Chapter 8

CONDITIONING OF STRUCTURAL MATRICES

8.1 INTRODUCTION

The use of the digital computer for problems in structural analysis requires the solution of a large system of algebraic equations of the form

$$\mathbf{Ax} = \mathbf{b}, \tag{8-1}$$

as also cited at the opening of Chapter 7. This is true both for the force method and the displacement approach. Sometimes the solution of Eq.(8-1) changes greatly by small perturbation in matrix **A**. Then we say **A** is ill-conditioned with respect to this solution. The accuracy of the solution of Eq.(8-1) can be sensitive to the characteristics of the matrix **A**. Therefore it is important to study these characteristics and their inter-relationships with the source, propagation and distribution of possible errors. In doing so, better methods of problem formulation must be found and techniques for predicting, detecting and minimizing solution errors must be devised.The ill-conditioning of stiffness matrices for the displacement method of analysis was studied by Shah [166]. In his work, methods were suggested for improving conditioning of the stiffness matrices. A mathematical investigation of matrix error analysis is due to Rosanoff and Ginsburg [158]. In their work, it was shown that numerically unstable equations may arise in physically stable problems. Thus the need for routine measurement of matrix conditioning numbers associated with various patterns of formulation is emphasized. The effect of substructuring on conditioning of stiffness matrices was investigated by Grooms and Rowe [53] who concluded that substructuring does not significantly influence the solution accuracy of ill-conditioned systems. Filho [40] suggested an orthogonalization method for the best conditioning of flexibility and stiffness matrices; however, this is an impractical approach for multi-member complex structures.

Optimizing the conditioning of equilibrium equations when an algebraic force method is employed, was studied by Robinson and Haggenmacher [157]. For the combinatorial force method, studies have been limited to increasing the sparsity of cycle basis incidence matrices, Henderson [59] and Goodspeed and Martin [49] (see also Cassell [16] for a discussion on the latter reference). Recently, methods have been developed for selecting particular types of statical and kinematical bases leading to flexibility and stiffness matrices better conditioned than classical ones, Kaveh [89].

In structural engineering, one of the important sources of ill-conditioning is the use of members in a structure which have widely different stiffnesses (or flexibilities). The application of standard statical or kinematical bases (though optimal) leads to ill-conditioned structural matrices. In this chapter, methods are developed for generating special cycle and cut set bases corresponding to statical and kinematical bases which provide the best possible conditioning for flexibility and stiffness matrices, respectively.

8.2 CONDITION NUMBERS

In order to measure the conditioning of a matrix, various numbers are defined and employed in practice. Three commonly used condition numbers are defined in the following; they are simple and easy to use.

8.2.1 THE RATIO OF EXTREME EIGENVALUES

Eigenvalues and eigenvectors are related to the conditioning of matrices. The ratio of the extreme eigenvalues of a matrix $|\lambda_{max}| / |\lambda_{min}|$ can be taken as its condition number. It can easily be shown that the logarithm to the base ten of this condition number is roughly proportional to the maximum number of significant figures lost in inversion or in the solution of simultaneous equations. Thus the number of good digits, g, in the solution, is given by

$$g = p - \log(|\lambda_{max}| / |\lambda_{min}|) = p - PL. \qquad (8\text{-}2)$$

In this relationship $PL = \log(|\lambda_{max}| / |\lambda_{min}|)$ and p is a number which varies from machine to machine. For example, the IBM/360 uses approximately eight digits for single precision and 16 digits for double precision calculations. It should be mentioned that the above estimate is conservative, and experience shows that PL is one digit on the safe side. The importance of this condition number justifies more explanation and a simple numerical example.

Symmetric matrices can be written as a linear combination of rank one matrices as

$$\mathbf{A} = \sum_{\mathbf{i}=1}^{\mathbf{n}} \lambda_i \mathbf{v}_i \mathbf{u}_i^t, \tag{8-3}$$

and

$$\mathbf{A}^{-1} = \sum_{\mathbf{i}=1}^{\mathbf{n}} (1/\lambda_i) \mathbf{v}_i \mathbf{u}_i^t, \tag{8-4}$$

with $\mathbf{v}_\mathbf{i}^\mathbf{t} \mathbf{u}_i = 1$ for i=1,...,n. In the above equations λ_i is the ith eigenvalue and $\mathbf{v}_i$ is the corresponding eigenvector of **A**, and $\mathbf{u}_i$ is the ith eigenvector of $\mathbf{A}^{-1}$ · Eq.(8-3) shows that the rank one matrices of the eigenvectors enter the matrix **A** in amount proportional to their respective eigenvalues. The lower mode of **A** becomes weakly represented as the ratio of extremal eigenvalues becomes large. Specifically, as a first approximation for each power of 10 in the ratio $| \lambda_{max} | / | \lambda_{min} |$, the lower mode will lose about 1 decimal digit in a finite computer number set representation of the matrix. On the other hand, the lower mode of **A** is the upper mode of $\mathbf{A}^{-1}$, because the coefficients of the linear combination (the eigenvalues) are inverted. Therefore, inverting matrices without some feel for their conditioning can lead to wrong solutions. Consider a 2×2 matrix such as

$$\mathbf{A} = \begin{bmatrix} 1/9 & 1/10 \\ 1/10 & 1/11 \end{bmatrix} = \begin{bmatrix} 0.11111111 & 0.10000000 \\ 0.10000000 & 0.09090909 \end{bmatrix}. \tag{8-5}$$

The eigenvalues and eigenvectors of **A** with 8 digits are

$$\begin{aligned} \lambda_1 &= 0.20151896 \qquad \mathbf{v}_1 = \mathbf{u}_1 = \begin{Bmatrix} 0.74178794 \\ 0.67063452 \end{Bmatrix}, \\ \lambda_2 &= 0.0005012437 \qquad \mathbf{v}_2 = \mathbf{u}_2 = \begin{Bmatrix} 0.67063452 \\ -0.74178794 \end{Bmatrix}, \end{aligned} \tag{8-6}$$

leading to $\lambda_1/\lambda_2 = 402.0379 = 10^{2.604}$. From Eq.(8-3) matrix **A** can be written as

$$
\begin{aligned}
\mathbf{A} &= \lambda_1 \mathbf{v}_1 \mathbf{u}_1^{\mathbf{t}} + \lambda_2 \mathbf{v}_2 \mathbf{u}_2^{\mathbf{t}} \\
&= 0.20151896 \begin{Bmatrix} 0.74178794 \\ 0.67063452 \end{Bmatrix} \begin{Bmatrix} 0.74178794 & 0.67063452 \end{Bmatrix} \\
&+ 0.0005012437 \begin{Bmatrix} 0.67063452 \\ -0.74178794 \end{Bmatrix} \begin{Bmatrix} 0.67063452 & -0.74178794 \end{Bmatrix} \\
&= \begin{bmatrix} 0.11088567 & 0.10024935 \\ 0.10024935 & 0.090633285 \end{bmatrix} + \begin{bmatrix} 0.00022543467 & -0.00024935298 \\ -0.00024935298 & 0.00027580902 \end{bmatrix} \\
&= \begin{bmatrix} 0.11111111 & 0.099999997 \\ 0.099999997 & 0.090909094 \end{bmatrix}.
\end{aligned} \tag{8-7}
$$

In forming this 8-digit approximation to the matrix, the component matrix $\lambda_2 \mathbf{v}_2 \mathbf{u}_2^{\mathbf{t}}$ which has three leading zeros in its elements, is truncated to about 5 digits. Therefore an 8-digit representation of the matrix **A** contains about 5 digits of information about the rank one matrix $\mathbf{v}_2 \mathbf{u}_2^{\mathbf{t}}$.

Similarly consider $\mathbf{A}^{-1}$ formed as

$$
\begin{aligned}
\mathbf{A}^{-1} &= \frac{1}{\lambda_1} \mathbf{v}_1 \mathbf{u}_1^{\mathbf{t}} + \frac{1}{\lambda_2} \mathbf{v}_2 \mathbf{u}_2^{\mathbf{t}} \\
&= \begin{bmatrix} 2.7305090 & 2.468944 \\ 2.4685944 & 2.2318030 \end{bmatrix} + \begin{bmatrix} 897.26951 & -992.46859 \\ -992.46859 & 1097.7681 \end{bmatrix} \\
&= \begin{bmatrix} 900.00002 & -990.00000 \\ -990.00000 & 1099.9999 \end{bmatrix} \approx \begin{bmatrix} 900 & -990 \\ -990 & 1100 \end{bmatrix}.
\end{aligned} \tag{8-8}
$$

Notice that the rank of matrix $\mathbf{v}_2 \mathbf{u}_2^{\mathbf{t}}$, which was only available to about 5 digits in the approximation of **A**, is the largest component of $\mathbf{A}^{-1}$. One should expect that five digits would be about the most one could obtain by numerically inverting the approximate matrix.

The true inverse can be obtained using rational number arithmetic, and is shown in Eq.(8-7) to the right of approximation sign. Using eight digit arithmetic, the approximate matrix is inverted, yielding

$$= \begin{bmatrix} 0.11111111 & 0.10000000 \\ 0.10000000 & 0.090909094 \end{bmatrix}^{-1} = \begin{bmatrix} 900.00089 & -990.00099 \\ -990.00099 & 1100.0011 \end{bmatrix}. \qquad (8\text{-}9)$$

The poorest terms in this approximate inverse are the off-diagonal terms which have barely 6 significant digits. For this matrix

$$\log_{10} |\lambda_{max}| \,/\, |\lambda_{min}| = \log_{10} 402.0379 = 2.604.$$

Therefore one should expect the approximate inverse to be limited to 8 _ 2.6 = 5.4 good digits. It should be mentioned that for positive definite and symmetric matrices the calculation of $|\lambda_{max}| \,/\, |\lambda_{min}|$ can be carried out by the power method, using Rayleigh´s Quotient. Since a structural matrix **A** is symmetric and positive definite, therefore the convergence of the procedure is ensured and the largest eigenvalue λ_{max} of **A** can easily be calculated. The largest eigenvalue of $\mathbf{A}^{-1}$ provides the smallest eigenvalue of **A**. This method becomes especially simple if the inverse of the matrix is obtained as part of the calculation. However, the inversion of **A** can be avoided by using the fact that if the eigenvalues of **A** are λ_{min}, ... , λ_{max} , then the eigenvalues of c**I** _ **A** are c_λ_{min},..., c_ λ_{max}. Therefore if constant c is greater than λ_{max}, then the largest eigenvalue of c**I** _ **A** will be c_λ_{min} . This provides a simple approach for evaluating λ_{min}. Simple computer programs for calculating λ_{min} and λ_{max} of a positive definite matrix are provided (see answer to problem 8.6).

Some other condition numbers used in this book are described in the following.

8.2.2 DETERMINANT OF A ROW-NORMALIZED MATRIX

A simple and workable measure of conditioning of a set of equations is to evaluate the determinant of the row-normalized matrix of the coefficients of the set. This means that each row of matrix **A**, say A_i, is divided by

$$[\, a_{i1}^2 + a_{i2}^2 + \ldots + a_{in}^2]^{1/2}\,. \qquad (8\text{-}10)$$

The magnitude of the determinant of the row-normalized **A**, denoted by PN, is a good measure for the conditioning of **A**. Obviously the magnitude of this determinant lies in the range

$$0 < PN \leq 1,$$

since **A** is necessarily positive definite. The matrix with perfect conditioning has PN = 1, which occurs in the case of orthogonal or diagonal matrices.

8.2.3 THE RATIO OF DETERMINANTS

Since the best conditioned matrix for inversion is a diagonal one, the following parameter may also be adopted as a practical means to measure the conditioning of a matrix **A**. Define

$$\in = \det[\mathbf{A}] - \det[A_{ii}], \qquad (8\text{-}11)$$

where $[A_{ii}]$ is a diagonal matrix consisting of the diagonal entries of [**A**] and det means determinant.

The value of $\in$ approaches zero for an ideally conditioned matrix. Therefore the following condition number, PDET, can be employed.

$$\mathrm{PDET} = \det[\mathbf{A}] / \det[A_{ii}]. \qquad (8\text{-}12)$$

For an ideally conditioned matrix, PDET should approach unity. This condition number is simple and very easy to calculate.

8.3 WEIGHTED GRAPH AND AN ADMISSIBLE MEMBER

The relative stiffnesses (or flexibilities) of members of a structure can be considered as positive integers associated with the members of the graph model of a structure, resulting in a *weighted graph*.

Let S be the model of a frame structure and $\mathbf{k}_{m_i}$ denote the stiffness matrix of an element m_i in a global coordinate system selected for the structure. A *weight* can be defined for m_i, using the diagonal entries k_{ii} of $\mathbf{k}_{m_i}$ as

$$W(m_i) = \Sigma k_{ii} = 2\,(\alpha_1 + \alpha_4^z + \alpha_3^z), \qquad (8\text{-}13)$$

where $\alpha_1 = \dfrac{EA}{L}$, $\alpha_4^z = \dfrac{12EI}{L^3}$ and $\alpha_3^z = \dfrac{4EI}{L}$.

A different weight employing the square roots of the diagonal entries of $\mathbf{k}_{m_i}$ can also be used

$$W(m_i) = \Sigma\sqrt{k_{ii}} = 2\,[(\alpha_1)^{1/2} + (\alpha_4^z)^{1/2} + (\alpha_3^z)^{1/2}]. \tag{8-14}$$

Other weight functions may be defined for representing the relative stiffnesses of the members of S, as appropriate.

DEFINITION: Let the weight of members m_1, m_2, ..., $m_{M(S)}$ be defined by $W(m_1)$, $W(m_2)$,..., $W(m_{M(S)})$, respectively. A member m_i is called *F-admissible* if

$$W(m_i) \geq \frac{1}{\alpha} \sum_{j=1}^{M(S)} \frac{W(m_j)}{M(S)} \,, \tag{8-15}$$

where α is an integer number which can be taken as 2,3, ... We have used α=2; however, a complete study using other values of α is required. If a member is not F-admissible, it is called *inadmissible* or *S-admissible*.

8.4 OPTIMALLY CONDITIONED CYCLE BASES

In order to optimize the conditioning of flexibility matrices, special statical bases, correspondingly cycle bases possessing particular properties, must be selected.

A cycle basis is defined as an *optimally conditioned cycle basis* if:

(a) It is an optimal cycle basis, i.e. the number of nonzero entries of the corresponding cycle adjacency matrix is minimum, leading to a maximal sparsity of the flexibility matrix.

(b) The members of greatest weight of S are included in the overlaps of the cycles; i.e. the off diagonal terms of the corresponding flexibility matrix have the smallest possible magnitudes.

A weighted graph may have more than one optimal cycle basis. The one satisfying condition (b) is optimally conditioned. However, if no such a cycle basis exists, then a compromise should be found in satisfying conditions (a) and (b). In other words, a basis should be selected which partially satisfies both conditions. Since there is no algorithm for the formation of an optimal cycle basis, one should look only for a suboptimally conditioned cycle basis.

EXAMPLE: Consider a 3×3 grid as shown in Fig. 8.1(a), with the relative weights of the members being encircled. An optimal cycle basis of S contains 9 regional cycles (mesh basis) and corresponds to

$$L_T = \sum_{i=1}^{8} L(C^i \cap C_{i+1}) = 1+1+1+2+2+1+2+2=12.$$

The weight of the members contained in the overlaps is determined as

$$W_T = \sum_{i=1}^{8} W(C^i \cap C_{i+1}) =2+2+10+12+12+1+3+3 = 45,$$

where L_T and W_T are the *length* and *weight* of the overlaps of the selected cycles, respectively.

(a) A weighted graph S.

(b) An optimal cycle basis of S.

(c) A suboptimally conditioned cycle basis.

Fig. 8.1 A single layer rigid-jointed grid S.

A suboptimal cycle basis of S is illustrated in Fig. 8.1(c) for which

$$L'_T = \sum_{i=1}^{8} L(C^i \cap C_{i+1}) = 1+1+1+2+2+2+4+4=17.$$

The weight of the members contained in the overlaps is calculated as

$$W'_T = \sum_{i=1}^{8} W(C^i \cap C_{i+1}) = 2+2+10+12+12+14+16+16 = 84.$$

The weight of the overlaps of the selected cycles is considerably increased at the expense of some increase of their lengths, hence some decrease in the sparsity of its cycle adjacency matrix. Obviously W_T can be further increased; however, the decrease of sparsity will significantly influence the optimality of the cycle basis.

In this structure, the members of weight 1 are inadmissible according to the definition of the previous section, since $1 < \frac{1}{2} \times \frac{69}{24} = 1.43.$

8.4.1 FORMULATION OF THE PROBLEM

The problem of selecting an optimally conditioned cycle basis can be stated in the following mathematical form

$$\text{Min} \sum_{i=1}^{b1(\overline{S})-1} L(C^i \cap C_{i+1}),$$

and

$$\text{Max} \sum_{i=1}^{b1(\overline{S})-1} W(C^i \cap C_{i+1}),$$

where $\overline{S}$ is a contracted S as defined in Chapter 6, and $C^i = \bigcup_{j=1}^{i} C_j$.

As can be seen, the problem is a multi-objective optimization problem, and the following algorithms are designed such that both objective functions are partially satisfied simultaneously.

8.5 SUBOPTIMALLY CONDITIONED CYCLE BASES

In this section three algorithms are developed for the selection of suboptimally conditioned cycle bases of a weighted graph. On each selected cycle three or six S.E.Ss are formed depending on S being a planar or a space frame, respectively. The condition number of the flexibility matrix corresponding to the selected statical basis is obtained using the methods of Section 8.2.

8.5.1 ALGORITHMS

ALGORITHM A

This algorithm uses the chords of a special spanning tree to ensure the independence of the selected cycles. In order to avoid the inclusion of inadmissible chords in intersections of the cycles, such chords are not added to the set of members to be used for generation of the cycles of S.

Step 1. Select the centre "O" of S with a graph or algebraic graph theoretical method as described in Chapter 7.

Step 2. Generate an SRT using the members of highest weights, i.e.

2.1 Take all members incident with O and assign "1" to the other ends.

2.2 Find all members incident with nodes denoted by "1" and order them in ascending magnitude of their weights.

2.3 Select the tree members from the above ordered members, and assign "2" to the other ends.

Step 3. Repeat Step 2 as many times as needed until all the nodes of S are spanned and an SRT is formed.

Step 4. Order the members incident with "1" in ascending magnitude of weight and use the members of maximal weight as the chord of the first minimal length cycle. If this chord is an F-admissible one, add it to the list of the tree members, and denote this list by T^c.

Step 5. Generate the second shortest length cycle on the second maximal weight member incident with "1" using the members of T^c. Again add the chord to T^c if it is

F-admissible. Continue this process until all the chords incident with the nodes labelled as "1" are used.

Step 6. Repeat Steps 4 and 5 for all the nodes labelled by "2". Repeat this process sequentially for all the nodes labelled by 3,4,...,k, until a basis is selected.

This algorithm generates suboptimally conditioned cycle bases, and has the following advantages compared with the algorithm for generating a fundamental cycle basis.

(a) Starting node at the centre of S, limits the length of the generated cycles.

(b) Employing the used chords in the formation of cycles: reduces the length of the selected cycles.

(c) Forbidding the addition of F-inadmissible chords: prevents the inclusion of weak members in the overlaps of the cycles.

(d) Using members of highest weight in each stage of generating an SRT: leaves the weaker members as chords, which can be excluded because of inadmissibility.

One can select an spanning tree of maximal weight employing the Greedy Algortihm (see Chapter 9), in place of an SRT of maximal weight with respect to the centre node of S; however, in general, longer cycles will then be selected corresponding to a cycle adjacency matrix of less sparsity.

An improvement may be achieved by comparison of the centre node (or nodes) and adjacent nodes to select a node of higher average weight as a starting node. The average weight of a node is taken as

$$\Sigma \text{ weights of the members incident with } n_i / \deg n_i.$$

This improvement is due to the inclusion of all the members of the root node in T^c.

EXAMPLE: In the following a simple grid is considered, and the drawback of using an spanning tree of maximal weight compared with an SRT of maximal weight rooted at the centre node O is illustrated. The inadmissible members are shown in dashed lines, and the selected trees are illustrated in bold lines. Using an spanning tree results in much longer cycles, corresponding to a less sparse cycle adjacency matrix **D**. This in turn leads to a conditioning of **G** which in general is worse than the result obtained by an SRT.

(a) A cycle basis generated using an SRT. (b) A cycle basis using an spanning tree.

Fig. 8.2 Comparison of two different cycle bases.

ALGORITHM B

In this algorithm, the formation of disjoint cycles is permitted, to enable the generation of short cycles. However, the simplicity of the independence check in Algorithm A can not be maintained.

Step 1. Assign weights to the members of S and order them in ascending order of their weights.

Step 2. Select a member of minimal weight m_1^g and form a cycle of maximal possible weight out of the existing minimal length cycles on m_1^g.

Step 3. Select the next admissible cycle of minimal length on the second unused member of minimal weight, having the maximum possible weight and containing members from S - m_1^g.

Step 4. Repeat the process of Step 3 selecting an admissible cycle C_j of minimal length which has maximal possible weight, containing members from $S - \bigcup_{i=1}^{j-1} m_i^g$.

Step 5. Repeat Step 4 until a set of $b_1(S)$ independent cycles forming a suboptimally conditioned cycle basis is formed.

In this algorithm also the formation of disjoint cycles enables the formation of short cycles at early stages of the algorithm; however, excluding m_i^g after forming the corresponding cycle leads to the generation of longer cycles in later stages.

EXAMPLE: Consider a simple 3×3 grid with weights assigned to its members, as shown in Fig. 8.3(a).

(a) A weighted graph. (b) The generated cycle basis.

Fig. 8.3 A 3×3 grid and the selected cycle basis.

The cycles generated are depicted in Fig. 8.3(b), in which 7 cycles of length 4 are formed. However, for cycles 8 and 9, because of excluding $\bigcup_{i=1}^{7} m_i^g$, the generated cycles are long, leading to a less sparse **D** matrix.

ALGORITHM C

This algorithm is a modified version of Algorithm 3 presented in Chapter 6 (p.175) for selecting a suboptimal cycle basis of S, in which the relative stiffnesses of the members are also taken into account.

Step 1. Contract S into S′ by replacing all paths with nodes of degree 2 by a single member. If a path contains an F-inadmissible member, then the replaced member will also be taken as F-inadmissible.

Step 2. Calculate the incidence number and cycle length number of the members of S.

Step 3. Start with a member of the least cycle length number and generate a minimal weight cycle C_1 on this member. The weight of a cycle in this algorithm is taken as the sum of the incidence numbers of its members.

Step 4. Generate the second admissible cycle of minimal weight C_2 on the next member of the least cycle length number. If $C_1 \cap C_2$ contains an F-inadmissible member, and $C_1 \oplus C_2$ does not contain such a member, then replace C_2 by $C_1 \oplus C_2$ otherwise take C_2 as the second cycle of the basis.

Step 5. Subsequently select the kth admissible smallest weight cycle C_k on the unused member by the least cycle length number. If $C^{k-1} \cap C_k$ contains an F-inadmissible member, and $C_k \oplus C_j$ does not have such a member, then replace C_k by $C_k \oplus C_j$, otherwise take C_k as the kth cycle. In the above relationship C_j are all the generated cycles adjacent to C_k.

Step 6. The process of Step 5 should be continued as far as the generation of admissible minimal weight cycles is possible. After a member has been used as many times as its incidence number, before each extra usage, increase the incidence number of such a member by unity.

Step 7. On an unused member of the least length number, generate one admissible cycle of the smallest weight. This cycle is not a minimal weight cycle, otherwise it would have been selected at Step 6. Such a cycle is known as a subminimal weight cycle. Again a process similar to Step 5 should be performed for possible interchange of the cycle, and the incidence numbers should be updated for each extra usage. Now Step 6 should be repeated, since the formation of the new subminimal weight cycle may have altered the admissibility condition of the other cycles, and the selection of further minimal weight cycles may now have become possible.

Step 8. Repeat Step 7, selecting minimal and subminimal weight cycles with the process of combining for better conditioning, until $b_1(S') = b_1(S)$ cycles are generated.

Step 9. A reverse process to that of the contraction performed in Step 1 transforms the selected cycle basis of S′ to that of S.

This algorithm is implemented on a PC and the improvements obtained on the conditioning of the flexibility matrices by using this method are studied by some examples. The results are compared with those of Algorithm 3 of Chapter 6.

8.5.2 EXAMPLES

EXAMPLE 1: A three - storey frame is considered, as shown in Fig. 8.4. Three cases are studied using two types of member properties:

Type 1 $A_1 = 0.00106\text{m}^2$ $I_1 = 0.00000171\text{m}^4$

Type 2 $A_2 = 0.00970\text{m}^2$ $I_2 = 0.0001961\text{m}^4$.

(a) (b) (c)

Fig. 8.4 Three - storey frames with different member properties.

The elastic modulus of the material is taken as $E = 2.1 \times 10^8$ kN/m^2 and all the members have L=3m. Type 1 members are shown in normal lines and type 2 members are illustrated in bold lines.

Algorithm 3 of Chapter 6 is applied to these frames and in all the cases regional cycles are formed as an optimal (minimal) cycle basis. For each cycle 3 S.E.Ss are generated, and $\mathbf{B}_1$ and the corresponding flexibility matrices $\mathbf{G}$ are formed. The condition numbers for these matrices are listed in Table 8.1.

Type	PL	PN	PDET
(a)	1.971889	9.934432E_6	7.932374E_4

(b)	3.611656	9.322886E_11	4.257163E_8
(c)	3.692658	6.496687E_13	4.944418E_10

Table 8.1 Condition numbers for Example 1.

Algorithm C of this chapter selected the following cycles as a suboptimally conditioned cycle basis:

For (a) C_1 = (7,9,1), C_2 = (4,5,8,3) and C_3 = (2,8,6,9).

For (b) C_1 = (7,9,1), C_2 = (4,5,8,3) and C_3 = (2,8,6,7,1).

For (c) C_1 = (7,9,1), C_2 = (4,5,8,3) and C_3 = (2,8,6,7,1).

The corresponding flexibility matrices have condition numbers as listed in Table 8.2.

The improvements in PN and PDET are apparent and a slight increase in PL shows the inconsistency of the three different condition numbers.

Type	PL	PN	PDET
(a)	1.971889	9.934432E_6	7.932374E_4
(b)	4.160444	1.438844E_7	1.224165E_5
(c)	3.883811	1.915509E_7	2.248523E_5

Table 8.2 Improved condition numbers for Example 1.

EXAMPLE 2: A two - storey frame with three bays is considered, as shown in Fig. 8.5. The same member properties are used and three cases are studied. The calculated condition numbers are listed in Table 8.3. For this purpose Algorithm 3 of Chapter 6 is used and the selected cycle bases are optimal for all three cases.

(a) (b) (c)

Fig. 8.5 Two - storey frames with different member types.

Type	PL	PN	PDET
(a)	2.95416	2.870534E_14	7.828693E_10
(b)	4.504203	5.23706E_28	1.030332E_24
(c)	4.311532	5.23706E_28	8.155067E_22

Table 8.3 Condition numbers for Example 2.

Algorithm C (Section 8.5.3) is applied, and the selected cycles for each case are illustrated in Fig. 8.6. The corresponding flexibility matrices have the condition numbers as listed in Table 8.4.

Type	PL	PN	PDET
(a)	2.942885	3.07853E_14	7.828629E_10
(b)	3.770917	9.75069E_18	2.023529E_14
(c)	3.742143	1.04041E_14	1.356510E_9

Table 8.4 Improved condition numbers for Example 2.

Fig. 8.6 Selected cycle bases using Algorithm C.

The considerable improvement is due to the formation of suboptimal cycle bases used in place of optimal cycle bases. It should be noted that these comparisons are made against the best existing algorithm, since sparsity itself has a great influence on the conditioning of flexibility matrices.

8.6 OPTIMALLY CONDITIONED CUT SET BASES

In order to optimize the conditioning of the stiffness matrices, special cut set bases must be used in the formation of kinematical bases.

A cut set basis with the following properties is defined as an *optimally conditioned cut set basis*:

(a) It is an optimal cut set basis; i.e. the number of nonzero entries of its cut set adjacency matrix and the corresponding number of nonzero entries of its stiffness matrix are minimum.

(b) The members of lowest weight of S are included in the overlaps of the cut sets; i.e. the off diagonal terms of the corresponding stiffness matrix have the smallest possible magnitudes.

A weighted graph may or may not have an optimally conditioned cut set basis. However, if such an basis does not exist or cannot be found, then a compromise should be found to satisfy the above two conditions; i.e. a basis which satisfies both conditions partially should be selected.

8.6.1 MATHEMATICAL FORMULATION OF THE PROBLEM

The cardinality of a cut set basis for a connected graph is given by

$$\rho(S) = N(S) - 1. \qquad (8\text{-}16)$$

The problem of finding an optimally conditioned cut set basis can then be stated as

Select a cut set basis $\{C^*_1, C^*_2, \ldots, C^*_{\rho(S)}\}$ such that

$$L_s = \text{Min} \sum_{i=1}^{\rho(S)-1} L(C^{*i} \cap C^*_{i+1}),$$

and

$$W_s = \text{Min} \sum_{i=1}^{\rho(S)-1} W(C^{*i} \cap C^*_{i+1}), \qquad (8\text{-}17)$$

where $C^{*i} = \bigcup_{j=1}^{i} C^*_j$, L denoting the length and W indicating the weight of the members of $C^{*i} \cap C^*_{i+1}$, respectively.

Again we have a multi-objective optimization problem, which is not so easy to solve. Therefore we design an algorithm which is practical and satisfies partially the required conditions.

8.7 SUBOPTIMALLY CONDITIONED CUT SET BASES

A fundamental cut set basis of a graph can easily be generated using each branch of a spanning tree as the generator of a cut set. A more common cut set basis, employed in the displacement method of structural analysis, is a cocycle basis of S. For this basis each element simply isolates a node of S, except the ground node.

Although a cocycle basis corresponds to a rather sparse cut set adjacency matrix, other cut set bases corresponding to more sparse cut set adjacency matrices, leading to more sparse stiffness matrices, can be generated. As an example, consider a frame model S as depicted in Fig. 8.7(a) for which a cocycle basis and a cut set basis are selected, as illustrated in Figs 8.7(b) and (c), respectively. The patterns of the corresponding cut

(a) (b) (c)

Fig. 8.7 A planar frame S, a cocycle basis and a cut set basis of S.

set adjacency matrices are shown in the next page using * for nonzero entries.

$$\mathbf{D}_1^* = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & * & * & * & * & & \\ 2 & * & * & & * & & \\ 3 & * & & * & * & * & * \\ 4 & * & * & * & * & & * \\ 5 & & & * & & * & * \\ 6 & & & * & * & * & * \end{array} \qquad \mathbf{D}_2^* = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & * & * & & & & \\ 2 & * & * & * & & & \\ 3 & & * & * & * & & \\ 4 & & & * & * & * & \\ 5 & & & & * & * & * \\ 6 & & & & & * & * \end{array}$$

$$\chi(\mathbf{D}_1^*) = 24 \qquad \chi(\mathbf{D}_2^*) = 16$$

It will be realised that sparser stiffness matrices can be generated using suitable cut set bases than by employing the traditional cocycle basis, Ref. [95].

In order to keep the off-diagonal terms small, the members in the overlaps of the cut sets should be as flexible as possible; i.e. lower weight members should be included in the overlaps. In the following three algorithms are designed for the formation of suboptimal cut set bases of the graph model of the structures.

8.7.1 ALGORITHMS

The formation of a cocycle basis of a graph model S is simple and straight foreword. For this purpose the members incident with each free node (except the selected datum node) are taken as an element of the basis. Repeating this operation for all the free nodes, completes the process of the generation.

ALGORITHM A

Step 1. Generate a spanning tree of maximal weight . Order its members (branches) in ascending magnitude of weight.

Step 2. Use a branch of the least weight, form the selected tree and form the first fundamental cut set on this branch

Step 3. Form the next fundamental cut set on the unused branch of the least weight.

Step k. Repeat Step 3 for the other unused branches until ρ(S) = N(S) − 1 independent cut sets forming a basis is generated.

ALGORITHM B

Step 1. Form a cocycle basis; denote the selected cocycles by C^{*1}.

Step 2. Take the first cocycle C_i^* of C^{*1} and combine with the remaining cocycles of C^{*1}. For each cocycle C_j^* (j=2, ... ,ρ(S)) satisfying the following condition, replace C_j^* with $C_i^* \oplus C_j^*$.

$$\text{Condition: } (L_{s2} < L_{s1}) \text{ or } (L_{s2} = L_{s1} \text{ and } W_{s2} < W_{s1}),$$

where $L_{s1}(W_{s1})$ and $L_{s2}(W_{s2})$ indicate the lengths (weights) before and after the application of the combining process. The new set of cocycles and/or cut sets are denoted by C^{*2}.

Step 3. Take C_2^* of C^{*2} and repeat a process similar to that of Step 2.

Step. k. Take C_k^* of C^{*k-1} and combine with the elements of C^{*k-1}. The process terminates when k becomes equal to ρ(S).

ALGORITHM C

This algorithm is the same as Algorithm B, with the difference that the corresponding condition is replaced by the following one:

$$\text{Condition: } (W_{s2} < W_{s1}) \text{ or } (W_{s2} = W_{s1} \text{ and } L_{s2} < L_{s1}).$$

The selected bases are suboptimal and contain elements with lower weight members leading to kinematical bases corresponding to small off-diagonal terms for stiffness matrices.

8.7.2 EXAMPLE

A one bay four - storey planar truss is considered as shown in Fig. 8.8, with cross - sections being designated by A_i. Typical member cross - sections are

$A_1 = 20$ cm2, $A_2 = 10$ cm2, $A_3 = 5$ cm2, $A_4 = 4$ cm2, and $E = 2.1 \times 10^4$kN/cm2.

(a) A planar truss. (b) The graph model S.

Fig. 8.8 A planar truss and its graph model S.

The patterns of the cut set bases adjacency matrices are illustrated in the following.

$$\begin{array}{ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & * & * & * & & & & & \\
2 & * & * & * & * & * & & & \\
3 & * & * & * & * & * & & & \\
4 & & * & * & * & * & * & & \\
5 & & & * & * & * & * & * & \\
6 & & & & * & * & * & * & * \\
7 & & & & & * & * & * & * \\
8 & & & & & & * & * & *
\end{array}$$

Pattern of $\mathbf{D}^*$ by a cocycle basis.

$$\begin{array}{ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & * & & & & * & * & . & . \\
2 & & & * & * & & & * & * \\
3 & & * & * & * & & * & * & * \\
4 & & * & * & * & & * & * & * \\
5 & * & & & & * & * & & \\
6 & * & * & & * & * & * & & \\
7 & & & * & * & & & * & * \\
8 & & & * & * & & & * & *
\end{array}$$

Pattern of $\mathbf{D}^*$ by Algorithm A.

$$\begin{array}{ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & * & * & & & & & & \\
2 & * & * & * & & & & & \\
3 & & * & * & * & & & & \\
4 & & & * & * & * & & & \\
5 & & & & * & * & * & & \\
6 & & & & & * & * & * & \\
7 & & & & & & * & * & * \\
8 & & & & & & & * & *
\end{array}$$

Pattern of $\mathbf{D}^*$ by Algorithm B.

$$\begin{array}{ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & * & * & & & & & . & . \\
2 & * & * & * & & & & & \\
3 & & * & * & * & & & & \\
4 & & & * & * & * & & & \\
5 & & & & * & * & * & & \\
6 & & & & & * & * & * & * \\
7 & & & & & & * & * & * \\
8 & & & & & & * & * & *
\end{array}$$

Pattern of $\mathbf{D}^*$ by Algorithm C.

The condition numbers of stiffness matrices, the sparsity, and the magnitudes of L_s and W_s for the selected cut set bases are illustrated in Table 8.5.

Algorithm	PL	PN	PDET	$\chi(\mathbf{D}^*)$	L_s	W_s
Coc. basis	2.720131	2.296570E–7	9.284245E–6	34	13	75936.5
A	2.145762	1.093639E–5	1.134908E–3	32	12	48048.7
B	2.502612	8.509724E–4	9.406178E–3	22	7	46200.0
C	2.245613	2.702627E–4	3.457425E–3	24	8	36400.0

Table 8.5 Comparison of the condition numbers and sparsities.

The execution time for the formation of the selected cut set bases (T_C) and the corresponding stiffness matrices (T_K) are are presented in Table 8.6.

Time	Cocycle basis	A	B	C
T_C	0.00	0.88	0.76	o.88
T_K	0.43	0.65	0.48	0.60

Table 8.6 Comparison of the computational time.

Although the sparsity of stiffness matrices **K** can be improved by the formation of special cut set bases in place of cocycle bases, the improvements, in general, are not significant. On the other hand the conditioning of **K** can be improved by employing appropriate cut set bases. Algorithm B improves the conditioning of the stiffness matrices, maintaining the sparsity of the stiffness matrices. This improvement is more significant for Algorithms A and C, although the sparsity of **K** is not maintained.

EXERCISES

8.1 For the grid shown in the following, the relative stiffnesses are encircled. Find a suboptimally conditioned cycle basis of this model.

8.2 Find the condition numbers PL, PDET and PN of the stiffness matrix of the following planar frame.

8.3 Repeat Exercise 8.2 for the flexibility matrix of the same structure.

8.4 Study the effect of bandwidth reduction on the stiffness matrices of structures and find out whether this effect is significant. Illustrate this fact with a simple matrix chosen arbitrarily.

8.5 Use Algorithms A,B and C to find suboptimally conditioned cycle bases for the following weighted graphs. The numbers 1 and 2 show the member types as given in Section 8.5.4.

(a) (b)

8.6 Write a computer program to calculate the largest and the smallest eigenvalues for adjacency matrices of graphs.