CHAPTER 10

A Graph-Theoretical Approach for Configuration Processing

10.1 INTRODUCTION

For a large system, configuration processing is one of the most tedious and time-consuming parts of the analysis. Different methods have been proposed for configuration processing and data generation, among which the formex algebra of Nooshin [180,181] is perhaps the most general and powerful tool for this purpose. Behravesh et al. [9] employed set theory and showed that some concepts of set algebra can be used to build up a general method for describing the interconnection patterns of structural systems. There are many other references on the field of data generation; however, most of them are prepared for specific classes of problem. For example, many algorithms have been developed and successfully implemented on mesh or grid generation, a complete review of which may be found in a paper by Thacker [232] and a book by Thomson et al. [234].

In this chapter, a graph-theoretical approach is presented for configuration processing, which uses similar concepts developed for the set theoretical and formex algebraic methods of Refs. [9,180,181], with the difference that it avoids the use of new terminology, and employs the most elementary definitions of graph theory. Four basic functions are presented here, which consist of translation, rotation, reflection and projection functions. The algebraic representations described here not only provide the topological properties of a structure, but also provide simple means for obtaining the geometrical properties of structures using simple transformations. Examples of skeletal structures are included to illustrate the simplicity of the concepts presented in this chapter. The application of the transformations is extended to the generation of finite element meshes.

10.2 ALGEBRAIC REPRESENTATION OF A GRAPH

Various algebraic representations of graphs were studied in Chapter 1. These consisted of matrices such as the node-member incidence matrix, adjacency matrix, member list, and in particular the following compact member list, containing the end nodes of the members of a graph in a single row.

As an example, the interconnection pattern of a graph S, shown in Figure 10.1 can be represented by,

$$(1,2\ ;\ 1,3\ ;\ 2,3\ ;\ 3,4),$$

or, in a form compatible with the notation used in this chapter:

$$\{[1,2],[1,3],[2,3],[3,4]\}.$$

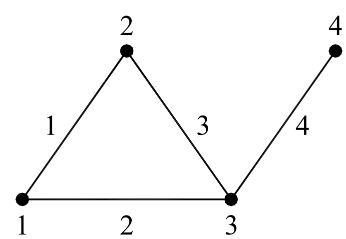


Fig. 10.1 A graph S.

In this scheme, effectively a set of members is specified; however, the common node numbers provide the interconnections of the members.

All these representations specify the corresponding graph up to isomorphism. In structural engineering, however, a structural model should uniquely be represented. Therefore some additional information must be provided. As an example, the coordinates of nodes in Euclidean space can be given. In such a case, however, the geometry of the structure will also be involved. To have only the connectivity, a weaker formulation will be sufficient. For this purpose, one may use an integer coordinate system for specifying the interconnection of S. Further simplicity can be achieved if a rectilinear grid system is employed. Such a system may be a 1, 2 or 3-dimensional coordinate system, an example of which is depicted in Figure 10.2.

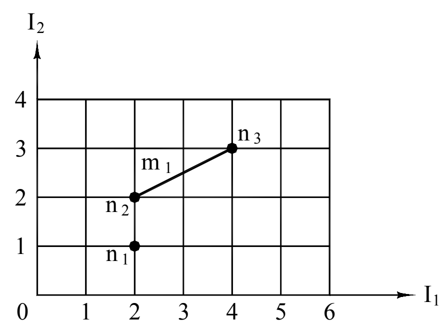


Fig. 10.2 A 2-dimensional rectilinear grid system.

From now on, any member of the set of such systems will be referred to as an *integer coordinate system* (ICS). An ICS has grid lines and grid points, as shown in Figure 10.2.

A single node of a graph will be shown by a grid point. A member of a graph will be specified by its end nodes. As an example,

$$n_1 = (2,1)$$

represents the node n_1, and

$$m_1 = [(2,2),(4,3)]$$

is an algebraic representation of the member m_1, as illustrated in Figure 10.2. This is a directed member, and, if no orientation is assigned, then one can represent m_1 also as,

$$m_1 = [(4,3),(2,2)],$$

i.e. an arbitrary order can be used for the end nodes of m_1.

In general, it is preferable to use the first quarter of an ICS in order to have positive integers as the coordinates of the nodes. However, if for some reasons the complete ICS is preferred, then both positive and negative values will be present. As an example, the algebraic representation of m_1, which is a symmetric member with respect to I_2, is given as:

$$m_1 = [(-1,2),(1,2)].$$

A subgraph (or a graph) can also be represented in an ICS by specifying its members, and the integer coordinates of its end nodes, Figure 10.3. As an example, the subgraph S_1 can be represented as:

$$S_1 = \{[(2,3),(3,2)],[(3,2),(2,1)],[(2,1),(1,1)]\}.$$

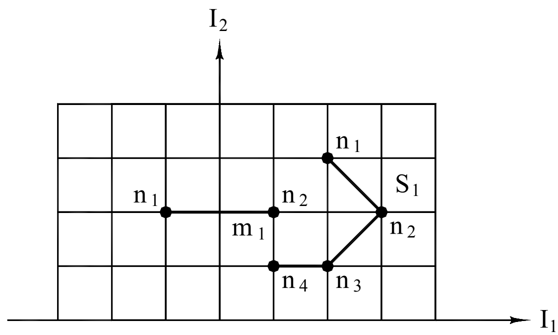


Fig. 10.3 A member and a subgraph in an ICS.

To simplify the notation and the language, from now on we will not distinguish between a graph (or subgraph) and the corresponding algebraic representation.

It is interesting to note the difference between the above representation and the previous scheme which specifies the graph up to isomorphism; i.e.

$$S_1 = \{[1,2],[2,3],[3,4]\}.$$

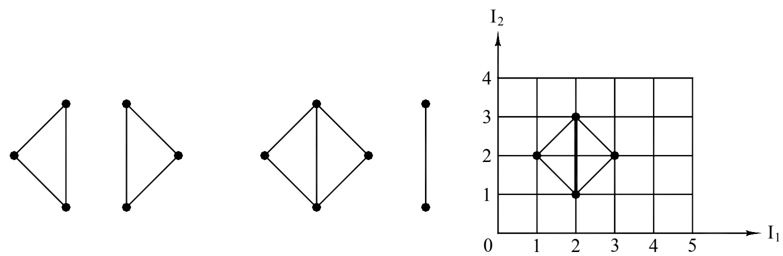
In fact, the integer coordinates of each node guarantee the unique representation of S_1.

An algebraic representation such as the one above allows one to assign weights with nodes, members, and subgraphs of a graph. These weights can be specified loads assigned to the nodes, relative stiffnesses associated with members, or group properties of substructures. This can be done in the same way that numbers were assigned to nodes, members and cycles of a graph in Chapter 1. However, in this chapter it is unnecessary to consider such factors, since only the interconnections of the structural models are of interest in configuration processing.

10.3 REPRESENTATIONS OF OPERATIONS ON GRAPHS

10.3.1 ADDITION OF TWO SUBGRAPHS

Consider a subgraph S_1, which is joined to S_2 through a member, as shown in Figure 10.4(a), resulting in $S_1 \cup S_2$:



S_1 S_2 $S_1 \cup S_2$ $S_1 \cap S_2$

(a) (b)

Fig. 10.4 Addition of two subgraphs.

Representing $S_1 \cup S_2$ in an ICS, one writes:

$$S_1 = \{[(2,1),(2,3)],[(2,3),(1,2)],[(1,2),(2,1)]\},$$

and

$$S_2 = \{[(2,1),(3,2)],[(2,1),(2,3)],[(3,2),(2,3)]\}.$$

The algebraic structure of $S_1 \cup S_2$ is obtained by omitting the repeated intersection of two subgraphs. Therefore,

$$S_1 \rightarrowtail S_2 = \{[(2,1),(2,3)],[(2,3),(1,2)],[(1,2),(2,1)],[(2,1),(3,2)],[(3,2),(2,3)]\},$$

in which one $[(2,1),(2,3)]$ is deleted in composition. From now on, such a representation will be denoted by $S_1 \rightarrowtail S_2$. If a series of compositions as,

$$S_1 \cup S_2 \cup S_3 \cup \ldots \cup S_m = \bigcup_{i=1}^{m} S_i, \tag{10-1}$$

is performed, then the corresponding algebraic representation will be shown by a different summation sign, as the following, in order to emphasis the composition nature of the operation:

$$S_1 \cup S_2 \cup \ldots \cup S_m = \sum_{i=1}^{m} S_i \, . \qquad (10\text{-}2)$$

In order to avoid the extra operation of deleting the algebraic representation of common members, composition may be considered through the common nodes only, i.e. S_1 and S_2 may be considered as illustrated in Figure 10.5. In this composition, $S_1 \cap S_2$ contains no member, and the composition of two subgraphs is simplified and consists of simple addition of their algebraic representations.

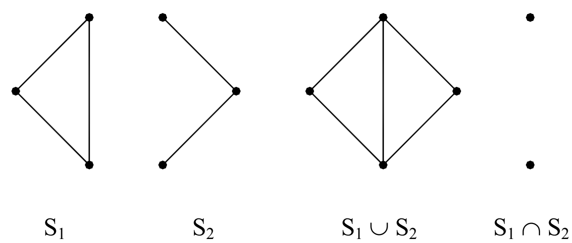


Fig. 10.5 A simple composition.

10.3.2 SUBTRACTION OF TWO SUBGRAPHS

Consider a subgraph S_1 as shown in Figure 10.6. Take S_2 as the star of the node n_2. A subgraph $S_1 - S_2$ is a graph containing the star of n_4. This subtraction in algebraic form is very simple and contains the deletion of the members of the star of n_2, i.e. for,

$$S_1 = \{[(1,1),(3,1)],[(3,1),(3,3)],[(3,3),(1,3)],[(1,3),(1,1)]\},$$

and $$S_2 = \{[(1,1),(3,1)],[(3,1),(3,3)]\},$$

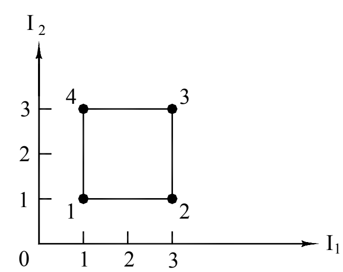


Fig. 10.6 Subtraction of two subgraphs.

the algebraic representation of this subtraction is given by:

$$S_1 - S_2 = \{[(3,3),(1,3)],[(1,3),(1,1)]\}.$$

Cut-outs frequently occur in structures. Sometimes, in order to keep the regularity of the structure, it is advantageous to generate the entire structural model and then subtract (delete) certain parts. This can be achieved by subtracting the stars within the cut-out. As an example, for S with cut-out shown in dashed lines (Figure 10.7), the main structure, S_m, can be formulated as:

$$S_m = S - \sum_{i=1}^{4} \text{Star } n_i. \tag{10-3}$$

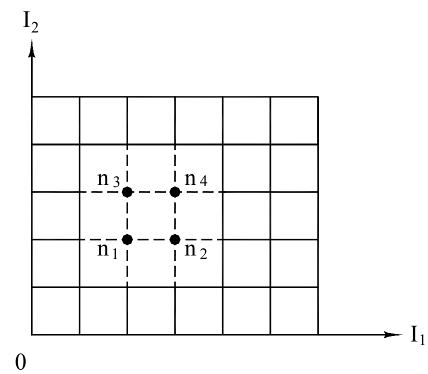


Fig. 10.7 A plane grid with a cut-out.

10.4 SPECIAL GRAPHS

A *null* graph containing only isolated nodes, denoted by S_0, can be represented by its nodes. As an example, the null graph S_0 of Figure 10.8 can be represented as:

$$S_0 = \{(1,1),(2,1),(3,1),(4,1),(1,2),(2,2),(3,2),(4,2),(1,3),(2,3),(3,3),(4,3)\}.$$

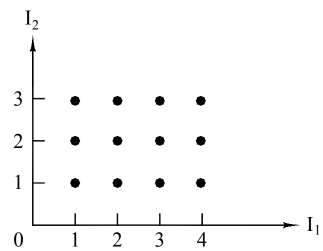


Fig. 10.8 A null graph S_0.

As a second example, the null graph containing the isolated nodes n_1, n_2, n_3 and n_4 of Figure 10.7 can be written as,

$$S_{20} = \{(2,2),(3,2),(2,3),(3,3)\},$$

where S_{20} is the null graph of S_2. Conversely S_2 may be considered as Star (S_{20}) in S, where

$$\text{Star}\,(S_{20}) = \bigcup_{i=1}^{N(S_{20})} \text{Star}\, n_i \quad \text{where } n_i \in S_2, \tag{10-4}$$

and $N(S_{20})$ is the number of isolated nodes of S_{20}. Therefore, in Figure 10.7, the grid with the cut-out can be represented as $S_m = S_1 - \text{Star}\,(S_{20})$.

The null graph, or the null graph of a graph, plays an important role in structural analysis. As an example, the null graph of a part of a structural model may represent the boundary nodes, or those nodes which are loaded in the structure; and therefore compact representation of such nodes is advantageous in data generation. The null graph of a graph S_i will be denoted by S_{i0}.

A similar convention can be applied to a complete graph. Since in a complete graph, all the nodes are connected to each other by distinct members, one can specify only its null graph with an extra command to indicate the completeness of the graph. For example, COM (S_{k0}) can be used to refer to the complete graph S_k.

10.5 SOME FUNCTIONS FOR CONFIGURATION PROCESSING

Many practical structures have a regular and repeated pattern, which enables their generation using simple functions. As an example, S in Figure 10.9 can be generated by translating S_1, shown in bold lines, in the I_1 direction; i.e., if the coordinates of the nodes of S_1 are each increased by 2 units, the adjacent subgraph

S_2 will be obtained. A similar operation on S_2 results in S_3, and the same operation on S_3 forms S_4, completing the generation of $S = \bigcup_{i=1}^{4} S_i$.

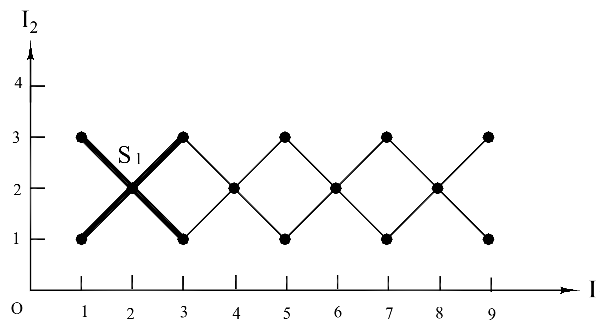


Fig. 10.9 A typical S_1 for generating S by translation.

A second example is depicted in Figure 10.10, in which S is generated by rotating a typical subgraph S_1 shown in bold lines. Rotation about O by 90^o leads to S_2, and a similar operation on S_2 results in S_3, etc. Therefore, $S = S_1 \cup S_2 \cup S_3 \cup S_4$ can be easily generated.

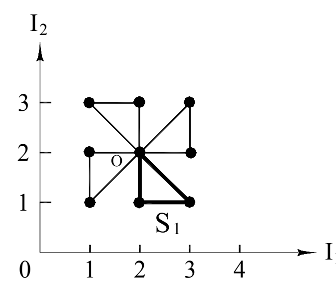


Fig. 10.10 A typical S_1 for generating S by rotation.

An operation by means of which S_j is obtained from S_i can be shown as,

$$S_j = \phi \mid S_i, \tag{10-5}$$

where ϕ can be considered as a mapping of S_i into S_j, or it may be taken as a function which defines the relationship between S_i and S_j. The following functions are defined for ϕ:

(a) If $S_j = \phi \mid S_i$, then $S_i = \phi^{-1} \mid S_j$, i.e. the inverse of function ϕ exists.

(b) The identity function is supposed to map S_i onto itself. This function is denoted by ϕ^0: i.e. $S_i = \phi^0 \mid S_i$.

(c) Repeated application of ϕ on S_i will be denoted by $\phi^r \mid S_i$, which means ϕ has been applied r times, and each time ϕ operates on the result of the previous step.

(d) The composition of functions ϕ^m and ϕ^n is equivalent to ϕ^{m+n}.

In the following, various functions and their significances are described, which in the main follow those of Ref. [180].

10.5.1 TRANSLATION FUNCTIONS

A translation function operating on S_i to produce S_j is denoted by,

$$S_j = \phi \mid S_i = \text{TRAN}\ (h,q) \mid S_i, \tag{10-6}$$

in which h = 1, 2 and 3 depending on translation taking place along I_1, I_2 and I_3, respectively. q is the number of integer units to be added to each coordinate of algebraic representation of S_i in direction h. This operation can be stated as follows:

Let a node of S_i be denoted by (I_1,I_2,I_3) and that of S_j be denoted by (I'_1, I'_2, I'_3). The function TRAN (h,q) maps the nodes of S_i onto those of S_j such that, for each node,

$$I'_i = I_i \quad \text{for} \quad i=1,2,3 \quad \text{except} \quad i=h,$$

and

$$I'_h = I_h + q. \tag{10-7}$$

Obviously, q should be taken as an integer, if the new node is supposed to be a grid point.

Example: Consider a graph S_1 as shown in Figure 10.11.

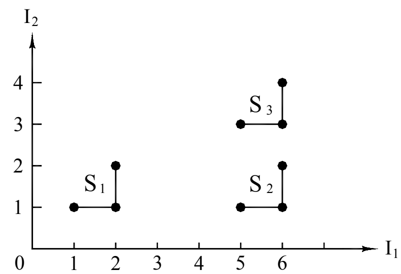


Fig. 10.11 Subgraphs obtained by translations.

S_1 can be written as:

$$S_1 = \{[(1,1),(2,1)],[(2,1),(2,2)]\}.$$

Similarly, S_2 may be represented as:

$$S_2 = \mathrm{TRAN}(1,4) \mid S_1 = \{[(5,1),(6,1)],[(6,1),(6,2)]\},$$

Then S_3 can be obtained from S_2 by:

$$S_3 = \mathrm{TRAN}(2,2) \mid S_2 = \{[(5,3),(6,3)],[(6,3),(6,4)]\}.$$

Composition of the above functions results in S_3, when S_1 is given, i.e.

$$S_3 = \mathrm{TRAN}(2,2) \mid \mathrm{TRAN}(1,4) \mid S_1 = \{[(5,3),(6,3)],[(6,3),(6,4)]\}.$$

Example: A simple rectangular grid is formulated. Consider a graph as shown in Figure 10.12.

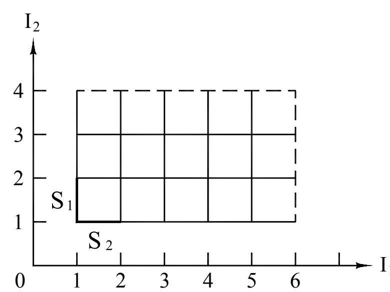


Fig. 10.12 A rectangular grid obtained by pure translations.

Different units (subgraphs) can be considered for generation. Naturally, one should consider the smallest unit for which the translation functions can also be written in a simple form. Let:

$$S_1 = \{[(1,1),(1,2)]\} \text{ and } S_2 = \{[(1,1),(2,1)]\}.$$

The members of the vertical edge of S containing S_1 can be represented as:

$$\text{TRAN }(2,0)|S_1 = S_1,\ \text{TRAN }(2,1)|S_1 \text{ and TRAN }(2,2)|S_1.$$

The above three members can collectively be shown as:

$$\sum_{j=0}^{2} \text{TRAN }(2,j)|\ S_1.$$

Translation of the resulting subgraph in the I_1 direction completes the formation of all the vertical members,i.e.

$$S_V = \sum_{i=0}^{5} \text{TRAN }(1,i)\ |\ \sum_{j=0}^{2} \text{TRAN }(2,j)\ |\ S_1,$$

The horizontal members can be formed by a similar approach leading to:

$$S_H = \sum_{j=0}^{3} \text{TRAN }(2,j)\ |\ \sum_{i=0}^{4} \text{TRAN }(1,i)\ |\ S_2.$$

Naturally, $S = S_V + S_H$,

$$S = \sum_{i=0}^{5} \text{TRAN }(1,i)\ |\sum_{j=0}^{2} \text{TRAN }(2,j)\ |\ \{[(1,1),(1,2)]\} +$$
$$\sum_{j=0}^{3} \text{TRAN }(2,j)\ |\sum_{i=0}^{4} \text{TRAN }(1,i)\ |\ \{[(1,1),(2,1)]\}.$$

As an alternative formulation, one may consider $S_1 \cup S_2 = S_3$ as a generating unit,

$$S_3 = \{[(1,1),(1,2)],[(1,1),(2,1)]\}.$$

Then
$$S_4 = \sum_{j=0}^{2} \text{TRAN }(2,j)\ |\sum_{i=0}^{4} \text{TRAN }(1,i)\ |\ S_3,$$

generating all members except the upper and right-hand side boundary members, which can be generated as:

$$S_5 = \text{TRAN }(1,5) \mid \sum_{j=0}^{3} \text{TRAN }(2,j) \mid S_1 + \text{TRAN }(2,3) \mid \sum_{i=0}^{4} \text{TRAN }(1,i) \mid S_2.$$

Therefore, $S = S_4 + S_5$ generates the entire model S.

Translation functions have the following properties:

1. TRAN (h_1,q_1) | TRAN(h_2,q_2) | S_i = TRAN (h_2,q_2) | TRAN (h_1,q_1) | S_i; i.e. these functions are commutative.

2. TRAN $(h,q)^{-1}$ | S_i = TRAN $(h, -q)$| S_i.

3. TRAN (h,q_1) | TRAN (h,q_2) | S_i = TRAN (h,q_1+q_2) | S_i.

   Similarly, TRAN $(h,q)^k$ | S_i = TRAN (h,kq) | S_i.

10.5.2 ROTATION FUNCTIONS

This function rotates a subgraph S_i with respect to a specified node to obtain S_j, and is denoted as:

$$S_j = \text{ROT }(h_1,h_2,q_1,q_2) \mid S_i. \tag{10-8}$$

Let (I_1,I_2,I_3) be a typical node of S_i and (I'_1, I'_2, I'_3) be the integer coordinates of the nodes of S_j. The rotation function maps the nodes of S_i onto those of S_j such that, for each node,

$$\begin{aligned} &I'_{h1} = q_2 + q_1 - I_{h2} \ \text{ for } \ i = h_1, \\ &I'_{h2} = q_2 - q_1 + I_{h1} \ \text{ for } \ i = h_2, \\ &\text{and } I'_{h3} = I_i \ \text{for the remaining value of i.} \end{aligned} \tag{10-9}$$

Obviously, the numbers q_1 and q_2 should be such that their sum and difference become integers, unless an even power of the function is used.

Example: Consider S_1 as shown in Figure 10.13. This subgraph can be represented as:

$$S_1 = \{[(1,1),(2,1)],[(2,1),(2,2)]\}.$$

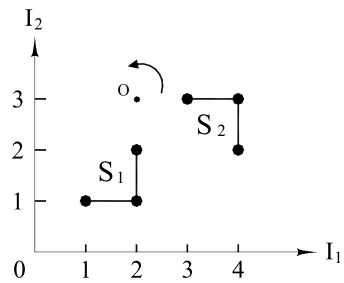


Fig. 10.13 A subgraph S_1 rotated to S_2.

For ROT (1,2,2,3), we have,

$$S_2 = \text{ROT}\ (1,2,2,3) \mid S_1 = \{[(4,2),(4,3)],[(4,3),(3,3)]\},$$

which is depicted in Figure 10.13.

The rotation function ROT (h_1,h_2,q_1,q_2) rotates S_i through $\pi/2$ about an axis perpendicular to the $I_{h1} - I_{h2}$ plane and intersects this plane at a point whose coordinates are q_1 and q_2. The sense of rotation is such that a rotation of I_{h1} through $\pi/2$ about the origin maps the positive side of I_{h1} onto that of I_{h2}. ROT $(h_1,h_2,q_1,q_2)^k| S_i$ represents the rotation of S_i by $k\pi/2$.

For a general case, when the rotation is an arbitrary angle β, simple formulation can be made. Consider two points $A(I_1,I_2)$ and $A'(I_1',I_2')$ as shown in Figure 10.14. The centre of rotation is taken as $O'(q_1,q_2)$. The following relations are obvious:

$$\begin{aligned} \overrightarrow{OA} &= \overrightarrow{OO'} + \overrightarrow{O'A} \\ \overrightarrow{OA'} &= \overrightarrow{OO'} + \overrightarrow{O'A'} \end{aligned} \tag{10-10}$$

Projecting the vectors OA and OA' on I_1 and I_2 axis,

$$\begin{aligned} q_1 + L\cos\alpha &= I_1 \\ q_2 + L\sin\alpha &= I_2 \end{aligned} \tag{10-11}$$

where L is the length of $O'A$, which is the same as $O'A'$, since rotation preserves the length. Using Eq. (10-10) and projecting the components of the vectors OA and OA' on I_1 and I_2 leads to:

$$\begin{aligned} q_1 + L\cos(\alpha+\beta) = I_1' = q_1 + L\cos\alpha\cos\beta - L\sin\alpha\sin\beta \\ q_2 + L\sin(\alpha+\beta) = I_2' = q_2 + L\sin\alpha\cos\beta - L\sin\beta\cos\alpha \end{aligned} \tag{10-12}$$

Combining Eq. (10-11) and Eq. (10-12) leads to:

$$\begin{aligned} I_1' = q_1 + (I_1 - q_1)\cos\beta - (I_2 - q_2)\sin\beta \\ I_2' = q_2 - (I_2 + q_2)\cos\beta + (I_1 - q_1)\sin\beta \end{aligned} \tag{10-13}$$

Functions for a general rotation are obtained as:

$$\begin{aligned} I_1' = q_1 - q_1\cos\beta + q_2\sin\beta + I_1\cos\beta - I_2\sin\beta \\ I_2' = q_2 - q_2\cos\beta + q_1\sin\beta - I_2\cos\beta + I_1\sin\beta \end{aligned} \tag{10-14}$$

Substituting β = 90° leads to the special case of Eq. (10-9).

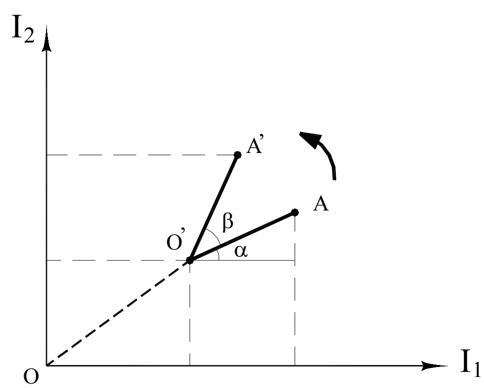
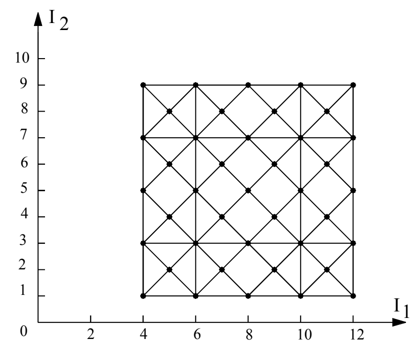
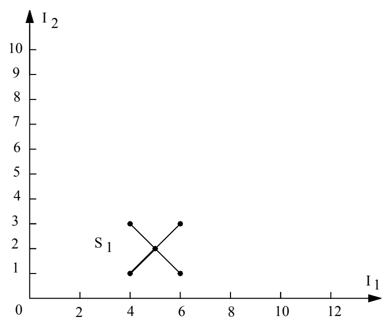


Fig. 10.14 General rotation.

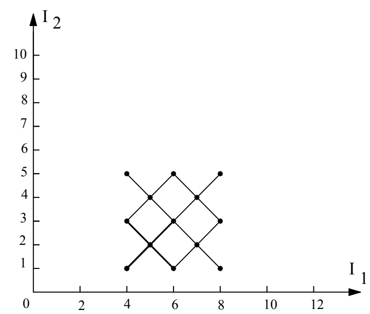
Example: The graph model of a planar truss, as shown in Figure 10.15(a), can be generated using rotation functions as follows:



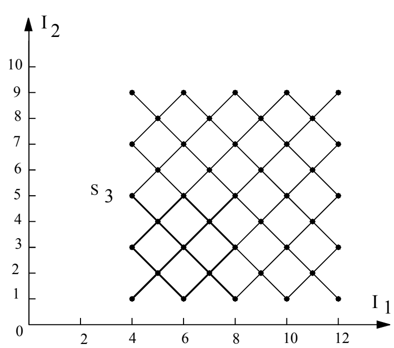
(a) The graph model S.



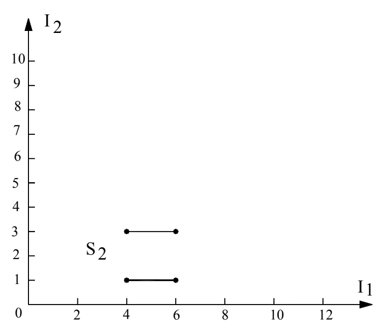
(b) S_1 is rotated.



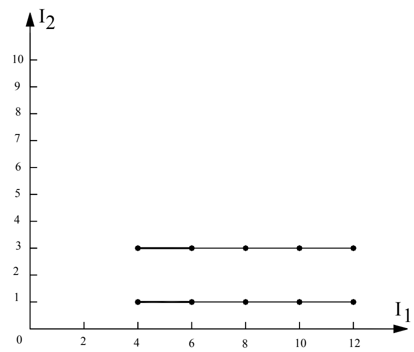
(c) The subgraph of (b) is rotated.



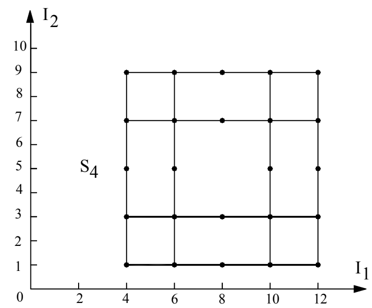
(d) S_3.



(e) S_2 is rotated.



(f) The subgraph of (e) is rotated.



(g) S_4.

Fig. 10.15 A planar truss generated by pure rotations.

Consider $S_1 = \{[(4,1),(5,2)]\}$ and $S_2 = \{[(4,1),(6,1)]\}$.

First S_3 and S_4, shown in Figures 10.14 (b and g), are formulated as:

$$S_3 = \sum_{k=0}^{3} \mathrm{ROT}\,(1,2,8,5)^k \mid \sum_{j=0}^{3} \mathrm{ROT}\,(1,2,6,3)^j \mid \sum_{i=0}^{3} \mathrm{ROT}\,(1,2,5,2)^i \mid S_1,$$

and $$S_4 = \sum_{k=0}^{3} \mathrm{ROT}\,(1,2,8,5)^k \mid \sum_{j=6}^{8} \mathrm{ROT}\,(1,2,j,3)^2 \mid \mathrm{ROT}(1,2,5,2)^2 \mid S_2.$$

Now, for the entire model S, we have:

$$S = S_3 + S_4$$

Rotation functions have the following properties:

1. Rotation functions are not in general commutative, except for some special cases.

2. For an integer k, $\mathrm{ROT}\,(h_1,h_2,q_1,q_2)^{4k}|S_i = S_i$; i.e. $\mathrm{ROT}\,(h_1,h_2,q_1,q_2)^{4k}$ is an identity function.

3. The inverse of a rotation function is the cube of itself; i.e. $\mathrm{ROT}\,(h_1,h_2,q_1,q_2)^{-1}|\, S_i = \mathrm{ROT}\,(h_1,h_2,q_1,q_2)^3|\, S_i$.

10.5.3 REFLECTION FUNCTIONS

This function reflects a subgraph S_i, and finds S_j as:

$$S_j = \mathrm{REF}\ (h,q) \mid S_i\,, \tag{10-15}$$

Like the previous functions, a reflection function REF (h,q) maps nodes of S_i onto those of S_j by the following rule:

$$I'_h = 2q - I_h \quad \text{for } i = h,$$

and

$$I'_i = I_i \text{ for the remaining i s.} \tag{10-16}$$

Obviously, 2q should be integer if the new coordinates are intended to be integers.

Example: Different reflections of S_1 are illustrated in Figure 10.16:

$$S_1 = \{[(2,1),(2,2)],[(2,2),(1,2)],[(1,2),(2,1)]\}.$$

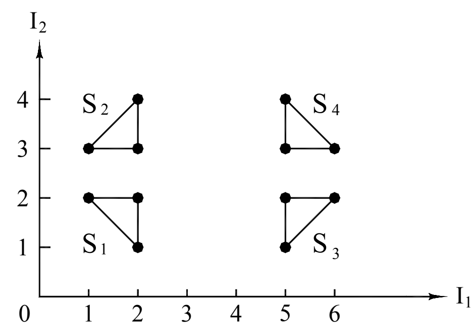


Fig. 10.16 Subgraphs generated by reflections.

Now $S_2 = \mathrm{REF}\ (2,2.5) \mid S_1 = \{[(2,4),(2,3)],[(2,3),(1,3)],[(1,3),(2,4)]\}$,

and $S_3 = \mathrm{REF}\ (1,3.5) \mid S_1 = \{[(5,1),(5,2)],[(5,2),(6,2)],[(6,2),(5,1)]\}$.

Similarly, $S_4 = \mathrm{REF}\ (2,2.5) \mid S_3 = \{[(5,4),(5,3)],[(5,3),(6,3)],[(6,3),(5,4)]\}$.

For a reflection function, S_j is the mirror image of S_i with respect to a plane which is normal to the I_h axis and intersects it at a point with $I_h = q$.

Example: The graph shown in Figure 10.17 can be formulated using a single member as follows:

Take $S_1 = \{[(2,1)],[(1,2)]\}$ as a generating subgraph. Then:

$$S = \sum_{i=0}^{1} \mathrm{REF}\,(2,2)^i \mid \sum_{j=3}^{5} \mathrm{REF}\,(1,j) \mid \sum_{k=0}^{1} \mathrm{REF}\,(1,2)^k \mid S_1.$$

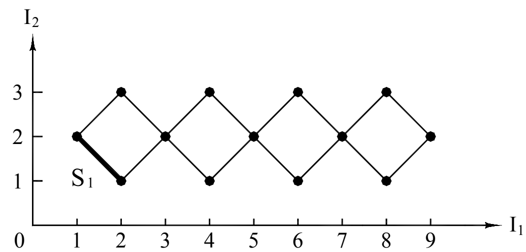


Fig. 10.17 A graph generated by pure reflections.

Reflection functions have the following properties:

1. For $h_1 \neq h_2$, REF $(h_1,q_1) \mid$ REF $(h_2,q_2) \mid S_i =$ REF $(h_2,q_2) \mid$ REF $(h_1,q_1) \mid S_i$.
2. REF $(h,q)^{-1} \mid S_i =$ REF $(h,q) \mid S_i$, also REF $(h,q)^{2k} \mid S_i = S_i$

   and REF $(h,q)^{2k+1} \mid S_i =$ REF $(h,q) \mid S_i$.

10.5.4 PROJECTION FUNCTIONS

A projection function is defined as,

$$\mathrm{PROJ}\,(h,q) \mid S_i \tag{10-17}$$

which is a mapping of S_i onto S_j such that:

$$I'_h = q \text{ for } i=h$$

and

$$I'_i = I_i \text{ for remaining i s } (i=1,2,3). \tag{10-18}$$

Naturally, q should be integer if the nodes of S_j are preferred to be on the grid points.

Example: The projections of S_1, S_2 and S_3 are illustrated in Figure 10.18.

$$S_1 = \{[(1,1)],[(2,2)]\},$$

$$S_2 = \text{PROJ}\ (1,4) \mid S_1 = \{[(4,1)],[(4,2)]\},$$

$$S_3 = \{[(2,3),(3,3)],[(3,3),(2,4)],[(2,4),(2,3)]\},$$

and $$S_4 = \text{PROJ}\ (2,1) \mid S_3 = \{[(2,1),(3,1)],[(3,1),(2,1)],[(2,1),(2,1)]\}.$$

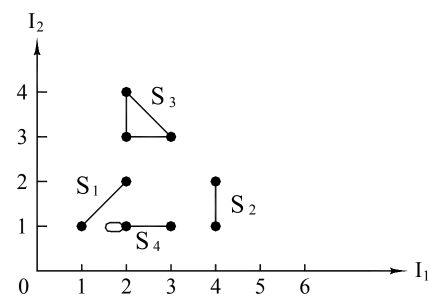


Fig. 10.18 Projections of subgraphs.

For a projection function S_j = PROJ (h,q) | S_i, S_j is obtained from S_i by projecting S_i onto a plane that is perpendicular to the I_h axis and that intersects the plane at a point where $I_h = q$.

Projection functions have the following properties:

1. A projection function has no inverse;

2. For $h_1 \neq h_2$, PROJ (h_1,q_1)|PROJ (h_2,q_2) | S_i = PROJ (h_2,q_2)|PROJ (h_1,q_1)|S_i;

3. PROJ (h,q_k) | ... | PROJ (h,q_2) | PROJ (h,q_1) | S_i = PROJ (h,q_k) | S_i.

It should be noted that the useful functions are by no means limited to the four types introduced in this section. One can define functions which are more suitable for a particular application. Many such functions are defined in Ref. [136].

Compact algebraic representation of a structural model is one of the aims of the methods presented in this chapter. For this purpose, it may be necessary to use a combination of functions, which are presented or can be defined. However, optimal representation of an arbitrary configuration in the sense of compactness is an open problem.

Example: In the following example, a combination of translation and rotation functions is used for algebraic representation of a structural model.

Let S be a double-layer grid as shown in Figure 10.19. A 3-dimensional integer coordinate system is used for its presentation.

The top layer of S is represented as:

$$\mathrm{TH} = \sum_{i=0}^{4} \mathrm{TRAN}(1,2i) \mid \sum_{j=0}^{5} \mathrm{TRAN}(2,2j)[(1,1,1),(3,1,1)],$$

$$\mathrm{TV} = \sum_{i=0}^{5} \mathrm{TRAN}(1,2i) \mid \sum_{j=0}^{4} \mathrm{TRAN}(2,2j)[(1,1,1],(1,3,1)].$$

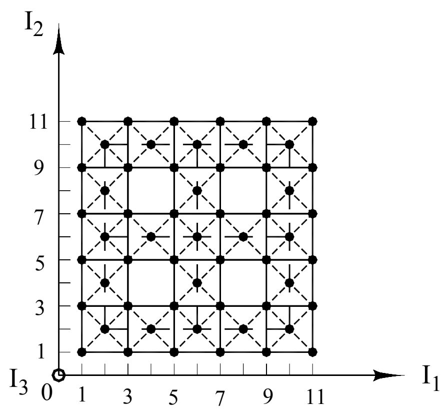


Fig. 10.19 A double-layer grid S.

The bottom layer can be formulated as:

$$\mathrm{BH} = \sum_{i=0}^{3} \mathrm{TRAN}(1,2i) \mid \sum_{j=0}^{2} \mathrm{TRAN}(2,4j) \mid [(2,2,0),(4,2,0)],$$

$$\mathrm{BV} = \sum_{i=0}^{2} \mathrm{TRAN}(1,4i) \mid \sum_{j=0}^{3} \mathrm{TRAN}(2,2j) \mid [(2,2,0),(2,4,0)].$$

The rest of the model can be represented as:

$$\mathrm{BT} = \sum_{i=0}^{3} \mathrm{ROT}(1,2,2,2)^{i} \mid [(1,1,1),(2,2,0)],$$

$$BT1 = \sum_{i=0}^{4} TRAN(1,2i) \mid \sum_{j=0}^{2} TRAN(2,4j) \mid BT,$$

$$BT2 = \sum_{i=0}^{2} TRAN(1,4i) \mid \mid TRAN\ (2,2) \mid BT,$$

The algebraic representation of the entire model can now be shown as:

$$S = TH + TV + BH + BV + BT1 + BT2.$$

10.6 GEOMETRY OF STRUCTURES

The algebraic representations described in the previous sections not only provide the topological properties of a structure, but also provide simple means for obtaining other properties. As an example, the geometrical properties of a structure can be obtained using simple transformations. All that is needed is to establish a relation between the integer coordinate employed and the coordinate system chosen for geometrical presentation of the structure. This coordinate system can be an orthogonal Cartesian coordinate, a polar, a cylindrical or any other curvilinear coordinate system.

Example: Consider a grid model as shown in Figure 10.20.

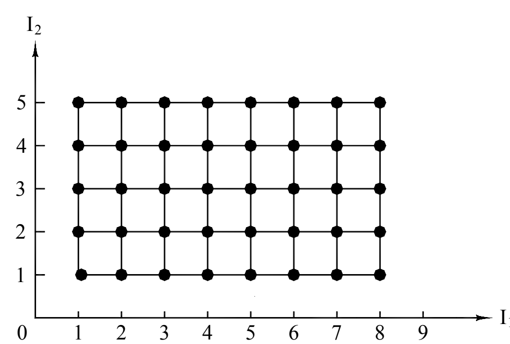


Fig. 10.20 A planar grid S.

Let x and y be the geometric coordinates of a typical joint of the structure with respect to a selected two-dimensional Cartesian coordinate system. Consider the following transformation:

$$x = 3I_1 \text{ and } y = 2I_2. \tag{10-19}$$

Then the geometry of S will be a graph as shown in Figure 10.21.

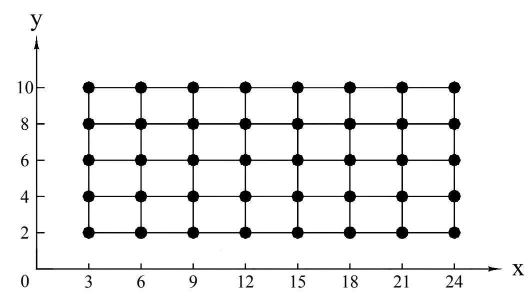


Fig. 10.21 Geometrical properties of S.

Now consider the same graph and use a polar coordinate system with the following transformation:

$$r = I_2 + 1 \text{ and } \theta = (I_1 - 1)\frac{\pi}{7}. \tag{10-20}$$

Then S becomes a configuration as illustrated in Figure 10.22.

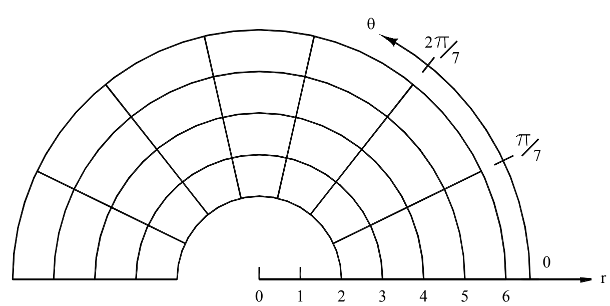


Fig. 10.22 Grid S mapped to a polar coordinate system.

The same model can be mapped into a three-dimensional cylindrical coordinate system with the following simple transformation:

$$r = 20, \quad \theta = \frac{\pi}{15} I_2 \quad \text{and} \quad z = 3I_1. \tag{10-21}$$

The resulting configuration is depicted in Figure 10.23.

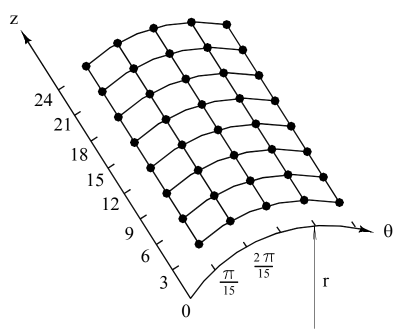


Fig. 10.23 Grid S mapped to a cylindrical coordinate system.

It should be noted that the connectivity coordinate system defined in Chapter 7 can easily be combined with the concepts presented in this chapter, to obtain a favourable nodal numbering of a structure, Kaveh [91].

10.7 EXTENSION TO HYPERGRAPHS

For skeletal structures, the model can be represented as a graph; however, there are other types of structure having more than two nodes per member. Finite element models are examples of this kind. For these structures, the model can be considered as a hypergraph, Berge [13]. A *hypergraph* consists of a set of nodes, and a set of members with a relation of incidence which associates some nodes with each member. A special case is when two nodes are associated with each member; then a hypergraph becomes a graph. Many of the methods presented thus far stay valid for hypergraphs as well.

Example: A finite element mesh consisting of triangular elements is considered as shown in Figure 10.24. This model can be formulated as follows:

A typical element m_l can be represented as:

$$m_l = \{[(1,1),(2,1),(2,2)]\}.$$

Similarly:

$$m_2 = \{[(1,1),(2,2),(1,2)]\}.$$

These elements may be shown as:

$$S_1 = m_1 + m_2 = \{[(1,1),(2,1),(2,2)],[(1,1),(2,2),(1,2)]\}.$$

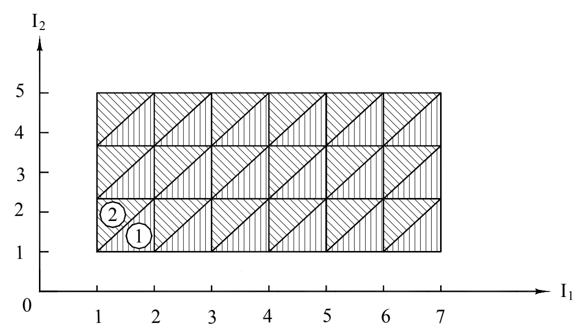


Fig. 10.24 A hypergraph as the model of a finite element mesh.

Using translation functions, the algebraic representation of the entire FEM becomes:

$$S = \sum_{i=0}^{5} \mathrm{TRAN}(1,i) \Big| \sum_{j=0}^{2} \mathrm{TRAN}(2,j) \,\Big|\, S_1.$$

Therefore, mesh generation can be considered as a special case of the configuration processing discussed in this chapter.

Example: A frame structure is considered with its shear wall modelled as triangular finite elements, Figure 10.25.

The algebraic representation of the model can be written as:

$$F = \sum_{i=0}^{2} \mathrm{TRAN}(1,2i) \Big| \sum_{j=0}^{4} \mathrm{TRAN}(2,2j) \,\Big|\, \{[(1,1),(1,3)],[(1,3),(3,3)]\},$$

$$W = \sum_{i=0}^{1} \mathrm{TRAN}(1,2i) \Big| \sum_{j=0}^{4} \mathrm{TRAN}(2,2j) \,\Big|\, \sum_{k=0}^{3} \mathrm{ROT}(1,2,8,2)^k [(7,1),(8,2),(9,1)],$$

$$S_h = F + W,$$

and $$S = \sum_{i=0}^{1} REF(1,11)^i \Big| \; S_h .$$

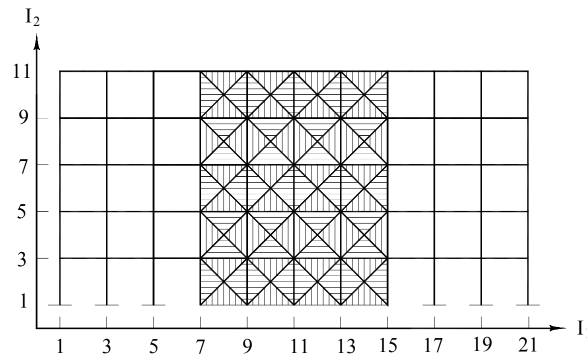


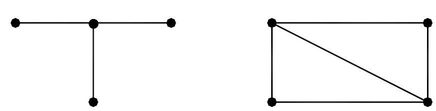
Fig. 10.25 A frame with a shear wall.

It can be concluded that, simple definitions and concepts of the theory of graphs provide a powerful tool for a complete formulation of configuration processing. The ideas are by no means limited to topological graphs, but can be applied equally to abstract graphs and hypergraphs, preserving the generality of the formulation.

Finally, it should be mentioned that the functions presented in this chapter can easily be programmed and employed in data generation of large-scale structures. Other functions suitable for particular applications can also be formulated and used.

EXERCISES

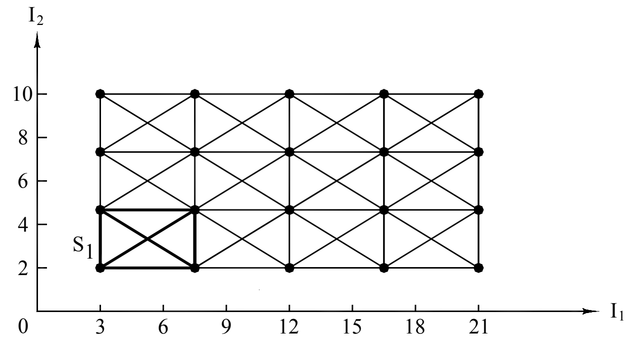
10.1 Write the algebraic representation of the following graphs in an arbitrary rectilinear ICS:



10.2 Plot the graph corresponding to the following algebraic representation in an ICS:

$$S = \sum_{i=0}^{5} \text{TRAN}\,(1,3i) \mid \sum_{j=0}^{5} \text{TRAN}\,(2,j) \mid \{[(1,1),(2,2)],[(3,1),(2,2)]\}.$$

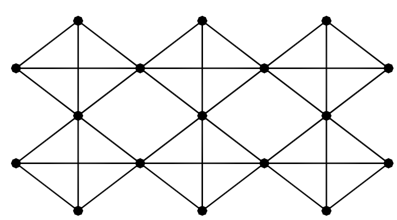
10.3 Write the algebraic representation of the following structural model using translation functions:



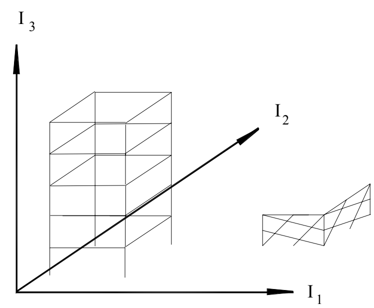
10.4 Plot the graph model of the following algebraic representation in an arbitrary ICS:

$$S = \sum_{i=0}^{3} \text{ROT}(1,2,6,6)^{i} \big| \{[(6,4),(2,2)],[(4,6),(2,2)]\}.$$

10.5 Write the algebraic representation of the following structural model in an arbitrary ICS, using reflection functions:



10.6 Using translation and rotation functions, write the algebraic representation of the following tower structure in the given three-dimensional coordinate system.



10.7 Write a computer program to generate a configuration with the translation functions of Section 10.5.1.

10.8 Repeat 10.7 with the rotation functions of Section 10.5.2.

10.9 Repeat 10.7 with the reflection functions of Section 10.5.3.

10.10 Repeat 10.7 with the projection functions of Section 10.5.4.