CHAPTER 6

Matrix Force Method

6.1 INTRODUCTION

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 non-linear analysis and optimisation has been neglected.

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

1. Topological force methods, 2. Algebraic force methods,

3. Mixed algebraic-combinatorial force methods, 4. Integrated force method.

Topological methods have been developed by Henderson [76] and Maunder [169] for rigid-jointed skeletal structures using manual selection of the cycle bases of their graph models. Methods suitable for computer programming are due to Kaveh [89,93,107]. These topological methods are generalized to cover all types of skeletal structures, such as rigid-jointed frames, pin-jointed planar trusses and ball-jointed space trusses [94].

Algebraic methods have been developed by Denke [36], Robinson [204], Topçu [238], Kaneko et al. [87], Soyer and Topçu [221] and mixed algebraic-topological methods have been used by Gilbert et al. [58], Coleman and Pothen [29-30], and Pothen [196].

The integrated force method has been developed by Patnaik [177-179], in which the equilibrium equations and the compatibility conditions are satisfied simultaneously in terms of the force variables.

In this chapter, topological methods using graph-theoretical concepts are mainly presented. However, algebraic methods and mixed algebraic-topological approaches are also briefly discussed.

6.2 FORMULATION

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.

Consider a structure S with M members and N nodes, which is γ(S) times statically indeterminate. Select γ(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:

$$\mathbf{q} = \{q_1, q_2, ..., q_{\gamma(S)}\}^t . \tag{6-1}$$

Remove the constraints corresponding to redundants, in order to obtain a 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,

$$\mathbf{p} = \{p_1, p_2, ..., p_n\}^t , \tag{6-2}$$

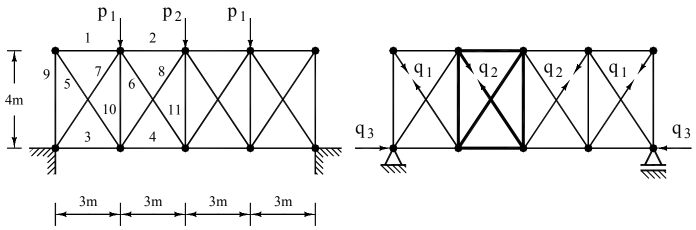
where n is the number of components for applied nodal loads.

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

$$\mathbf{r} = \mathbf{B}_0\mathbf{p} + \mathbf{B}_1\mathbf{q}, \tag{6-3}$$

where $\mathbf{B}_0$ and $\mathbf{B}_1$ are rectangular matrices each having m rows, and n and γ(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*.

As an example, consider a planar truss, as shown in Figure 6.1(a), which is 5 times statically indeterminate. EA is taken to be the same for all the members.



(a) A planar truss. (b) Selected unknown forces.

Fig. 6.1 A statically indeterminate planar truss.

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 Figure 6.1(b). 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=1,2,3), respectively:

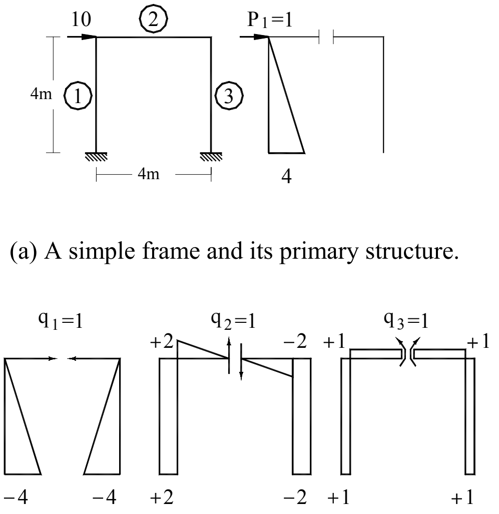
$$\mathbf{B}_0^t = \begin{bmatrix} 0 & -3/4 & 3/4 & 3/4 & 0 & 0 & -5/4 & 0 & 0 & 1 & 0 \\ 0 & -3/8 & 3/8 & 3/4 & 0 & 0 & -5/8 & -5/8 & 0 & 1/2 & 1 \end{bmatrix},$$

and

$$\mathbf{B}_1^t = \begin{bmatrix} -3/5 & 0 & -3/5 & 0 & 1 & 0 & 1 & 0 & -4/5 & -4/5 & 0 \\ 0 & -3/5 & 0 & -3/5 & 0 & 1 & 0 & 1 & 0 & -4/5 & -4/5 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The columns of $\mathbf{B}_1$ 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, Figure 6.1(b).

As a second example, consider a portal frame shown in Figure 6.2(a), which is 3 times statically indeterminate.



(a) A simple frame and its primary structure.

(b) Three self-equilibrating systems.

Fig. 6.2 A statically indeterminate frame.

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:

$$\mathbf{B}_0^t = \begin{bmatrix} -4 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

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, Figure 6.2(b). Notice that three self-equilibrating stress systems are formed on a 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*, Ref. [78].

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 3-storey frame as shown in Figure 6.3(a). A cut system as shown in Figure 6.3(b) corresponds to a statical basis containing three self-equilibrating stress systems formed on each element of the cycle basis shown in Figure 6.3(b). However, the same particular solution can be employed with a statical basis formed on the cycles of the basis shown in Figure 6.3(c).

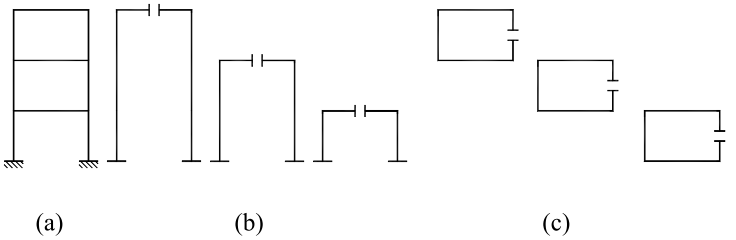


Fig. 6.3 A 3-storey frame with different cut systems.

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 **p** and bi-actions $q_i = 1$ (i=1,2, ... , $\gamma(S)$), or by using existing information.

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

$$\mathbf{u} = \mathbf{F}_m\mathbf{r} = \mathbf{F}_m\mathbf{B}_0\mathbf{p} + \mathbf{F}_m\mathbf{B}_1\mathbf{q} . \tag{6-4}$$

In matrix form:

$$[\mathbf{u}] = [\mathbf{F}_m][\mathbf{B}_0 \;\; \mathbf{B}_1]\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix}. \tag{6-5}$$

From the principle of virtual work:

$$[\mathbf{v}] = \begin{bmatrix} \mathbf{B}_0^t \\ \mathbf{B}_1^t \end{bmatrix} [\mathbf{u}]. \tag{6-6}$$

Combining Eq. (6-5) and Eq. (6-6) results in,

$$\begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_c \end{bmatrix} = \begin{bmatrix} \mathbf{B}_0^t \\ \mathbf{B}_1^t \end{bmatrix} [\mathbf{F}_m][\mathbf{B}_0 \ \mathbf{B}_1] \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}, \tag{6-7}$$

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,

$$\begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_c \end{bmatrix} = \begin{bmatrix} \mathbf{B}_0^t\mathbf{F}_m\mathbf{B}_0 & \mathbf{B}_0^t\mathbf{F}_m\mathbf{B}_1 \\ \mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_0 & \mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}, \tag{6-8}$$

or

$$\mathbf{v}_0 = (\mathbf{B}_0^t\mathbf{F}_m\mathbf{B}_0)\mathbf{p} + (\mathbf{B}_0^t\mathbf{F}_m\mathbf{B}_1)\mathbf{q}, \tag{6-9}$$

and

$$\mathbf{v}_c = (\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_0)\mathbf{p} + (\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_1)\mathbf{q}. \tag{6-10}$$

Consider now the compatibility conditions as:

$$\mathbf{v}_c = \mathbf{0}. \tag{6-11}$$

The above equation together with Eq. (6-10) leads to:

$$\mathbf{q} = -(\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_1)^{-1}(\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_0)\mathbf{p}. \tag{6-12}$$

Substituting in Eq. (6-9) yields:

$$\mathbf{v}_0 = [\mathbf{B}_0^t\mathbf{F}_m\mathbf{B}_0 - \mathbf{B}_0\mathbf{F}_m\mathbf{B}_1(\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_1)^{-1}\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_0]\mathbf{p}, \tag{6-13}$$

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

$$\mathbf{r} = [\mathbf{B}_0 - \mathbf{B}_1(\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_1)^{-1}\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_0]\mathbf{p}, \tag{6-14}$$

where $\mathbf{G} = (\mathbf{B}_1^t\mathbf{F}_m\mathbf{B}_1)$ is known as the *flexibility matrix* of the structure.

For an efficient force method, the matrix **G** should be:

(a) sparse;

(b) well-conditioned;

(c) well-structured, i.e. narrowly banded.

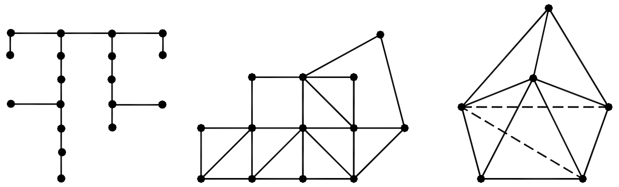
In order to provide the properties (a) and (b) for **G**, the structure of $\mathbf{B}_1$ should be carefully designed, since the pattern of $\mathbf{F}_m$ for a given discretization is unchanged; i.e. a suitable statical basis should be selected. This problem is treated in different forms by various 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 Chapter 8.

6.3 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.

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 as defined in Chapter 1.

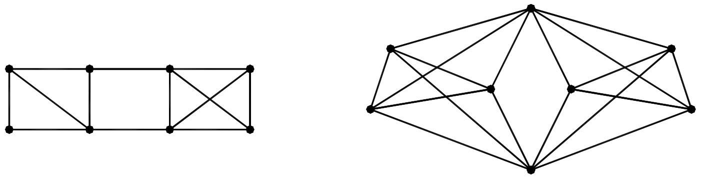
Obviously, a structure whose model is a γ-tree is statically determinate when γ(S) describes the degree of static indeterminacy of the structure. The ensuing stress resultants can uniquely be determined everywhere in the structure by equilibrium only. Examples of γ-trees for different γ functions are shown in Figure 6.4.



(a) $\gamma(S)=3M-3N+3$. (b) $\gamma(S)=M-2N+3$. (c) $\gamma(S)=M-3N+6$.

Fig. 6.4 Examples of γ-trees.

Notice that $\gamma(T) = 0$ does not guarantee the rigidity of a γ-tree. For example, the graphs depicted in Figure 6.5 both satisfy $\gamma(T) = 0$; however, neither is rigid.

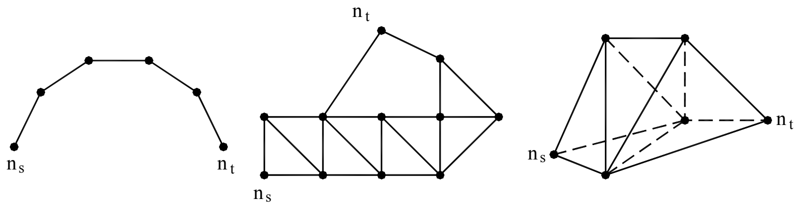


(a) $\gamma(S)=M-2N+3$. (b) $\gamma(S)=M-3N+6$.

Fig. 6.5 Structures satisfying $\gamma(T)=0$ which are not rigid.

Definition 2: A member of $S - T$ is called a γ-*chord* of T.

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.

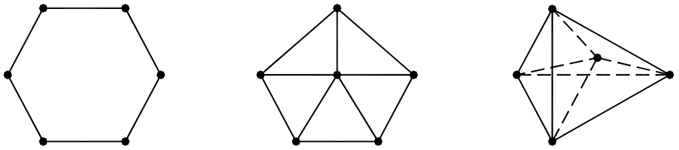


(a) $\gamma(S)=\alpha(M-N+1)$. (b) $\gamma(S)=M-2N+3$. (c) $\gamma(S)=M-3N+6$.

Fig. 6.6 Examples of γ-paths.

As an example, the graphs shown in Figure 6.6 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 C_k, denoted by $L(C_k)$, is called the *length* of C_k. Examples of γ-cycles are shown in Figure 6.7.



(a) $\gamma(S)=\alpha(M-N+1)$. (b) $\gamma(S)=M-2N+3$. (c) $\gamma(S)=M-3N+6$.

Fig. 6.7 Examples of γ-cycles.

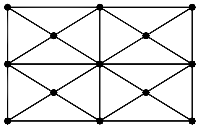
Definition 5: Let m_i be a γ-chord of T. Then $T \cup m_i$ contains a γ-cycle C_i which is defined as a *fundamental γ-cycle* of S with respect to T. Using the Intersection Theorem of Chapter 2 (Eq. (2-9)), it can easily be shown that,

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

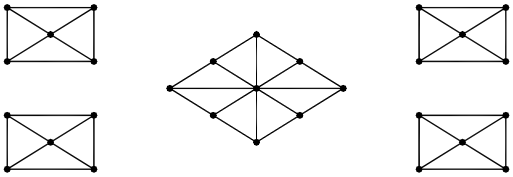
indicating the existence of a γ-cycle in $T \cup m_i$. For a rigid T, the corresponding fundamental γ-cycle is also rigid, since the addition of an extra member between the existing nodes of a graph cannot destroy the rigidity. A fundamental γ-cycle can be obtained by omitting all the removable subgraphs 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 shown in Figure 6.8.



(a) A planar truss S.



(b) A generalized cycle basis of S.

Fig. 6.8 A planar truss S, and the elements of a GCB of S.

Definition 7. A *generalized cycle basis-member incidence matrix* **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 **D**, which is an $\eta(S)\times\eta(S)$ matrix when undirected γ-cycles are considered; then the negative entries of **C** become positive.

6.4 MINIMAL AND OPTIMAL GENERALIZED CYCLE BASES

A matrix is called *sparse* if many of its entries are zero.

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

$$L(C) = \sum_{i=1}^{\eta(S)} L(C_i) . \tag{6-15}$$

Obviously $\chi(\mathbf{C}) = L(C)$ and a minimal GCB can be defined as a basis which corresponds to minimum $\chi(\mathbf{C})$. A GCB for which L(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 $\chi(\mathbf{CC}^t)$ 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 **C** is the number of row j such that:

(a) $j\in\{i+1,i+2,...,\eta(S)\}$,

(b) $C_i \cap C_j \neq \varnothing$, 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:

$$\chi(\mathbf{D}) = \eta(S) + 2 \sum_{i=1}^{\eta(S)-1} \sigma_i(\mathbf{C}) . \tag{6-16}$$

This relationship shows the correspondence of a GCB incidence matrix **C** and that of its GCB adjacency matrix. In order to minimize $\chi(\mathbf{D})$, the value of $\sum_{i=1}^{\eta(S)-1} \sigma_i(\mathbf{C})$

should be minimized, since η(S) is a constant for a given structure S; i.e. γ-cycles with a minimum number of overlaps should be selected.

6.5 PATTERN EQUIVALENCE OF FLEXIBILITY AND CYCLE ADJACENCY MATRICES

Matrix $\mathbf{B}_1$ containing a statical basis, in partitioned form, is pattern equivalent to $\mathbf{C}^t$. Similarly $\mathbf{B}_1^t\,\mathbf{F}_m\,\mathbf{B}_1$ is pattern equivalent to $\mathbf{CIC}^t$ or $\mathbf{CC}^t$. 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 $\mathbf{CC}^t$.

As an example, if a sparse matrix **G** is required, this can be achieved by increasing the sparsity of $\mathbf{CC}^t$. Similarly for a banded **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 $\mathbf{CC}^t$ is often smaller than that of **G**. For example, for a space frame the dimension of $\mathbf{CC}^t$ is six-fold and for a planar frame three-fold smaller than that of **G**. Therefore the optimisation process becomes much simpler when combinatorial properties are used.

2) The entries of **C** and $\mathbf{CC}^t$ are elements of Z_2 and therefore easier to operate on, compared to $\mathbf{B}_1$ and **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.

6.6 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 Chapter 9 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. $L(C_1) \leq L(C_i)$ 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 $C - \{C_1,C_2,...,C_{k-1}\}$ which is independent of C_1, C_2,..., C_{k-1} and $L(C_k) \leq L(C_i)$ for all $C_i \in C - \{C_1,C_2,..., C_{k-1} \}$.

After η(S) steps, a minimal GCB will be selected by this process, a proof can be found in Kaveh [94].

6.7 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 γ(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.

6.7.1 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 GCB adjacency matrices.

6.7.2 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 are discussed in Chapter 7.

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 $T \cup m_1 \cup m_2 \cup ... \cup m_{k-1}$ (not through m_k). Such a γ-path together with m_k forms a γ-cycle.

Step 6: Repeat Step 5 until η(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 [89,108].

6.7.3 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 $C^k = C_1 \cup C_2 \cup ... \cup C_k$ is called *admissible* if

$$\gamma(C^k \cup C_{k+1}) = \gamma(C^k) + a, \tag{6-17}$$

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 that 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.

6.8 FORCE METHOD FOR THE ANALYSIS OF RIGID-JOINTED STRUCTURES

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,

$$\gamma(S) = \alpha b_1(S) = \alpha[M(S) - N(S) + b_0(S)], \tag{6-18}$$

where $\alpha = 3$ or 6 depending on the structure being a planar or a space frame, respectively.

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 4 or 6 S.E.Ss on each element of a cycle basis of S.

In order to obtain a flexibility matrix of maximal sparsity, a cycle basis corresponding to minimum $\chi(\mathbf{D})$ is needed, i.e. an optimal cycle basis should be formed. However, because of the complexity or this problem, most of the research has been concentrated on minimal cycle basis selection, except those of Refs. [97,98], which minimize the overlaps of the cycles rather than their length only.

Methods for 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 [78] and Kaveh [89,108]. Graph-theoretical methods suitable for computer applications, were developed by Kaveh [93,96, 108]. For completeness of the discussion, both groups of methods are described in this section. The first group, however, requires some terminology from algebraic and combinatorial topology for which the reader may refer to Cooke and Finney [31]; otherwise Section 6.8.1 can be ignored.

6.8.1 CYCLE BASES SELECTION: TOPOLOGICAL METHODS

Cycle selection procedures described in this section use the concept of embedding the geometric realization of S into another polyhedron whose dissection has dimension equal to 2. The idea originates from a planar graph embedded in R^2, in which the cycles bounding finite regions form an efficient basis (mesh basis) for the first cycle group $Z_1(S,R)$.

The object is to extend this approach to embedding S on polyhedrons and manifolds with certain properties. These properties are measured by using the homology groups $H_p(K,R)$ of the underlying complex, which measure, roughly speaking, the number of independent p-dimensional holes of K. In other words, they measure the extent to which K has non-bounding p-cycles.

A 2-Dimensional Polyhedron Embedding. Let S be the mathematical model of a structure, which is a simple graph (1-complex). The underlying polyhedron or geometric realization of S is often denoted by |S|. However, for simplicity it is also denoted by S.

An *embedding* f: S→ P is a homemorphism of S into polyhedron P. An embedding is called a *2-cell embedding* if the components of [P−f(S)] are all 2-cells. If the 2-cells are regular, then embedding is called a *regular 2-cell embedding*. Let f (S) be dissected into a 1-complex isomorphic to the dissection of S. Then f(S) and the components of [P−f(S)] form a dissection of P into a 2-dimensional cell complex K.

Admissible Embeddings. The cycles bounding the 2-cells of K are known as *regional cycles*. An *admissible embedding* f of S is one for which the regional cycles form a set of $b_1(S)$ independent cycles from $Z_1(K,R)$. The necessary and sufficient condition for a 2-cell embedding f: S → P to be admissible is that the homology groups with real coefficients are trivial. In the case of 2-cell embedding, this condition holds when first and second Betti number of K are zero. Using concepts from algebraic topology, the admissibility condition can be stated as follows:

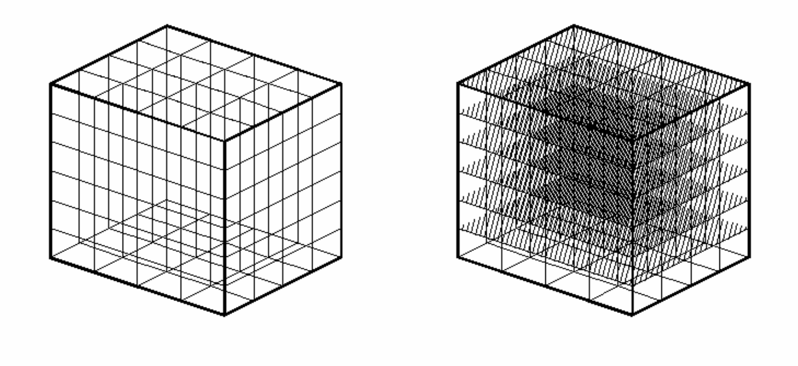
> The necessary and sufficient condition for f: S → P to be admissible is that the corresponding K be acyclic.

A regular complex K is called *acyclic* if all homology groups of K are trivial. Equivalently, any cycle of $Z_1(K,R)$ bounds in K; i.e. $Z_1(K,R) = B_1(K,R)$.

It is easy to show that a contractible complex is acyclic; hence a contractible embedding is admissible. If K is collapsible, then it is obviously contractible. Thus a collapsible embedding is also admissible.

A graph can be viewed as the 1-skeleton of a 3-complex. An n-cell is called *collapsible* if it can be shrunk into the remainder of its n−1 cells through a free n−1 cell. If a 3-complex can be collapsed into a point, then it is called *collapsible*. A collapsible 3-complex can be used for the formation of a cycle basis of its 1-skeleton. This can be achieved by collapsing all the 3-cells through free 2-cells or 2-cells being freed in subsequent steps.

Example: Consider a space graph S as shown in Figure 6.9(a). S can be viewed as the 1-skeleton of a 3-complex as depicted in Figure 6.9(b). After collapsing all the 3-cells through the shaded 2-cells, the bounding cycles of the remaining 2-complex (81 cycles) form a cycle basis of S.



(a) A space graph S. (b) S embedded on a 3-complex

Fig. 6.9 A space graph and its collapsible embedding.

Modified Manifold Embedding: Edmond's permutation technique provides a method for a 2-cell embedding of S in an orientable 2-manifold M. The choice of node permutations in this method is made arbitrarily, which may lead to a manifold with an unnecessarily large genus. The *genus* of graph is the minimum number of handles of a sphere on which the graph can be embedded. In order to reduce the genus of **M**, Duke [41] developed a reduction technique for transforming a 2-cell embedding **M** into **M***, where the genus of **M*** is one less than that of **M**. Youngs [253] developed an algorithm for minimal embedding by considering all the possible sets of node permutations of **S**, leading to genus of **S**. This algorithm is very lengthy and impractical. Kaveh [95] developed a practical method in which the node permutations are determined in the process of embedding S. However, this algorithm does not always lead to minimal embeddings.

A different approach may also be employed which is based on an intuitive regular embedding of S on a manifold. For an embedding f: S → P which dissects polyhedron **M** into 2-complex **K**, **M** is a manifold if:

1. Each 1-cell of K is incident with exactly two 2-cells.

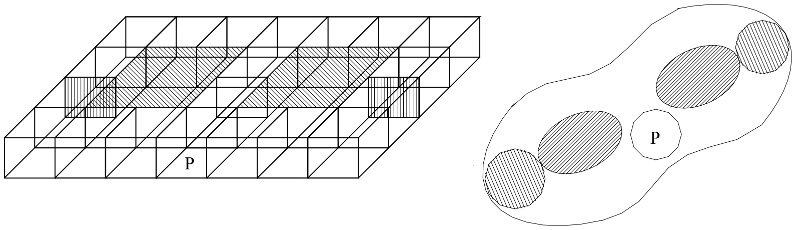
2. All the 2-cells and 1-cells of K having a particular 0-cell and a face can be ordered in a sequence so that the consecutive cells are incident.

3. K can be oriented so that $\sum_{i} \bar{z}_i = 0$, where $\bar{z}_i$ is a regional cycle.

If M is manifold, then $b_2(K) = 1$ and $b_1(K) = 2\Psi$, where Ψ is the genus of **M**, i.e.

M is homeomorphic to a sphere with Ψ handles. Thus f is not admissible, but can be modified by adding 2Ψ appropriate fillings and one perforation of order 2 to make it admissible.

Example: Consider a hollow box S embedded on a sphere with two handles, i.e. a double torus, as shown in Figure 6.10(a). Modifications are made by four proper fillings and one perforation (P) of order 2, as illustrated schematically in Figure 6.10 (b).



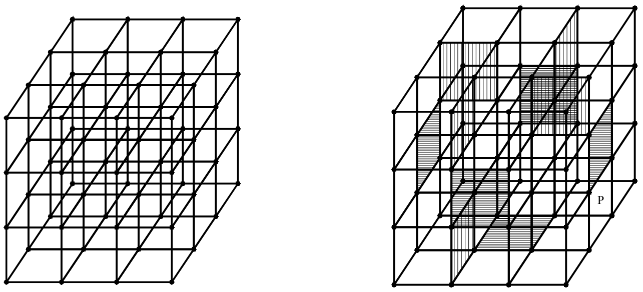
(a) S and its embedding. (b) Schematic illustration of the modified torus.

Fig. 6.10 An admissible embedding of S.

Seventy-nine cycles of length 4 and two cycles of length 8 are selected as a minimal cycle basis of S. However, for multi-member complex structures, this method is by no means practical.

Example: Let S be the graph model of a 3-storey frame as shown in Figure 6.11(a). The first storey can be embedded on a torus and the third storey on a second torus. Four tubes can be added to accommodate the columns of the second storey. Identifying these through one of the tubes results in a sphere with five handles as depicted in Figure 6.11(b). Once the embedding is achieved, modifications can easily be made. In the schematic representation, fillings are shaded and the necessary perforation is denoted by P, Figure 6.11(b).

In a manifold embedding, the quality of the selected cycle basis depends upon the genus of the manifold on which S is embedded. Thus it is ideal to have a minimal embedding. However, little is known about an efficient approach to carry out such an embedding.



(a) Graph model of a space frame. (b) Schematic illustration of the embedding.

Fig. 6.11 A modified manifold embedding of a space graph.

Embedding S on a Union of Discs: S can be considered as the union of some planar connected non-separable subgraphs of S. The process of such an embedding is as follows:

Step 1: Identify a planar subgraph S_1 and embed it on a disc d_1 whose dissection K_1 is isomorphic to S_1.

Step 2: The second subgraph S_2 is identified such that the corresponding K_2 has a 2-cell with a free 1-face and $|K_1| \cap |K_2|$ is a connected subspace of the frontier of d_1.

Step 3: The process step 2 is continued, and at the *i*th step K_i is joined to $K^{i-1} = \bigcup_{j=1}^{i-1} K_j$, with K_i having a free 1-face and $|K_i| \cap |K^{i-1}|$ being a connected subspace of the frontier of d_{i-1} Obviously K^i is collapsible to K^{i-1}.

The process will be terminated when all 1-cells of S are embedded in $K^q = \bigcup_{j=1}^{q} K_j$ which is collapsible, since:

$$K = K^q \to K^{q-1} \to \ldots \to K^i \to \ldots \to K^1 = K_1.$$

Example: Consider a space frame S as illustrated in Figure 6.12. This model is embedded on three discs K_1-K_3 as shown in the Figure, resulting in a cycle basis consisting of 56 regional cycles of length 4.

It is ideal to embed S in a complex K with a minimum number of discs for all possible collapsible embeddings. This number is known as the *thickness* of the graph S, denoted by t(S), Ref. [178]. Hence an embedding should be performed with q as the thickness of S. Again only partial results are available about the thickness of the graphs. Any approach to embedding on a smaller number of discs would be advantageous for reducing the overlaps of the cycles.

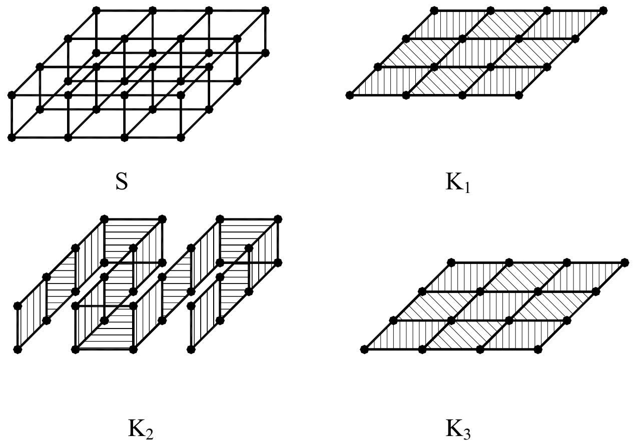


Fig. 6.12 A space graph and the identified discs K_1-K_3.

Remark: The computer implementation of the methods of this section for selecting a cycle basis may become uneconomical from engineering point of view; however, the study of these methods provides a firm background for designing many graph-theoretical approaches. The visualization of these methods helps to develop algorithms having different useful characteristics. As an example, the expansion process of the next section may be viewed as an embedding of a graph on the elements of a disc space, with discs having certain properties. Similarly, one can design an algorithm for embedding a graph on the elements of a ball space, with balls having specified properties (the reverse process of a collapsible embedding).

6.8.2 CYCLE BASES SELECTION: GRAPH-THEORETICAL METHODS

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 [225] and improved by Zykov [257]. Many practical algorithms for selecting subminimal cycle bases have been developed by Kaveh [93] and Cassell et al. [21]. Similar methods have been presented by Hubicka and Syslø [83] claiming the formation of a minimal cycle basis of a graph. Kolasinska [140] found a counter example to Hubicka and Syslø´s algorithm. A similar conjecture was made by Kaveh [93] for planar graphs; however, a counter example has been given by Kaveh and Roosta [114]. Horton [82] presented a polynomial time algorithm to find minimal cycle bases of graphs. Kaveh and Rahami used an algebraic graph-theoretical approach [133].

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 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 Figure 6.13. The SRTs are shown in bold lines. Notice that the generation of SRTs is terminated as soon as n_c has been found.

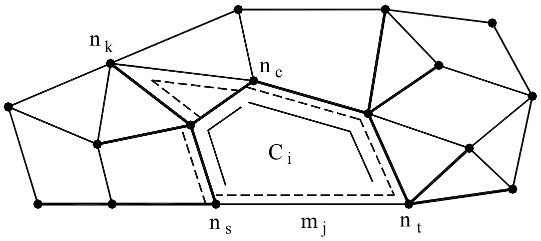


Fig. 6.13 A minimal cycle on a member.

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, Figure 6.13.

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.

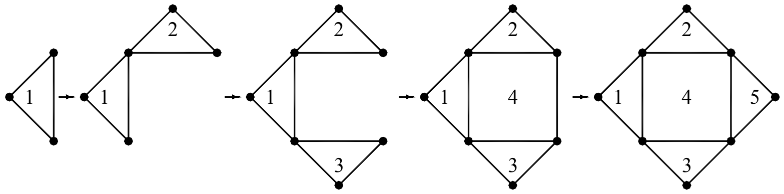


Fig. 6.14 A graph S and selected cycles.

In order to overcome the above difficulty, Kaveh [93] 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 Figure 6.14.

Hubicka and Syslø [83] 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 [114],

Finally, Horton [82] 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)×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 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-theretical 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 [89].

A third method, which is also graph-theretical, consists of using admissible cycles. Consider the expansion process,

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

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

$$b_1(C^{k+1}) = b_1 (C^k \cup C_{k+1}) = b_1 (C^k) + 1. \qquad (6\text{-}19)$$

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

(1) $A_{k+1} = C^k \cap C_{k+1} = \varnothing$, where $\varnothing$ is an empty intersection.

(2) $\overline{b}_1 (A_{k+1}) = r - s$, where r and s are the numbers of components of C^{k+1} and C^k, respectively.

(3) $\overline{b}_1 (A_{k+1}) = 0$ when C^k and C^{k+1} are connected (r=s).

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

As an example, the sequence of cycle selection in Figure 6.15 are as specified by their numbers.

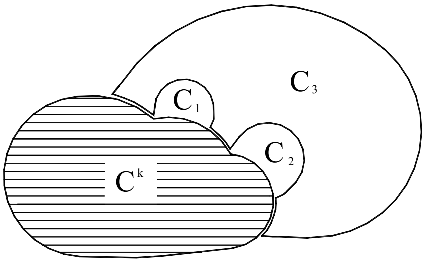


Fig. 6.15 A cycle and its bounded cycles.

A different approach, is suggested by Hubicka and Syslø, in which,

$$b_1(C^{k+1}) = b_1(C^k) + p, \tag{6-20}$$

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 1974)

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-theretical methods discussed in Chapter 7.

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 Section 6.10.1.

Algorithm 2 (Kaveh 1974)

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 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:

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

(ii) 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 1976)

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.

Algorithm 4 (Horton 1987)

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.

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 to be used in the Greedy Algorithm. The use of matroids and the Greedy Algorithm has been suggested by Kaveh [89,93,105], and they have been employed by Lawler [157] and Kolasinska [140].

Example 1: Consider a planar graph S as shown in Figure 6.16, 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:

$C_1 = (1,2,3)$, $C_2 = (1,8,9)$, $C_3 = (2,6,3)$, $C_4 = (2,5,6)$, $C_5 = (1,4,5,2)$

$C_6 = (1,7,5,2)$, $C_7 = (8,6,2,1)$, $C_8 = (10,8,6,3,11)$.

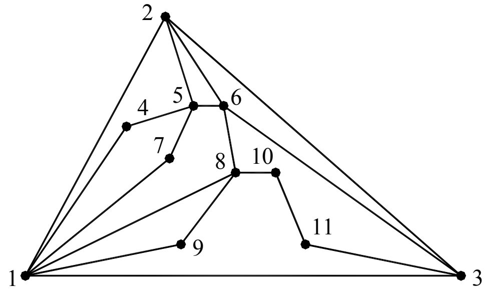
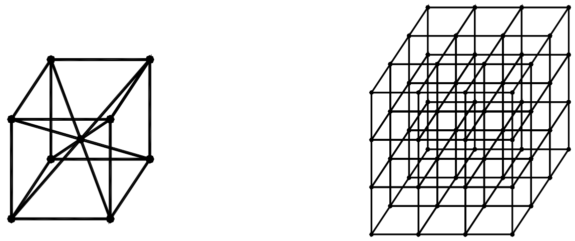


Fig. 6.16 A planar graph S.

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 Figure 6.17(a). For S_i there are 12 members joining 8 corner nodes, and a central node joined to these corner nodes. The model S is shown in Figure 6.17(b), in which some of the members are omitted for clarity in the diagram. For this graph, $b_1(S) = 270$.



(a) A typical S_i (i=1,...,27). (b) S with some omitted members.

Fig. 6.17 A space frame S.

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 Figure 6.18. 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 cycles,

C_7 = (1,2,10,5,6,1), C_8 = (2,3,4,5,10,2) 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. [114].

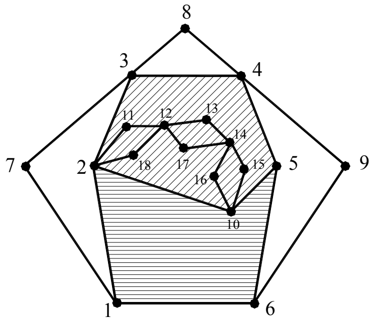
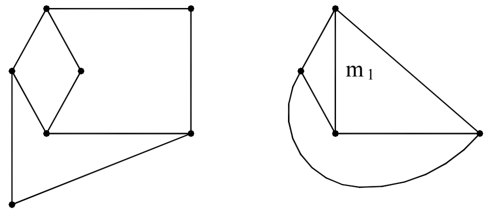


Fig. 6.18 A planar graph S.

6.8.2.1 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. $b_1(S) = b_1(S')$. As an example, S is contracted to S' in Figure 6.19.



(a) S. (b) S'.

Fig. 6.19 S and its contracted graph S'.

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 $\mathbf{CC}^t$.

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, respectively. 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 $q' = q+l$ (l 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,

$$I_{jk} = p' + 1 \quad \text{and} \quad I_{jk}^{c} = (q + p'q') / (1 + p'), \tag{6-21}$$

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 with the smallest IN, 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.

6.8.2.2 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 α+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 α+1 followed by the formation of possible α-sided cycles. This process is repeated until no admissible cycles of length α and α+1 can be found. Denote the generated cycles by C^n.

Step 4: Select an admissible cycle C_{n+1} of length α+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 α+1 can be generated. Repeat Step 3 until no cycles of length α, α+1 or α+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 slightly inferior to those of the direct method A.

6.8.2.3 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 **D**.

Example 1: Consider S as the graph model of a space frame with $b_1(S) = 12$, as shown in Figure 6.20. Hence, 12 independent cycles should be selected as a basis. Algorithm 3 selects a minimal cycle basis containing the following cycles:

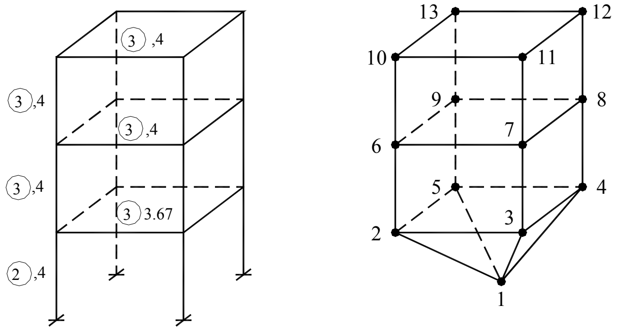
C_1=(1,2,3), C_2=(1,2,5), C_3=(1,3,4), C_4=(1,5,4), C_5=(2,3,6,7),
C_6=(3,4,7,8), C_7=(4,5,8,9), C_8=(6,7,8,9), C_9=(7,8,11,12),
C_{10}=(6,7,10,11), C_{11}=(9,8,12,13), C_{12}=(10,11,12,13),

which corresponds to:

$$\chi(\mathbf{C}) = 4\times3+8\times4 = 44,$$

and

$$\chi(\mathbf{D}) = 12+2\times23 = 58.$$



(a) A space structure. (b) The graph model S of the structure.

Fig. 6.20 A space frame, and CLNs and INs of its members.

Using Algorithm A leads to the formation of a similar basis with the difference that C'_8=(6,9,10,13) is generated in place of C_8=(6,7,8,9), which leads to:

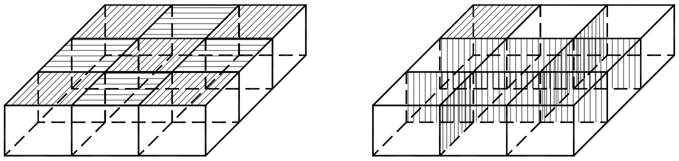
$$\chi(\mathbf{C}') = 4\times3+8\times4 = 44,$$

$$\chi(\mathbf{D}') = 12+2\times20 = 52.$$

The CLNs and INs of the members used in this algorithm are illustrated in Figure 6.20(a).

Example 2: In this example S is a space structure with $b_1(S) = 33$, as shown in Figure 6.21(a). Both Algorithms 3 and A, select 33 cycles of length 4, i.e. a minimal cycle basis with $\chi(\mathbf{C}) = 4\times33=132$ is obtained. The basis selected by Algorithm 3, contains (in the worst case) all 3-sided cycles of S, except those which are shaded in Figure 6.21(a), with $\chi(\mathbf{D}) = 233$.

Algorithm A selects all 3-sided cycles of S except those shaded in Figure 6.21(b), with $\chi(\mathbf{D})=190$. It will be noticed that, for structures containing nodes of higher degrees, considerable improvement is obtained by the use of Algorithm A.



(a) A minimal cycle basis. (b) A suboptimal cycle basis.

Fig. 6.21 Minimal and suboptimal cycle bases of S.

Example 3: Consider a space frame as shown in Figure 6.22, for which $b_1(S)=10$. The minimal cycle basis selected by Algorithm 3, consists of,

$$C_1= (1,2,3),\ C_2= (4,5,6),\ C_3= (7,8,9),\ C_4= (10,11,12),$$

$$C_5= (1,2,5,4),\ C_6= (2,3,6,5),\ C_7= (4,5,8,7),\ C_8= (5,6,9,8)\ ,$$

$$C_9= (7,8,11,10),\ C_{10}= (8,9,12,11),$$

corresponding to $\chi(\mathbf{C}) = 4\times3+6\times4 = 36$ and $\chi(\mathbf{D}) = 10+2[0+0+0+2+3+3+ 4+3+4] = 10+2\times19 = 48$.

However, the following non-minimal cycle basis has a higher sparsity for **C**, and leads to a more sparse **D**,

$$C_1= (1,2,3),\ C_2= (1,2,5,4),\ C_3= (2,3,6,5),\ C_4= (1,3,6,4),$$

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),

for which $\chi(C') = 2\times3+8\times4 = 38$ corresponding, to $\chi(\mathbf{D}') = 10+2[1+2+3+1+2+3+1+2+2] = 10+2\times17 = 44$.

The above cycle basis can further be improved by selecting the following basis:

C_1= (1,2,3), C_2= (1,2,4,5), C_3= (2,3,5,6), C_4= (4,5,6),

C_5= (4,6,7,9), C_6= (5,6,8,9), C_7= (7,8,9), C_8= (7,8,10,11),

C_9= (7,9,10,12), C_{10}= (10,11,12),

for which $\chi(\mathbf{C}') = 2\times3+8\times4 = 38$ corresponding to $\chi(\mathbf{D}') = 10+2[1+2+2+1+2+2+1+3+2] = 10+2\times16 = 42$.

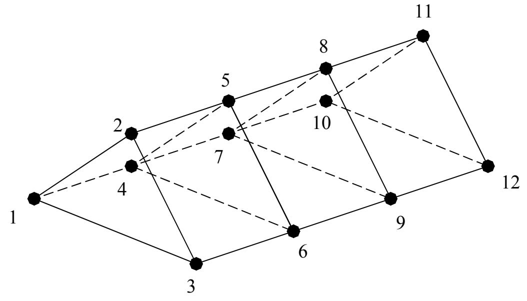


Fig. 6.22 A space frame S.

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

6.8.3 FORMATION OF $\mathbf{B}_0$ AND $\mathbf{B}_1$ MATRICES

In order to generate the elements of a $\mathbf{B}_0$ 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 Figure 6.23(a), two SR subtrees are generated, Figure 6.23(b).

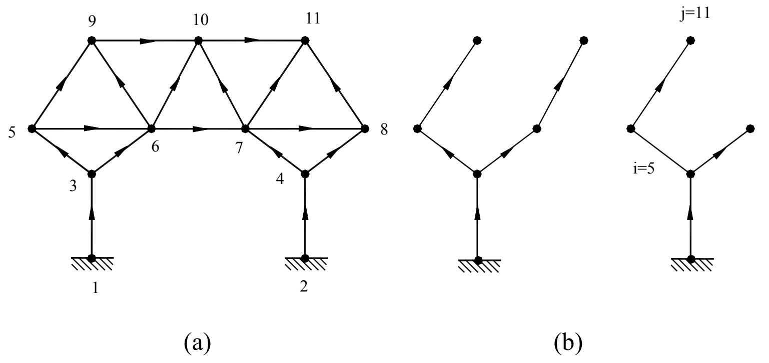


Fig. 6.23 S and two of its SR subtrees.

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 $\mathbf{B}_0$: This is a 6M(S)×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.

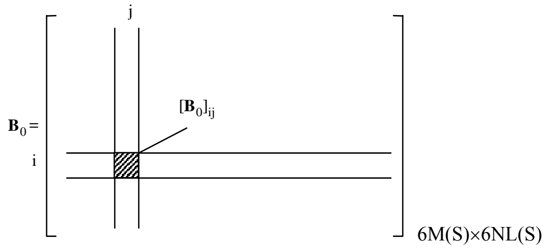
The coefficients of $\mathbf{B}_0$ can be obtained by considering the transformation of each joint load to the ground node of the corresponding subtree. $[\mathbf{B}_0]_{ij}$ for member i and node j is given by a 6×6 submatrix as,

$$[\mathbf{B}_0]_{ij} = \alpha_{ij}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -\Delta z & \Delta y & 1 & 0 & 0 \\ \Delta z & 0 & -\Delta x & 0 & 1 & 0 \\ -\Delta y & \Delta x & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6-22}$$

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 Δ_{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:



(6-23)

MATRIX $\mathbf{B}_1$: This is a $6M(S)\times 6b_1(S)$ 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 Figure 6.24.

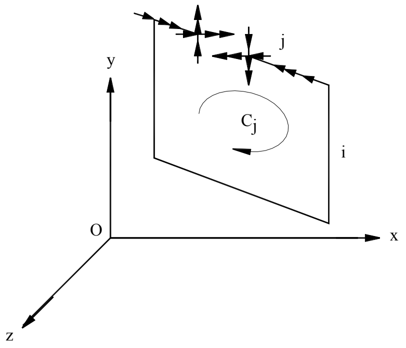


Fig. 6.24 A cycle and the considered bi-action at a cut.

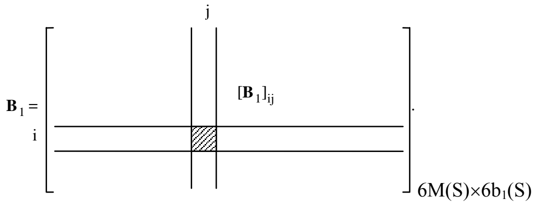
The internal forces under the application of each bi-action are a self-equilibrating stress system As for the matrix $\mathbf{B}_0$, a submatrix $[\mathbf{B}_1]_{ij}$ of $\mathbf{B}_1$ is a 6×6 submatrix, the columns of which show the internal forces at the lower numbered end of member i under the application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

$$[\mathbf{B}_1]_{ij} = \beta_{ij} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -\Delta z & \Delta y & 1 & 0 & 0 \\ \Delta z & 0 & -\Delta x & 0 & 1 & 0 \\ -\Delta y & \Delta x & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (6\text{-}24)$$

in which Δx, Δy and Δz are the differences of the coordinates x, y and z of the beginning node of the generator j and the beginning node of the member i. The orientation coefficient Δ_{ij} is defined as:

$$\beta_{ij} = \begin{cases} +1 \text{ if member i has the same orientation of the cycle generated on j,} \\ -1 \text{ if member i has the reverse orientation of the cycle generated on j,} \\ 0 \text{ if member i is not in the cycle whose generator is j.} \end{cases}$$

The pattern of $\mathbf{B}_1$ containing $[\mathbf{B}_1]_{ij}$ submatrices is shown in the following:



(6-25)

Subroutines for the formation of $\mathbf{B}_0$ and $\mathbf{B}_1$ matrices are included in the program presented in Ref. [112].

Example 1: A four by four planar frame is considered as shown in Figure 6.25.

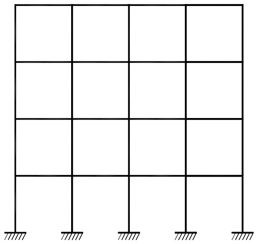
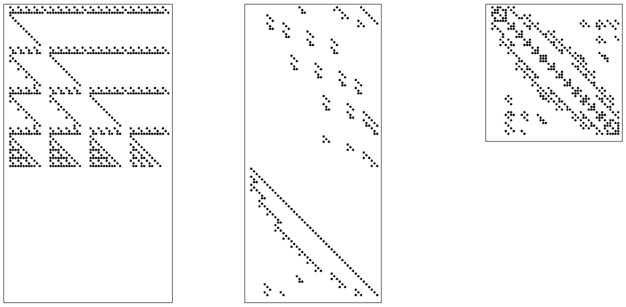


Fig. 6.25 A four by four planar frame S.

The patterns of $\mathbf{B}_0$, $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ formed on selected SRT and the elements of the cycle basis selected by any of the methods of the previous section are depicted in Figure 3.26, corresponding to $\chi\,(\mathbf{B}_1) = 241$ and $\chi\,(\mathbf{B}_1^t\mathbf{B}_1) = 388$.



(a) Pattern of $\mathbf{B}_0$ (b) Pattern of $\mathbf{B}_1$. (c) Pattern of $\mathbf{B}_1^t\mathbf{B}_1$.

Fig. 6.26 Patterns of $\mathbf{B}_0$, $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ matrices for S.

Example 2: A one-bay three-storey frame is considered as shown in Figure 6.27.

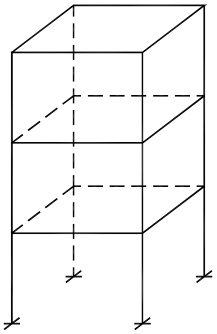
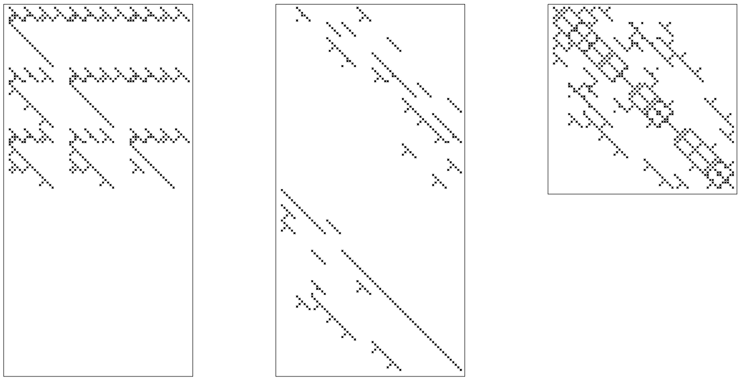


Fig. 6.27 A simple space frame S.

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-theretical algorithms of the previous section are shown in Figure 6.28, corresponding to $\chi\,(\mathbf{B}_0) = 478$, $\chi\,(\mathbf{B}_1) = 310$ and $\chi\,(\mathbf{B}_1^t\mathbf{B}_1)$ =562.



(a) Pattern of $\mathbf{B}_0$. (b) Pattern of $\mathbf{B}_1$. (c) Pattern of $\mathbf{B}_1^t\mathbf{B}_1$.

Fig. 6.28 Patterns of $\mathbf{B}_0$, $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ matrices for S.

The rest of the operations are routine work in the matrix analysis of structures. The interested reader may refer to standard textbooks such as those of McGuire and Gallagher [172], Prezemienecki [197], or Meek [173].

6.9 FORCE METHOD FOR THE ANALYSIS OF PIN-JOINTED PLANAR TRUSSES

The methods described in Section 6.8 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. [108]. 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.

6.9.1 ASSOCIATE GRAPHS FOR SELECTION OF A SUBOPTIMAL GCB

Let S be the model of a planar truss with triangulated panels, as shown in Figure 6.29. 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.

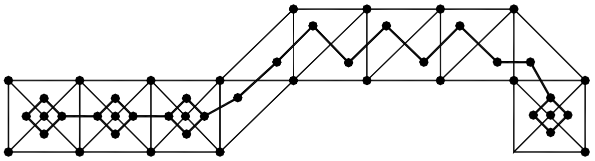


Fig. 6.29 A planar truss S and its associate graph A(S).

If S has some cut-outs, as shown in Figure 6.30, 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.

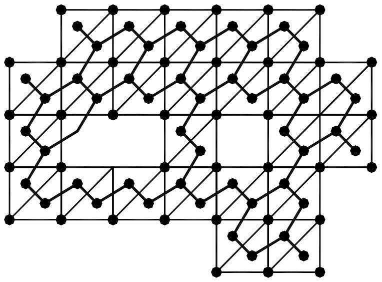


Fig. 6.30 S with two cut-outs and its A(S).

Theorem A: For a fully triangulated truss (except for the exterior boundary), as in Figure 6.30, the dimension of a statical basis γ(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.

$$\gamma(S) = N_i(S) = b_1[A(S)]. \qquad (6\text{-}26)$$

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:

$$b_1[A(S)] = N_i(S).$$

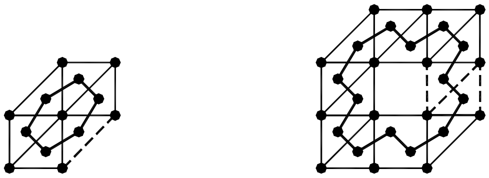
For trusses which are not fully triangulated, as described in Chapter 2, we have:

$$\gamma(S) = N_i(S) - M_c(S). \qquad (6\text{-}27)$$

A Cycle of A(S) and the Corresponding γ-Cycle of S: In Figure 6.31(a), 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. Typical γ-cycles of S are shown by continuous lines, and their γ-chords are depicted in dashed lines.



(a) A type C_I cycle.

(b) A type C_{III} cycle.

Fig. 6.31 Two different types of cycle.

Figure 6.38(b) shows a truss unit with one cut-out. In general, if a cut-out is an m-gone, then the completion of the triangulation requires m–3 members. Instead, m internal nodes will be created, increasing the DSI by m. Hence Eq. (6-27) 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 Figure 6.31(b). 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 3-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 Figure 6.30, for which γ(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 + 3×2 = 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.

6.9.2 A BIPARTITE GRAPH FOR SELECTION OF A SUBOPTIMAL GCB

Let S be a planar truss consisting of rectangular panels with or without diagonal members, an example of which is shown in Figure 6.32(a). The bipartite graph B(S) of S is constructed as follows:

Associate one with each row of panels and denote them by $r_1, r_2, \ldots, r_m$. Similarly, with each column of panels, associate one node and denote them by $c_1, c_2, \ldots, c_n$, as illustrated in Figure 6.32(b). Connect r_i to c_j if the corresponding panel in S has a diagonal member. The graph obtained in this manner is called the *bipartite graph* B(S) of S.

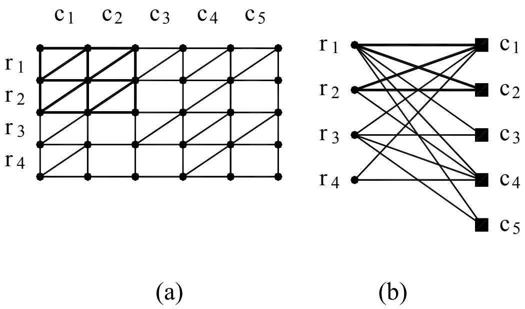


Fig. 6.32 A graph S and its bipartite graph B(S).

Theorem B: The degree of statical indeterminacy of a planar truss S, is the same as the first Betti number of its bipartite graph B(S), i.e.

$$\gamma(S) = b_1[B(S)]. \tag{6-28}$$

Proof: For a truss of rectangular shape with rectangular panels and without diagonal members, the number of members $\overline{M}$ and nodes $\overline{N}$ are as:

$$\overline{M} = [4+3(n-1)]+(m-1)[4+3(n-1)-n] = m+2mn+n,$$

$$\overline{N} = (m+1)(n+1).$$

For M' extra diagonal members, $\overline{M}$ is increased by M(B(S)), which is the same as the number of members of B(S).

By definition, the number of nodes of B(S) is N(B(S)) = m+n,

and M(S) = m+2mn+n+M(B(S) = $\overline{N}$ = mn+m+n+1

Therefore:

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

$$= m + 2mn + n + M(B(S)) - 2(mn + m + n + 1) + 3$$

$$= M(B(S)) - N(B(S)) + 1 = b_1[B(S)].$$

A Cycle of B(S) and the Corresponding γ-Cycle of S: A cycle of B(S) corresponds to a subgraph of S which contains a γ-cycle. In Figure 6.32, a cycle of B(S) and its corresponding subgraph containing a γ-cycle of S are illustrated in bold lines. For this formation, r_1 and r_2 are specified as r_{min} and r_{max}. All c_is which have links with two r_ks with $r_{min} \leq r_k \leq r_{max}$, should be considered in the formation of the corresponding γ-cycle. For this cycle, only c_1 and c_2 have such properties.

Therefore, the subgraph of S situated in row r_1 and r_2, and columns c_1 and c_2 are those which contain a γ-cycle. For simplicity, a cycle and its γ-cycle are denoted by,

$$C_i(r_1,r_2,c_1,c_2) \text{ and } \gamma\text{-}C_i(r_1,r_2,c_1,c_2), \tag{6-29}$$

and for $C_j(r_3,r_4,c_1,c_4)$, we have $\gamma - C_j(r_3,r_4,c_1,c_2,c_3,c_4)$.

This means that a γ-cycle can be formed on a subgraph of S, which contains row 3 and 4, and columns 1, 2, 3 and 4 of S.

General Cases: For an irregularity in the pattern of the boundary, as that of Figure 6.33, the above method can still be used. The additional part contains $\overline{M} = 2\overline{N}$ members, leaving γ(S) = M(S) − 2N(S)+3 unchanged. On the other hand, M(B(S)) and N(B(S)) are not affected. Hence b_1[B(S)] = M (B(S)) − N(B(S)) + 1 is unaltered, and γ(S) = b_1[B(S)] still holds.

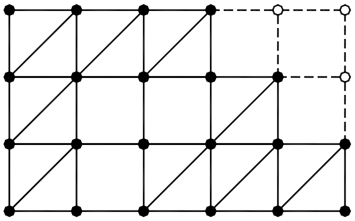


Fig. 6.33 A truss model S with irregular boundary.

Algorithm BB

Step 1: Construct the bipartite graph B(S) of S.

Step 2: Select a subminial cycle basis of B(S).

Step 3: Find the subgraphs of S corresponding to the cycle of the selected basis of B(S) at Step 2.

Now one S.E.S. can be formed on each subgraph of S containing a γ-cycle of S.

For localizing the S.E.Ss, the cycle for which $|r_{min} - r_{max}| + |c_{max} - c_{min}|$ has the smallest value should be chosen. The value $|c_{max} - c_{min}|$ can be obtained once r_{max} and r_{min} are determined. Due to the simplicity of the structure of B(S), the formation of cycle of short lengths leading γ-cycles of S becomes almost trivial.

Example: Let S be the graph model of a planar truss, as shown in Figure 6.32(a). The bipartite graph B(S) of S is constructed as in Figure 6.32(b). For this structure,

$$\gamma(S) = N_i(S) - M_c(S) = 12 - 6 = 6,$$

and six γ-cycles are selected. The following cycles of B(S) and the corresponding subgraphs of S containing the elements of a GCB are obtained:

$$C_1(r_1, r_2, c_1, c_2) \qquad \gamma - C_1(r_1, r_2, c_1, c_2)$$

$$C_2(r_1, r_2, c_1, c_4) \qquad \gamma - C_2(r_1, r_2, c_1, c_2, c_3, c_4)$$

$$C_3(r_2, r_3, c_1, c_4) \qquad \gamma - C_3(r_1, r_3, c_1, c_2, c_3, c_4)$$

$$C_4(r_3, r_4, c_1, c_4) \qquad \gamma - C_4(r_3, r_4, c_1, c_2, c_3, c_4)$$

$$C_5(r_1, r_3, c_1, c_5) \qquad \gamma - C_5(r_1, r_2, c_3, c_1, c_2, c_3, c_4, c_5)$$

$$C_6(r_1, r_3, c_3, c_5) \qquad \gamma - C_6(r_1, r_2, c_3, c_3, c_4, c_5)$$

Finally, it should be noted that Algorithm AA is more suitable for triangulated trusses, with or without cut-outs, while Algorithm BB is more appropriate for trusses with rectangular panels, with or without diagonal members.

6.10 FORCE METHOD OF ANALYSIS FOR GENERAL STRUCTURES

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 a general structure, the underlying graph or hypergraph of a self-equilibrating stress system has not yet been properly defined, and much research has still to be done. Algebraic methods, on the other hand, 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 these methods are the larger storage requirements, and the higher number of operations.

These difficulties can partially be overcome by employing combinatorial approaches within the algebraic methods, whenever such tools are available, and their use can lead to some simplifications.

6.10.1 ALGEBRAIC METHODS

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

$$\mathbf{Ar} = \mathbf{p}, \tag{6-30}$$

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

The member forces can be written as

$$\mathbf{r} = \mathbf{B}_0\mathbf{p} + \mathbf{B}_1\mathbf{q}, \tag{6-31}$$

where $\mathbf{B}_0$ is an m×n matrix such that $\mathbf{AB}_0$ is an n×n identity matrix, and $\mathbf{B}_1$ is an m×t matrix such that $\mathbf{AB}_1$ is an n×t zero matrix. $\mathbf{B}_1$ 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 **r** minimize the quadratic form,

$$\tfrac{1}{2}\mathbf{r}^t\mathbf{F}_m\mathbf{r}, \tag{6-32}$$

subject to the constraint as in Eq. (6-30). $\mathbf{F}_m$ is an m×m block diagonal element flexibility matrix. Using Eq. (6-31), it can be seen that **q** must satisfy the following equation,

$$(\mathbf{B}_1^t\,\mathbf{F}_m\mathbf{B}_1)\mathbf{q} = -\mathbf{B}_1^t\,\mathbf{F}_m\mathbf{B}_0\mathbf{p}, \tag{6-33}$$

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 **q** from Eq. (6-33), **r** can be found using Eq. (6-31). The structure of **G**, is again important, and its sparsity, bandwidth and conditioning govern the efficiency of the force method. For the sparsity of **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 **A**. For the moment, let **A** be partitioned so that,

$$\mathbf{AP} = [\mathbf{A}_1 , \mathbf{A}_2], \tag{6-34}$$

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

$$\mathbf{B}_1 = \mathbf{P}\begin{bmatrix} -\mathbf{A}_1^{-1}\mathbf{A}_2 \\ \mathbf{I} \end{bmatrix}. \tag{6-35}$$

$$\mathbf{AB}_1 = \begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}\begin{bmatrix} -\mathbf{A}_1^{-1}\mathbf{A}_2 \\ \mathbf{I} \end{bmatrix} = 0.$$

Obviously, a permutation **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×n identity matrix **I** in the first columns of **A** by column changes and a sequence of n pivots. This procedure can be expressed as,

$$\mathbf{G}_n\mathbf{G}_{n-1} \ldots \mathbf{G}_2\mathbf{G}_1\mathbf{AP} = [\mathbf{I} , \mathbf{M}], \tag{6-36}$$

where G_i is the *i*th pivot matrix and **P** is an m×m column permutation matrix (so $\mathbf{P}^t = \mathbf{P}$) and **I** is an n×n identity matrix, and **M** is an n×t matrix. A pivot matrix $\mathbf{G}_i$ is a diagonal matrix whose *i*th diagonal entry is the *i*th pivot. We denote $\mathbf{G}_n\mathbf{G}_{n-1} \ldots \mathbf{G}_2\mathbf{G}_1$ by **G**, where **G** is a diagonal matrix whose diagonal entries are the scaling constants. Scaling, means simply to perform row operations.

Equation (6-36) can then be written as,

$$\mathbf{GAP} = [\mathbf{I}\ ,\ \mathbf{M}], \tag{6-37}$$

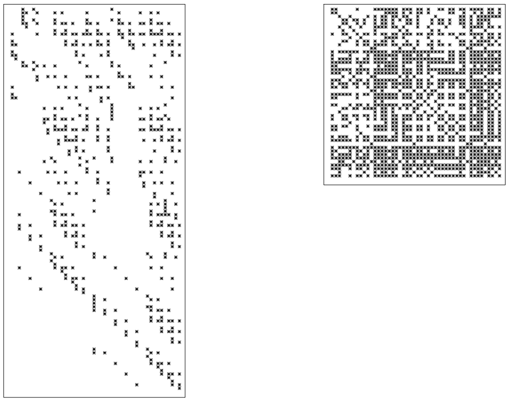
or

$$\mathbf{AP} = \mathbf{G}^{-1}[\mathbf{I}\ ,\ \mathbf{M}] = [\mathbf{G}^{-1}\ ,\ \mathbf{G}^{-1}\mathbf{M}], \tag{6-38}$$

which can be regarded as Gauss-Jordan factorization of **A**, and

$$\mathbf{B}_0 = \overline{\mathbf{P}}\begin{bmatrix}\mathbf{G}\\ \mathbf{0}\end{bmatrix} \quad \text{and} \quad \mathbf{B}_1 = \overline{\mathbf{P}}\begin{bmatrix}-\mathbf{M}\\ \mathbf{1}\end{bmatrix}. \tag{6-39}$$

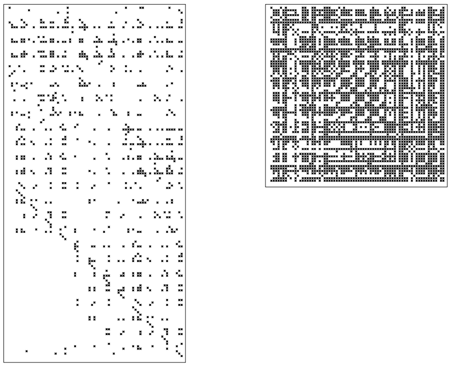
Example 1: The four by four planar frame of Figure 6.25 is reconsidered. The patterns of $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ formed by the Gauss-Jordan elimination method, are depicted in Figure 6.34, corresponding to $\chi(\mathbf{B}_1) = 491$ and $\chi(\mathbf{B}_1^t\mathbf{B}_1) = 1342$.



(a) Pattern of $\mathbf{B}_1$. (b) Pattern of $\mathbf{B}_1^t\mathbf{B}_1$.

Fig. 6.34 Patterns of $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ matrices for S.

Example 2: The three-storey frame of Figure 6.27 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 Figure 6.35, corresponding to $\chi(\mathbf{B}_1) = 483$ and $\chi(\mathbf{B}_1^t\mathbf{B}_1)$ =1592.



(a) Pattern of $\mathbf{B}_1$. (b) Pattern of $\mathbf{B}_1^t\mathbf{B}_1$.

Fig. 6.35 Patterns of $\mathbf{B}_1$ and $\mathbf{B}_1^t\,\mathbf{B}_1$ matrices for S.

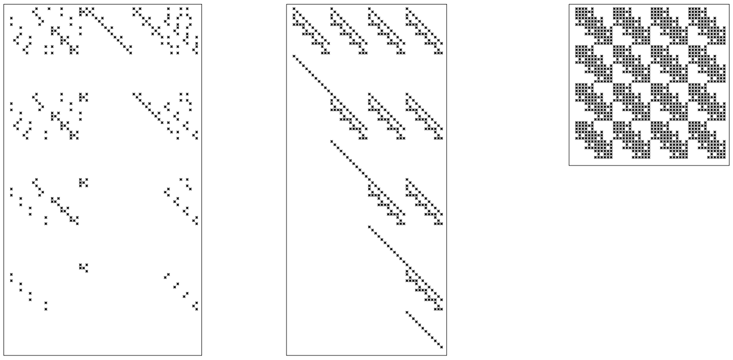
LU Decomposition Method*:* Using the **LU** decomposition method, one obtains the **LU** factorization of **A** as:

$$\mathbf{PA} = \mathbf{LU} \qquad \mathbf{U}\overline{\mathbf{P}} = [\mathbf{U}_1, \mathbf{U}_2], \tag{6-40}$$

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

$$\mathbf{B}_0 = \overline{\mathbf{P}}\begin{bmatrix} \mathbf{U}_1^{-1}\mathbf{L}^{-1}\mathbf{P} \\ \mathbf{0} \end{bmatrix} \text{ and } \mathbf{B}_1 = \overline{\mathbf{P}}\begin{bmatrix} -\mathbf{U}_1^{-1}\mathbf{U}_2 \\ \mathbf{I} \end{bmatrix}. \tag{6-41}$$

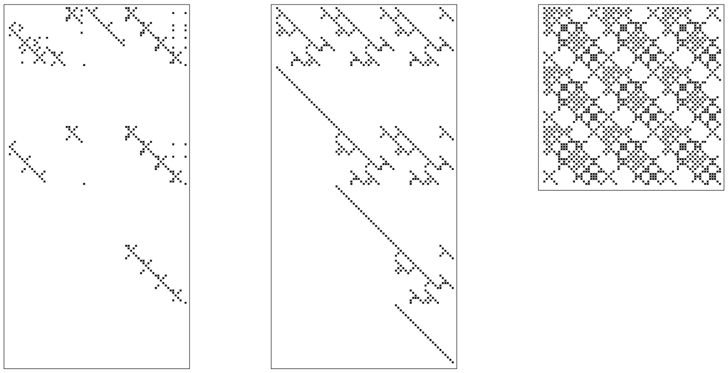
Example 1: The four by four planar frame of Figure 6.25 is re-considered. The patterns of $\mathbf{B}_0$, $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ formed by the **LU** factorization method, are depicted in Figure 6.36. The sparsity for the corresponding matrices are χ ($\mathbf{B}_0$) = 280, χ($\mathbf{B}_1$) = 408 and χ($\mathbf{B}_1^t\mathbf{B}_1$) =1248.



(a) Pattern of $\mathbf{B}_0$. (b) Pattern of $\mathbf{B}_1$. (c) Pattern of $\mathbf{B}_1^t\mathbf{B}_1$.

Fig. 6.36 Patterns of $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ matrices for S.

Example 2: The three-storey frame of Figure 6.27 is re-considered, and the **LU** factorization method is used. The patterns of $\mathbf{B}_0$, $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ matrices formed are shown in Figure 6.37, corresponding to $\chi(\mathbf{B}_0) = 287$, $\chi(\mathbf{B}_1) = 504$ and $\chi(\mathbf{B}_1^t\mathbf{B}_1)$ =1530.



(a) Pattern of $\mathbf{B}_0$. (b) Pattern of $\mathbf{B}_1$. (c) Pattern of $\mathbf{B}_1^t\mathbf{B}_1$.

Fig. 6.37 Patterns of $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ matrices for S.

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

$$\mathbf{AP} = \mathbf{Q}\,[\mathbf{R}_1\,,\,\mathbf{R}_2], \tag{6-42}$$

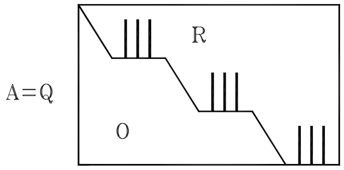
where **P** is again a permutation matrix, and $\mathbf{R}_1$ is an upper triangular matrix of order n. $\mathbf{B}_1$ can be obtained as:

$$\mathbf{B}_1 = \mathbf{P}\begin{bmatrix} -\mathbf{R}_1^{-1}\mathbf{R}_2 \\ \mathbf{I} \end{bmatrix}. \tag{6-43}$$

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. [74] 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.

Write the matrix **A** = $(a_1,a_2,...,a_n)$ by columns. A *start column* is a column such that the ranks of $(a_1,a_2,...,a_{s-1})$ and $(a_1,a_2,...,a_s)$ 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



(6-44)

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**.

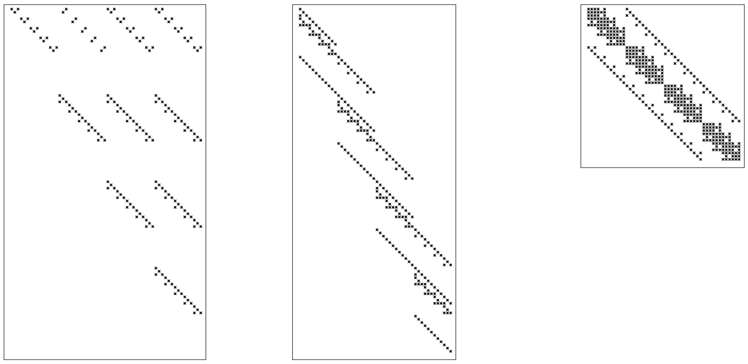
The Turn-back method finds one null vector for each start column a_s by "turning back" from column s to find the smallest k for which columns $a_s,a_{s-1},...,a_{s-k}$ are linearly dependent. The null vector has a non-zero only in position s−k through s. Thus, if k is small for most of the start columns, 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. [238], the basic idea had also been used in Refs. [89,93,94]. Since this correspondence simplifies the understanding of the Turn-back method, it is briefly described in the following.

For the Algorithm 1 of Section 6.8.2, 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 [89,105]. Therefore, the idea used in Algorithm 1 and its generalization for the formation of generalized cycle bases in Refs. [89,94] seem to constitute a similar idea to that of the algebraic Turn-back method.

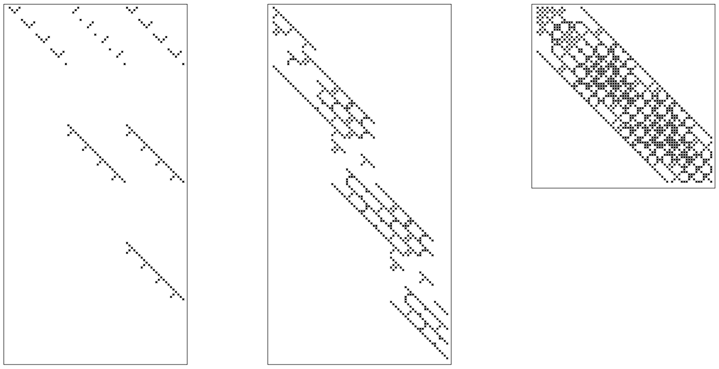
Example 1: The four by four planar frame of Figure 6.25 is re-considered. The patterns of $\mathbf{B}_0$, $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ formed by the Turn-back LU factorization method are depicted in Figure 6.38, corresponding to $\chi(\mathbf{B}_0) = 175$, $\chi(\mathbf{B}_1) = 240$ and $\chi(\mathbf{B}_1^t\mathbf{B}_1) = 408$.



(a) Pattern of $\mathbf{B}_0$ (b) Pattern of $\mathbf{B}_1$. (c) Pattern of $\mathbf{B}_1^t\mathbf{B}_1$.

Fig. 6.38 Patterns of $\mathbf{B}_0$, $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ matrices for S.

Example 2: The three-storey frame of Figure 6.27 is re-considered, and the Turn-back LU factorization method is used. The patterns of $\mathbf{B}_0$, $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ matrices formed are shown in Figure 6.39, corresponding to $\chi(\mathbf{B}_0) = 160$, $\chi(\mathbf{B}_1) = 476$ and $\chi(\mathbf{B}_1^t\mathbf{B}_1) = 984$.



(a) Pattern of $\mathbf{B}_0$ (b) Pattern of $\mathbf{B}_1$. (c) Pattern of $\mathbf{B}_1^t\mathbf{B}_1$.

Fig. 6.39 Patterns of $\mathbf{B}_0$, $\mathbf{B}_1$ and $\mathbf{B}_1^t\mathbf{B}_1$ matrices for S.

A comparative study of various force methods has been made in Ref. [112].

Many algorithms have been developed for selection of null bases, and the interested reader may refer to Refs. [29,58,87,196,238].

6.11 OPTIMAL PLASTIC ANALYSIS AND DESIGN OF FRAMES

The problem of plastic analysis and design of rigid-jointed skeletal structures was cast in the form of a mathematical programming with a linear objective function subjected to linear constraint by Charnes et al. [23], as early as 1951. Further progress in this field is due to Baker and Heyman [5], Horne [81], Thierauf [233], Tam [230], Mokhtar-zadeh and Kaveh [176] among many others. Considerable progress has been made in the last decades; a complete list of references may be found in Munro [177], and Livesley [160].

In this section, efficient programs are developed for optimal safe plastic analysis and design of frames. These problems are formulated in linear programming form, and graph-theretical approaches are used for the formation of highly sparse equilibrium equations as constraints of the optimisation approach.

6.11.1 FORMULATION

The following assumptions are made to simplify the formulation of the plastic analysis and design of frames:

1. Equilibrium equations are referred to the un-deformed geometry of the structure.

2. Plastic hinges are considered at critical section, with unlimited ductility.

3. The loads are assumed to increase proportionally.

4. Constraints are related only to bending moment yield conditions, and to the design considerations.

5. The effect of axial forces and shear forces is neglected.

For a static approach, the distribution of the internal forces at critical sections can be written as:

$$\mathbf{r} = \mathbf{B}_0\mathbf{p} + \mathbf{B}_1\mathbf{q}, \tag{6-45}$$

Consider a two-storey frame as shown in Figure 6.40 for which $\gamma(S) = 6$, and the number of critical sections is 11, as marked.

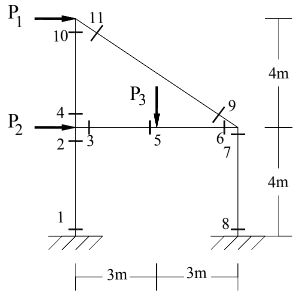


Fig. 6.40 A two-storey frame and its critical sections.

Cuts can be introduced at the end of two beams, to transform the model into a determinate one, in the form of two subtrees. The distribution of moments at the critical sections (M_1, M_2, … , M_{11}) of these subtrees, can then be calculated under the unit values of the applied loads P_1, P_2, P_3 and bi-actions R_1, R_2, R_3, R_4, R_5 and R_6.

Using Eq. (6-45), the bending moments corresponding to internal loads can be expressed as:

$$\mathbf{M}_r = \mathbf{M}_0\mathbf{p} + \mathbf{M}_1\mathbf{q}. \tag{6-46}$$

For the considered example, $\mathbf{M}_0$ and $\mathbf{M}_1$ are 11×3 and 11×6 matrices, respectively. In general, the dimensions of these matrices are $c \times l$ and $c \times k$, respectively, where c, l and k are the number of critical sections, number of applied load components and the number of redundants of the structure, respectively.

6.11.2 OPTIMAL SAFE PLASTIC ANALYSIS OF FRAMES

In the safe analysis of frames, the least value of the load factor λ should be found subjected to the equilibrium and yield conditions. Consider $\mathbf{M}_p$ as a $c \times l$ vector containing the full plastic moments of the members at critical sections. The mathematical formulation of the problem can be stated as

$$\text{Maximize } \lambda$$

subject to

$$\left|\lambda(\mathbf{M}_0\mathbf{P}+\mathbf{M}_{\mathrm{fe}})+\mathbf{M}_1\mathbf{X}\right| \leq \mathbf{TM}_{\mathrm{p}}, \tag{6-47}$$

where **T** is a Boolean matrix consisting of 0 and 1 entries. The $t_{ij}=1$ if the *j*th plastic moments has to be considered in limiting the bending moment at section i, and $t_{ij}=0$ otherwise. $\mathbf{M}_{\mathrm{fe}}$ contains the fixed end moments, added due to the presence of span loads.

Equation (6-47) can be written as:

$$-\mathbf{TM}_{\mathrm{p}} \leq \lambda(\mathbf{M}_0\mathbf{P}+\mathbf{M}_{\mathrm{fe}})+\mathbf{M}_1\mathbf{X} \leq \mathbf{TM}_{\mathrm{p}}. \tag{6-48}$$

For unrestricted variables **X**, the following transformation is considered to obtain non-negative optimisation variables $\overline{\mathbf{X}}$:

$$\overline{\mathbf{X}} = \mathbf{X}+\mathbf{\Omega}, \tag{6-49}$$

in which Ω is a large positive number. Therefore:

$$-\mathbf{TM}_{\mathrm{p}} \leq \lambda(\mathbf{M}_0\mathbf{P}+\mathbf{M}_{\mathrm{fe}})+\mathbf{M}_1(\overline{\mathbf{X}}-\Omega) \leq \mathbf{TM}_{\mathrm{p}}. \tag{6-50}$$

Denoting,

$$\mathbf{M}_{\mathrm{a}} = \mathbf{M}_0\mathbf{P} + \mathbf{M}_{\mathrm{fe}}, \tag{6-51}$$

the constraints become,

$$\begin{bmatrix} \mathbf{M}_{\mathrm{a}} & \mathbf{M}_1 \\ -\mathbf{M}_{\mathrm{a}} & -\mathbf{M}_1 \end{bmatrix} \begin{bmatrix} \lambda \\ \overline{\mathbf{X}} \end{bmatrix} \leq \begin{bmatrix} \mathbf{M}_{\mathrm{b}} \\ \mathbf{M}_{\mathrm{c}} \end{bmatrix}, \tag{6-52}$$

with

$$\overline{\mathbf{X}} \geq 0,$$

and

$$\mathbf{M}_{\mathrm{b}} = \mathbf{TM}_{\mathrm{p}} + \mathbf{\Omega M}_1,$$

$$\mathbf{M}_{\mathrm{c}} = \mathbf{TM}_{\mathrm{p}} + \mathbf{\Omega M}_1.$$

The dimension of the coefficient matrix of the constraints is 2c×(ρ+1).

6.11.3 OPTIMAL SAFE PLASTIC DESIGN

For optimal design, if the model contains a single prismatic member of design variable d_1, then clearly the design objective could be to minimize subject to the imposed constraints. If two prismatic members of lengths L_1 and L_2 were being fixed throughout design variables d_1 and d_2, then the importance of the design variables would be controlled by those lengths. In deciding how best to allocate structural material between the two members, one could weight the design variables by their lengths, and this suggests a convenient linearized objective function Z as

$$Z = L_1 d_1 + L_2 d_2. \tag{6-53}$$

This can be generalized to a structure with g design variables as,

$$Z = \sum_{i=1}^{g} L_i d_i, \tag{6-54}$$

where L_i is the length of all the members with design variable d_i, and g is the total number of design variables (groups), respectively.

It is justified that the minimization of Z can approximately be achieved by considering a linearized objective function W as the total weight of the structure as,

$$W = \sum_{i=1}^{g} L_i M_{pi}, \tag{6-55}$$

where M_{pi} is the plastic moment of a member of group i. The above function can be written in matrix form as

$$W = \mathbf{L}^t \mathbf{M}_p. \tag{6-56}$$

Now the problem of optimal plastic design can be stated as:

Minimize W

subject to:

$$\left|\lambda(\mathbf{M}_0\mathbf{P} + \mathbf{M}_{fe}) + \mathbf{M}_1\mathbf{X}\right| \le \mathbf{T}\mathbf{M}_p. \tag{6-57}$$

Since the redundants may have negative values, therefore the following transformation is performed on Eq. (6-57) to avoid such negative values in the mathematical programming:

$$\mathbf{Y} = \boldsymbol{\Omega}+\mathbf{X},$$

$$-\mathbf{TM}_\mathrm{p} \leq \lambda(\mathbf{M}_0\mathbf{P}+\mathbf{M}_\mathrm{fe})+\mathbf{M}_1(\mathbf{Y}-\boldsymbol{\Omega}) \leq \mathbf{TM}_\mathrm{p}. \tag{6-58}$$

Denoting,

$$\mathbf{M}_\mathrm{d} = (\mathbf{M}_0\mathbf{P}+\mathbf{M}_\mathrm{fe}-\boldsymbol{\Omega}\mathbf{M}_1). \tag{6-59}$$

the constraints in matrix form become,

$$\begin{bmatrix} \mathbf{M}_1 & -\mathbf{T} \\ -\mathbf{M}_1 & -\mathbf{T} \end{bmatrix} \begin{bmatrix} \mathbf{Y} \\ \mathbf{M}_\mathrm{p} \end{bmatrix} \leq \begin{bmatrix} -\mathbf{M}_\mathrm{d} \\ \mathbf{M}_\mathrm{d} \end{bmatrix}, \tag{6-60}$$

with

$$\mathrm{Y} \geq 0 \text{ and } \mathbf{M}_\mathrm{p} \geq 0. \tag{6-61}$$

The dimension of the coefficient matrix of the constraints is 2c×(ρ+g).

6.11.4 FORMATION OF $\mathbf{M}_0$ AND $\mathbf{M}_1$ MATRICES

A cycle basis is first selected, using the any algorithm of Section 6.8.2.1. The generators of the selected cycles are then cut to transform the graph model into a spanning forest. The structure corresponding to this forest is statically determinate and can be used as a primary structure, to form the particular solution. This forest transfers the joint and span loads to the ground tree nodes.

The orientation assigned to each member, is taken from the first end (lower numbered node) to the last end (higher numbered node). For each subtree of the forest, the orientation is in the direction of its natural expansion (growth) from its ground node (root).

The moment produced at the end of member i, when load at joint j is applied, is given as,

$$\mathbf{M}_{0ij} = \begin{bmatrix} \mathbf{M}_{0ij}^{1} \\ \mathbf{M}_{0ij}^{2} \end{bmatrix} = \alpha_{ij} \begin{bmatrix} \delta_{y1} & -\delta_{x1} & 1 \\ \delta_{y2} & -\delta_{x2} & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ m_z \end{bmatrix}, \tag{6-62}$$

where p_x, p_y and m_z are the components of the applied load on node j, and δ_x and δ_y are the difference in coordinates x and y of the two ends of the member i from node j, respectively. The indices 1 and 2 show the first end node and the last end node of the corresponding parameters, and α_{ij} is defined as Section 6.8.3.

The position of $\mathbf{M}_{0ij}$ in $\mathbf{M}_0$ is similar to that of the position of $\mathbf{B}_{0ij}$ in $\mathbf{B}_0$ according to Eq. (6-23).

For the presence of span loads, the effect of the transferred bending moments at their corresponding critical sections is also considered.

In order to form the $\mathbf{M}_1$ matrix, every cycle is oriented as its generator, and cut at the first end of its generator. Bi-actions are then applied at the two cut ends. The bending moments at the first and last ends of member i, due to bi-actions at generator j, are given as

$$\mathbf{M}_{1ij} = \begin{bmatrix} \mathbf{M}_{1j}^{1} \\ \mathbf{M}_{1ij}^{2} \end{bmatrix} = \beta_{ij} \begin{bmatrix} \delta_{y1} & -\delta_{x1} & 1 \\ \delta_{y2} & -\delta_{x2} & 1 \end{bmatrix} \begin{bmatrix} X_x \\ X_y \\ X_z \end{bmatrix}, \tag{6-63}$$

in which X_x, X_y and X_z are the components of bi-actions on the first end of the generator j, and δ_x and δ_y are the difference in coordinates x and y of the two end nodes of member i from the first end of the generator j, respectively. β_{ij} is the orientation factor similar to Eq. (6-24).

The position of $\mathbf{M}_{1ij}$ in $\mathbf{M}_1$ is similar to that of the position of $\mathbf{B}_{1ij}$ in $\mathbf{B}_1$ according to Eq. (6-25).

For extra critical sections under span loads, the effects of redundants at these sections should also be calculated.

6.11.5 STANDARD MATHEMATICAL PROGRAMMING FORMULATION

In order to transform the problems of optimal plastic analysis and design to standard forms, the inequalities involved should be transformed to equalities. For this purpose some slack variables Y_s and Z_s are introduced. For plastic analysis, variables Y_s are add to inequalities, and

$$\begin{bmatrix} \mathbf{M}_a & \mathbf{M}_l & \mathbf{A}_l \\ -\mathbf{M}_a & -\mathbf{M}_l & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \frac{\lambda}{\mathbf{X}} \\ \mathbf{Y}_s \end{bmatrix} = \begin{bmatrix} \mathbf{M}_b \\ \mathbf{M}_c \end{bmatrix} \tag{6-64}$$

is obtained. For optimal plastic design, variables Z_s are added to inequalities and hence the following form is obtained:

$$\begin{bmatrix} \mathbf{M}_a & -\mathbf{T} & \mathbf{B}_l \\ -\mathbf{M}_a & -\mathbf{T} & \mathbf{B}_2 \end{bmatrix} \begin{bmatrix} \lambda \\ \mathbf{M}_p \\ \mathbf{Z}_s \end{bmatrix} = \begin{bmatrix} -\mathbf{M}_d \\ \mathbf{M}_d \end{bmatrix}. \tag{6-65}$$

In the mathematical programming problems, constraints corresponding to Eq. (6-52) and Eq. (6-60), the right-hand constants must be positive. Therefore in general form, the conditions ≤ and ≥, four inequalities may result. Transformation of these conditions to a standard form of LP problem requires slack and artificial variables to be considered in the left-hand side of the constraints. In the above equations $\mathbf{A}_1$ and $\mathbf{A}_2$ or $\mathbf{B}_1$ and $\mathbf{B}_2$ are matrices having +1 and −1 entries depending on a typical entry (i,j) of these matrices, corresponding to *j*th slack or artificial variable, respectively, and zero otherwise.

Once the standard forms are obtained, any program on simplex method can be used for the solution.

6.11.6 NUMERICAL RESULTS

Example 1: A two-storey frame is considered as shown in Figure 6.41. Four groups of plastic moments are taken as design variables. An optimal design is performed and the following results are obtained:

For the columns of the first floor M_{p1}=3.0kN.m, and columns of the second floor M_{p2}=1.5kN.m. For beams of the first and second floor, M_{p3}=4.5kN.m and M_{p4}=1.5kN.m are selected, respectively.

The total minimum weight is calculated as 51.0kN.m^2.

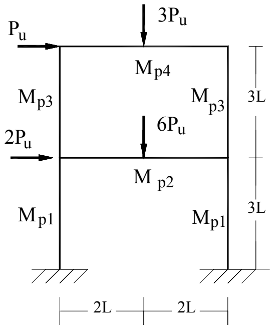


Fig. 6.41 A two-storey frame.

Example 2: A two-span frame is considered as shown in Figure 6.42. Four groups of plastic moments are taken as design variables. An optimal design is performed and the following results are obtained:

For the columns M_{p3}=27kN.m, and for beams M_{p1}=77kN.m and M_{p2}=59kN.m, are selected, respectively. The total minimum weight is calculated as 1059 kN.m^2.

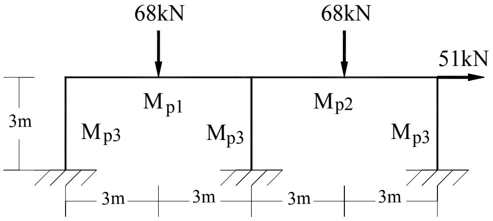


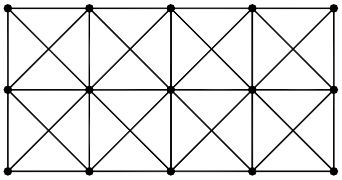
Fig. 6.42 A two-span frame.

The methods presented in this section are simple and efficient and require exactly the same data as a kinematic approach. These methods are applicable to structures of arbitrary configuration, and can easily be modified for structures with non-prismatic members.

EXERCISES

6.1 For each set of integers a, b and c of Table 2.1, draw an arbitrary γ-tree and a γ-cycle.

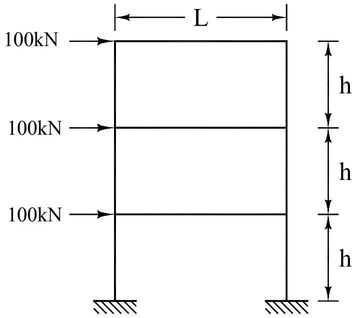
6.2 Construct a γ-tree for the following graph when it is viewed as the graph model of a planar truss:



6.3 In Exercise 4.2, select a fundamental γ-cycle basis of S and form its γ-cycle adjacency matrix.

6.4 Find a graph for which Algorithm 3 fails to select a minimal cycle basis. Repeat this exercise for Algorithm 2.

6.5 Form $\mathbf{B}_0$ and $\mathbf{B}_1$ matrices, by selecting a suitable SRT and cycle basis for the following planar frame:



6.6 Form $\mathbf{B}_0$ and $\mathbf{B}_1$ for the planar truss of Exercise 4.2, when it is supported in a statically determinate fashion. Choose the support nodes arbitrarily.

6.7 Prove the minimality of the cycle basis selected by Horton´s algorithm.

6.8 Find a counter-example for the minimality of the cycle basis selected by Algorithm 3 for planar structures.

6.9 Why do the regional cycles of a planar graph form a cycle basis (mesh basis)?

6.10 Use a combinatorial method, to generate minimal and suboptimal cycle bases for the following graphs: