate graph of S , denoted by $A(S)$, is a graph whose nodes are in a one-to-one correspondence with triangular panels of S , and two nodes of $A(S)$ are connected by a member if the corresponding panels have a common member in S .

Fig. 2.34 S with two cut-outs and its A(S)

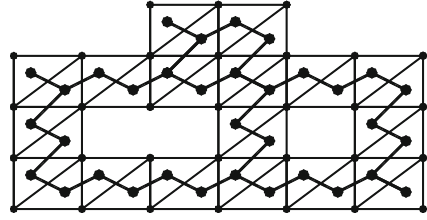
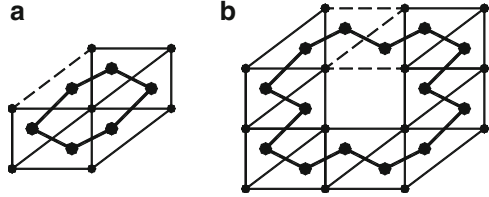


Fig. 2.35 Two different types of cycles. (a) A type C_I cycle. (b) A type C_{III} cycle



If S has some cut-outs, as shown in Fig. 2.34, then its associate graph can still be formed, provided that each cut-out is surrounded by triangulated panels.

For trusses containing adjacent cut-outs, a cut-out with cut-nodes in its boundary, or any other form violating the above-mentioned condition, extra members can be added to S. The effect of such members should then be included in the process of generating its self-equilibrating stress systems.

Theorem A. For a fully triangulated truss (except for the exterior boundary), as in Fig. 2.33, the dimension of a statical basis $\gamma(S)$ is equal to the number of its internal nodes, which is the same as the first Betti number of its associate graph, i.e.

$$\gamma(S) = N_i(S) = b_1[A(S)]. \quad (2.69)$$

Proof. Let M' and N' be the numbers of members and nodes of A(S), respectively. By definition,

$$N' = R(S) - 1,$$

and $M' = M_i(S) = M(S) - M_e(S) = M(S) - N_e(S) = M(S) - [N(S) - N_i(S)]$.

Thus: $b_1[A(S)] = M' - N' + 1 = M(S) - [N(S) - N_i(S)] - R(S) + 1 + 1 = 2 - R(S) + M(S) - N(S) + N_i(S)$.

By Euler's polyhedron formula, we have:

$$2 - R(S) + M(S) - N(S) = 0.$$

Therefore:

For trusses which are not fully triangulated, we have:

$$\gamma(S) = N_i(S) - M_c(S).$$

A Cycle of A(S) and the Corresponding γ -Cycle of S. In Fig. 2.35a, a triangulated truss and its associate graph, which is a cycle, are shown for which

$$\gamma(S_i) = N_i = 1 = b_1[A(S)].$$

Since C_1 of A(S) corresponds to one γ -cycle of S, it is called a *type I cycle*, denoted by C_I . A γ -cycle of S is shown by continuous lines, and its γ -chords are depicted in dashed lines.

Figure 2.35b shows a truss unit with one cut-out. In general, if a cut-out is an m -gon, then the completion of the triangulation requires $m-3$ members. Instead, m internal nodes will be created, increasing the DSI by m . Hence Eq. 2.68 yields,

$$\gamma(S) = m - (m - 3) = 3,$$

while: $b_1[A(S)] = 1$.

However, in this case S contains three γ -cycles. A γ -path P and three γ -chords (dashed lines) are depicted in Fig. 2.35b. Obviously $P \cup m_i$ ($i = 1, 2, 3$) form three γ -cycles which correspond to a cycle of type C_{III} of A(S). Thus two types of cycles C_I and C_{III} should be recognized in A(S) and an appropriate number of γ -cycles will then be generated.

Algorithm AA

Step 1: Construct the associate graph A(S) of S.

Step 2: Select a mesh basis of A(S), using an appropriate cycle selection algorithm.

For fully triangulated S, Algorithms 1–3 generate cycle bases with three-sided elements.

Step 3: Select the γ -cycles of S corresponding to the cycles of A(S). One γ -cycle for each cycle of type C_I , and three γ -cycles for each cycle of type C_{III} should be chosen.

Once a GCB is selected, on each γ -cycle one self-equilibrating stress system can easily be formed. Therefore, a statical basis with localized self-equilibrating stress systems will be obtained.

Example. Let S be the graph model of a planar truss, as shown in Fig. 2.34, for which $\gamma(S) = 12$. For A(S), six cycles of length 6 of type C_I and two cycles of lengths 18 and 26 of type C_{III} are selected. Therefore, the total of $6 + 3 \times 2 = 12$ γ -cycles of S are obtained. On each γ -cycle one self-equilibrating stress system is constructed, and a statical basis consisting of localized self-equilibrating stress systems is thus obtained.

2.6.2 Minimal GCB of a Graph

Theoretically a minimal GCB of a graph can be found using the Greedy Algorithm developed for matroids. This will be discussed in Kaveh [15, 20] after matroids have been introduced, and here only the algorithm is briefly outlined.

Consider the graph model of a structure, and select all of its γ -cycles. Order the selected γ -cycles in ascending order of length. Denote these cycles by a set C . Then perform the following steps:

Step 1: Choose a γ -cycle C_1 of the smallest length, i.e. $L(C_1) < L(C_i)$ for all $C_i \in C$

Step 2: Select the second γ -cycle C_2 from $C - \{C_1\}$ which is independent of C_1 and $L(C_2) \leq L(C_i)$ for all γ -cycles of $C - \{C_1\}$.

Step k: Subsequently choose a γ -cycle C_k from $C - \{C_1, C_2, \dots, C_{k-1}\}$ which is independent of C_1, C_2, \dots, C_{k-1} and $L(C_k) \leq L(C_i)$ for all $C_i \in C - \{C_1, C_2, \dots, C_{k-1}\}$.

After $\eta(S)$ steps, a minimal GCB will be selected by this process, a proof of which can be found in Kaveh [19].

2.6.3 Selection of a Subminimal GCB: Practical Methods

In practice, three main difficulties are encountered in an efficient implementation of the Greedy Algorithm. These difficulties are briefly mentioned in the following:

1. Selection of some of the γ -cycles for some $\gamma(S)$ functions.
2. Formation of all of the γ -cycles of S .
3. Checking the independence of γ -cycles.

In order to overcome the above difficulties, various methods are developed. The bases selected by these approaches correspond to very sparse GCB adjacency matrices, although these bases are not always minimal.

Method 1. This is a natural generalization of the method for finding a fundamental cycle basis of a graph, and consists of the following steps:

Step 1: Select an arbitrary γ -tree of S , and find its γ -chords.

Step 2: Add one γ -chord at a time to the selected γ -tree to form fundamental γ -cycles of S with respect to the selected γ -tree.

The main advantage of this method is the fact that the independence of γ -cycles is guaranteed by using a γ -tree. However, the selected γ -cycles are often quite long, corresponding to highly populated CCB adjacency matrices.

Method 2. This is an improved version of Method 1, in which a special γ -tree has been employed and each γ -chord is added to γ -tree members after being used for formation of a fundamental γ -cycle.

- Step 1: Select the centre “O” of the given graph. Methods for selecting such a node will be discussed in Chap. 5.
- Step 2: Generate a shortest route γ -tree rooted at the selected node O, and order its γ -chords according to their distance from O. The *distance* of a member is taken as the sum of the shortest paths between its end nodes and O.
- Step 3: Form a γ -cycle on the γ -chord of the smallest distance number, and add the used γ -chord to the tree members, i.e. form $T \cup m_1$.
- Step 4: Form the second γ -cycle on the next nearest γ -chord to O, by finding a γ -path in $T \cup m_1$ (not through m_2). Then add the second used γ -chord m_2 to $T \cup m_1$ obtaining $T \cup m_1 \cup m_2$.
- Step 5: Subsequently form the k th γ -cycle on the next unused γ -chord nearest to O, by finding a γ -path in the $T \cup m_1 \cup m_2 \cup \dots \cup m_{k-1}$ (not through m_k). Such a γ -path together with m_k forms a γ -cycle.
- Step 6: Repeat Step 5 until $\eta(S)$ of γ -cycles are selected.

Addition of the used γ -chords to the γ -tree members leads to a considerable reduction in the length of the selected γ -cycles, while maintaining the simplicity of the independence check.

In this algorithm, the use of an SRT, orders the nodes and members of the graph. Such an ordering leads to fairly banded member-node incidence matrices. Considering the columns corresponding to tree members as independent columns, a base is effectively selected for the cycle matroid of the graph, Kaveh [34].

Method 3. This method uses an expansion process, at each step of which one independent γ -cycle is selected and added to the previously selected ones. The independence is secured using an admissibility condition defined as follows.

A γ -cycle C_{k+1} added to the previous selected γ -cycles $C^k = C_1 \cup C_2 \cup \dots \cup C_k$ is called *admissible* if,

$$\gamma(C^k \cup C_{k+1}) = \gamma(C^k) + a, \quad (2.70)$$

where “a” is the coefficient defined in Table 2.1. The algorithm can now be described as follows.

- Step 1: Select the first γ -cycle of minimal length C_1 .
- Step 2: Select the second γ -cycle of minimal length C_2 which is independent of C_1 , i.e. select the second admissible γ -cycle of minimal length.
- Step k: Subsequently, find the k th admissible γ -cycle of minimal length. Continue this process until $\eta(S)$ independent γ -cycles forming a subminimal GCB are obtained.

A γ -cycle of minimal length can be generated on an arbitrary member by adding a γ -path of minimal length between the two end nodes of the member (not through the member itself). The main advantage of this algorithm is avoiding the formation of all γ -cycles of S and also the independence control, which becomes feasible by graph theoretical methods.

The above methods are elaborated for specific $\gamma(S)$ functions in subsequent sections, and examples are included to illustrate their simplicity and efficiency.

2.7 Algebraic Force Methods of Analysis

Combinatorial methods for the force method of structural analysis have been presented in previous sections. These methods are very efficient for skeletal structures and in particular for rigid-jointed frames. For general structures, the underlying graph of self-equilibrating stress systems will be discussed in Chaps. 6 and 7. Algebraic methods can be formulated in a more general form to cover different types of structures such as skeletal structures and finite element models. The main drawbacks of pure algebraic methods are the larger storage requirements, and the higher number of operations.

2.7.1 Algebraic Methods

Consider a discrete or discretized structure S , which is assumed to be statically indeterminate. Let \mathbf{r} denote the m -dimensional vector of generalized independent element (member) forces, and \mathbf{p} the n -vector of nodal loads. The equilibrium conditions of the structure can then be expressed as,

$$\mathbf{A}\mathbf{r} = \mathbf{p}, \quad (2.71)$$

where \mathbf{A} is an $n \times m$ *equilibrium matrix*. The structure is assumed to be rigid, and therefore, \mathbf{A} has a full rank, i.e. $t = m - n > 0$, and $\text{rank } \mathbf{A} = n$.

The member forces can be written as,

$$\mathbf{r} = \mathbf{B}_0\mathbf{p} + \mathbf{B}_1\mathbf{q}, \quad (2.72)$$

where \mathbf{B}_0 is an $m \times n$ matrix such that $\mathbf{A}\mathbf{B}_0$ is an $n \times n$ identity matrix, and \mathbf{B}_1 is an $m \times t$ matrix such that $\mathbf{A}\mathbf{B}_1$ is an $n \times t$ zero matrix. \mathbf{B}_0 and \mathbf{B}_1 always exist for a structure and in fact many of them can be found for a structure. \mathbf{B}_1 is called a *self-stress matrix* as well as *null basis matrix*. Each column of \mathbf{B}_1 is known as a *null vector*. Notice that the null space, null basis and null vectors correspond to complementary solution space, statical basis and self-equilibrating stress systems, respectively, when S is taken as a general structure.

Minimizing the potential energy requires that \mathbf{r} minimize the quadratic form,

$$\frac{1}{2}\mathbf{r}^t\mathbf{F}_m\mathbf{r}, \quad (2.73)$$

subject to the constraint as in Eq. 2.71. \mathbf{F}_m is an $m \times m$ block diagonal element flexibility matrix. Using Eq. 2.72, it can be seen that \mathbf{q} must satisfy the following equation.

$$(\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_1)\mathbf{q} = -\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_0\mathbf{p}, \quad (2.74)$$

where $\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_1 = \mathbf{G}$ is the *overall flexibility matrix of the structure*. Computing the redundant forces \mathbf{q} from Eq. 2.49, \mathbf{r} can be found using Eq. 2.9. The structure of \mathbf{G} is again important and its sparsity, bandwidth and conditioning govern the efficiency of the force method. For the sparsity of \mathbf{G} one can search for a sparse \mathbf{B}_1 matrix, which is often referred to as the sparse null basis problem.

Many algorithms exist for computing a null basis \mathbf{B}_1 of a matrix \mathbf{A} . For the moment let \mathbf{A} be partitioned so that,

$$\mathbf{A}\mathbf{P} = [\mathbf{A}_1, \mathbf{A}_2], \quad (2.75)$$

where \mathbf{A}_1 is $n \times n$ and non-singular, and \mathbf{P} is a permutation matrix that may be required in order to ensure that \mathbf{A}_1 is non-singular. One can write:

$$\mathbf{B}_1 = \mathbf{P} \begin{bmatrix} -\mathbf{A}_1^{-1}\mathbf{A}_2 \\ \mathbf{I} \end{bmatrix}. \quad (2.76)$$

By simple multiplication it becomes obvious that:

$$\mathbf{A}\mathbf{B}_1 = [\mathbf{A}_1 \quad \mathbf{A}_2] \begin{bmatrix} -\mathbf{A}_1^{-1}\mathbf{A}_2 \\ \mathbf{I} \end{bmatrix} = \mathbf{0}.$$

A permutation \mathbf{P} that yields a non-singular \mathbf{A}_1 matrix can be chosen purely symbolically, but this says nothing about the possible numerical conditioning of \mathbf{A}_1 and the resulting \mathbf{B}_1 .

In order to control the numerical conditioning, pivoting must be employed. There are many such methods based on various matrix factorizations, including the Gauss-Jordan elimination, **QR**, **LU**, **LQ** and Turn-back method. Some of these methods are briefly studied in the following:

Gauss-Jordan Elimination Method. In this approach one creates an $n \times n$ identity matrix \mathbf{I} in the first columns of \mathbf{A} by column changes and a sequence of n pivots. This procedure can be expressed as,

$$\mathbf{G}_n\mathbf{G}_{n-1} \dots \mathbf{G}_2\mathbf{G}_1\mathbf{A}\mathbf{P} = [\mathbf{I}, \mathbf{M}], \quad (2.77)$$

Fig. 2.36 Patterns of B_1 and $B_1^t B_1$ matrices for S using Gauss-Jordan elimination method, Example 1. (a) Pattern of B_1 . (b) Pattern of $B_1^t B_1$

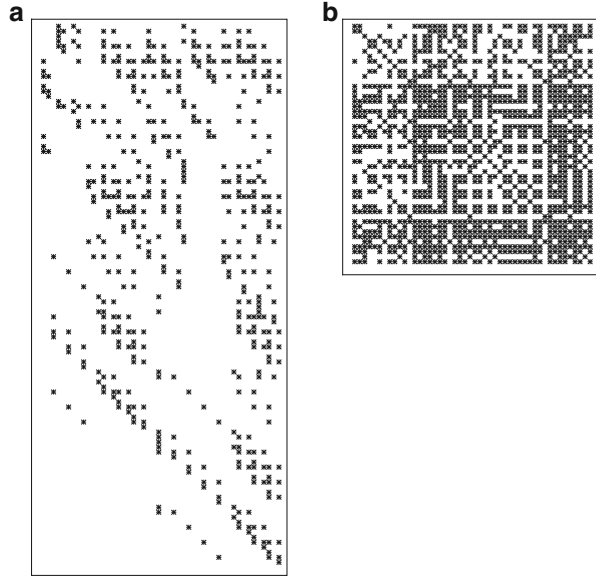
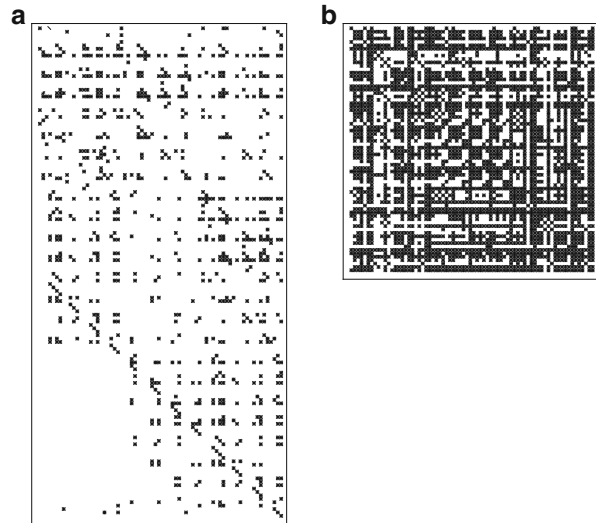


Fig. 2.37 Patterns of B_1 and $B_1^t B_1$ matrices for S using Gauss-Jordan elimination method, Example 2. (a) Pattern of B_1 . (b) Pattern of $B_1^t B_1$



where G_i is the i th pivot matrix and P is an $m \times m$ column permutation matrix (so $P^t = P$) and I is an $n \times n$ identity matrix, and M is an $n \times t$ matrix. Denoting $G_n G_{n-1} \dots G_2 G_1$ by G we have,

Fig. 2.38 Patterns of B_1 and $B_1^t B_1$ matrices for S using LU decomposition method, Example 1.

(a) Pattern of B_1 .
(b) Pattern of $B_1^t B_1$

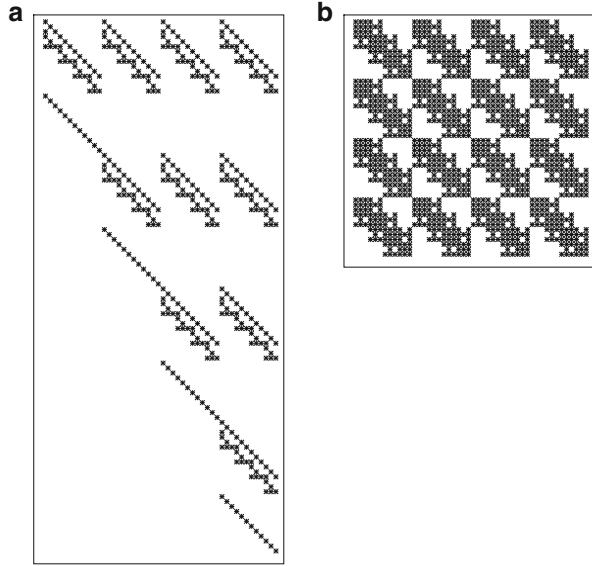
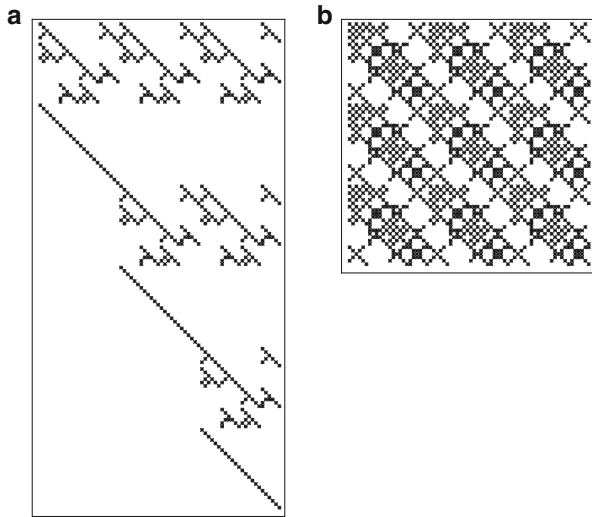


Fig. 2.39 Patterns of B_1 and $B_1^t B_1$ matrices for S using LU decomposition method, Example 2.

(a) Pattern of B_1 .
(b) Pattern of $B_1^t B_1$



$$\mathbf{GAP} = [\mathbf{I}, \mathbf{M}], \quad (2.78)$$

$$\text{or} \quad \mathbf{AP} = \mathbf{G}^{-1}[\mathbf{I}, \mathbf{M}] = [\mathbf{G}^{-1}, \mathbf{G}^{-1}\mathbf{M}], \quad (2.79)$$

which can be regarded as Gauss-Jordan factorization of \mathbf{A} , and:

$$\mathbf{B}_0 = \bar{\mathbf{P}} \begin{bmatrix} \mathbf{G} \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{B}_1 = \bar{\mathbf{P}} \begin{bmatrix} -\mathbf{M} \\ \mathbf{I} \end{bmatrix} \quad (2.80)$$

Example 1. The four by four planar frame of Fig. 2.24 is reconsidered. The patterns of \mathbf{B}_1 and $\mathbf{B}_1^t \mathbf{B}_1$ formed by the Gauss-Jordan elimination method are depicted in Fig. 2.36, corresponding to $\chi(\mathbf{B}_1) = 491$ and $\chi(\mathbf{B}_1^t \mathbf{B}_1) = 1342$.

Example 2. The three-story frame of Fig. 2.24 is re-considered, and the Gauss-Jordan elimination method is used. The patterns of \mathbf{B}_1 and $\mathbf{B}_1^t \mathbf{B}_1$ matrices formed are shown in Fig. 2.37, corresponding to $\chi(\mathbf{B}_1) = 483$ and $\chi(\mathbf{B}_1^t \mathbf{B}_1) = 1592$.

LU Decomposition Method. Using the LU decomposition method, one obtains the LU factorization of \mathbf{A} as,

$$\mathbf{PA} = \mathbf{LU} \quad \text{and} \quad \mathbf{U}\bar{\mathbf{P}} = [\mathbf{U}_1, \mathbf{U}_2], \quad (2.81)$$

\mathbf{P} and $\bar{\mathbf{P}}$ are again permutation matrices of order $n \times n$ and $m \times m$, respectively. Now \mathbf{B}_0 and \mathbf{B}_1 can be written as:

$$\mathbf{B}_0 = \bar{\mathbf{P}} \begin{bmatrix} \mathbf{U}_1^{-1} \mathbf{L}^{-1} \mathbf{P} \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{B}_1 = \bar{\mathbf{P}} \begin{bmatrix} -\mathbf{U}_1^{-1} \mathbf{U}_2 \\ \mathbf{I} \end{bmatrix}. \quad (2.82)$$

Example 1. The four by four planar frame of Fig. 2.24 is re-considered. The patterns of \mathbf{B}_1 and $\mathbf{B}_1^t \mathbf{B}_1$ formed by the LU factorization method are depicted in Fig. 2.38. The sparsity for the corresponding matrices are $\chi(\mathbf{B}_1) = 408$ and $\chi(\mathbf{B}_1^t \mathbf{B}_1) = 1248$.

Example 2. The three-storey frame of Fig. 2.24 is re-considered, and the LU factorization method is used. The patterns of \mathbf{B}_1 and $\mathbf{B}_1^t \mathbf{B}_1$ matrices formed are shown in Fig. 2.39, corresponding to $\chi(\mathbf{B}_1) = 504$ and $\chi(\mathbf{B}_1^t \mathbf{B}_1) = 1530$.

QR Decomposition Method. Using a QR factorization algorithm with column pivoting yields,

$$\mathbf{AP} = \mathbf{Q}[\mathbf{R}_1, \mathbf{R}_2], \quad (2.83)$$

where \mathbf{P} is again a permutation matrix, and \mathbf{R}_1 is an upper triangular matrix of order n . \mathbf{B}_1 can be obtained as:

$$\mathbf{B}_1 = \mathbf{P} \begin{bmatrix} -\mathbf{R}_1^{-1} \mathbf{R}_2 \\ \mathbf{I} \end{bmatrix}. \quad (2.84)$$

Turn-back LU Decomposition Method. Topçu developed a method, the so-called Turn-back LU procedure, which is based on LU factorization and often results in highly sparse and banded \mathbf{B}_1 matrices. Heath et al. [35] adopted this method for use with QR factorization. Due to the efficiency of this method, a brief description of their approach will be presented in the following.

Fig. 2.40 Patterns of B_1 and $B_1^T B_1$ matrices for S using Turn-back LU decomposition method, Example 1. (a) Pattern of B_1 . (b) Pattern of $B_1^T B_1$

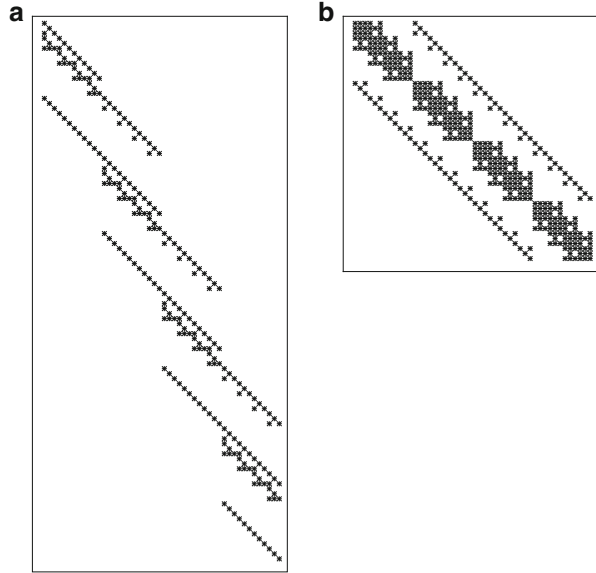
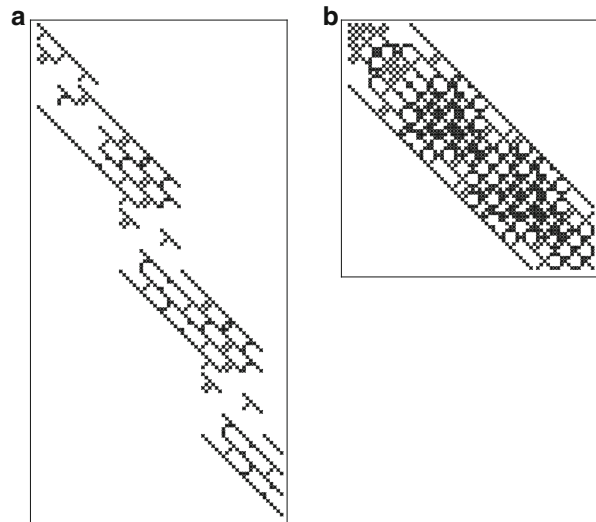


Fig. 2.41 Patterns of B_1 and $B_1^T B_1$ matrices for S using Turn-back LU decomposition method, Example 2. (a) Pattern of B_1 . (b) Pattern of $B_1^T B_1$



Write the matrix $A = (a_1, a_2, \dots, a_n)$ by columns. A *start column* is a column such that the ranks of $(a_1, a_2, \dots, a_{s-1})$ and (a_1, a_2, \dots, a_s) are equal. Equivalently, a_s is a start column if it is linearly dependent on lower-numbered columns. The coefficients of this linear dependency give a null vector whose highest numbered non-zero is in position s . It is easy to see that, the number of start columns is $m - n = t$, the dimension of the null space of A .

The start column can be found by performing a **QR** factorization of **A**, using orthogonal transformations to annihilate the subdiagonal non-zeros. Suppose that in carrying out the **QR** factorization we do not perform column interchanges but simply skip over any columns that are already zero on and below the diagonal. The result will then be a factorization of the form:

$$A=Q \begin{array}{|c|} \hline \begin{array}{c} \diagup \\ \text{R} \\ \diagdown \\ \text{O} \end{array} \\ \hline \end{array} \quad (2.85)$$

The start columns are those columns where the upper triangular structure jogs to the right; that is, a_s is a start column if the highest non-zero position in column s of **R**, is no larger than the highest non-zero position in earlier columns of **R**.

The Turn-back method finds one null vector for each start column a_s , by “turning back” from column s to find the smallest k for which columns $a_s, a_{s-1}, \dots, a_{s-k}$ are linearly dependent. The null vector has a non-zero only in position $s-k$ throughs. Thus if k is small for most of the start columns, then the null basis will have a small profile. Notice that the turn-back operates on **A**, and not on **R**. The initial **QR** factorization of **A** is used only to determine the start columns, and then discarded.

The null vector that Turn-back finds from start column a_s may not be non-zero in position s . Therefore, Turn-back needs to have some way to guarantee that its null vectors are linearly independent. This can be accomplished by forbidding the left-most column of the dependency for each null vector from participating in any later dependencies. Thus, if the null vector for start column a_s has its first non-zero in position $s-k$, every null vector for a start column to the right of a_s will be zero in position $s-k$.

Although the term “Turn-back” is introduced in Ref. [7], the basic idea had also been used in Refs. [36]. Since this correspondence simplifies the understanding of the Turn-back method, it is briefly described in the following.

For the Algorithm 1 of Sect. 2.3, the use of an SRT orders the nodes and members of the graph simultaneously, resulting in a fairly banded member-node incidence matrix **B**. Considering the columns of **B** corresponding to tree members as independent columns, effectively a cycle is formed on each ordered chord (start column) by turning back in **B** and establishing a minimal dependency, using the tree members and previously used chords. The cycle basis selected by this process forms a base for the cycle matroid of the graph, as it is described in Kaveh [37]. Therefore, the idea used in Algorithm 1 and its generalization for the formation of a generalized cycle bases in Ref. [38] seems to constitute a similar idea to that of the algebraic Turn-back method.

Example 1. The four by four planar frame of Fig. 2.24 is re-considered. The patterns of **B**₁ and **B**₁^{t**B**₁ formed by the Turn-back LU factorization method are depicted in Fig. 2.40, corresponding to $\chi(\mathbf{B}_1) = 240$ and $\chi(\mathbf{B}_1^t \mathbf{B}_1) = 408$.}

Example 2. The four by four planar frame of Fig. 2.24 is re-considered, and the Turn-back LU factorization method is used. The patterns of \mathbf{B}_1 and $\mathbf{B}_1^t \mathbf{B}_1$ matrices formed are shown in Fig. 2.41, corresponding to $\chi(\mathbf{B}_1) = 476$ and $\chi(\mathbf{B}_1^t \mathbf{B}_1) = 984$.

A comparative study of various force methods has been made in Ref. [30].

Many algorithms have been developed for selection of null bases, and the interested reader may refer to Refs. [38, 39].

References

1. Henderson JC de C, Bickley WG (1955) Statical indeterminacy of a structure. *Aircr Eng* 27:400–402
2. Maunder EWA (1971) Topological and linear analysis of skeletal structures. Ph.D. thesis, London University, IC
3. Kaveh A (1992) Recent developments in the force method of structural analysis. *Appl Mech Rev* 45:401–418
4. Langefors B (1961) Algebraic topology and elastic networks, SAAB TN49. Linköping
5. Denke PH (1962) A general digital computer analysis of statically indeterminate structures, NASA-TD-D-1666
6. Robinson J (1973) Integrated theory of finite element methods. Wiley, New York
7. Topçu A (1979) A contribution to the systematic analysis of finite element structures using the force method (in German). Doctoral dissertation, Essen University
8. Kaneko I, Lawo M, Thierauf G (1982) On computational procedures for the force methods. *Int J Numer Method Eng* 18:1469–1495
9. Gilbert JR, Heath MT (1987) Computing a sparse basis for the null space. *SIAM J Algebra Discr Method* 8:446–459
10. Coleman TF, Pothén A (1987) The null space problem II; algorithms. *SIAM J Algebra Discr Method* 8:544–561
11. Patnaik SN (1986) Integrated force method versus the standard force method. *Comput Struct* 22:151–164
12. Kaveh A, Jahanshahi M (2006) An efficient program for cycle basis selection and bandwidth optimization. *Asian J Civil Eng* 7(1):95–109
13. Kaveh A, Daei M (2010) Suboptimal cycle bases of graphs using an ant colony system algorithm. *Eng Comput* 27(4):485–494
14. Timoshenko S, Young DH (1945) Theory of structures. McGraw-Hill, New York
15. Kaveh A (1974) Application of topology and matroid theory to the analysis of structures. Ph.D. thesis, London University, IC
16. Kaveh A (1988) Topological properties of skeletal structures. *Comput Struct* 29:403–411
17. Mauch SP, Fenves SJ (1967) Release and constraints in structural networks. *J Struct Div ASCE* 93:401–417
18. Müller-Breslau H (1912) Die graphische Statik der Baukonstruktionen. Alfred Kröner Verlag, 1907, und Leipzig
19. Kaveh A (2004) Structural mechanics: graph and matrix methods, 3rd edn. Research Studies Press, Baldock
20. Henderson JC de C, Maunder EWA (1969) A problem in applied topology. *J Inst Math Appl* 5:254–269
21. Kaveh A (1976) Improved cycle bases for the flexibility analysis of structures. *Comput Method Appl Mech Eng* 9:267–272
22. Kaveh A (1988) Suboptimal cycle bases of graphs for the flexibility analysis of skeletal structures. *Comput Method Appl Mech Eng* 71:259–271

23. Stepanec GF (1964) Basis systems of vector cycles with extremal properties in graphs. *Uspekhi Mat Nauk* 19:171–175 (in Russian)
24. Zykov AA (1969) Theory of finite graphs. Nuaka, Novosibirsk (in Russian)
25. Hubicka E, Syslø MM (1975) Minimal bases of cycles of a graph. In: Fiedler M (ed) Recent advances in graph theory. Academia Praha, Prague, pp 283–293
26. Kaveh A, Roosta GR (1994) Revised Greedy algorithm for the formation of minimal cycle basis of a graph. *Commun Numer Method Eng* 10:523–530
27. Horton JD (1987) A polynomial time algorithm to find the shortest cycle basis of a graph. *SIAM J Comput* 16:358–366
28. Lawler EL (1976) Combinatorial optimization; networks and matroids. Holt, Rinehart and Winston, New York
29. Kolasinska E (1980) On a minimum cycle basis of a graph. *Zastos Math* 16:631–639
30. Kaveh A, Mokhtar-zadeh A (1993) A comparative study of the combinatorial and algebraic force methods. In: Proceedings of the Civil-Comp93, Edinburgh, pp 21–30
31. Brusa L, Riccio F (1989) A frontal technique for vector computers. *Int J Numer Method Eng* 28:1635–1644
32. Prezemiesniecki JS (1968) Theory of matrix structural analysis. McGraw-Hill, New York
33. Pestel EC, Leckie FA (1963) Matrix methods in elastomechanics. McGraw-Hill, New York
34. Kaveh A (1993) Matroids applied to the force method of structural analysis. *Z Angew Math Mech* 73:T355–T357
35. Heath MT, Plemmons RJ, Ward RC (1984) Sparse orthogonal schemes for structural optimization using the force method. *SIAM J Sci Stat Comput* 5:514–532
36. Cassell AC (1976) An alternative method for finite element analysis; a combinatorial approach to the flexibility method. *Proc R Soc Lond A* 352:73–89
37. Kaveh A (1979) A combinatorial optimization problem; optimal generalized cycle bases. *Comput Method Appl Mech Eng* 20:39–52
38. Coleman TF, Pothén A (1986) The null space problem I; complexity. *SIAM J Algebra Disc Method* 7:527–537
39. Plemmons RJ, White RE (1990) Substructuring methods for computing the null space of equilibrium matrices. *SIAM J Matrix Anal Appl* 11:1–22

Computational Structural Analysis and Finite Element
Methods

Kaveh, A.

2014, XVI, 432 p. 305 illus., 90 illus. in color., Hardcover

ISBN: 978-3-319-02963-4