UNIVERSITY OF CAPE TOWN

DEPARTMENT OF CIVIL ENGINEERING

RONDEBOSCH, CAPE TOWN, SOUTH AFRICA

LIMIT ANALYSIS AND SHAKEDOWN IN PLANE FRAMES AND PLANE STRESS PROBLEMS

by

A.S. Douglas

A thesis submitted in partial fulfilment
of the requirements for the degree
Master of Science in Engineering

September 1977

DECLARATION OF CANDIDATE

I, Andrew Sholto Douglas, hereby declare that this thesis is my own work and that it has not been submitted for a degree at another university.

Signed by candidate

September 1977

BUT 620 DOU

ABSTRACT

A method is developed for the determination of the shakedown load factor for elastic, perfectly plastic plane frames subjected to cyclic loading or random loads varying between fixed limits. The essential feature of the method is the employment of an automatic force method of elastic analysis which provides self-stress systems for the frame. This in turn permits the ready formulation of the compatibility requirements which are imposed on plastic hinge rotations. As a result, the analysis proceeds with data input which is comparable to straightforward kinematic analysis. A preliminary study of the generalisation of this approach to the limit analysis of plane stress problems is also given.

ACKNOWLEDGEMENTS

The writer wishes to thank Professor J.B. Martin for his guidance throughout the thesis, and in particular for his help and constructive criticism on the manuscript.

The writer is indebted to the Council for Scientific and Industrial Research, the University of Cape Town and the Ninham Shand Trust Fund for financial assistance during this project.

Thanks are also due to Lee Behm, for typing the manuscript, to Rod Douglas for the diagrams, to Davide Hanan for help with the proof-reading, to Harold Cable for reproducing copies of the thesis and to my wife for her help and support especially towards the end of the project.

LIST OF SYMBOLS

Special Symbols

{ }	a column vector
[ ]	a matrix
T	superscript, the transpose of a vector or matrix
| |	determinant of a matrix, or the modulus of a scalar
•	(above a symbol) derivative with respect to time

Lower Case Symbols

k	number of plastic deformation positions
kj	number of joints (or nodes)
kℓ	number of loading parameters
km	number of members (or elements)
ℓ	length
m,n	dimensions of matrices and vectors
t	time
u,v	displacements

Upper Case Symbols

E	Youngs modulus
I	Moment of Inertia
M	Moment
M_o	Limit Moment

N	Axial Force
P,Q	Forces
W	Work
X,Y	Cartesian Coordinate Directions

Greek

γ	angle of inclination
θ	plastic hinge rotation
δ	axial extension
λ	arbitrary constant
κ	curvature
σ	stress
ε	strain
ϕ	rotation
Γ	load parameter
Σ	summation

Matrices and Vectors

$[A]$	static matrix
$[\bar{A}]$	row-echelon form of $[A]$
$[A']$	dependent columns of $[\bar{A}]$
$[B]$	deformation matrix
$[F_e]$	element flexibility matrix
$[F]$	global flexibility matrix
$[I]$	identity matrix
$\{\ell\}$	individual compatibility requirements
$[L]$	set of compatibility requirements

$\{N_e\}$	element internal force vector
$\{N\}$	internal force vector
$\{\bar{N}\}$	reordered force vector
$\{N_I\}$	internal forces (not redundant)
$\{N'\}$	internal forces (redundant)
$\{r\}$	self-stress systems
$\{s\}$	self equilibrating moment systems
$[S_e]$	element stiffness matrix
$[S]$	global stiffness matrix
$\{u_a\}$	displacements at node a
$\{u\}$	general displacements
$\{\delta_e\}$	element deformations
$\{\delta\}$	general deformations
$\{\varepsilon_e\}$	element strains
$\{\varepsilon\}$	general strains
$\{\sigma_e\}$	element stresses
$\{\sigma\}$	general stresses
$\{\sigma_o\}$	yield stresses
$[\Delta]$	diagonal matrix of element areas

<u>Subscripts</u>

i,j	positions
p	plastic
int	internal
ext	external
max	maximum
min	minimum

N Axial Force

P,Q Forces

W Work

X,Y Cartesian Coordinate Directions

Greek

γ angle of inclination

θ plastic hinge rotation

δ axial extension

λ arbitrary constant

κ curvature

σ stress

ε strain

ϕ rotation

Γ load parameter

Σ summation

Matrices and Vectors

$[A]$ static matrix

$[\bar{A}]$ row-echelon form of $[A]$

$[A']$ dependent columns of $[\bar{A}]$

$[B]$ deformation matrix

$[F_e]$ element flexibility matrix

$[F]$ global flexibility matrix

$[I]$ identity matrix

$\{\ell\}$ individual compatibility requirements

$[L]$ set of compatibility requirements

$\{N_e\}$	element internal force vector
$\{N\}$	internal force vector
$\{\bar{N}\}$	reordered force vector
$\{N_I\}$	internal forces (not redundant)
$\{N'\}$	internal forces (redundant)
$\{r\}$	self-stress systems
$\{s\}$	self equilibrating moment systems
$[S_e]$	element stiffness matrix
$[S]$	global stiffness matrix
$\{u_a\}$	displacements at node a
$\{u\}$	general displacements
$\{\delta_e\}$	element deformations
$\{\delta\}$	general deformations
$\{\varepsilon_e\}$	element strains
$\{\varepsilon\}$	general strains
$\{\sigma_e\}$	element stresses
$\{\sigma\}$	general stresses
$\{\sigma_o\}$	yield stresses
$[\Delta]$	diagonal matrix of element areas

Subscripts

i,j	positions
p	plastic
int	internal
ext	external
max	maximum
min	minimum

Superscripts

E	elastic
C	cycle
S	shakedown
*	limit analysis

LIST OF CONTENTS

			page
Declaration			(i)
Abstract			(ii)
Acknowledgements			(iii)
List of Symbols			(iv)
List of Contents			(viii)
CHAPTER 1:		INTRODUCTION	1
CHAPTER 2:		ELASTIC SOLUTION FOR PLANE FRAMES	4
	2.1	Formulation	4
	2.2	Computational Approach	15
CHAPTER 3:		SHAKEDOWN IN PLANE FRAMES	22
	3.1	Introduction	22
	3.2	Formulation	25
	3.3	Computational Approach	31
CHAPTER 4:		PLANE FRAME NUMERICAL EXAMPLES	34
		Example 4.1	34
		Example 4.2	36
		Example 4.3	38
		Example 4.4	40
		Conclusions	41

			page
CHAPTER	5:	ELASTIC SOLUTION FOR PLANE STRESS PROBLEMS	42
	5.1	Introduction	42
	5.2	Formulation	43
	5.3	Computational Approach	52
CHAPTER	6:	LIMIT ANALYSIS IN PLANE STRESS PROBLEMS	55
	6.1	Introduction	55
	6.2	Formulation	57
	6.3	Computational Approach	59
CHAPTER	7:	PLANE STRESS NUMERICAL EXAMPLES	62
		Example 7.1	62
		Conclusions	64
REFERENCES			65
APPENDIX	I:	Relationship between the statics matrix [A] and the deformation matrix [B] for plane frames	67
APPENDIX	II:	Relationship between the statics matrix [A] and the deformation matrix [B] for plane stress problems with constant strain triangular elements	68
APPENDIX	III:	Computer program for the shakedown analysis of plane frames	69
		Example of data input and computer printout for example 4.1(c)	87
		Explanation of data input	88
APPENDIX	IV:	Computer program for the limit analysis of plane stress problems	94
		Example of data input for limit analysis of plane stress problems	109
		Explanation of data input	110

CHAPTER 1

INTRODUCTION

Design procedures for steel structures are increasingly taking cognisance of the details of the response of the structure beyond the elastic range. For structures composed of a material which can be idealised as elastic, perfectly plastic, particular significance is given to the magnitude of static loads which lead to unconstrained plastic flow in the structure, and to loading cycles which lead to failure as the result of alternating plastic deformation at any one point in the structure or unbounded deformations resulting from plastic flow occurring during each cycle of loading. The determination of the largest static load multipliers for which flow will not occur is referred to as the limit analysis problem, while the determination of the largest multiplier applied to any set of cyclic loads for which the steady state response of the structure is elastic is referred to as the shakedown analysis problem.

Analytical methods for the direct determination of the limit load multiplier and the shakedown load multiplier are well established (see, for example, Baker, Horne and Heyman [2], Hodge [5], Martin [9], Massonnet and Save [10] and Neal [11]). Limit analysis, it should be noted, can be considered as the special case of shakedown analysis when the load cycle degenerates to a single load state.

Less success has been achieved, however, in the formulation of efficient numerical method of analysis for both the limit and shakedown analysis of complex problems. Both problems can be reduced to programming problems in which an objective function must be minimised (or maximised) subject to sets of equality or inequality constraints. Numerical difficulties arise most commonly from the number of constraints, leading to large and cumbersome solution routines.

This thesis will be concerned with kinematic methods of analysis. Consideration is given first to the shakedown analysis of plane frames, where the starting point is the work of Cohn, Ghosh and Parimi [4]. The shakedown problem in plane frames is a linear programming problem; the objective function is formulated in terms of the potential plastic hinge rotations which might occur in the frame, and the constraints involve compatibility requirements imposed on the plastic hinge rotations. The formulation of the constraints has caused some difficulty; Cohn et al specify independent basic mechanisms of flow for the structure, a procedure which is time consuming insofar as data preparation is concerned, and which excludes failure due to alternating plasticity. While the latter point is not a serious omission, in the sense that the alternating plasticity shakedown load factor can be readily determined from an elastic analysis, it is clearly preferable if the incremental collapse and alternating plasticity shakedown load factors could be determined simultaneously.

In the first part of the thesis a new formulation of a numerical procedure for the kinematic method of shakedown analysis is presented in which the data input is in the form conventionally required for finite element analysis. In essence, the formulation involves a preliminary automatic force method elastic analysis of the frame. The force method analysis provides the elastic response, which is required in any case, and more importantly it provides the independent self-stress systems or self equilibrating internal force systems for the frame. These self-stress systems are subsequently employed to express the compatibility constraints imposed on the plastic hinge rotations in the shakedown analysis, thus permitting a rapid, simple and efficient formulation of the linear programming problem.

The essential feature of the force method of analysis presented is that it is automatic in the sense that redundants (or releases) are not chosen by the analyst, as is usual in the force method for frames (see, for example, Livesley [8]). The automatic force method of elastic analysis employed is similar to the work of Robinson [14] and Robinson and Regl [13], particularly in the sense that the data input is as straightforward as that required in kinematic elastic analysis. However, there are distinct differences in approach, notably in the equilibrium equations and in the formulation of compatibility equations.

In the second part of the thesis a preliminary study of the possibility of the extension of the ideas developed for plane frames to continuum problems is presented. The problem of plane stress is treated, with a simple linearised yield condition, and attention is confined to the limit analysis problem.

The extension depends essentially on the generalisation of the automatic force method of elastic analysis for the frame problem to the plane stress problem. A finite element discretisation, involving constant strain triangular elements, of the plane stress problem is adopted; hence the generalisation of the automatic force method is not strictly a stress method but rather a solution of the dual of the elastic kinematic finite element problem. The self-stress systems so generated are, however, appropriate for the formulation of elastic and plastic compatibility equations for the elastic analysis and the limit analysis problem respectively. In view of the linearised yield conditions, the limit analysis problem is a linear programming problem.

CHAPTER 2

ELASTIC SOLUTION FOR PLANE FRAMES

2.1 Formulation

We consider the elastic analysis of a plane frame with km labelled members and kj labelled joints or nodes. The structure is discretised by choosing the nodes in such a way that all members are straight, prismatic and unloaded between nodes. Thus all loads must act at the joints and any distributed load must be broken up into a series of point loads. The frame lies in a global cartesian co-ordinate system (shown in Figure 2.1) in which rotation is positive in the anticlockwise direction.

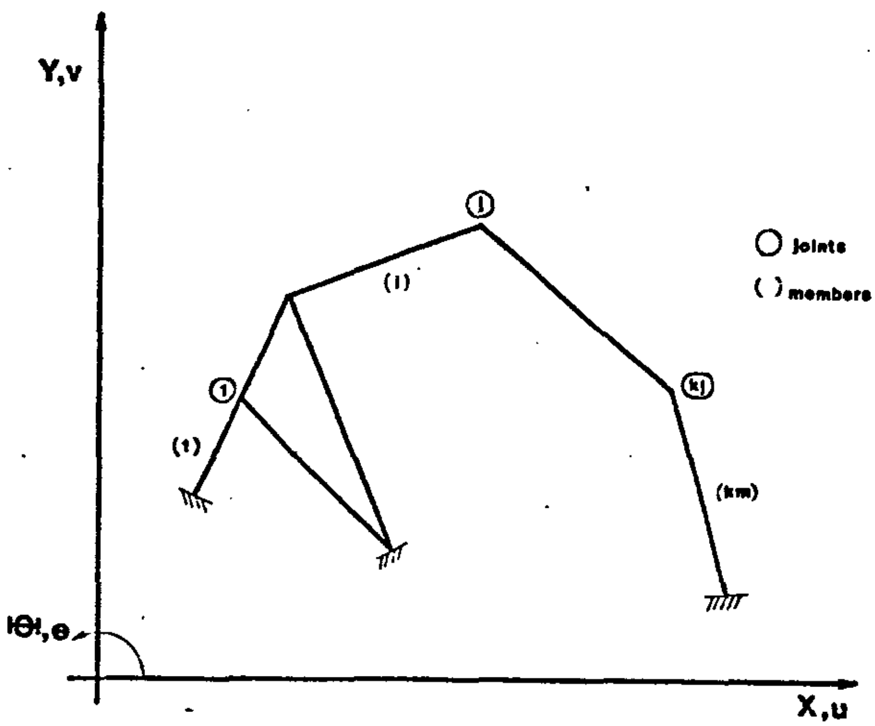

Figure 2.1 General Plane Frame

Each of the km ordered members is subjected to the internal forces shown in Figure 2.2. Owing to the fact that all members are straight and unloaded, the shear force is constant and may be expressed as the sum of the moments at each end divided by the length of the member.

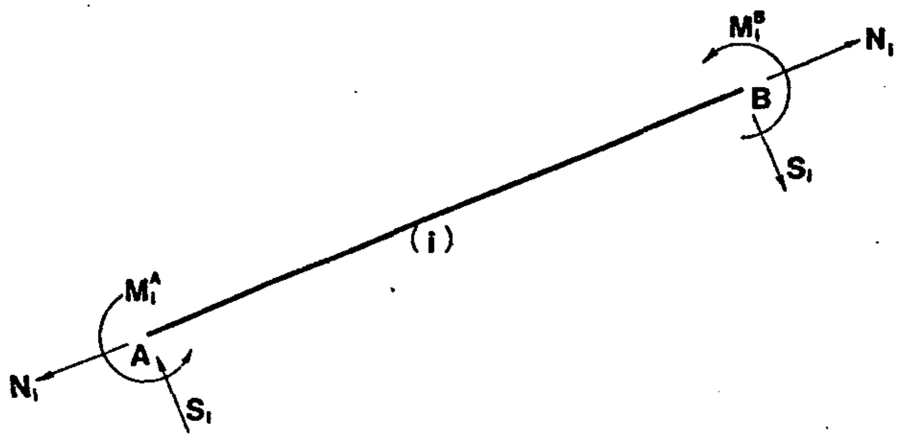

Figure 2.2 Member Forces

The independent member forces associated with each member (or element) may be represented by the (3 × 1) column vector $\{N_e\}_i$, where

$$\{N_e\}_i = \{N_i, M_i^A, M_i^B\}^T. \tag{2.1}$$

N_i is the axial force in member i and M_i^A and M_i^B are the moments at joints A and B respectively in member i. The (m × 1) column vector $\{N\}$ of all internal forces may be defined by

$$\{N\} = \{\{N_e\}_1^T, \{N_e\}_2^T, \ldots, \{N_e\}_i^T, \ldots, \{N_e\}_{km}^T\}^T. \tag{2.2}$$

Each member will suffer deformations conjugate to the member forces {N} as shown in Figure 2.3.

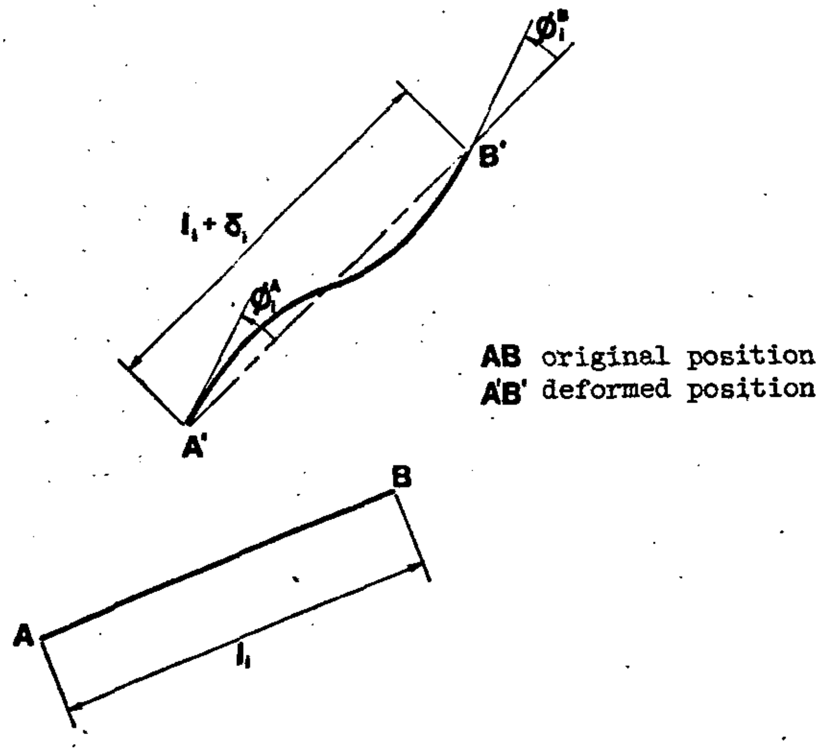

Figure 2.3 Member Deformation

The member deformations may be represented by the (3 × 1) column vector $\{\delta_e\}_i$, where

$$\{\delta_e\}_i = \{\delta_i, \phi_i^A, \phi_i^B\}^T. \tag{2.3}$$

δ_i is the axial extension of member i and ϕ_i^A and ϕ_i^B are the rotations of member i at joints A and B respectively, relative to $\overrightarrow{A'B'}$, the line joining the terminal nodes. The (m × 1) column vector {δ} of all member deformations may be defined by

$$\{\delta\} = \left[\{\delta_e\}_1^T, \{\delta_e\}_2^T, \ldots, \{\delta_e\}_i^T, \ldots, \{\delta_e\}_{km}^T\right]^T. \tag{2.4}$$

Let the (n × 1) column vector {u} represent the ordered unconstrained displacement components at each of the kj ordered unconstrained joints in the global cartesian coordinate system shown in Figure 2.1. It is assumed that the frame is supported by restraining some, or all, of the X, Y and θ displacement components to be zero at certain nodes. Hence

$$\{u\} = \{u_1, v_1, \theta_1, \ldots, u_i, v_i, \theta_i, \ldots, u_{kj}, v_{kj}, \theta_{kj}\}^T, \qquad (2.5)$$

where u_i, v_i, θ_i are the displacement components of joint i in the X, Y, θ directions respectively.

Similarly the (n × 1) column vector {P} represents the similarly ordered external load components at each node, in the same cartesian coordinate system (shown in Figure 2.4). These may be defined as

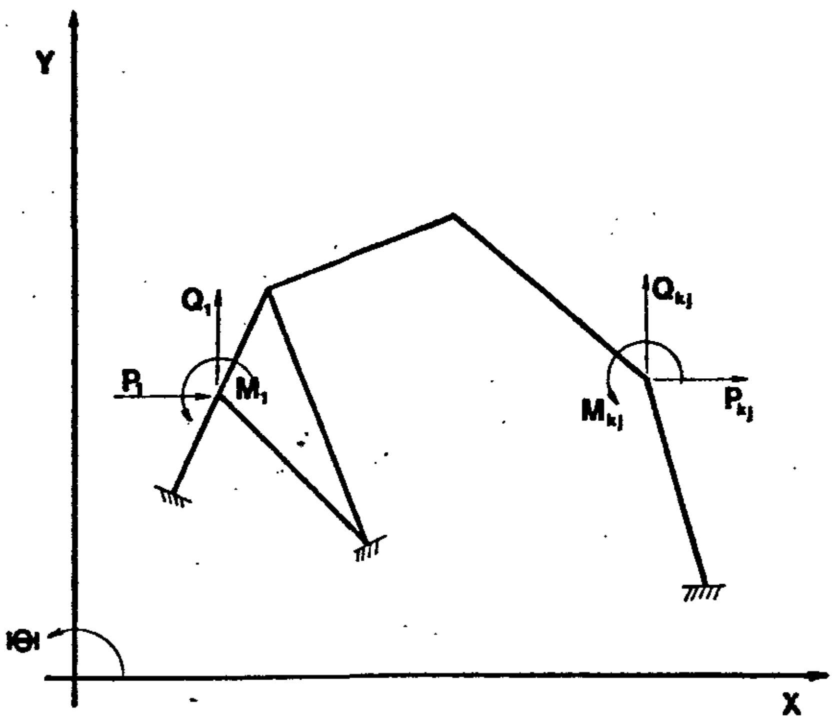

Figure 2.4 Frame Loading

$$\{P\} = \{P_1,Q_1,M_1,\ldots,P_i,Q_i,M_i,\ldots,P_{kj},Q_{kj},M_{kj}\}^T. \tag{2.6}$$

P_i,Q_i are components of the applied forces at node i in the X and Y directions respectively, while M_i is the applied moment at node i in direction θ.

The member forces $\{N_e\}_i$, defined in equation (2.1), can be related to the member deformations $\{\delta_e\}_i$, equation (2.3), by the (3 × 3) non-singular member stiffness matrix $[S_e]_i$. Thus

$$\{N_e\}_i = [S_e]_i\{\delta_e\}_i, \tag{2.7}$$

where

$$[S_e]_i = \begin{bmatrix} \frac{A_iE_i}{\ell_i} & 0 & 0 \\ 0 & \frac{4E_iI_i}{\ell_i} & \frac{2E_iI_i}{\ell_i} \\ 0 & \frac{2E_iI_i}{\ell_i} & \frac{4E_iI_i}{\ell_i} \end{bmatrix}. \tag{2.8}$$

A_i, I_i and E_i are the cross sectional area, moment of inertia about an axis normal to the plane of the frame and Youngs modulus for the member i, while ℓ_i is the length of member i.

We may thus relate the ordered member forces {N} to the similarly ordered member deformations $\{\delta\}$ by the tridiagonal nonsingular global stiffness matrix [S]. Thus

$$\underset{m\times 1}{\{N\}} = \underset{m\times m}{[S]}\ \underset{m\times 1}{\{\delta\}}, \tag{2.9}$$

where

$$[S] = \begin{bmatrix} [S_e]_1 & & & & 0 \\ & \ddots & & & \\ & & [S_e]_i & & \\ & & & \ddots & \\ 0 & & & & [S_e]_{km} \end{bmatrix}. \tag{2.10}$$

The inverse of the stiffness matrix [S] is the flexibility matrix [F], so that

$$\{\delta\} = [F]\{N\} \tag{2.11}$$

is obtained on inverting equation (2.9).
[F] is also tridiagonal and may be similarly partitioned to give

$$[F] = \begin{bmatrix} [F_e]_1 & & & & \\ & \ddots & & & \\ & & [F_e]_i & & \\ & & & \ddots & \\ & & & & [F_e]_{km} \end{bmatrix}. \tag{2.12}$$

Each element flexibility matrix may be found by writing the corresponding element stiffness matrix, i.e.

$$[F_e]_i = \begin{bmatrix} \frac{\ell_i}{A_i E_i} & 0 & 0 \\ 0 & \frac{\ell_i}{3E_i I_i} & -\frac{\ell_i}{6E_i I_i} \\ 0 & -\frac{\ell_i}{6E_i I_i} & \frac{\ell_i}{6E_i I_i} \end{bmatrix} = [S_e]_i^{-1}. \tag{2.13}$$

ℓ_i, E_i, A_i and I_i have been defined previously in equation (2.8).

From the geometry of the structure we may define the element deformation matrix $[B_{ab} \vdots B_{ba}]_i$ which relates the member deformations $\{\delta_e\}_i$ to the joint displacements, (see Figure 2.5), $\{u_a\}$ and $\{u_b\}$ where a and b are the terminal joints of member i. Thus

$$\{\delta_e\}_i = [B_{ab} \vdots B_{ba}] \begin{Bmatrix} u_a \\ \hline u_b \end{Bmatrix}, \tag{2.14}$$

where

$$[B_{ab} \vdots B_{ba}] = \left[\begin{array}{ccc|ccc} -\cos\gamma_i & -\sin\gamma_i & 0 & \cos\gamma_i & \sin\gamma_i & 0 \\ -\frac{\sin\gamma_i}{\ell_i} & \frac{\cos\gamma_i}{\ell_i} & 1 & \frac{\sin\gamma_i}{\ell_i} & -\frac{\cos\gamma_i}{\ell_i} & 0 \\ -\frac{\sin\gamma_i}{\ell_i} & \frac{\cos\gamma_i}{\ell_i} & 0 & \frac{\sin\gamma_i}{\ell_i} & -\frac{\cos\gamma_i}{\ell_i} & 1 \end{array}\right].$$

(2.15)

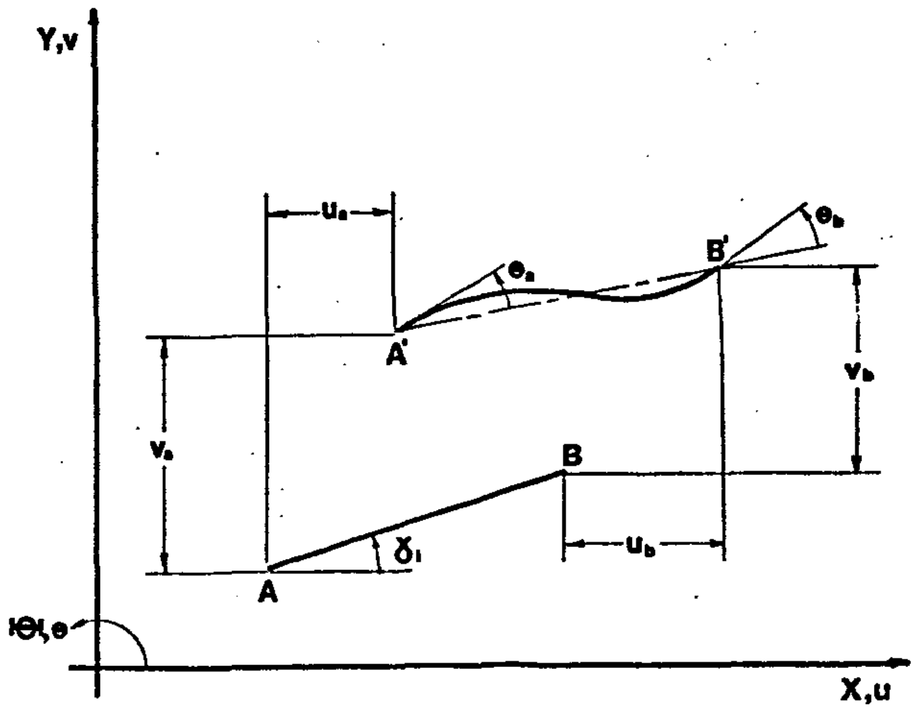

Figure 2.5 Joint Displacements and Member Deformation

The member deformations $\{\delta_e\}_i$ have been defined in equation (2.3), while the joint displacements $\{u_a\}$ & $\{u_b\}$ may be defined as

$$\{u_a\} = \{u_a, v_a, \theta_a\}^T. \qquad (2.16)$$

and

$$\{u_b\} = \{u_b, v_b, \theta_b\}^T. \qquad (2.17)$$

u_a, v_a, θ_a are the displacements at end A, while u_b, v_b, θ_b are the displacements at end B, both in directions X, Y, θ respectively. The orientation of member AB is defined by γ_i, the angle between the

X axis and $\overrightarrow{AB}$ measured from the X axis in direction θ, as shown in Figure 2.3.

We can now assemble the global deformation matrix [B], which relates all the member deformations $\{\delta\}$ to the unconstrained joint displacements $\{u\}$, in

$$\underset{m\times 1}{\{\delta\}} = \underset{m\times n}{[B]}\ \underset{n\times 1}{\{u\}} , \tag{2.18}$$

by expanding equation (2.14), for each element, to fit into equation (2.18). We can assemble [B] row by row, for each element, for example

$$\begin{Bmatrix} \delta_i \\ \phi_i^A \\ \phi_i^B \end{Bmatrix} = \begin{bmatrix} \\ \ldots 0 \ldots [B_{ab}] \ldots 0 \ldots [B_{ba}] \ldots 0 \ldots \\ \\ \end{bmatrix} \cdot \begin{Bmatrix} \vdots \\ 0 \\ \vdots \\ \{u_a\} \\ \vdots \\ 0 \\ \vdots \\ \{u_b\} \\ \vdots \\ 0 \\ \vdots \end{Bmatrix} . \tag{2.19}$$

The joint equilibrium equations may be formulated in terms of the member end forces $\{N\}$ acting at the joints and the loading $\{P\}$, at the same joints. These equations make up the coefficients of the statics matrix [A], hence

$$\underset{n\times m}{[A]}\ \underset{m\times 1}{\{N\}} = \underset{n\times 1}{\{P\}}.$$

It can be shown, using virtual work (see Appendix I), that

$$[A]^T = [B]. \qquad (2.21)$$

Necessary and sufficient conditions that the structure is statically determinate are that [A] is square and non-singular, i.e. $m = n$ and $|A| \neq 0$. If [A] is not square then a necessary and sufficient condition that the structure is stable is that Rank $[A] = n$. We limit our attention here to stable problems and since statically determinate problems can be solved using equations (2.20) alone, we concentrate now on how to deal with statically indeterminate problems. In statically indeterminate structures it is not possible to solve uniquely for the member forces {N} in terms of the loads {P}, i.e. equation (2.20) does not yield a unique solution for {N}. It is necessary to add to equation (2.20) (m - n) compatibility equations expressed in terms of the ordered unknown member forces {N}.

We can formulate these compatibility equations in terms of the (m - n) independent solutions to the homogeneous equations,

$$\underset{n \times m}{[A]}\ \underset{m \times 1}{\{N\}} = \underset{n \times 1}{\{0\}} . \qquad (2.22)$$

The general solution of equation (2.22) is

$$\{\bar{N}\} = \lambda_1\{r_1\} + \lambda_2\{r_2\} + \ldots + \lambda_{(m-n)}\{r_{(m-n)}\}, \qquad (2.23)$$

where λ_i, i = 1 to (m - n) are arbitrary constants and $\{r_i\}$, i = 1 to (m - n) are (m - n) linearly independent force systems.

These independent force systems may be recognised as a set of residual force systems, or self-stress systems, i.e. sets of member forces in equilibrium with zero loads. Considering the jth self-stress system, we see that on using the principle of virtual work and substituting from equations (2.18) and (2.21)

$$\{\delta\}^T\{r_j\} = \{u\}^T[B]^T\{r_j\} = \{u\}^T[A]\{r_j\} = 0. \qquad (2.24)$$

Transposing and using equation (2.11)

$$\{r_j\}^T\{\delta\} = \{r_j\}^T[F]\{N\} = \{\ell_j\}^T\{N\} = 0, \qquad (2.25)$$

where

$$\{\ell_j\}^T = [F]\{r_j\}^T. \qquad (2.26)$$

Let the matrix [L] be defined as

$$[L] = [\{\ell_1\}\{\ell_2\}\ldots\{\ell_{(m-n)}\}]. \qquad (2.27)$$

We can now assemble m equations (n equilibrium and (m - n) compatibility) for {N} using equations (2.20) and (2.25) and (2.27). Thus

$$\underset{m\times m}{\begin{bmatrix} [A] \\ \hline [L]^T \end{bmatrix}} \underset{m\times 1}{\{N\}} = \underset{m\times 1}{\begin{Bmatrix} P \\ \hline 0 \end{Bmatrix}} . \qquad (2.28)$$

Equations (2.28) admit a unique solution for {N}.

2.2 Computational Approach

The method used computationally to solve equation (2.27) is illustrated here by considering the simple example shown in Figure 2.6.

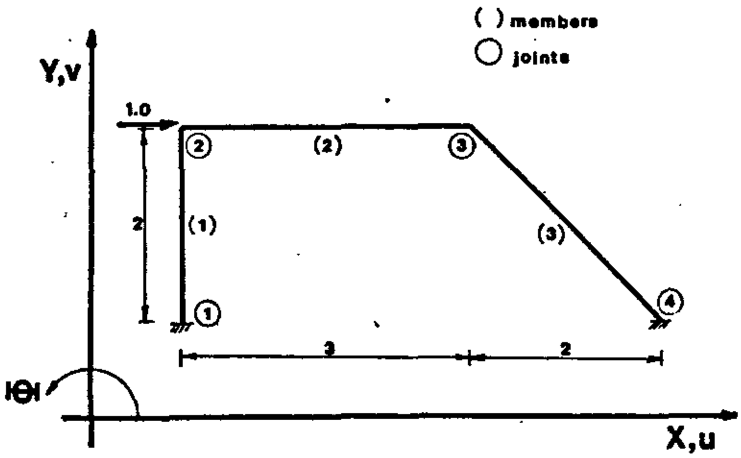

Figure 2.6 Illustrative Example

The unknown internal force vector {N} may be built up using equations (2.1) and (2.2) as follows:

$$\{N\} = \left\{\{N_e\}_1^T;\ \{N_e\}_2^T;\ \{N_e\}_3^T\right\} \tag{2.29}$$

$$= \{N_1;\ M_1^A;\ M_1^B;\ N_2;\ M_2^A;\ M_2^B;\ N_3;\ M_3^A;\ M_3^B\}^T. \tag{2.30}$$

The known external load vector {P} can be formulated using equation (2.6) to give:

$$\{P\} = \{P_2;\ Q_2;\ M_2;\ P_3;\ Q_3;\ M_3\}^T \tag{2.31}$$

$$= \{1,0;\ \ 0,0;\ \ 0,0;\ \ 0,0;\ \ 0,0;\ 0,0\}^T \tag{2.32}$$

From the geometry of the structure we may assemble the statics matrix [A] by transposing the deformation matrix [B] (equation (2.20)). The [B] matrix is made up by considering each member in turn, equation (2.14) being expanded to be consistent with equations (2.18). Equation (2.14) is given explicitly for member 3 as

$$[B_{34} \vdots B_{43}] = \left[\begin{array}{ccc|ccc} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{4} & \frac{1}{4} & 1 & -\frac{1}{4} & -\frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & -\frac{1}{4} & -\frac{1}{4} & 1 \end{array}\right] \tag{2.33}$$

Assembling the $[B_{ab} \vdots B_{ba}]$ matrices for each of the three members using equations (2.14), (2.15) and (2.18) and transposing (equation (2.21)) we arrive at equation (2.20). For this example we may write out equation (2.20) explicitly as:

$$\begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -\frac{1}{\sqrt{2}} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{\sqrt{2}} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} N_1 \\ M_1^A \\ M_1^B \\ N_2 \\ M_2^A \\ M_2^B \\ N_3 \\ M_3^A \\ M_3^B \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.34)$$

Notice that those equations relating the unknown member forces to the unknown reactions have been deleted. These deleted equations are complementary to those relating the member deformations to the constrained joint displacements.

By Gaussian row operations only we may arrive at the row-echelon form (see Noble [12]) of [A], $[\overline{A}]$. The row-echelon form is unique (ibid), and independent of the actual sequence of row operations used to produce this form. For the illustrative example being considered

$$[\bar{A}] = \left[\begin{array}{cccccccc} 1 & & & & & & ,71 & ,25 & ,25 \\ & 1 & & & & & 3,5 & -1,2 & -,25 \\ & & 1 & & & & ,21 & 1,7 & ,75 \\ & & & 1 & & & -,71 & ,25 & ,25 \\ & & & & 1 & & -,21 & -1,7 & -,75 \\ & & & & & 1 & 0 & 1,0 & 0 \end{array}\right].$$

(2.35)

We notice that the Rank of [A] = 6, which indicates that the structure is stable. Notice also that $[\bar{A}]$ may be written,

$$[\bar{A}] = [I \vdots A'] \tag{2.36}$$

where [I] is the (n × n) identity matrix. In most cases it is necessary to reorder the columns of $[\bar{A}]$ to arrive at equation (2.36). The internal force vector {N} will then have to be similarly reordered to give $\{\bar{N}\}$, where

$$\{\bar{N}\} = \{N_I \vdots N'\}^T. \tag{2.37}$$

$\{N_I\}$ are those internal forces associated with the columns in $[\bar{A}]$ which form the identity matrix, {N'} are those forces associated with the columns which are linearly dependent on the columns of the identity matrix.

Equation (2.20) may thus, on reducing [A] to row echelon form, and on reordering $[\bar{A}]$ (equation (2.36)), be written

$$[I\,\vdots\,A'] \begin{Bmatrix} N_I \\ \overline{N'} \end{Bmatrix} = \{\bar{P}\}. \qquad (2.38)$$

$\{\bar{P}\}$ is the result of all the row operations necessary to get from $[A]$ to $[\bar{A}]$ applied to $[P]$.

We are now in a position to extract the (m - n) self-stress systems from the homogeneous equations (2.22). Equations (2.22) may be rewritten, after reducing [A] to row-echelon form and after using equations (2.36), (2.37) and (2.38) as

$$\underset{(n\times n)}{[I]}\ \underset{(n\times 1)}{\{N_I\}} = -\underset{(n\times(m-n))}{[A']}\ \underset{((m-n)\times 1)}{\{N'\}}\ . \qquad (2.39)$$

The self-stress systemsare obtained by successively putting one element of $\{N'\}$ equal to unity and the remainder equal to zero, solving for $\{N_I\}$ and then reconstituting $\{N\}$ from equation (2.37).

Computationally it is more efficient to obtain the self-stress systems directly from $[\bar{A}]$. This may be achieved by identifying those columns not associated with the identity matrix and expanding these columns with (m - n) additional elements in the row position corresponding to the column position of the linearly dependent columns. The additional element is zero except where row and column position coincide, where it is - 1.

Adopting this method a consistent set of self-stress systems $\{r_j\}$ may be identified. In the illustrative example being considered there are three independent force systems in equilibrium with zero load, viz:

$$\{r_1\} = \{,71 \quad 3,5 \quad ,21 \quad -,71 \quad -,21 \quad 0 \quad -1 \quad 0 \quad 0\}^T, \qquad (2.40a)$$

$$\{r_2\} = \{,25 \quad -1,2 \quad 1,7 \quad ,25 \quad -1,7 \quad 1,0 \quad 0 \quad -1 \quad 0\}^T, \qquad (2.40b)$$

$$\{r_3\} = \{,25 \quad -,25 \quad ,75 \quad ,25 \quad -,75 \quad 0 \quad 0 \quad 0 \quad -1\}^T. \qquad (2.40c)$$

We are now able to formulate the compatibility equations, using equations (2.13) and (2.26). In the illustrative example of Figure 2.6, Youngs modulus, the member cross-sectional areas and the moment of inertia are the same for each member. The axial stiffness AE = 1 and flexural ridgidity El = 2,5 for each member. The element flexibility matrix for member 3 may be written explicitly as an example,

$$[F_e]_3 = \begin{bmatrix} \sqrt{2} & 0 & 0 \\ 0 & \frac{\sqrt{2}}{7,5} & -\frac{\sqrt{2}}{15} \\ 0 & -\frac{\sqrt{2}}{15} & \frac{\sqrt{2}}{7,5} \end{bmatrix} \qquad (2.41)$$

Using equation (2.26) we may formulate the (m - n) compatibility conditions. These conditions, for our example, are

$$\{\ell_1\} = \{,25 \quad -,22 \quad ,18 \quad -,37 \quad -,15 \quad ,08 \quad -,5 \quad 0 \quad 0\}^T, \qquad (2.42a)$$

$$\{\ell_2\} = \{,28 \quad -,31 \quad -,31 \quad ,42 \quad -,5 \quad ,42 \quad 0 \quad -,21 \quad ,10\}^T, \qquad (2.42b)$$

$$\{\ell_3\} = \{,33 \quad -,11 \quad -,11 \quad ,5 \quad -,2 \quad ,10 \quad 0 \quad ,13 \quad ,25\}^T. \qquad (2.42c)$$

Notice that all the compatibility equations have been multiplied by a positive constant to improve the conditioning of the final equations.

$$\{r_1\} = \{,71 \quad 3,5 \quad ,21 \quad -,71 \quad -,21 \quad 0 \quad -1 \quad 0 \quad 0\}^T, \qquad (2.40a)$$

$$\{r_2\} = \{,25 \quad -1,2 \quad 1,7 \quad ,25 \quad -1,7 \quad 1,0 \quad 0 \quad -1 \quad 0\}^T, \qquad (2.40b)$$

$$\{r_3\} = \{,25 \quad -,25 \quad ,75 \quad ,25 \quad -,75 \quad 0 \quad 0 \quad 0 \quad -1\}^T. \qquad (2.40c)$$

We are now able to formulate the compatibility equations, using equations (2.13) and (2.26). In the illustrative example of Figure 2.6, Youngs modulus, the member cross-sectional areas and the moment of inertia are the same for each member. The axial stiffness AE = 1 and flexural ridgidity El = 2,5 for each member. The element flexibility matrix for member 3 may be written explicitly as an example,

$$[F_e]_3 = \begin{bmatrix} \sqrt{2} & 0 & 0 \\ 0 & \frac{\sqrt{2}}{7,5} & -\frac{\sqrt{2}}{15} \\ 0 & -\frac{\sqrt{2}}{15} & \frac{\sqrt{2}}{7,5} \end{bmatrix} \qquad (2.41)$$

Using equation (2.26) we may formulate the (m - n) compatibility conditions. These conditions, for our example, are

$$\{\ell_1\} = \{,25 \quad -,22 \quad ,18 \quad -,37 \quad -,15 \quad ,08 \quad -,5 \quad 0 \quad 0\}^T, \qquad (2.42a)$$

$$\{\ell_2\} = \{,28 \quad -,31 \quad -,31 \quad ,42 \quad -,5 \quad ,42 \quad 0 \quad -,21 \quad ,10\}^T, \qquad (2.42b)$$

$$\{\ell_3\} = \{,33 \quad -,11 \quad -,11 \quad ,5 \quad -,2 \quad ,10 \quad 0 \quad ,13 \quad ,25\}^T. \qquad (2.42c)$$

Notice that all the compatibility equations have been multiplied by a positive constant to improve the conditioning of the final equations.

These compatibility equations are added to equation (2.38), which, if the columns of $[\bar{A}]$ and elements of $\{N\}$ remain in their original order, may be written

$$[\bar{A}]\{N\} = \{\bar{P}\}, \tag{2.43}$$

to give, by equations (2.25) and (2.27),

$$\begin{bmatrix} \bar{A} \\ \cdots \\ L^T \end{bmatrix} \{N\} = \begin{Bmatrix} \bar{P} \\ \cdots \\ 0 \end{Bmatrix} \tag{2.44}$$

Equation (2.44) admits a unique solution for $\{N\}$, where

$$\{N\} = \{0{,}1242;\ 1{,}3692;\ 0{,}4508;\ -0{,}0900;\ -0{,}4508;\ 0{,}0783;\ -0{,}1515;\ -0{,}0783;\ 0{,}0100\}^T, \tag{2.45}$$

for the illustrative example.

CHAPTER 3

SHAKEDOWN IN PLANE FRAMES

3.1 Introduction

In this Chapter we examine the response of elastic-perfectly plastic frames to cyclic loads. As in the frames considered in Chapter 2, we assume here that all members are straight, prismatic and unloaded. The loads act only at the nodes and all dynamic effects are ignored.

Since the constituent material is assumed to be elastic-perfectly plastic, the moment curvature relationship may be idealised as shown in Figure 3.1. Plastic deformation in the form of hinges will occur wherever the bending moment diagram equals the plastic moment M_o. The axial load in any member is assumed not to affect the plastic moment M_o.

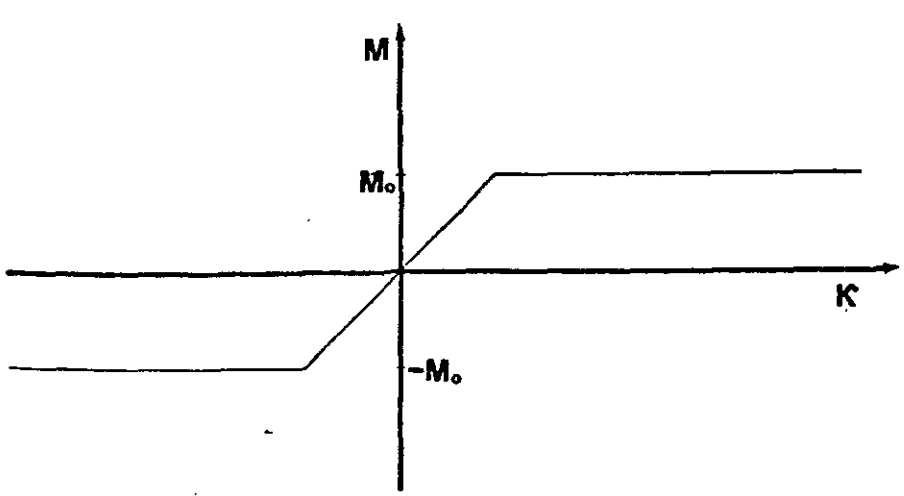

Figure 3.1 Moment Curvature Relationship

The loading applied to many types of structure will vary considerably with time. Most civil engineering structures are subjected to working loads which will operate independently of each other between certain predetermined limits. For example, consider a structure subjected to two discrete load types, or load parameters, wind load and imposed load. Both loads are functions of time and may be represented by $\Gamma_1(t)$ and $\Gamma_2(t)$ respectively. Suppose that we know only that

$$\Gamma_1^- \leq \Gamma_1(t) \leq \Gamma_1^+ , \qquad (3.1a)$$

$$\Gamma_2^- \leq \Gamma_2(t) \leq \Gamma_2^+ . \qquad (3.1b)$$

At any instant the actual loads on the structure can be represented by a point in load space. This point is confined by inequalities (3.2) to lie within the domain shown in Figure 3.2.

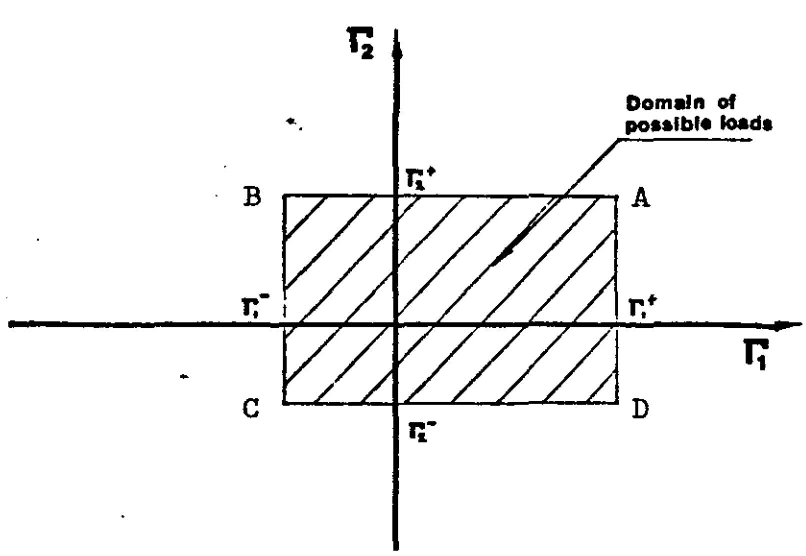

Figure 3.2 Domain of Possible Loads in Load Space

We examine now the behaviour of the structure if the loads repeatedly follow the cycle which forms the boundary of the domain of possible loads (ABCDA in Figure 3.2). If, for the first cycle, the response of the structure is entirely elastic, then it will remain elastic for all future load cycles. If, however, some plastic deformation occurs, then one of two possibilities exists. Either plastic deformation continues to take place or, after a certain number of transient cycles, a purely elastic steady state response is set up. In the latter case the structure is said to have shaken down.

It is convenient to subdivide further the former case where continued plastic deformation takes place. Firstly the plastic deformation may increase monotonically at hinges at different points in the structure, i.e. any plastic hinge rotation takes place in the same sense. Unbounded deformation may thus occur and the structure is said to have failed by incremental collapse. Secondly, the plastic deformation may be non-monotonic; this usually occurs at a single hinge point and the hinge rotations will take place in opposite senses during a cycle. These reversals of stress will cause low cycle fatigue and the structure is said to have failed owing to alternating plasticity.

It can be shown (for example see Martin [9]) that if the structure will shake down for the load cycle which circumscribes the domain of possible loads, then it will shake down for all possible load cycles within that domain. Thus it is sufficient to consider only the circumscribing cycles.

We introduce here the load factor Γ, which is a scalar multiplier for all loads in the load space. By increasing Γ we increase the domain of all possible loads (see Figure 3.3). The shakedown load factor Γ^s is equal to the maximum value of Γ for which the structure will shake down.

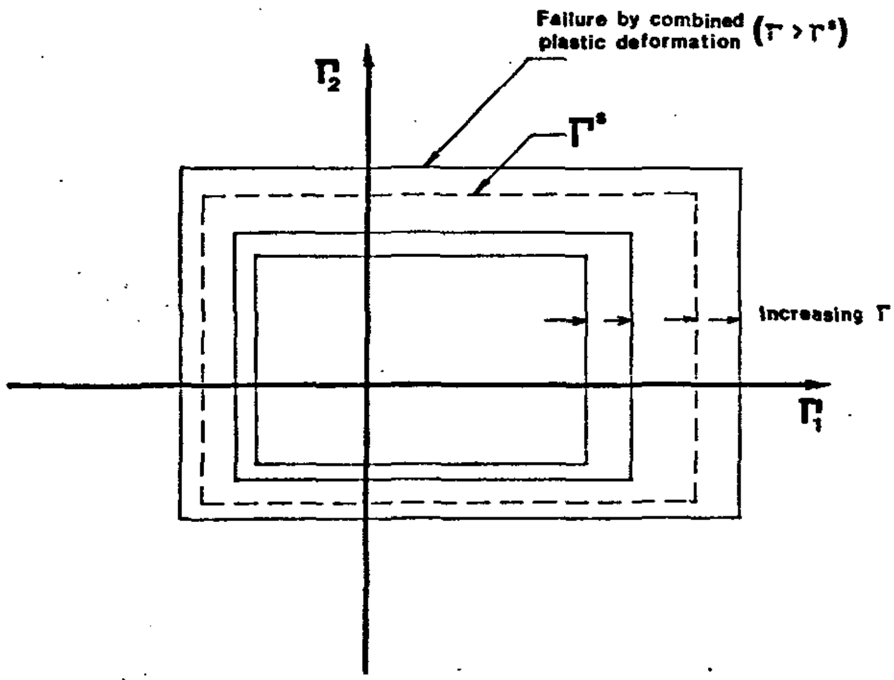

Figure 3.3 Increasing Γ in Load Space

3.2 Formulation

It can be shown (for example see Martin [9]) that the shakedown theorems may be written in general programming form. The kinematic theorem states that the shakedown load factor Γ^s is given by the least value of

$$\Gamma = \frac{W_p^c}{W_{ext}^c} \tag{3.2}$$

during a kinematically admissible deformation cycle. W_p^c is the plastic work done over the cycle and W_{ext}^c is the work done by the external loads over the cycle.

Without loss of generality, the kinematic shakedown theorem may be restated as follows; the shakedown load factor Γ^s is given by the least value of

$$\Gamma = W_p^c \tag{3.3a}$$

subject to the condition that

$$W_{ext}^c = 1 \tag{3.3b}$$

and that the deformation cycle is kinematically admissible.

Let us consider first the plastic work done over a cycle, W_p^c. In the plane frame structures being dealt with, all plastic work is in the form of plastic hinge rotations, which take place at k labelled hinge positions. The plastic hinge rotation over a cycle at position j, θ_j, can be defined by

$$\theta_j = \int_c \dot{\theta}_j(t)dt \tag{3.4a}$$

where $\theta_j(t)$ is the rotation rate at time t and c indicates that the integral is over one cycle. The work done by the hinges over one load cycle can therefore be given by the work equation

$$W_p^c = \sum_{j=1}^{k} \int_c M_{oj}|\dot{\theta}_j(t)|dt. \tag{3.4b}$$

For incremental collapse, all hinge rotations are monotonic and $\dot{\theta}_j(t)$ therefore does not change sign. In this case

$$\int_c M_{oj}|\dot{\theta}_j(t)|dt = M_{oj}|\theta_j| \qquad (3.5)$$

for each plastic hinge. However for alternating plasticity θ_j is not monotonic and equation (3.5) is not valid. We therefore introduce two variables θ_j^+ and θ_j^-, such that

$$\theta_j = \theta_j^+ - \theta_j^-, \quad \theta_j^+ \geq 0, \quad \theta_j^- \geq 0. \qquad (3.6)$$

Whenever any hinge rotation takes place at joint j through the cycle, either θ_j^+ increases or θ_j^- increases. At no stage during the cycle can θ_j^+ or θ_j^- decrease. Therefore

$$\dot{\theta}_j^+(t) \geq 0 \quad \text{and} \quad \dot{\theta}_j^-(t) \geq 0$$

throughout the cycle. Since we measure θ_j^+ and θ_j^- only at the end of any cycle we can write only that

$$\int_c M_{oj}|\dot{\theta}_j(t)|dt \leq M_{oj}\theta_j^+ + M_{oj}\theta_j^- , \qquad (3.7)$$

since

$$|\theta_j| \leq \theta_j^+ + \theta_j^-.$$

However, we note that, we are minimising equation (3.4) and hence equation (3.7) will be an equality. We can therefore write;

$$W_p^c = \sum_{j=1}^{k} M_{oj}\theta_j^+ + \sum_{j=1}^{k} M_{oj}\theta_j^- . \qquad (3.8)$$

Secondly, let us evaluate the work done by the external loads over a cycle. This cannot be achieved readily for most classes of structures but in the case of frames a few observations lead us to an evaluable expression for the external work. We may use the principle of virtual work (for example see Martin [9]) to show that the external work done over a cycle is equal to the integral, over the cycle, of the elastic moments under the loads for that part of the cycle, multiplied by the sum of the elastic deformation rates associated with the residual stress field and the plastic deformation rates, $\dot{\theta}_j$. Since the elastic deformation rates are induced by the residual moments, which are in equilibrium with zero load, we can show, again by using the principle of virtual work, that the work done on these elastic deformations by the elastic moments is zero. Because the hinge rotations take place only at k specific joints and are zero elsewhere, the external work may now be written as

$$W_{ext}^c = \sum_{j=1}^{k} \int_c M_j^E(t)\dot{\theta}_j dt, \qquad (3.9)$$

where $M_j^E(t)$ is the elastic response to the load cycle.

The evaluation of equation (3.9) remains a complex problem. However, let us again introduce the variables θ_j^+ and θ_j^- subject to conditions (3.6). Let M_{maxj}^E and M_{minj}^E be such that

$$M^E_{maxj} = \text{maximum} \quad \left(M^E_j(t)\right), \tag{3.10a}$$

$$M^E_{minj} = \text{minimum} \quad \left(M^E_j(t)\right), \tag{3.10b}$$

over a cycle.

Thus

$$\sum_{j=1}^{k} \int_c M^E_j(t)\dot{\theta}_j dt \leq \sum_{j=1}^{k} [M^E_{maxj}\theta^+_j - M^E_{minj}\theta^-_j] \tag{3.11}$$

because of inequalities (3.9).

Suppose we now define a new programming problem, that of finding the least value of

$$\Gamma = W^c_p \tag{3.12a}$$

subject to $$\sum_{j=1}^{k} [M^E_{maxj}\theta^+_j - M^E_{minj}\theta^-_j] = 1 \tag{3.12b}$$

and $(\theta^+_j - \theta^-_j)$ kinematically admissible. For this new problem, the least value of $\Gamma \leq \Gamma^s$ because of inequality (3.11) and condition (3.12b).

However, it can be shown (for example see Martin [9]), that, if the limiting deformation cycle is correct, then inequality (3.11) is in fact an equality. Thus the least value of $\Gamma = \Gamma^s$ for the correct collapse mode, but may be conservative for incorrect modes.

Finally we need to ensure that the deformation cycle is kinematically admissible. If the frame is statically indeterminate to degree (m - n), as in Chapter 2, we may find (m - n) linearly independent self equilibrating moment systems $\{s_i\}$, where

$$\{s_i\} = \{s_{i1},\ldots,s_{ij},\ldots,s_{ik}\}^T. \tag{3.13}$$

Necessary and sufficient conditions that the deformation cycle is kinematically admissible are that the rotations, θ_j, obey the virtual work relations

$$\sum_{j=1}^{k} s_{ij}\theta_j = \sum_{j=1}^{k} s_{ij}(\theta_j^+ - \theta_j^-) = 0 \tag{3.14}$$

for i = 1 to (m - n).

We may now state the kinematic theorem, by equations (3.3), (3.4), (3.5), (3.8), (3.11) and (3.14) as follows; Γ^s is equal to the least value of

$$\Gamma = \sum_{j=1}^{k} M_{oj}\theta_j^+ + \sum_{j=1}^{k} M_{oj}\theta_j^- \tag{3.15a}$$

subject to

$$\sum_{j=1}^{k} M^E_{maxj}\theta_j^+ - \sum_{j=1}^{k} M^E_{minj}\theta_j^- = 1 \tag{3.15b}$$

and

$$\sum_{j=1}^{k} s_{ij}\theta_j^+ - \sum_{j=1}^{k} s_{ij}\theta_j^- = 0 \tag{3.15c}$$

for i = 1 to (m - n).

3.3 Computational Approach

Since all members are assumed to be straight, prismatic and unloaded between nodes, the bending moment diagram is linear, with maxima and minima occurring at the nodes. The nature of the bending moment diagram and the moment curvature relationship ensures that plastic hinges are confined to sections adjacent to the nodes. Thus each member will have two possible hinge positions, one at each end.

Firstly, we need the coefficients, M_{oj}, of the objective function (equation (3.15a)). These must be input as data with the other section properties necessary for the elastic solution. Secondly, the coefficients of the external work constraint (equation (3.15b)) must be evaluated. We require the extreme values of the moment range for the load cycle. This may be achieved by linearising the load cycle and considering the apices of the linearised load cycle in turn. It is therefore necessary to have the elastic solution for each of the load conditions $\{P_j\}$ represented by the $k\ell$ apices of the load cycle. By expanding equation (2.28) to accommodate the $k\ell$ load vectors $\{P_j\}$ and their corresponding solutions $\{N_j\}$ we have

$$\begin{bmatrix} A \\ \cdots \\ L^T \end{bmatrix} [\{N_1\},\ldots,\{N_{k\ell}\}] = \left[\begin{Bmatrix} P_1 \\ \cdots \\ 0 \end{Bmatrix}, \ldots, \begin{Bmatrix} P_{k\ell} \\ \cdots \\ 0 \end{Bmatrix} \right], \qquad (3.16)$$

which enables us to solve uniquely for $\{N_j\}$, $j = 1$ to $k\ell$. The extreme moment values at each point, M^E_{maxj} and M^E_{minj}, may thus be found simply by scanning the solution $\{N_j\}$ for the maximum and minimum moments at each point. Note that $\{N_j\}$ contain the member end moments conjugate to the plastic hinge rotation at each end, and the member axial force, which is ignored in this part of the analysis.

3.3 Computational Approach

Since all members are assumed to be straight, prismatic and unloaded between nodes, the bending moment diagram is linear, with maxima and minima occurring at the nodes. The nature of the bending moment diagram and the moment curvature relationship ensures that plastic hinges are confined to sections adjacent to the nodes. Thus each member will have two possible hinge positions, one at each end.

Firstly, we need the coefficients, M_{oj}, of the objective function (equation (3.15a)). These must be input as data with the other section properties necessary for the elastic solution. Secondly, the coefficients of the external work constraint (equation (3.15b)) must be evaluated. We require the extreme values of the moment range for the load cycle. This may be achieved by linearising the load cycle and considering the apices of the linearised load cycle in turn. It is therefore necessary to have the elastic solution for each of the load conditions $\{P_j\}$ represented by the $k\ell$ apices of the load cycle. By expanding equation (2.28) to accommodate the $k\ell$ load vectors $\{P_j\}$ and their corresponding solutions $\{N_j\}$ we have

$$\begin{bmatrix} A \\ \cdots \\ L^T \end{bmatrix} [\{N_1\},\ldots,\{N_{k\ell}\}] = \left[\begin{Bmatrix} P_1 \\ \cdots \\ 0 \end{Bmatrix}, \ldots, \begin{Bmatrix} P_{k\ell} \\ \cdots \\ 0 \end{Bmatrix} \right], \qquad (3.16)$$

which enables us to solve uniquely for $\{N_j\}$, $j = 1$ to $k\ell$. The extreme moment values at each point, M^E_{maxj} and M^E_{minj}, may thus be found simply by scanning the solution $\{N_j\}$ for the maximum and minimum moments at each point. Note that $\{N_j\}$ contain the member end moments conjugate to the plastic hinge rotation at each end, and the member axial force, which is ignored in this part of the analysis.

Lastly, the compatibility conditions $\{s_j\}$ are needed. These were generated by the particular method of elastic solution used in Chapter 2. It is for this reason alone that this elastic method is used. Equation (2.23) yields a set of (m - n) linearly independent selfstress systems $\{r_j\}$. This set, however, includes axial forces, which we may ignore and form (m - n) linearly independent self equilibrating moment systems, $\{s_j\}$, by ignoring the axial force terms of $\{r_j\}$.

Equations (3.15) may thus be written in vector form as follows; Γ^s is equal to the least value of

$$\Gamma = \{M_o\}^T\{\theta^+\} + \{M_o\}^T\{\theta^-\} \qquad (3.17a)$$

subject to

$$\{M^E_{max}\}\{\theta^+\} - \{M^E_{min}\}^T\{\theta^-\} = 1 \qquad (3.17b)$$

and $$\{s_j\}\{\theta^+\} - \{s_j\}\{\theta^-\} = 0 \qquad (3.17c)$$

for i = 1 to (m - n).

Notice that both the objective function and the constraints are linear and that the variables are all non-negative. We may thus solve the linear programming problem (equations (3.17)) by the simplex algorithm.

For this purpose the subroutine SIMPLX, written by the University of Wisconsin Computer Centre, is used. Slight modifications were introduced to reduce the amount of core storage needed.

After the self-stress system $\{r_j\}$ and the elastic solutions $\{N_j\}$ have been computed, little extra time is needed for the more complex shakedown solution. For the 22 member problem tested, the elastic solution used 26,9 seconds of CPU (Central Processor Unit) time, while the full shakedown solution required 30,2 seconds CPU time, an increase of approximately 12%.

Some previous researchers (for example, Cohn, Ghosh and Parimi [4]) have formulated the compatibility requirements in terms of the independent incremental collapse mechanisms, which automatically excluded the possibility of identifying alternating plasticity. However, it is of importance to note here that the solution to the linear programming problem (3.17) yields the correct shakedown load factor Γ^s, owing to either collapse mode. This is because the variables θ_j^+ and θ_j^- allow non-monotonic rotations and because the compatibility requirements are formulated generally, and not in terms of incremental collapse mechanisms.

A program was written in FORTRAN IV for the Univac 1106 at the University of Cape Town Computer Centre embodying the approach defined above. The internal logic of the program will not be described, but a listing of the program and examples of data input and computer output are given in Appendix III. Results of several illustrative problems are given in the following Chapter.

CHAPTER 4

PLANE FRAME NUMERICAL EXAMPLES

Example 4.1

The single storey single bay frame shown in Figure 4.1 was analysed by Martin [9] using a single load state (point A in Figure 4.2) and two piecewise linear load cycles (ABOEA and ACDEA in Figure 4.2).

The results of the program written using the presented method of analysis are compared with Martin's results (with $\ell = M_o = 1$) in Table 4.1. In all cases, the correct failure mechanism and collapse load factors were obtained.

TABLE 4.1

Load State	Martin	Present Analysis
$H = 1$ $V = 1$ (point A)	$\Gamma^* = 3,000$ Mechanism as in Figure 4.3	$\Gamma^* = 3,000$ Mechanism as in Figure 4.3
$0 \leq H \leq 1$ $0 \leq V \leq 1$ (ABOEA)	$\Gamma^s \doteq 2,857$ Mechanims as in Figure 4.3	$\Gamma^s = 2,857$ Mechanism as in Figure 4.3
$-1 \leq H \leq 1$ $0 \leq V \leq 1$ (ACDEA)	$\Gamma^s = 2,759$ Alternating plasticity at node 1	$\Gamma^s = 2,759$ Alternating plasticity at node 1

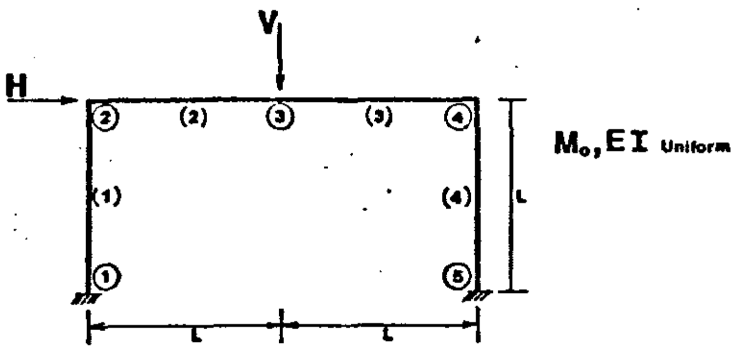

Figure 4.1 Single Storey Single Bay Frame

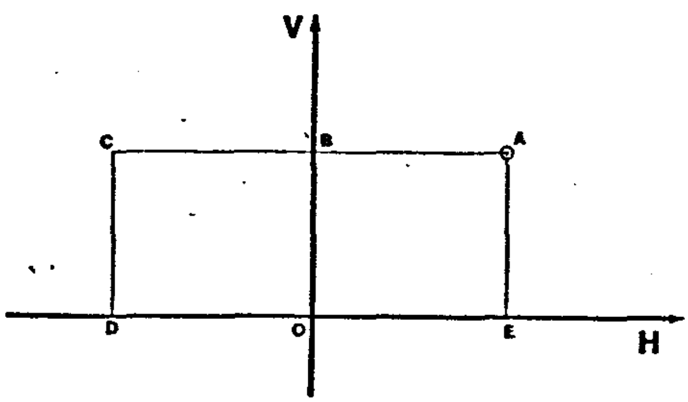

Figure 4.2 Load Cycles for Example 4.1

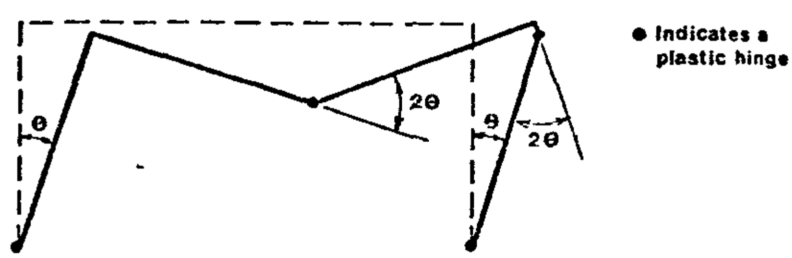

Figure 4.3 Failure Mechanism for Example 4.1

Example 4.2

The double storey single bay frame shown in Figure 4.4 was analysed by Cohn, Ghosh and Parimi [4] using a single load state (point A in Figure 4.5) and a piecewise linear load cycle (ABOCA in Figure 4.5).

The results of the program written are compared with those of Cohn, Ghosh and Parimi (using $M_o = I_1 = \ell = 1$) in Table 4.2. In both cases the correct failure mechanisms were obtained. The 2 % difference in the shakedown load factor can be attributed to the fact that Cohn et al do not take axial stiffness into account for their elastic analysis while the presented analysis does.

TABLE 4.2

Load State	Cohn, Ghosh & Parimi	Present Analysis
$H = 1$ $V = 3$ (point A)	$\Gamma^* = 0{,}533$ Mechanism as in Figure 4.6	$\Gamma^* = 0{,}533$ Mechanism as in Figure 4.6
$0 \leq H \leq 1$ $0 \leq V \leq 3$ (ABOCA)	$\Gamma^s = 0{,}509$ Mechanism as in Figure 4.6	$\Gamma^s = 0{,}520$ Mechanism as in Figure 4.6

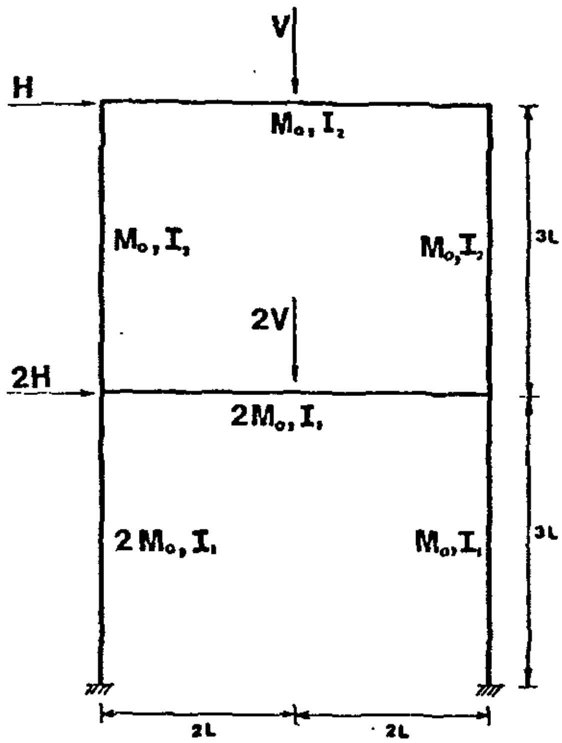

Figure 4.4 Single Bay Double Storey Frame

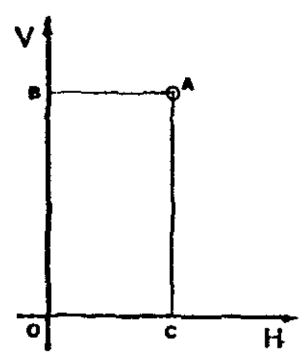

Figure 4.5 Load Cycle for Example 4.2

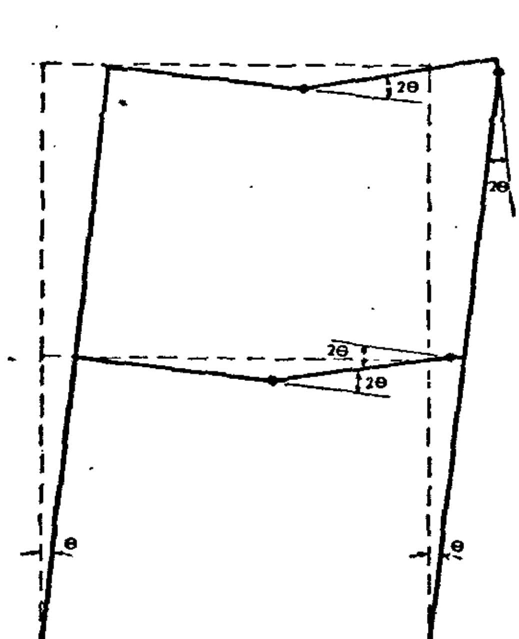

Figure 4.6 Failure Mechanism for Example 4.2

Example 4.3

The single storey double bay frame shown in Figure 4.7(a) was analysed by Baker, Horne and Heyman [2] using the piecewise linear load cycle shown in Figure 4.8. Notice that the second bay is subject to a uniform distributed load. On analysing this frame by the present method it was necessary to represent the distributed load as a series of point loads, as shown in Figure 4.7(b).

The results of the program written are compared with those of Baker et al (with $I_1 = \ell = M_o = 1$) in Table 4.3. The correct collapse mechanism is obtained, and the 5,5% difference in the shakedown load factor can be attributed to both the influence of axial stiffness and the effect of the approximated loading on the elastic bending moment diagram.

TABLE 4.3

Load State	Baker Horne & Heyman	Present Analysis
$0 \leq H \leq 1$	$\Gamma^s = 2,30$	$\Gamma^s = 2,173$
$0 \leq P \leq 2$	Mechanism as in	Mechanism as in
$0 \leq Q \leq 4$	Figure 4.9	Figure 4.9

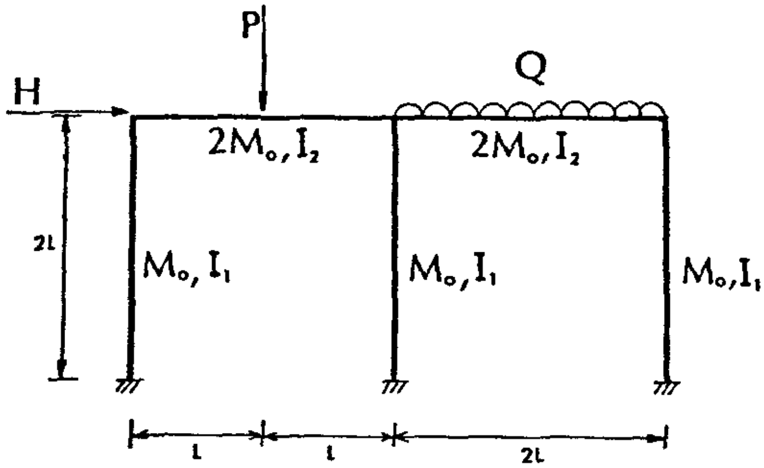

Figure 4.7(a) Single Storey Double Bay Frame (Baker et al)

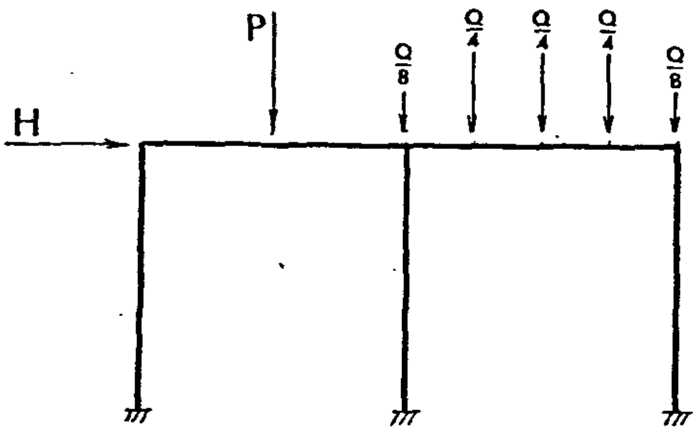

Figure 4.7(b) Single Storey Double Bay Frame (Present Analysis)

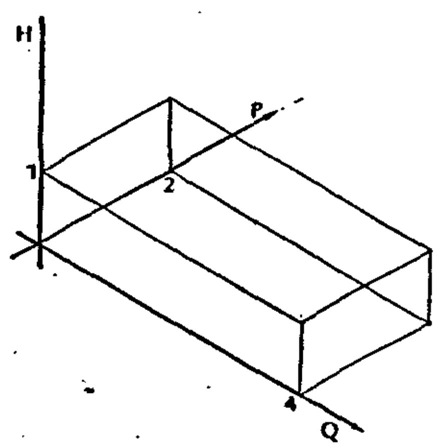

Figure 4.8 Circumscribing Load Space

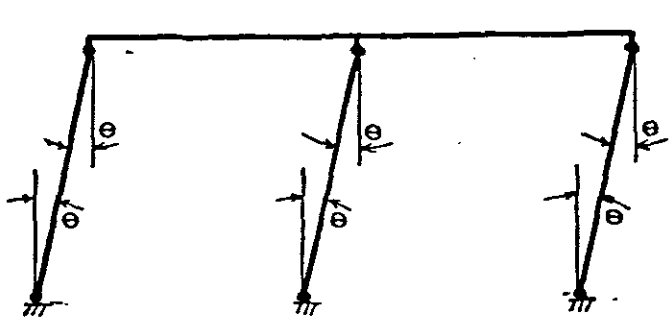

Figure 4.9 Failure Mechanism for Example 4.3

Example 4.4

The three-bay pitched portal frame, typical of factory buildings, shown in Figure 4.10 was analysed using the program written to indicate the scope of the program.

The collapse mechanism for the load space shown in Figure 4.11 is shown in Figure 4.12. The elastic solution used 23,6 seconds of CPU time while the full shakedown solution used 25,5 seconds, an increase of 8%.

The shakedown load factor $\Gamma^S = 2,19$

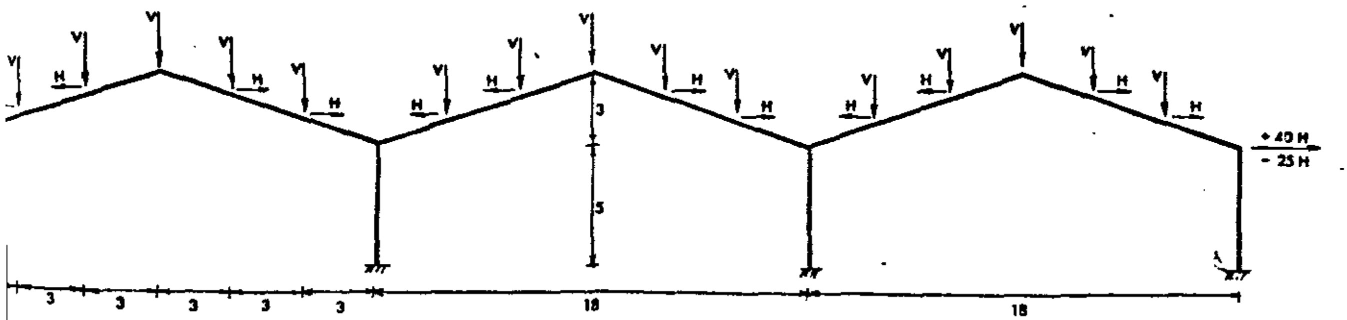

FIGURE 4.10 Three-bay Frame Example 4.4

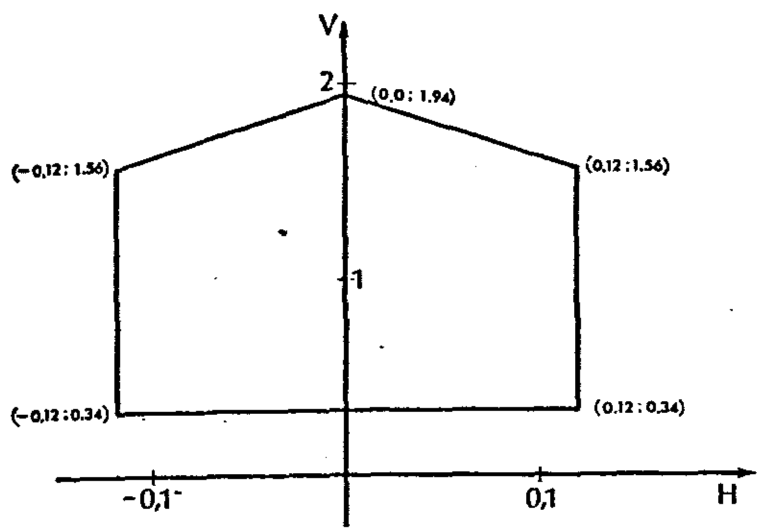

FIGURE 4.11 Loadspace for Example 4.4

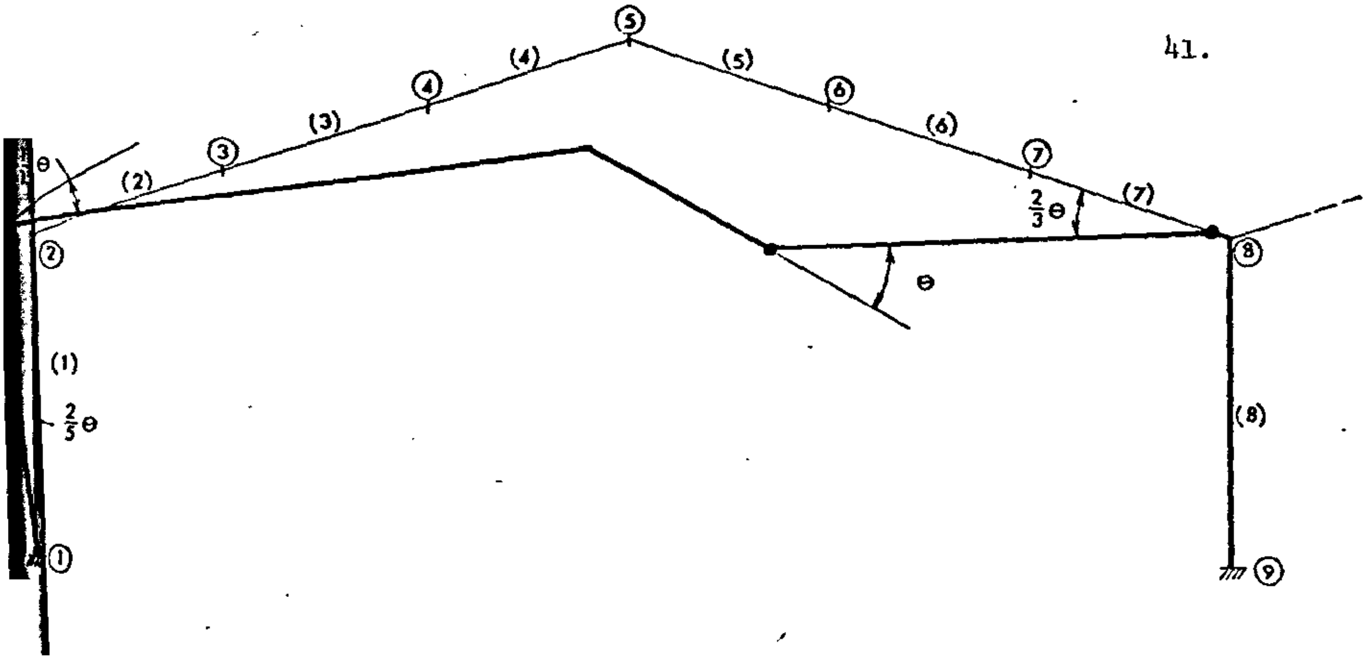

Figure 4.12 End Bay Collapse Mechanism for Example 4.4

Conclusions

The examples given in this Chapter indicate the potential of the method described in Chapters 2 and 3 in analysing frame structures under variable repeated loading. It is evident that the program written will be able to solve the shakedown (or limit analysis) problem in plane frames for most practical situations even if the structure is more complex than Example 4.4. However, it should be noted that in multi-storey frame structures stability is usually the governing factor in determining the load carrying capacity of the structure. The program written does not take stability into account and is therefore applicable to single or double storey structures of any number of bays.

The program employs certain storage and time saving techniques. However, it is felt that better efficiency could be obtained by storing the statics matrix [A] in banded form. The linearly dependent columns identified during the row-echelonisation process may be larger than the bandwidth and provision therefore has to be made for these columns. Also, the compatibility requirements [L] will not be within the bandwidth and this fact will have to be taken into account.

5.2 Formulation

Consider a sheet of unit thickness for which the plane stress assumptions are valid. The plate is discretised into km triangles, the kj apices of which are called nodes. The loading acts only in the global XY plane of the plate. No singularities in loading are permitted but continuous loads must be approximated by loading which acts only at the nodes.

We assume continuity of displacement across element boundaries and linear variation of displacement within each triangular element. The displacement components of the three apices are therefore sufficient to describe the displacement at any point within the triangle. We can therefore write the general displacements within any triangle as

$$u = \alpha_1 + \alpha_2 X + \alpha_3 Y, \qquad (5.1a)$$

$$v = \alpha_4 + \alpha_5 X + \alpha_6 Y, \qquad (5.1b)$$

where α are constant coefficients for each triangle. The displacements at the apices a, b and c of the general acute angled triangle may be related to the coefficients by substituting the coordinates of the apices into equations (5.1), (see Figure 5.1). Hence

$$\begin{bmatrix} u_a \\ u_b \\ u_c \\ v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} 1 & X_a & Y_a & 0 & 0 & 0 \\ 1 & X_b & Y_b & 0 & 0 & 0 \\ 1 & X_c & Y_c & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & X_a & Y_a \\ 0 & 0 & 0 & 1 & X_b & Y_b \\ 0 & 0 & 0 & 1 & X_c & Y_c \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix}, \qquad (5.2)$$

or

$$\{u_e\} = [C]\{\alpha\}. \tag{5.3}$$

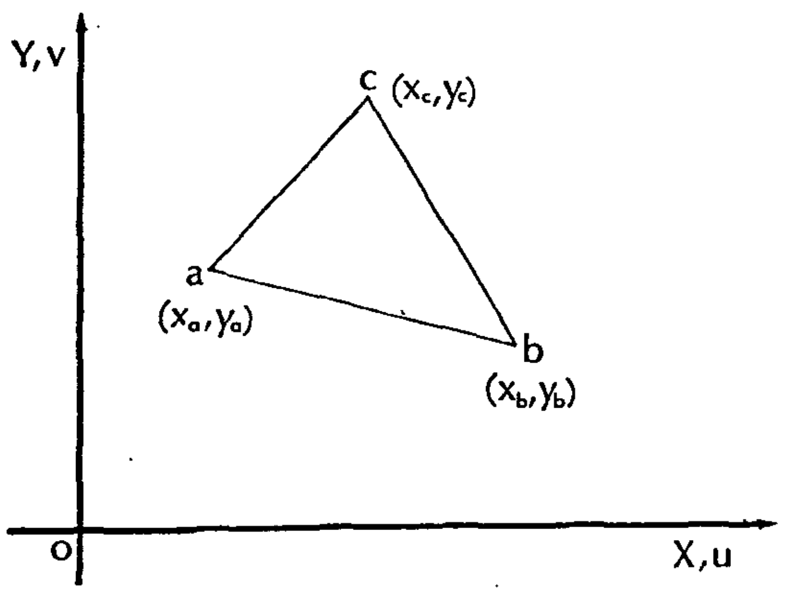

Figure 5.1 General Acute Angled Triangle

The strain displacement relations

$$\varepsilon_{xx} = \frac{\partial u}{\partial X} = \alpha_2 , \tag{5.4a}$$

$$\varepsilon_{yy} = \frac{\partial v}{\partial Y} = \alpha_6 , \tag{5.4b}$$

and

$$\varepsilon_{xy} = \frac{\partial u}{\partial Y} + \frac{\partial v}{\partial X} = \alpha_3 + \alpha_5, \tag{5.4c}$$

may be written in matrix form as

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \{\alpha\}, \qquad (5.5)$$

or

$$\{\varepsilon_e\} = [D]\{\alpha\} . \qquad (5.6)$$

By inverting equation (5.3) and using equation (5.6)

$$\{\varepsilon_e\} = [D]\{\alpha\} = [D][C]^{-1}\{u_e\} = [B_e]\{u_e\}. \qquad (5.7)$$

Where

$$[B_e] = [D][C]^{-1} \qquad (5.8)$$

$$= \frac{1}{\mu} \begin{bmatrix} Y_b-Y_c & Y_c-Y_a & Y_a-Y_b & 0 & 0 & 0 \\ 0 & 0 & 0 & X_c-X_b & X_a-X_c & X_b-X_a \\ X_c-X_b & X_a-X_c & X_b-X_a & Y_b-Y_c & Y_c-Y_a & Y_a-Y_b \end{bmatrix}, \qquad (5.9)$$

and

$$\mu = |C| = X_a(Y_b-Y_c) + X_b(Y_c-Y_a) + X_c(Y_a-Y_b). \qquad (5.10)$$

Let the (n × 1) column vector {u} represent the ordered unconstrained displacement components of the kn unconstrained nodes, i.e.

$$\{u\} = \{u_1, v_1, \ldots, u_j, v_j, \ldots, u_{kn}, v_{kn}\}. \tag{5.11}$$

Let the (m × 1) column vector {ε} represent the ordered strain component of each element, i.e.

$$\{\varepsilon\} = \left\{\{\varepsilon_e\}_1^T, \ldots, \{\varepsilon_e\}_j^T, \ldots, \{\varepsilon_e\}_{kn}^T\right\}^T. \tag{5.12}$$

We can now, by expanding and reordering equation (5.7) set up the general strain displacement relationships

$$\underset{(m\times 1)}{\{\varepsilon\}} = \underset{(m\times n)}{[B]}\;\underset{(n\times 1)}{\{u\}}, \tag{5.13}$$

by considering each element in turn. For example, if element j has apices a, b and c then

$$\{\varepsilon_{ej}\} = \frac{1}{\mu}\begin{bmatrix} & Y_b - Y_c & 0 & & Y_c - Y_a & 0 & & Y_a - Y_b & 0 \\ \cdots & 0 & X_c - X_b & \cdots & 0 & X_a - X_c & \cdots & 0 & X_b - X_a \\ & X_c - X_b & Y_b - Y_c & & X_a - X_c & Y_c - Y_a & & X_b - X_a & Y_a - Y_b \end{bmatrix} \begin{bmatrix} \vdots \\ u_a \\ v_a \\ \vdots \\ u_b \\ v_b \\ \vdots \\ u_c \\ v_c \\ \vdots \end{bmatrix} \tag{5.14}$$

where μ has been defined previously in equation (5.10).

From the stress strain relationships in each element,

$$\varepsilon_{xx} = \sigma_{xx}/E - \nu\sigma_{yy}/E , \qquad (5.15a)$$

$$\varepsilon_{yy} = \sigma_{yy}/E - \times\sigma_{xx}/E , \qquad (5.15b)$$

$$\varepsilon_{xy} = 2(1 + \nu)\sigma_{xy}/E , \qquad (5.15c)$$

we can formulate the element flexibility matrix $[F_e]$ as

$$[F_e] = \begin{bmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 2+2\nu \end{bmatrix} \qquad (5.16)$$

We can therefore write

$$\{\varepsilon_e\} = [F_e]\{\sigma_e\}, \qquad (5.17)$$

where

$$\{\sigma_e\} = \{\sigma_{xx}, \sigma_{yy}, \sigma_{xy}\}^T. \qquad (5.18)$$

Formulation of the flexibility matrix, which relates the strain components $\{\varepsilon\}$ to the stress components $\{\sigma\}$, for the whole sheet is now possible, i.e.

$$\underset{(m\times 1)}{\{\varepsilon\}} = \underset{(m\times m)}{[F]} \underset{(m\times 1)}{\{\sigma\}} , \qquad (5.19)$$

where

$$[F] = \begin{bmatrix} [F_e]_1 & & \\ & \ddots & \\ & & [F_e]_2 \\ & & & \ddots \\ & & & & [F_e]_3 \end{bmatrix} , \qquad (5.20)$$

and

$$\{\sigma\} = \left\{\{\sigma_e\}_1^T, \ldots, \{\sigma_e\}_j^T, \ldots, \{\sigma_e\}_{km}^T\right\}^T . \qquad (5.21)$$

Transposing equation (5.19) we have

$$\{\sigma\} = [F]^{-1}\{\varepsilon\} = [S]\{\varepsilon\}. \qquad (5.22)$$

From equations (5.16) and (5.20) [F] is symmetric and therefore [S] will also be symmetric.

Conjugate to the nodal displacements {u} we may formulate the (n × 1) column vector {P} of the loads which, although acting only at the nodes, approximate the actual boundary tractions and body forces. We may define {P} by

$$\{P\} = \{P_1, Q_1, \ldots, P_j, Q_j, \ldots, P_{kj}, Q_{kj}\}^T , \qquad (5.23)$$

where P_j and Q_j are the forces acting on node j in the global X and Y directions respectively.

It is usual, at this stage, to formulate the displacement method of solution. This approach is shown here briefly. By equating the internal work done by the sheet to the external work done by the loads we may state that the integral over the sheet of the generalised stress field multiplied by the generalised strain field must equal the external loads multiplied by their corresponding displacements. Since we are dealing with the plane stress case for which both stress and strain are constant within each element and since we have approximated the external loads by forces acting on the nodes, we can write

$$\sum_{j=1}^{km} \{\sigma_e\}^T\{\varepsilon_e\} \int_e dv = \{u\}^T\{P\}. \tag{5.24}$$

Since each element is of unit thickness, the integral in equation (5.24) is merely the area of each element. We can formulate the $(m \times m)$ diagonal matrix $[\Delta]$ of element areas, so that $[\Delta]$ is compatible with both the vectors $\{\sigma\}$ and $\{\varepsilon\}$ by placing the area of each element on the diagonal consecutively three times in the correct order. We may thus write equation (5.24) in matrix form as,

$$\{\sigma\}^T[\Delta]\{\varepsilon\} = \{u\}^T\{P\}. \tag{5.25}$$

From equations (5.13) and (5.22) we may write

$$\{\varepsilon\}^T[S]^T[\Delta][B]\{u\} = \{u\}^T\{P\}. \tag{5.26}$$

Since [S] is symmetric it is unaltered by transposition.

Transposing equation (5.13) and substituting in equation (5.26) we have

$$\{u\}^T[B]^T[S][\Lambda][B]\{u\} = \{u\}^T\{P\} \tag{5.27}$$

and hence

$$[B]^T[S][\Lambda][B]\{u\} = \{P\} \tag{5.28}$$

Equation (5.28) allows a unique solution for $\{u\}$. Using equation (5.13) we may obtain the strains $\{\varepsilon\}$. The relation between the strains $\{\varepsilon\}$ and the stresses $\{\sigma\}$, equation (5.22), allows a unique solution for the stresses $\{\sigma\}$.

Although this method of solution is usually more efficient in the numerical solution, we will continue here with the method employed in Chapter 2. The reason for this is that we intend to use (see Chapter 6) the self-stress systems generated for the limit analysis problem.

We therefore set up the equilibrium equations which relate the element stresses $\{\sigma\}$ to the nodal loads $\{P\}$,

$$\underset{n\times m}{[A]}\ \underset{m\times 1}{\{\sigma\}} = \underset{n\times 1}{\{P\}}. \tag{5.29}$$

These equations are not set up using equilibrium principles, which would ensure continuity of force across element boundaries, but they are set up using the relationship (see Appendix II) between the statics matrix [A] and the deformation matrix [B] for plane stress, constant strain problems,

$$\underset{(n\times m)}{[B]^T} \; \underset{(m\times m)}{[\Delta]} = \underset{(n\times m)}{[A]}. \qquad (5.30)$$

The stresses used in equation (5.29) are therefore mathematically defined by the strain-displacement relationships, rather than physically defined. This distinction did not occur in the case of plane frame problems where the exact elastic solution may be obtained by either the force or displacement method of approach.

As in Chapter 2 equation (5.29) is not sufficient to solve uniquely for $\{\sigma\}$. Again the general solution to the homogeneous equations,

$$[A]\{\sigma\} = \{0\}, \qquad (5.31)$$

lead us to the compatibility requirements. Since the general solution of equation (5.31) may be written

$$\{\bar{\sigma}\} = \lambda_1\{r_1\} +\ldots+ \lambda_j\{r_j\} +\ldots+ \lambda_{(m-n)}\{r_{(m-n)}\}, \qquad (5.32)$$

where λ_j are arbitrary constants, the (m-n) vectors $\{r_j\}$ may be recognised as self-stress systems. Considering the jth self-stress system, we see that on using the principle of virtual work and substituting from equations (5.13) and (5.30)

$$\{\varepsilon\}^T[\Delta]\{r_j\} = \{u\}^T[B]^T[\Delta]\{r_j\} = \{u\}^T[A]\{r_j\} = 0 \qquad (5.33)$$

Transposing (note that since $[\Delta]$ is diagonal it is not affected) and using equation (5.19)

$$\{r_j\}^T[\Delta]\{\varepsilon\} = \{r_j\}^T[\Delta][F]\{\sigma\} = \{\ell_j\}^T\{\sigma\} = 0, \quad (5.34)$$

where

$$\{\ell_j\}^T = \{r_j\}^T[\Delta][F]. \quad (5.35)$$

Let the [L] matrix be defined as

$$[L] = [\{\ell_1\}\{\ell_2\}\ldots\{\ell_{(m-n)}\}]. \quad (5.36)$$

We can now assemble m equations (n equilibrium and (m-n) compatibility) for {N} using equations (5.29) and (5.34). Thus

$$\begin{bmatrix} [A] \\ \cdots \\ [L]^T \end{bmatrix} \{\sigma\} = \begin{Bmatrix} \{P\} \\ \cdots \\ \{0\} \end{Bmatrix} . \quad (5.37)$$

Equations (5.37) admit a unique solution for $\{\sigma\}$.

5.3 Computational Approach

The method used computationally to solve equations (5.37) is similar to that used in Chapter 2 to solve for equation (2.27). We will therefore illustrate only briefly the formulation of the problem.

Consider the deep beam shown in Figure 5.2, which has been discretised into 5 elements, the apices of which define 7 nodes. The loading shown is that used in the computations and is therefore an approximation to the real continuous loading.

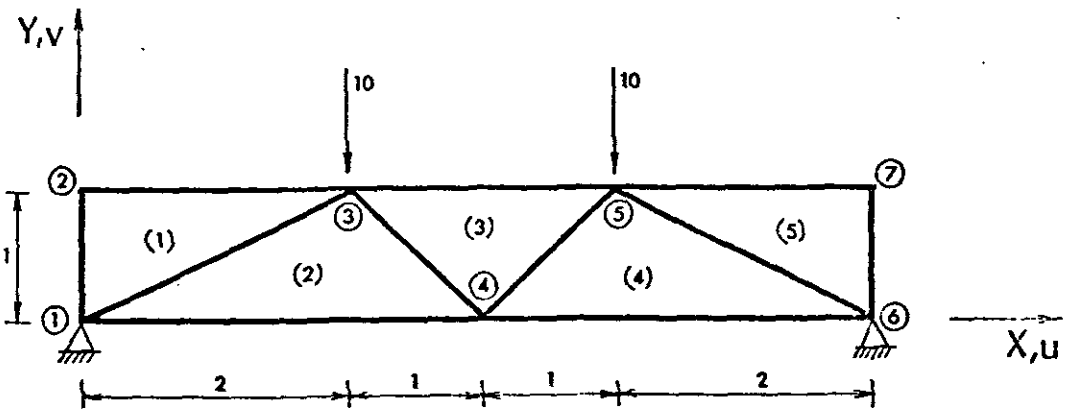

Figure 5.2 Simple Deep Beam

From the geometry of the structure we may, by using equations (5.14) and (5.30), assemble the statics matrix [A]. Setting up the load vector we will have

$$P = \{0;\ 0;\ 0;\ -10;\ 0;\ 0;\ 0;\ -10;\ 0;\ 0\}^T. \quad (5.38)$$

Equations (5.29) can now be formulated. As in Chapter 2 we now generate the self equilibrating stress systems which will enable us to arrive at the compatibility requirements.

By simple Gaussian row operations we may arrive at the row-echelon form of [A], $[\bar{A}]$. Performing identical operations on {P}, the equilibrium equations

$$[\bar{A}]\{\sigma\} = \{\bar{P}\}, \tag{5.39}$$

still hold. Again $[\bar{A}]$ may be written in the form

$$[\bar{A}] = [I \vdots A'] \tag{5.40}$$

by reordering the columns of $[\bar{A}]$. Similar reordering of the stresses $\{\sigma\}$ allows us (as in equation (2.39)) to write

$$[I]\{\sigma_I\} = -[A']\{\sigma'\}. \tag{5.41}$$

By the process outlined in Chapter 2 we may obtain the self-stress systems $\{r_j\}$. Using equations (5.19) and (5.35) we may formulate the compatibility requirements in terms of the stresses. Returning the columns of $[\bar{A}]$ and the stress vector $\{\sigma\}$ to their original order we can, using equations (5.34) and (5.36) arrive at

$$\begin{bmatrix} [\bar{A}] \\ \cdots \\ [L]^T \end{bmatrix} \{\sigma\} = \begin{Bmatrix} \{\bar{P}\} \\ \cdots \\ \{0\} \end{Bmatrix} \tag{5.42}$$

Equations (5.42) admit the unique displacement method solution for $\{\sigma\}$. For the example being considered

$$\{\sigma\} = \{-5{,}7\ ;\ -1{,}4\ ;\ -2{,}8\ ;\ -3{,}8\ ;\ -12{,}5\ ;\ -4{,}6\ ;\ -19{,}6\ ;\ 20{,}4\ ;\ 0{,}0\ ;\ -3{,}8\ ;\ -12{,}5\ ;\ 4{,}6\ ;\ -5{,}7\ ;\ -1{,}4\ ;\ 2{,}8\}^T \tag{5.43}$$

CHAPTER 6

LIMIT ANALYSIS IN PLANE STRESS PROBLEMS

6.1 Introduction

In this Chapter we explore the possibility of using the extra information given in the previous section to calculate the limit or collapse load in continuum structures. As a preliminary study we will consider plane stress problems of the type described earlier, i.e. a sheet of unit thickness which has been discretised into km triangular element, the apices of which formkj nodes.

The constituent material is assumed to be elastic-perfectly plastic and homogeneous throughout. We will use a simple, but typical, piecewise linear yield condition, i.e;

$$-\sigma_o \leq \sigma_{xxj} \leq \sigma_o, \qquad (6.1a)$$

$$-\sigma_o \leq \sigma_{yyj} \leq \sigma_o, \qquad (6.1b)$$

$$-\sigma_o \leq \sigma_{xyj} \leq \sigma_o, \qquad (6.1c)$$

where σ_o is the yield stress and σ_{xxj} and σ_{yyj} are the direct stresses and σ_{xyj} is the shear stress at any point j respectively.
This yield condition is only a rough approximation of the true yield condition but it may, however, be refined by the addition of more linear constraints. For convenience we introduce the (m × 1) column vector

$$\{\sigma_o\} = \{\sigma_o;\ldots;\sigma_o;\ldots;\sigma_o\}^T. \tag{6.2}$$

The inequalities (6.1) representing the yield surface are shown graphically in Figure 6.1.

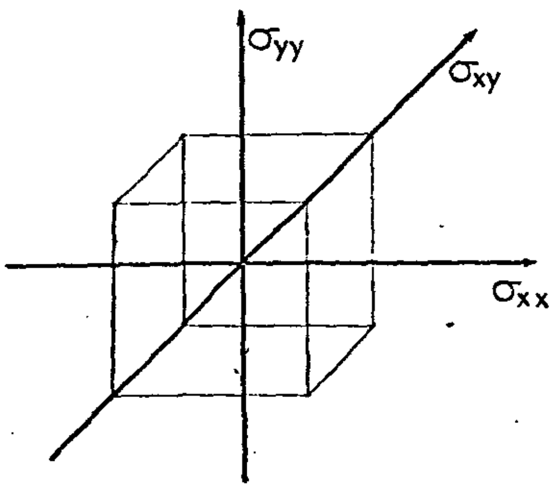

Figure 6.1 Yield Surface

The limit analysis problem is formulated in terms of the plastic strain rates at collapse. Note that in Chapter 3, for the shakedown problem in frames, we considered a finite interval of time, i.e. the interval for one load cycle, and therefore formulated the problem in terms of the plastic strains over one cycle. At collapse the body will remain rigid everywhere except in those regions where flow is taking place.

Conjugate to the limit stresses, $\{\sigma_o\}$, we can formulate the (m × 1) column vector of plastic strain rates as

$$\{\dot{\varepsilon}^p\} = \{\{\dot{\varepsilon}^p_e\}^T_1 \ldots \{\dot{\varepsilon}^p_e\}^T_j \ldots \{\dot{\varepsilon}^p_e\}^T_{km}\}^T, \tag{6.3}$$

where

$$\{\dot{\varepsilon}_e^p\}_j = \{\dot{\varepsilon}_{xx}^p, \ \dot{\varepsilon}_{yy}^p, \ \dot{\varepsilon}_{xy}^p\}_j^T . \tag{6.4}$$

6.2 Formulation

It can be shown (for example see Martin [9]) that the limit analysis theorems may be written in general programming form. The kinematic theorem states that the collapse load factor Γ^* is the least value of

$$\Gamma = \dot{W}_{int} / \dot{W}_{ext} , \tag{6.5}$$

Subject to the strain rate field being kinematically admissible $\dot{W}_{int}$ and $\dot{W}_{ext}$ are the internal energy dissipation rate and the external work rate respectively, under the set of loads being considered.

The internal energy dissipation rate, at collapse, is equal to the integral over the sheet of the plastic strain rates multiplied by their corresponding yield stresses. Since we have a simplified yield condition with constant strain rates in each element we can write that

$$\dot{W}_{int} = \sum_{j=1}^{km} \{\sigma_o\}_j^T \{|\dot{\varepsilon}_j^p|\}_j \int_{e_j} dv , \tag{6.6}$$

where $\{|\dot{\varepsilon}_e^p|\}_j$ are the moduli of the actual strain rates $\{\dot{\varepsilon}_e^p\}$.
If the sheet is of unit thickness we may write equation (6.6) in matrix form as

$$\dot{W}_{int} = \{\sigma_o\}^T[\Delta]\{|\dot{\varepsilon}^p|\}, \tag{6.7}$$

where $[\Delta]$ has been defined for equation (5.25).

The external work rate is equal to the generalised external loads multiplied by the generalised displacement rates. Using the principle of virtual velocities we can equate the external work rate with the integral over the structure of the elastic stresses due to the loads, multiplied by the strain rates compatible with the displacement rates. Since the elastic strain rates vanish at collapse only the plastic strain rates need be taken into account. Both the strain rates and their corresponding stresses will be constant in each element. We can now write that

$$\dot{W}_{ext} = \sum_{j=1}^{km} \{\sigma_e^E\}_j^T\{\dot{\varepsilon}_e^p\} \int_{e_j} dv , \tag{6.8}$$

where

$$\{\sigma_e^E\}_j = \{\sigma_{xx}^E, \ \sigma_{yy}^E, \ \sigma_{xy}^E\}_j^T , \tag{6.9}$$

and σ_{xx}^E, σ_{yy}^E and σ_{xy}^E are the solutions to equation (5.37) for element j under the loading considered. Equation (6.8) may also be written in matrix form

$$\dot{W}_{ext} = \{\sigma^E\}[\Delta]\{\dot{\varepsilon}^p\} , \tag{6.10}$$

where

$$\{\sigma^E\} = \{\{\sigma_e^E\}_1^T, \ldots, \{\sigma_e^E\}_j^T, \ldots, \{\sigma_e^E\}_{km}^T\}^T . \tag{6.11}$$

Using equations (6.7) and (6.10) we may, without loss of generality write the kinematic theorem as: Γ^* is the least value of

$$\Gamma = \{\sigma_o\}[\Delta]\{|\dot{\varepsilon}^p|\} \qquad (6.12a)$$

subject to $\{\sigma^E\}[\Delta]\{\dot{\varepsilon}^p\} = 1$ (6.12b)

and $\{\dot{\varepsilon}^p\}$ are kinematically admissible.

Necessary and sufficient conditions that $\{\dot{\varepsilon}^p\}$ are kinematically admissible are that they obey the virtual velocity relationships

$$\{r_j\}^T[\Delta]\{\dot{\varepsilon}^p\} = 0, \quad j = 1 \text{ to } m - n \qquad (6.13)$$

where $\{r_j\}$ $j = 1$ to $m - n$ are a set of linearly independent self-stress sytems, defined in equation (5.32). It is in order to obtain these conditions that the elastic analysis of Chapter 5 was introduced.

6.3 Computational Approach

As in Chapter 3 we will show how the programming problem (equations (6.12)) can be made suitable for solution by the simplex algorithm. We introduce two new variables ε_j^+ and ε_j^- such that

$$\dot{\varepsilon}_j^p = \dot{\varepsilon}_j^{p+} - \dot{\varepsilon}_j^{p-} \qquad (6.14)$$

and

Using equations (6.7) and (6.10) we may, without loss of generality write the kinematic theorem as: Γ^* is the least value of

$$\Gamma = \{\sigma_o\}[\Delta]\{|\dot{\varepsilon}^p|\} \qquad (6.12a)$$

subject to $\{\sigma^E\}[\Delta]\{\dot{\varepsilon}^p\} = 1$ (6.12b)

and $\{\dot{\varepsilon}^p\}$ are kinematically admissible.

Necessary and sufficient conditions that $\{\dot{\varepsilon}^p\}$ are kinematically admissible are that they obey the virtual velocity relationships

$$\{r_j\}^T[\Delta]\{\dot{\varepsilon}^p\} = 0, \quad j = 1 \text{ to } m - n \qquad (6.13)$$

where $\{r_j\}$ $j = 1$ to $m - n$ are a set of linearly independent self-stress sytems, defined in equation (5.32). It is in order to obtain these conditions that the elastic analysis of Chapter 5 was introduced.

6.3 Computational Approach

As in Chapter 3 we will show how the programming problem (equations (6.12)) can be made suitable for solution by the simplex algorithm. We introduce two new variables ε_j^+ and ε_j^- such that

$$\dot{\varepsilon}_j^p = \dot{\varepsilon}_j^{p+} - \dot{\varepsilon}_j^{p-} \qquad (6.14)$$

and

Using equations (6.7) and (6.10) we may, without loss of generality write the kinematic theorem as: Γ^* is the least value of

$$\Gamma = \{\sigma_o\}[\Delta]\{|\dot{\varepsilon}^p|\} \qquad (6.12a)$$

subject to $\{\sigma^E\}[\Delta]\{\dot{\varepsilon}^p\} = 1 \qquad (6.12b)$

and $\{\dot{\varepsilon}^p\}$ are kinematically admissible.

Necessary and sufficient conditions that $\{\dot{\varepsilon}^p\}$ are kinematically admissible are that they obey the virtual velocity relationships

$$\{r_j\}^T[\Delta]\{\dot{\varepsilon}^p\} = 0, \quad j = 1 \text{ to } m - n \qquad (6.13)$$

where $\{r_j\}$ $j = 1$ to $m - n$ are a set of linearly independent self-stress sytems, defined in equation (5.32). It is in order to obtain these conditions that the elastic analysis of Chapter 5 was introduced.

6.3 Computational Approach

As in Chapter 3 we will show how the programming problem (equations (6.12)) can be made suitable for solution by the simplex algorithm. We introduce two new variables ε_j^+ and ε_j^- such that

$$\dot{\varepsilon}_j^p = \dot{\varepsilon}_j^{p+} - \dot{\varepsilon}_j^{p-} \qquad (6.14)$$

and

$$\dot{\varepsilon}_j^{p+} \geq 0 , \quad \dot{\varepsilon}_j^{p-} \geq 0 , \qquad j = 1 \text{ to } m. \tag{6.15}$$

Therefore

$$|\dot{\varepsilon}_j^{p}| \leq \dot{\varepsilon}_j^{p+} + \dot{\varepsilon}_j^{p-} , \qquad j = 1 \text{ to } m. \tag{6.16}$$

In equation (6.16) the equality will hold only when one or both variables are zero. If both variables are non-negative then the inequality holds.

Equation (6.12a) may therefore be written

$$\Gamma \leq \{\sigma_o\}[\Delta]\{\dot{\varepsilon}^{p+}\} + \{\sigma_o\}[\Delta]\{\dot{\varepsilon}^{p-}\}, \tag{6.17}$$

however, since we will be minimising the right hand side of equation (6.17) the equality will hold.

The programming problem is now suitable for solution by the simplex algorithm. The objective function is linear, as are all the constraints and the variables are all non-negative. The kinematic theorem may therefore be stated as: the collapse load factor Γ^* is the least value of

$$\Gamma = \{\sigma_o\}^T[\Delta]\{\dot{\varepsilon}^{p+}\} + \{\sigma_o\}^T[\Delta]\{\dot{\varepsilon}^{p-}\} \tag{6.18a}$$

subject to $\{\sigma^E\}^T[\Delta]\{\dot{\varepsilon}^{p+}\} - \{\sigma^E\}^T[\Delta]\{\dot{\varepsilon}^{p-}\} = 1$ (6.18b)

and $\{r_j\}^T[\Delta]\{\dot{\varepsilon}^{p+}\} - \{r_j\}^T[\Delta]\{\dot{\varepsilon}^{p-}\} = 0 \quad j = 1 \text{ to } (m - n)$ (6.18c)

A FORTRAN program was written to illustrate the application of the method described above, and executed on the UNIVAC 1106 at the University of Cape Town Computer Centre. The final linear programming problem was solved by means of the SIMPLX package written by the University of Wisconsin Computer Centre. The internal logic of the program will not be described, but a listing of the program, together with examples of data input and computer output, is given in Appendix IV. Results for three specific problems solved by this program are discussed in the following Chapter.

CHAPTER 7

PLANE STRESS NUMERICAL EXAMPLES

Example 7.1

A deep beam was analysed using the three different triangular meshes shown in Figures 7.1(a), (b) and (c). Note that each successive mesh is a refinement of the previous mesh. Since we are using the kinematic method the limit load factor is an upper bound and should therefore decrease with each refining step. The computed limit load factors $\Gamma^*(a) = 0{,}5$; $\Gamma^*(b) = 0{,}375$; $\Gamma^*(c) = 0{,}375$ behave in this manner.

The times used for each analysis are given in Table 7.1

TABLE 7.1

Execution Time in Seconds

Example	Elastic Solution	Limit Analysis
7.1(a)	0,794	0,992
7.1(b)	4,113	14,34
7.1(c)	24,749	49,06

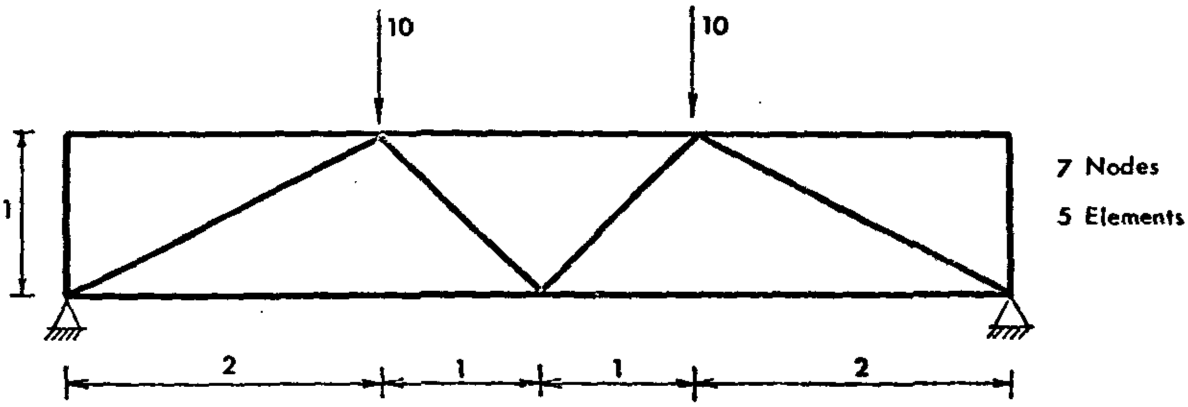

Figure 7.1(a) First Mesh

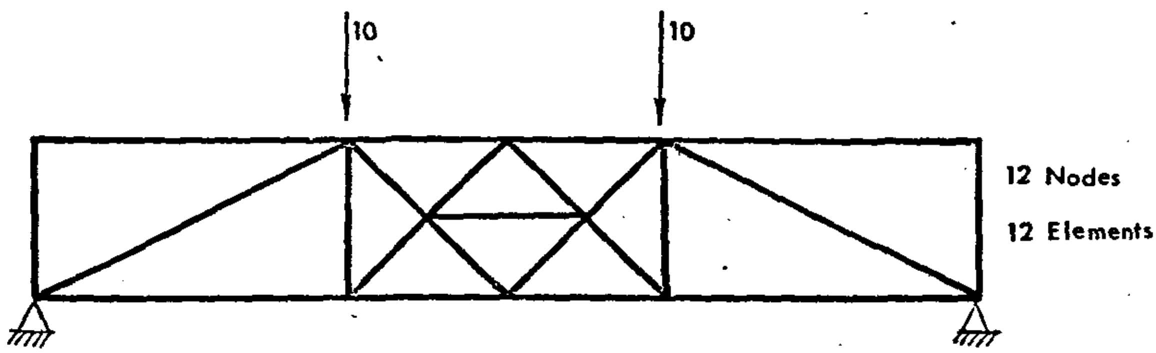

Figure 7.1(b) Second Mesh

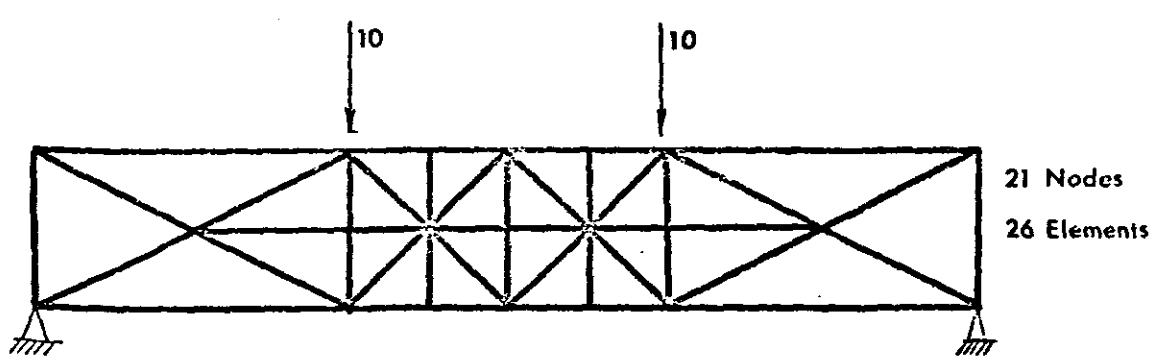

Figure 7.1(c) Third Mesh

Conclusions

The approach to limit analysis for plane stress outlined in Chapters 5 and 6 and illustrated above must be understood to be simply a preliminary study. Considerably more effort should be put into the extension of the method to more realistic yield conditions, and into the most efficient use of computing effort, before it can properly be compared with previous work.

A study of the literature (see, for example, Belytschko and Hodge [3], Koopman and Lance, [7], Anderheggen [1] and Hutula, [6]) shows, however, that numerical limit analysis of continuum problems is still in a process of development. While the problem will clearly emerge as a linear or non-linear programming problem, it is evident that formulations are required which are efficient in règard to data input, and which reduce the number of variables in the programming problem as much as possible. In this respect it is considered that the method of formulation suggested in this thesis is promising, and that it merits further development.

REFERENCES

1. Anderheggen, E. 'Finite Element Analysis Assuming Rigid-Ideal-Plastic Material Behaviour'. Water Annual Meeting, ASME, N.Y. December 1976 "Limit Analysis Using Finite Elements".

2. Baker, J.F., Horne, M.R. and Heyman, J. 'The Steel Skeleton' Vol II, 1956, Cambridge University Press.

3. Belytschko, T and Hodge, P.G. Jr. 'Plane Stress Limit Analysis by Finite Elements'. Journal of the Engineering Mechanics Division, ASCE, Vol 96, No EM 6, December 1970. pp 931 - 944.

4. Cohn, M.Z., Ghosh, S.K. and Parimi, S.R. 'Unified Approach to Theory of Plastic Structures'. Journal of the Engineering Mechanics Division, ASCE, Vol 98, No EM 5, October 1975. pp 1133 - 1158.

5. Hodge, P.G. Jr. 'Plastic Analysis of Structures'. 1959. Mc Graw-Hill.

6. Hutula, D.N. 'Finite Element Limit Anlysis of Two-dimensional Plane Structures'. Water Annual Meeting, ASME, N.Y. December 1976. "Limit Analysis Using Finite Elements".

7. Koopman, D.C.A. and Lance, R.H. 'On Linear Programming and Plastic Limit Analysis'. Journal of the Mechanics and Physics of Solids, Vol 13, pp 77 - 87.

8. Livesly, R.K. 'Matrix Methods of Structural Analysis'. 1964, Pergamon Press.

9. Martin, J.B. 'Plasticity: Fundamentals and General Results'. 1975. M.I.T. Press

10. Massonnet, C.E. and Save, M.A. 'Plastic Analysis and Design'. Vol 1, Beams and Frames. 1965. Blaisdell Publishing Company.

11. Neal, B.G. 'The Plastic Methods of Structural Analysis'. 1963. Chapman and Hall.

APPENDIX I:

Relationship between the statics matrix [A] and the deformation matrix [B] for plane frames

We consider here frame structures of the type defined in Chapter 2. The load vector {P}, internal force vector {N}, member deformation vector {δ} and nodal displacement vector {u}, have been defined in equations (2.6), (2.2), (2.4) and (2.5) respectively.

Let the internal forces {N} be in equilibrium with external loads {P}. Let an independent set of external displacements {u*} be compatible with the member deformations {δ*}.

Using the principle of virtual work we may state that

$$\{\delta^*\}^T\{N\} = \{u^*\}^T\{P\}. \tag{I.1}$$

Transposing equation (2.18) and using equation (2.20)

$$\{u^*\}^T[B]^T\{N\} = \{u^*\}^T[A]\{N\} \tag{I.2}$$

and therefore

$$[B]^T = [A]. \tag{I.3}$$

APPENDIX II

Relationship between the statics matrix [A] and the deformation matrix [B] for plane stress problems with constant strain triangular elements.

We consider here plane stress problems with constant strain triangular elements of the type defined in Chapter 5. The load vector {P}, the internal stress vector {σ}, the internal strain vector {ε} and the nodal displacement vector {u} have all been defined in equations (5.23), (5.21), (5.12) and (5.11) respectively.

Let the equilibrium equations be set up in such a way that the stresses ensure continuity of displacement across element boundaries. We may represent the loads by {P} and the resulting stresses by {σ}, subject to the above condition. Let an independent set of displacements {u*} be compatible with a set of strains {ε*}.

Using the principle of virtual work and by equation (5.25) we may state that

$$\{\varepsilon^*\}^T[\Delta]\{\sigma\} = \{u^*\}^T\{P\}. \qquad \text{(II.1)}$$

Transposing equation (5.13) and using equation (5.29)

$$\{u^*\}^T[B]^T[\Delta]\{\sigma\} = \{u^*\}[A]\{\sigma\} \qquad \text{(II.2)}$$

and therefore

$$[B]^T[\Delta] = [A]. \qquad \text{(II.3)}$$

APPENDIX III

Computer Program for the Shakedown Analysis of Plane Frames

```
DOUGLAS(1).SHAKDN
      CALL TIME(TIME10)
      CALL INPUT(NLOADC,MEMBRS,TIME10,NPI,NPF)
      CALL ASETUP
      CALL ROWELN
      CALL COMPAT(MEMBRS)
      CALL SOLVE
      CALL ELSOUT(NLOADC,TIME10)
      CALL LPPSET(NLOADC,MEMBRS,TIME10,NPI,NPF)
      STOP
      END
```

```
DOUGLAS(1).INPUT
      SUBROUTINE INPUT(NLOADC,MEMBRS,TIME10,NPI,NPF)
C*****************************************************************
C*    THIS SUBROUTINE READS IN ALL THE INPUT DATA.                *
C*****************************************************************
      COMMON/BLOCK1/A(10)
     +        /BLOCK2/E(50),AREA(50),IZ(50),MO(50),LENGTH(50)
     +        /BLOCK3/NRA,NCA,NCAL,NDR,NDC
     +        /BLOCK4/MEA(50),MEB(50)
     +        /BLOCK5/XCORD(50),YCORD(50)
     +        /BLOCK6/RF(200),CF(200)
      DIMENSION TITLE(12)
      INTEGER RF,CF
      REAL IZ,MO
      READ(8,101)TITLE
      READ(8,100)NODES,MEMBRS,NBDC,NLOADC,NJR,NMR
      DO 500 I=1,NODES
500   READ(8,100)XCORD(I),YCORD(I)
      DO 600  I=1,MEMBRS
600   READ(8,100)MEA(I),MEB(I),E(I),AREA(I),IZ(I),MO(I)
      DO 700 I=1,NBDC
700   READ(8,100)NBN,RF(3*NBN-2),RF(3*NBN-1),RF(3*NBN)
C*****************************************************************
C*       CALCULATE THE AMOUNT OF CORE STORAGE NEEDED              *
C*          FOR THE STATICS MATRIX (A).                           *
C*****************************************************************
      NRA  = 3*NODES
      NCA  = 3*MEMBRS
      NCAL = NCA + NLOADC
      ISA  = NCAL * NRA
      CALL MCORE(A(ISA))
C*****************************************************************
C*          READ IN THE LOAD VECTORS.                             *
C*****************************************************************
      DO 1000 I=1,NLOADC
      READ(8,100)NLOADS
      DO 1000 J=1,NLOADS
      READ(8,100)N,T1,T2,T3
      N = (3*N-3)*NCAL
      K = NCA + I
      A(N + K) = T1
      N = N+NCAL
      A(N + K) = T2
      N = N+NCAL
      A(N + K) = T3
1000  CONTINUE
C*****************************************************************
C*    READ IN THE JOINT AND MEMBER RELEASES                      *
C*****************************************************************
      IF(NJR .EQ. 0)GO TO 1200
      DO 1100 I=1,NJR
      READ(8,100)N
      RF(3*N) = -2
      DO 1100 J=1,MEMBRS
      IF(MEA(J) .EQ. N) CF(3*J-1) = -1
      IF(MEB(J) .EQ. N) CF(3*J  ) = -1
1100  CONTINUE
1200  IF(NMR .EQ. 0)GO TO 1250
      DO 1210 I=1,NMR
      READ(8,100)N,M
```

```
      IF(M .EQ. MEA(N)) CF(3*N-1) = -1
      IF(M .EQ. MEB(N)) CF(3*N  ) = -1
1210  CONTINUE
C*************************************************************
C*    CALCULATE THE NO OF DELETED ROWS (NDR) AND COLUMS(NDC)
C*************************************************************
1250  NDR = 0
      NDC = 0
      DO 1300 I=1,NRA
1300  IF(RF(I) .EQ. -1 .OR. RF(I) .EQ. -2) NDR = NDR +1
      DO 1400 I=1,NCA
1400  IF(CF(I) .EQ. -1) NDC = NDC +1
C***************************************************
C*    READ IN PROGRAMMING OUTPUT OPTIONS
C******************************************************
      READ(8,100)NPI,NPF
C*************************************************************
C*         PRINT OUT THE LOAD CASES BEING ANALISED.          *
C*************************************************************
      WRITE(5,200)TITLE
      DO 1500 I=1,NLOADC
      WRITE(5,210)I
      DO 1500 J=1,NODES
      K = (3*J-3)*NCAL+NCA+I
      M = K+NCAL
      N = M+NCAL
1500   WRITE(5,220)J,A(K),A(M),A(N)
      CALL TIME(TIME11)
      TIME11 = TIME11 - TIME10
      WRITE(5,230) TIME11
100   FORMAT()
101   FORMAT(12A6)
200   FORMAT(1H1,/,1H ,12A6)
210   FORMAT(//,1H ,' LOAD CASE ',I6,/,'  NODE',T30,'FORCE X'
     +,T50,'FORCE Y',T70,'MOMENT',/)
220   FORMAT(I6,T25,F11.4,T45,F11.4,T65,F11.4)
230   FORMAT(///,'  TIME FOR DATA INPUT = ',F8.4,'  SECONDS')
      RETURN
      END
```

```
DOUGLAS(1).ASETUP
         SUBROUTINE ASETUP
      C*************************************************************
      C*     THIS SUBROUTINE SETS UP THE STATICS MATRIX (A).       *
      C*************************************************************
         COMMON/BLOCK1/A(10)
     +        /BLOCK2/E(50),AREA(50),IZ(50),MO(50),LENGTH(50)
     +        /BLOCK3/NRA,NCA,NCAL,NDR,NDC
     +        /BLOCK4/MEA(50),MEB(50)
     +        /BLOCK5/XCORD(50),YCORD(50)
     +        /BLOCK6/RF(200),CF(200)
         REAL LENGTH,IZ,MO
         J = 0
         DO 1000 I=1,NBA,3
         J = J+1
         MAJ = MEA(J)
         MBJ = MEB(J)
         LENGTH(J) = SQRT((XCORD(MBJ)-XCORD(MAJ))**2+
     +                    (YCORD(MBJ)-YCORD(MAJ))**2)
         SGMA = (YCORD(MBJ) - YCORD(MAJ)) / LENGTH(J)
         CGMA = (XCORD(MBJ) - XCORD(MAJ)) / LENGTH(J)
         SL   = SGMA / LENGTH(J)
         CL   = CGMA / LENGTH(J)
         K = (3*MAJ-3)*NCAL + I
         A(K)   = -CGMA
         A(K+1) = -SL
         A(K+2) = -SL
      C
         K = (3*MAJ-2)*NCAL + I
         A(K)   = -SGMA
         A(K+1) = +CL
         A(K+2) = +CL
      C
         K = (3*MBJ-3)*NCAL + I
         A(K)   = +CGMA
         A(K+1) = +SL
         A(K+2) = +SL
      C
         K = (3*MBJ-2)*NCAL + I
         A(K)   = +SGMA
         A(K+1) = -CL
         A(K+2) = -CL
      C
         A(3*MAJ*NCAL-NCAL + I+1) = 1.0
         A(3*MBJ*NCAL-NCAL + I+2) = 1.0
      1000 CONTINUE
         RETURN
         E N D
```

DOUGLAS(1).ROWELN

```
      SUBROUTINE ROWELN
C**********************************************************************
C*    THIS SUBROUTINE REDUCES THE STATICS MATRIX
C*   (A) TO ITS ROW-ECHELON FORM SO THAT THE
C*    INDEPENDANT SELF-STRESS SYSTEMS MAY BE IDENTIFIED
C**********************************************************************
C
       COMMON/BLOCK1/A(10)
      +       /BLOCK3/NRA,NCA,NCAL,NDR,NDC
      +       /BLOCK6/RF(200),CF(200)
       INTEGER RF,CF
       DO 10 K=1,NCA
       IF(CF(K) .EQ. -1)GO TO 10
       CF(K) = 1
10     CONTINUE
       I = 0
       J = 1
       NRC = NRA-1
C**********************************************************************
C*     LOOK FOR A NON-ZERO ELEMENT IN COLUMN
C*    J BENEATH ROW I.
C**********************************************************************
20     IF(RF(I+1) .NE. -1 .AND. RF(I+1) .NE. -2)GO TO 25
       I = I+1
       IF (I .GE. NRA)GO TO 100
       GO TO 20
25     DO 30 K = I,NRC
       IF(RF(K+1) .EQ. -1 .OR. RF(K+1) .EQ. -2)GO TO 30
       IF(ABS(A(K*NCAL + J)) .GT. 1.0E-7)GO TO 40
30     CONTINUE
       IF(J .GE. NCA)GO TO 100
C**********************************************************************
C*    IF ALL THE ELEMENTS IN COLUMN J ARE ZERO BELOW
C*    ROW J THEN THIS COLUMN IS LINEARLY DEPENDANT
C*    ON THE PREVIOUS COLUMNS. FLAG THIS COLUMN.
C**********************************************************************
       CF(J) = 1
       J = J+1
       GO TO 20
C**********************************************************************
C*     IF THERE IS ONE NON-ZERO ELEMENT , LOOK FOR
C*     THE LARGEST ELEMENT IN COLUMN J.
C**********************************************************************
40     XMAX = 0.0
       DO 42 K=I,NRC
       T = ABS(A(K*NCAL + J))
       IF(RF(K+1) .EQ. -1 .OR. RF(K+1) .EQ. -2)GO TO 42
       IF( XMAX .GE. T)GO TO 42
       L = K
       XMAX = T
42     CONTINUE
C**********************************************************************
C*      IF THE LARGEST VALUE IN COLUMN J IS NOT IN
C*      ROW I, THEN SWOP ROW I WITH THE ROW IN WHICH
C*      LARGEST ELEMENT WAS FOUND.
C**********************************************************************
       IF(L .EQ. I)GO TO 60
       DO 50 K=I,NCAL
       T = A(L*NCAL + K)
```

```
      A(L*NCAL + K) = A(I*NCAL + K)
      A(I*NCAL + K) = T
50    CONTINUE
C*********************************************************************
C*    GET 1.0 IN POSITION I,J BY DIVIDING
C*    ROW I BY THE VALUE IN A(I,J).
C*********************************************************************
60    T = A(I*NCAL + J)
      DO 65 K=1,NCAL
      IF(CF(K) .EQ. -1)GO TO 65
      A(I*NCAL + K) = A(I*NCAL + K)/T
C*********************************************************************
C*    OPERATE ON ALL THE ROWS (EXCEPT ROW I) SO THAT
C*  . THEY HAVE ZEROES IN COLUMN J.
C*********************************************************************
65    CONTINUE
      DO 72 K=0,NRC
      IF(RF(K+1) .EQ. -1 .OR. RF(K+1) .EQ. -2)GO TO 72
      IF( K .EQ. I)GO TO 72
      K2 = K*NCAL
      T  = A(K2 + J)
      DO 70 L=1,NCAL
      IF(CF(L) .EQ. -1)GO TO 70
      A(K2 + L) = A(K2 + L) - T*A(I*NCAL + L)
      IF(ABS(A(K2 + L)) .LT. 1.0E-8) A(K2 + L) = 0.0
70    CONTINUE
72    CONTINUE
C*********************************************************************
C*    IF THE PROCESS IS COMPLETE, EXIT.
C*********************************************************************
      CF(J) = 0
      IF(I .GE. NRA .OR. J .GE. NCA)GO TO 100
      I = I+1
      J = J+1
      GO TO 20
100   RETURN
      E N D
```

```
DOUGLAS(1).COMPAT
      SUBROUTINE COMPAT(MEMBRS)
      COMMON/BLOCK1/A(10)
     +      /BLOCK2/E(50),AREA(50),IZ(50),MB(50),LENGTH(50)
     +      /BLOCK3/NRA,NCA,NCAL,NDR,NDC
     +      /BLOCK6/RF(200),CF(200)
      INTEGER RF,CF
      REAL LENGTH,IZ,MB
C*****************************************************************************
C*   EXPAND CORE STORAGE TO ACCOMODATE THE COMPATABILITY REQUIREMEN
C*****************************************************************************
      ISA = (NCA-NDC+NDR)*NCAL
      CALL MCORE(A(ISA))
C*****************************************************************************
C*     SET UP THE RESIDUAL FORCE SYSTEMS FROM THE LINEARLY       *
C*     DEPENDANT COLUMNS IN (A).                                  *
C****************************************************************
      KC = 0
      DO 20 I=1,NRA
      IF(RF(I) .EQ. -1 .OR. RF(I) .EQ. -2)GO TO 20
      KR = NRA-1
      KC = KC+1
   10 IF(CF(KC) .EQ.0)GO TO 12
      KC = KC+1
      GO TO 10
   12 DO 15 J=1,NCA
      IF(CF(J) .NE. 1)GO TO 15
      KR = KR+1
      A(KR*NCAL + KC) = A(I*NCAL-NCAL + J)
   15 CONTINUE
   20 CONTINUE
      KR = NRA-1
      DO 40 J=1,NCA
      IF(CF(J) .NE. 1)GO TO 40
      KR = KR+1
      A(KR*NCAL + J) = -1.0
   40 CONTINUE
      I = 0
      J2 = 0
      K3 = NCA-NDC+NDR-1
      JR = IFIX(NCA/15)
      IF(JR .LE. 0)GO TO 751
      DO 701 I=1,JR
      J1 = 15*(I-1) + 1
      J2 = J1 + 14
      WRITE(5,201)I
  201 FORMAT(1H1,' A MATRIX (BEFORE COMPAT) PAGE NO.',I3,/)
      WRITE(5,210)(CF(J),J=J1,J2)
      DO 701 K=0,K3
      K1 = K+1
      WRITE(5,220)K1,(A(K*NCAL + J),J=J1,J2)
  701  CONTINUE
  751  J1 = J2+1
      I = I+1
      WRITE(5,201)I
      WRITE(5,210)(CF(J),J=J1,NCA)
      DO 801 K = 0,K3
      K1 = K + 1
      WRITE(5,220)K1,(A(K*NCAL + J),J=J1,NCA)
  801  CONTINUE
```

```
C
C*********************************************************************
C*   STORE THE SHAKEDOWN CONSTRAINT EQUATIONS
C*********************************************************************
      CALL ERTRAN(6,'@ASG,T 12. . ')
      K3 = NCA-NDC+NDR-1
      K4 = 2
      DO 50 I=NRA,K3
      DO 50 J=1,NCA
      K4 = K4+1
      IF(  K4  .EQ. 3 )GO TO 45
      IF(CF(J) .EQ. -1)GO TO 50
      WRITE(12)A(I*NCAL + J)
      GO TO 50
45    K4 = 0
50    CONTINUE
      END FILE 12
      REWIND 12
C*********************************************************************
C*      MULTIPLY THE SELF STRESS SYSTEMS BY THE FLEXIBILITY
C*         MATRIX (E).
C*********************************************************************
      DO 60 I=NRA,K3
      KR = I*NCAL
      DO 60 J=1,MEMBRS
      K2 = 3*J-2
      K4 = K2+1
      K5 = K4+1
      A1 = A(KR + K4)
      A2 = A(KR + K5)
      F1 = LENGTH(J)/(3.0*E(J)*IZ(J))
      F2 = LENGTH(J)/(6.0*E(J)*IZ(J))
      A(KR + K2) = (A(KR + K2)*LENGTH(J)/(E(J)*AREA(J)))
      A(KR + K4) = (A1*F1 - A2*F2)
      A(KR + K5) = (A2*F1 - A1*F2)
60    CONTINUE
C
      DO 64 I=NRA,K3
      T1 = 0.0
      DO 62 J=1,NCAL
      T1 = AMAX1(T1,ABS(A(I*NCAL + J)))
62    CONTINUE
      T1 = 0.5/T1
      DO 64 J=1,NCAL
      A(I*NCAL + J) = A(I*NCAL + J) * T1
64    CONTINUE
C
      I = 0
      J2 = 0
      JR = IFIX(NCA/15)
      IF(JR .LE. 0)GO TO 75
      DO 70 I=1,JR
      J1 = 15*(I-1) + 1
      J2 = J1 + 14
      WRITE(5,200)I
      WRITE(5,210)(BF(J),J=J1,J2)
      DO 70 K=0,K3
      K1 = K+1
      WRITE(5,220)K1,(A(K*NCAL + J),J=J1,J2)
70    CONTINUE
```

```
C
C*********************************************************************
C*   STORE THE SHAKEDOWN CONSTRAINT EQUATIONS
C*********************************************************************
      CALL ERTRAN(6,'@ASG,T 12, , ')
      K3 = NCA-NDC+NDR-1
      K4 = 2
      DO 50 I=NRA,K3
      DO 50 J=1,NCA
      K4 = K4+1
      IF(  K4   .EQ. 3 )GO TO 45
      IF(CF(J) .EQ. -1)GO TO 50
      WRITE(12)A(I*NCAL + J)
      GO TO 50
45    K4 = 0
50    CONTINUE
      END FILE 12
      REWIND 12
C*********************************************************************
C*      MULTIPLY THE SELF STRESS SYSTEMS BY THE FLEXIBILITY
C*          MATRIX (F).
C*********************************************************************
      DO 60 I=NRA,K3
      KR = I*NCAL
      DO 60 J=1,MEMBRS
      K2 = 3*J-2
      K4 = K2+1
      K5 = K4+1
      A1 = A(KR + K4)
      A2 = A(KR + K5)
      F1 = LENGTH(J)/(3.0*E(J)*IZ(J))
      F2 = LENGTH(J)/(6.0*E(J)*IZ(J))
      A(KR + K2) = (A(KR + K2)*LENGTH(J)/(E(J)*AREA(J)))
      A(KR + K4) = (A1*F1 - A2*F2)
      A(KR + K5) = (A2*F1 - A1*F2)
60    CONTINUE
C
      DO 64 I=NRA,K3
      T1 = 0.0
      DO 62 J=1,NCAL
      T1 = AMAX1(T1,ABS(A(I*NCAL + J)))
62    CONTINUE
      T1 = 0.5/T1
      DO 64 J=1,NCAL
      A(I*NCAL + J) = A(I*NCAL + J) * T1
64    CONTINUE
C
      I = 0
      J2 = 0
      JR = IFIX(NCA/15)
      IF(JR .LE. 0)GO TO 75
      DO 70 I=1,JR
      J1 = 15*(I-1) + 1
      J2 = J1 + 14
      WRITE(5,200)I
      WRITE(5,210)(CF(J),J=J1,J2)
      DO 70 K=0,K3
      K1 = K+1
      WRITE(5,220)K1,(A(K*NCAL + J),J=J1,J2)
70    CONTINUE
```

```
C
C*****************************************************************
C*    STORE THE SHAKEDOWN CONSTRAINT EQUATIONS
C*****************************************************************
      CALL ERTRAN(6,'@ASG,T 12. . ')
      K3 = NCA-NDC+NDR-1
      K4 = 2
      DO 50 I=NRA,K3
      DO 50 J=1,NCA
      K4 = K4+1
      IF(  K4  .EQ. 3 )GO TO 45
      IF(CF(J) .EQ. -1)GO TO 50
      WRITE(12)A(I*NCAL + J)
      GO TO 50
45    K4 = 0
50    CONTINUE
      END FILE 12
      REWIND 12
C*****************************************************************
C*       MULTIPLY THE SELF STRESS SYSTEMS BY THE FLEXIBILITY
C*          MATRIX (F).
C*****************************************************************
      DO 60 I=NRA,K3
      KR = I*NCAL
      DO 60 J=1,MEMBRS
      K2 = 3*J-2
      K4 = K2+1
      K5 = K4+1
      A1 = A(KR + K4)
      A2 = A(KR + K5)
      F1 = LENGTH(J)/(3.0*E(J)*IZ(J))
      F2 = LENGTH(J)/(6.0*E(J)*IZ(J))
      A(KR + K2) = (A(KR + K2)*LENGTH(J)/(E(J)*AREA(J)))
      A(KR + K4) = (A1*F1 - A2*F2)
      A(KR + K5) = (A2*F1 - A1*F2)
60    CONTINUE
C
      DO 64 I=NRA,K3
      T1 = 0.0
      DO 62 J=1,NCAL
      T1 = AMAX1(T1,ABS(A(I*NCAL + J)))
62    CONTINUE
      T1 = 0.5/T1
      DO 64 J=1,NCAL
      A(I*NCAL + J) = A(I*NCAL + J) * T1
64    CONTINUE
C
      I = 0
      J2 = 0
      JR = IFIX(NCA/15)
      IF(JR .LE. 0)GO TO 75
      DO 70 I=1,JR
      J1 = 15*(I-1) + 1
      J2 = J1 + 14
      WRITE(5,200)I
      WRITE(5,210)(DF(J),J=J1,J2)
      DO 70 K=0,K3
      K1 = K+1
      WRITE(5,220)K1,(A(K*NCAL + J),J=J1,J2)
70    CONTINUE
```

```
75     J1 = J2+1
       I = I+1
       WRITE(5,200)I
       WRITE(5,210)(CF(J),J=J1,NCA)
       DO 80 K = 0,K3
       K1 = K + 1
       WRITE(5,220)K1,(A(K*NCAL + J),J=J1,NCA)
80     CONTINUE
200    FORMAT(1H1,'  A MATRIX SET-UP   PAGE NO: ',I3,/)
210    FORMAT(15I8)
220    FORMAT(I4,15E8.2)
       RETURN
       END
```

```
UGLAS(1).SOLVE
      SUBROUTINE SOLVE
C*********************************************************************
C*       THIS SUBROUTINE SOLVES FOR THE ELASTIC FORCES.
C*       IT USES GAUSS-JORDAN REDUCTION WITH PARTIAL PIVOTING.
C*********************************************************************
      COMMON/BLOCK1/A(10)
     +      /BLOCK3/NRA,NCA,NCAL,NDR,NDC
     +      /BLOCK6/RF(200),CF(200)
      INTEGER RF,CF
      I = 0
      J = I
      NRC = NCA-NDC+NDR-1
20    IF(RF(I+1) .NE. -1 .AND. RF(I+1) .NE. -2)GO TO 25
      I = I+1
      GO TO 20
25    DO 30 K = I,NRC
      IF(RF(K+1) .EQ. -1 .OR. RF(K+1) .EQ. -2)GO TO 30
      IF(ABS(A(K*NCAL + J)) .GT. 1.0E-7)GO TO 40
30    CONTINUE
C*********************************************************************
C*    IF ALL THE ELEMENTS IN COLUMN I ARE ZERO BELOW I
C*    THEN THE I TH COLUMN IS LINEARLY DEPENDANT ON THE
C*    PREVIOUS COLUMNS, HENCE WE HAVE A MECHANISM.
C*********************************************************************
      N = I/3
      M = J/3
      WRITE(5,600)N,M
      STOP
40    XMAX = 0.0
      DO 42 K=I,NRC
      T = ABS(A(K*NCAL + J))
      IF(RF(K+1) .EQ. -1 .OR. RF(K+1) .EQ. -2)GO TO 42
      IF(  XMAX  .GE.  T)GO TO 42
      L = K
      XMAX = T
42    CONTINUE
      IF(L .EQ. I)GO TO 60
      DO 50 K=I,NCAL
      T = A(L*NCAL + K)
      A(L*NCAL + K) = A(I*NCAL + K)
      A(I*NCAL + K) = T
50    CONTINUE
60    T = A(I*NCAL + J)
      DO 65 K=I,NCAL
      IF(CF(K) .EQ. -1)GO TO 65
      A(I*NCAL + K) = A(I*NCAL + K)/T
65    CONTINUE
      DO 72 K=0,NRC
      IF(RF(K+1) .EQ. -1 .OR. RF(K+1) .EQ. -2)GO TO 72
      IF(    K    .EQ.   I)GO TO 72
      K2 = K*NCAL
      T  = A(K2 + J)
      IF(T .EQ. 0)GO TO 72
      DO 70 L=I,NCAL
      IF(CF(L) .EQ. -1)GO TO 70
      A(K2 + L) = A(K2 + L) - T*A(I*NCAL + L)
      IF(ABS(A(K2 + L)) .LT. 1.0E-8) A(K2 + L) = 0.0
70    CONTINUE
72    CONTINUE
```

```
C**********************************************************************
C*       IF MATRIX REDUCTION IS COMPLETE, EXIT.
C**********************************************************************
      IF(I .GE. NRC .OR. J .GE. NCA)GO TO 100
      I = I+1
      J = J+1
      GO TO 20
  600 FORMAT(1H1,//,1H ,' THE AUGMENTED STATICS MATRIX IS',
     +' SINGULAR.',//,'  STRUCTURE IS A MECHANISM.',//
     +'  SOLUTION TERMINATED.',//,
     +' SINGULARITY OCCURS NEAR THE NODE ',I6,/,
     +'                              AND MEMBER  ',I6)
  100 RETURN
      E N D
```

```
DOUGLAS(1).ELSOUT
          SUBROUTINE ELSOUT(NLOADC,TIME10)
          COMMON/BLOCK1/A(10)
     +          /BLOCK2/E(50),AREA(50),IZ(50),MU(50),LENGTH(50)
     +          /BLOCK3/NRA,NCA,NCAL,NDR,NDC
     +          /BLOCK4/MEA(50),MEB(50)
     +          /BLOCK6/RF(200),CF(200)
          DIMENSION B(3)
          REAL LENGTH,IZ,MU
          INTEGER RF,CF
          WRITE(5,200)
          NRC = NCA-NDC+NDR
          DO 40 I=1,NLOADC
          M = 0
          WRITE(5,210)I
          WRITE(5,220)
          WRITE(5,230)
          K = 0
          DO 40 J=1,NRC
          IF(RF(J) .EQ. -1)GO TO 40
          K = K+1
          IF(RF(J) .NE. -2)GO TO 10
          B(K) = 0.0
          GO TO 20
   10     B(K) = A(J+NCAL-NCAL + NCA+I)
   20     IF( K  .LT.  3)GO TO 40
          M = M+1
          S = (B(2)+B(3))/LENGTH(M)
          WRITE(5,240)M,B(1),S,MEA(M),B(2),MEB(M),B(3)
          K = 0
   40     CONTINUE
          CALL TIME(TIME12)
          TIME12 = TIME12 - TIME10
          WRITE(5,250) TIME12
   C*************************************************************
   C*     FORMAT STATEMENTS FOLLOW.                             *
   C*************************************************************
   200    FORMAT(1H1,/,T40,' ELASTIC SOLUTION',/,T41,'================')
   210    FORMAT(//,T40,' L O A D   C A S E ',I3,/)
   220    FORMAT(T11,'AXIAL',T25,'SHEAR',T46,'END MOMENTS')
   230    FORMAT(' MEMBER',T11,'FORCE',T25,'FORCE',T34,'NODE',
         +T43,'MOMENT',T53,'NODE',T65,'MOMENT')
   240    FORMAT(I5,T7,F12.4,T20,F12.4,T34,I5,T40,F12.4,T53,I5,T60,F12.4
   250    FORMAT(///,'  TIME FOR ELASTIC SOLUTION = ',F8.4,
         +' SECONDS.')
          RETURN
          END
```

```
DOUGLAS(1).LPPSET
      C**********************************************************************
      C*    THIS SUBROUTINE SETS UP THE LINEAR PROGRAMMING PROBLEM.
      C**********************************************************************
            SUBROUTINE LPPSET(NLOADC,MEMBRS,TIME10,NPI,NPF)
            DIMENSION TOL(4),IFIX(20),JX(100),IOUT(4),X(100),Y(100),S(100)
           +,RHS(100),COST(100),T(100),EE(52,52),RI(50),ERR(4)
            COMMON/BLOCK1/A(10)
           +        /BLOCK2/E(50),AREA(50),IZ(50),MO(50),LENGTH(50)
           +        /BLOCK3/NRA,NCA,NCAL,NDR,NDC
           +        /BLOCK4/MEA(50),MEB(50)
           +        /BLOCK6/RF(200),CF(200)
            REAL LENGTH,IZ,MO
            INTEGER CF,RF
      C**********************************************************************
      C*     FOR THE SHAKEDOWN ANALYSIS, FIND THE MAXIMUM AND MINIMUM
      C*     MOMENTS AT EACH NODE. USE THESE AS THE FIRST CONSTRAINT.
      C**********************************************************************
            K1 = 1
            K2 = 2
            NRAC = NCA-NDC+NDR
            DO 20 I=1,NRAC
            K2 = K2+1
            IF( K2  .EQ.  3)GO TO 15
            IF(RF(I).EQ. -1 .OR. RF(I) .EQ. -2)GO TO 20
            XMAX = A(I*NCAL-NCAL + NCA+1)
            XMIN = A(I*NCAL-NCAL + NCA+1)
            IF(NLOADC .EQ. 1)GO TO 12
            DO 10 J=2,NLOADC
            XMAX = AMAX1(XMAX,A(I*NCAL-NCAL+NCA+J))
            XMIN = AMIN1(XMIN,A(I*NCAL-NCAL+NCA+J))
       10   CONTINUE
       12   RHS(K1   ) = +XMAX
            RHS(K1+1) = -XMIN
            K1 = K1+2
            GO TO 20
       15   K2 = 0
       20   CONTINUE
            NCC = K1-1
            NRC = NCA-NDC+NDR-NRA
            NRL = NRC+1
            NCM = NCC-1
      C*******************************************************************
      C*      CALCULATE THE COLLAPSE LOAD FACTOR FOR
      C*           ALTERNATING PLASTICITY .
      C*******************************************************************
            IF(NLOADC .LE. 1)GO TO 32
            K1 = 1
            ALTM = 1000.0
            DO 25 I = 1,MEMBRS
            IF(CF(3*I-1) .EQ. -1 .AND. CF(3*I) .EQ. -1)GO TO 25
            IF(CF(3*I-1) .EQ. -1)GO TO 22
            TEMP = 2.0*MO(I) / (RHS(K1)+RHS(K1+1))
            IF(ALTM .LT. TEMP)GO TO 24
            ALTM = TEMP
            KMAL = I
            KJAL = MEA(I)
       24   K1 = K1 + 2
       22   IF(CF(3*I) .EQ. -1)GO TO 25
            TEMP = 2.0*MO(I) / (RHS(K1) + RHS(K1+1))
```

```
      IF(ALTM .LT. TEMP)GO TO 26
      ALTM = TEMP
      KMAL = I
      KJAL = MEB(I)
26    K1 = K1 + 2
25    CONTINUE
      GO TO 36
C
32    ALTM = -1.0
C
36    DO 30 J=0,NCM
      A(1+J*NRL) = RHS(J+1)
30    CONTINUE
C*******************************************************************
C*   ADJUST CORE REQUIREMENTS FOR THE L.P.P. PROBLEM.
C*******************************************************************
      K1 = (NCA-NDC+NDR)*NCAL
      K2 = NCC*NRL
      IF( K1 .GT. K2)CALL LCORE(A(K2))
      IF( K1 .LT. K2)CALL MCORE(A(K2))
C************************************************************************
C*       SET UP THE OTHER CONSTRAINTS FROM THE STORED SELF STRESS
C*    SYSTEMS OBTAINED DURING THE ELASTIC ANALYSIS.
C************************************************************************
      DO 40 I=2,NRL
      DO 40 J=1,NCC,2
      READ(12)TEMP
      A(I + J*NRL-NRL) =  TEMP
      A(I + J*NRL)     = -TEMP
40    CONTINUE
C************************************************************************
C*       SET UP THE RIGHT HAND SIDE VECTOR  (RHS).
C************************************************************************
      RHS(1) = 1.0
      DO 60 I=2,NRL
      RHS(I) = 0.0
60    CONTINUE
C************************************************************************
C*       SET UP THE COST VECTOR.
C************************************************************************
      K2 = 1
      DO 80 I=1,MEMBRS
      IF(CF(3*I-1) .EQ. -1 .AND. CF(3*I) .EQ. -1)GO TO 80
      IF(CF(3*I-1) .EQ. -1 .OR . CF(3*I) .EQ. -1)GO TO 75
      COST(K2  ) = MO(I)
      COST(K2+1) = MO(I)
      COST(K2+2) = MO(I)
      COST(K2+3) = MO(I)
      K2 = K2+4
      GO TO 80
75    COST(K2  ) = MO(I)
      COST(K2+1) = MO(I)
      K2 = K2+2
80    CONTINUE
      IFIX( 1) = NRL
      IFIX( 2) = NCC
      IFIX( 3) = NRL
      IFIX( 4) = NCC
      IFIX( 5) = 0
      IFIX( 6) = 5000
```

```
      IFIX( 7) = 2*NRL
      IFIX( 8) = 4000
      IFIX( 9) = 0
      IFIX(10) = 0
      IFIX(11) = NPI
      IFIX(12) = NPF
      IFIX(13) = 0
      IFIX(14) = 1
      IFIX(15) = 52
      IFIX(16) = NCC
      IFIX(17) = 1
      IFIX(18) = 'SHAKDN'
      IFIX(19) = 0
      IFIX(20) = 0
C
      TOL(1) = 0.0
      TOL(2) = 0.0
      TOL(3) = 0.0
      TOL(4) = 0.0
C
C**********************************************************
C*   CALL THE LINEAR PROGRAMMING PACKAGE TO SOLVE FOR THE     *
C*   LOWEST COLLAPSE LOAD FACTOR.                              *
C**********************************************************
      CALL SIMPLX(T,RHS,COST,IFIX,TOL,OBJ,X,JX,PI,EE,ERR,IOUT,Y,S)
      CALL SKDNO(OBJ,X,JX,IOUT,ALTM,KMAL,KJAL,MEMBRS,NRL,NCC,TIMEIO
      RETURN
      END
```

DOUGLAS(1).SKDNO

```
      SUBROUTINE SKDNO(OBJ,X,JX,IOUT,ALTM,KMAL,KJAL,MEMBRS,NRL,NCC
     +                ,TIME10)
      DIMENSION JX(100),X(100),IOUT(4)
      COMMON/BLOCK3/NRA,NCA,NCAL,NDR,NDC
     +      /BLOCK4/MEA(50),MEB(50)
     +      /BLOCK6/RF(200),CF(200)
C
      WRITE(5,200)
      IF(IOUT(1) .EQ. 1) WRITE(5,211)
      IF(IOUT(1) .EQ. 2) WRITE(5,212)
      IF(IOUT(1) .EQ. 3) WRITE(5,213)
      IF(IOUT(1) .EQ. 4) WRITE(5,214)
      IF(IOUT(1) .EQ. 5) WRITE(5,215)
      IF(IOUT(1) .EQ. 6) WRITE(5,216)
C
      IF(ALTM .LE. -0.05)GO TO 3
C
      T = AMIN1(ALTM,OBJ)
      WRITE(5,220)OBJ
      WRITE(5,221)ALTM
      WRITE(5,222)T
      GO TO 4
C
    3 WRITE(5,220) OBJ
      WRITE(5,225)
C
    4 WRITE(5,230)
C
      T = 0.0
      DO 5 I=1,NRL
    5 T = AMAX1(T,X(I))
      DO 6 I=1,NRL
    6 X(I) = X(I)/T
C
      K1 = 1
      DO 10 I=1,MEMBRS
      WRITE(5,235)
      T1 = ' '
      IF(CF(3*I-1) .EQ. -1)GO TO 30
      DO 20 J=1,NRL
      IF(JX(J) .EQ. K1   )GO TO 34
      IF(JX(J) .EQ. K1+1)GO TO 36
   20 CONTINUE
      GO TO 32
   30 T1 = '*'
   32 WRITE(5,240)I,MEA(I),T1
      GO TO 22
C
   34 T1 = '+'
      T2 = X(J)
      GO TO 38
   36 T1 = '-'
      T2 = -X(J)
   38 WRITE(5,242)I,MEA(I),T1,T2
C
C
   22 K1 = K1 + 2
      T1 = ' '
      IF(CF(3*I) .EQ. -1)GO TO 40
```

```
      DO 25 J=1,NRL
      IF(JX(J) .EQ. K1  )GO TO 44
      IF(JX(J) .EQ. K1+1)GO TO 46
25    CONTINUE
      GO TO 42
40    T1 = '*'
42.   WRITE(5,250)MED(I),T1
      GO TO 50
44    T1 = '+'
      T2 = X(J)
      GO TO 48
46    T1 = '-'
      T2 = -X(J)
48    WRITE(5,252)MED(I),T1,T2
50    K1 = K1 + 2
10    CONTINUE
C
      IF(ALTM .LE. -0.05)GO TO 60
      WRITE(5,260)KMAL,KJAL
C
60    CALL TIME( TIME20 )
      T = TIME20 - TIME10
      WRITE(5,280) T
C*********************************************************************
C*    FORMAT  STATEMENTS  FOLLOW.                                    *
C*********************************************************************
200   FORMAT(1H1,T30,' S H A K E D O W N   S O L U T I O N.',/
     +,1H ,T31,36('='),///)
211   FORMAT(1H ,T20,' SOLUTION IS FEASIBLE AND OPTIMAL.')
212   FORMAT(1H ,T20,' NON - FEASIBLE SOLUTION.')
213   FORMAT(1H ,T20,' SOLUTION IS UNBOUNDED.')
214   FORMAT(1H ,T20,' ITERATION LIMIT REACHED.!)
215   FORMAT(1H ,T20,' ERROR IN INPUT TO LPP DETECTED.')
216   FORMAT(1H ,T20,' OPTIMUM NOT CHANGED IN LAST 4000 CYCLES.')
C
220   FORMAT(//,1H ,T20,' INCREMENTAL COLLAPSE LOAD FACTOR =',F8.4)
221   FORMAT(//,1H ,T20,' ALTERNATING PLASTICITY COLLAPSE LOAD',
     +' FACTOR =',F8.4)
222   FORMAT(///1H ,T20,' MINIMUM COLLAPSE LOAD FACTOR =',F8.4,/,
     +T21,6H('='))
225   FORMAT(///,1H ,T20,' ONLY A SINGLE LOAD CASE - SO WE HAVE'
     +,' ONLY THE LIMIT ANALYSIS PROBLEM.')
C
230   FORMAT(///,1H ,T20,' INCREMENTAL COLLAPSE LOAD MECHANISMS.',/
     +,T20,' HINGE POSITION',10X,'ROTATION',/,1H ,T20,
     +' MEMBER  JOINT')
235   FORMAT(1H )
240   FORMAT(1H ,T22,I4,2X,I4,A1)
242   FORMAT(1H ,T22,I4,2X,I4,A1,10X,F8.4)
250   FORMAT(1H ,T20,I4,A1)
252   FORMAT(1H ,T28,I4,A1,10X,F8.4)
C
260   FORMAT(//,T20,' ALTERNATING HINGE IN MEMBER:',I3,/
     +,T39,'AT  JOINT:',I3)
280   FORMAT(//,T21,' TOTAL EXECUTION TIME =',F9.4,' SECONDS.')
      RETURN
      END
```

```
      DO 25 J=1,NRL
      IF(JX(J) .EQ. K1   )GO TO 44
      IF(JX(J) .EQ. K1+1)GO TO 46
25    CONTINUE
      GO TO 42
40    T1 = '*'
42    WRITE(5,250)MEB(I),T1
      GO TO 50
44    T1 = '+'
      T2 = X(J)
      GO TO 48
46    T1 = '-'
      T2 = -X(J)
48    WRITE(5,252)MEB(I),T1,T2
50    K1 = K1 + 2
10    CONTINUE
C
      IF(ALTM .LE. -0.05)GO TO 60
      WRITE(5,260)KMAL,KJAL
C
60    CALL TIME( TIME20 )
      T = TIME20 - TIME10
      WRITE(5,280) T
C*****************************************************************
C*    FORMAT  STATEMENTS  FOLLOW.                                *
C*****************************************************************
200   FORMAT(1H1,T30,' S H A K E D O W N   S O L U T I O N.',/
     +,1H ,T31,36('='),///)
211   FORMAT(1H ,T20,' SOLUTION IS FEASIBLE AND OPTIMAL.')
212   FORMAT(1H ,T20,' NON - FEASIBLE SOLUTION.')
213   FORMAT(1H ,T20,' SOLUTION IS UNBOUNDED.')
214   FORMAT(1H ,T20,' ITERATION LIMIT REACHED.')
215   FORMAT(1H ,T20,' ERROR IN INPUT TO LPP DETECTED.')
216   FORMAT(1H ,T20,' OPTIMUM NOT CHANGED IN LAST 4000 CYCLES.')
C
220   FORMAT(//,1H ,T20,' INCREMENTAL COLLAPSE LOAD FACTOR =',F8.4)
221   FORMAT(//,1H ,T20,' ALTERNATING PLASTICITY COLLAPSE LOAD',
     +' FACTOR =',F8.4)
222   FORMAT(///1H ,T20,' MINIMUM COLLAPSE LOAD FACTOR =',F8.4,/,
     +T21,60('='))
225   FORMAT(///,1H ,T20,' ONLY A SINGLE LOAD CASE - SO WE HAVE'
     +,' ONLY THE LIMIT ANALYSIS PROBLEM.')
C
230   FORMAT(///,1H ,T20,' INCREMENTAL COLLAPSE LOAD MECHANISMS.',/
     +,T20,' HINGE POSITION',10X,'ROTATION',/,1H ,T20,
     +' MEMBER  JOINT')
235   FORMAT(1H )
240   FORMAT(1H ,T22,I4,2X,I4,A1)
242   FORMAT(1H ,T22,I4,2X,I4,A1,10X,F8.4)
250   FORMAT(1H ,T20,I4,A1)
252   FORMAT(1H ,T20,I4,A1,10X,F8.4)
C
260   FORMAT(//,T20,' ALTERNATING HINGE IN MEMBER:',I3,/
     +,T39,'AT  JOINT:',I3)
280   FORMAT(//,T21,' TOTAL EXECUTION TIME =',F9.4,' SECONDS.')
      RETURN
      END
```

Example of data input and computer printout for Example 4.1(c)

Data Input

```
ASD*DOUGLAS(1).SDATA3
LOAD CYCLE  (G,0) (G,G) (-G,G) (-G,0)
5,4,2,4,0,0
0.0,0.0
0.0,1.0
1.0,1.0
2.0,1.0
2.0,0.0
1,2,200000000.0,0.1,0.00001,1.0
2,3,200000000.0,0.1,0.00001,1.0
3,4,200000000.0,0.1,0.00001,1.0
4,5,200000000.0,0.1,0.00001,1.0
1,-1,-1,-1
5,-1,-1,-1
1
2,1.0,0.0,0.0
2
2,1.0,0.0,0.0
3,0.0,-1.0,0.0
2
2,-1.0,0.0,0.0
3,0.0,-1.0,0.0
1
2,-1.0,0.0,0.0
1,1
```

Explanation of Data Input

line 1 :	Title of example
line 2 :	Structural parameters kj, km, kr, kℓ, kjr, kmr
	kj = number of nodes
	km = number of members
	kr = number of restrained nodes
	kℓ = number of load points on load cycle
	kjr = number of node releases
	kmr = number of member releases
lines 3 to 7 :	Cartesian co-ordinate X, Y of each node
lines 8 to 11 :	Member end nodes (i.e. member incidences), E (Youngs Modulus), A (Cross-sectional Area), I (Moment of Inertia) and M_o (Limit Moment) of each member
lines 12 to 13 :	Number of the constrained node and the constraint in the X, Y, θ direction (- 1 = constrained; 0 = unconstrained)
line 14 :	Number of loaded nodes for the 1st load point
line 15 :	Loaded node number and load components (in the X, Y and θ directions)
line 16 :	Number of loaded nodes for the 2nd load point
lines 17 to 18 :	Loaded node numbers and load components
line 19 :	Number of loaded nodes for the 3rd load point
lines 20 to 21 :	Loaded node number and load components
line 22:	Number of loaded nodes for the 4th load point
line 23 :	Loaded node number and load components

line 24 : Intermediate and final printout indicators (1 = Printout Intermediate and/or Final Information from SIMPLX subroutine, 0 = suppress printout)

) CYCLE (G,0) (G,G) (-G,G) (-G,0)

AD CASE 1

DE	FORCE X	FORCE Y	MOMENT
1	.0000	.0000	.0000
2	1.0000	.0000	.0000
3	.0000	.0000	.0000
4	.0000	.0000	.0000
5	.0000	.0000	.0000

AD CASE 2

DE	FORCE X	FORCE Y	MOMENT
1	.0000	.0000	.0000
2	1.0000	.0000	.0000
3	.0000	-1.0000	.0000
4	.0000	.0000	.0000
5	.0000	.0000	.0000

AD CASE 3

DE	FORCE X	FORCE Y	MOMENT
1	.0000	.0000	.0000
2	-1.0000	.0000	.0000
3	.0000	-1.0000	.0000
4	.0000	.0000	.0000
5	.0000	.0000	.0000

AD CASE 4

DE	FORCE X	FORCE Y	MOMENT
1	.0000	.0000	.0000
2	-1.0000	.0000	.0000
3	.0000	.0000	.0000
4	.0000	.0000	.0000
5	.0000	.0000	.0000

E FOR DATA INPUT = .2828 SECONDS

ELASTIC SOLUTION

L O A D   C A S E   1

BER	AXIAL FORCE	SHEAR FORCE	NODE	END MOMENTS MOMENT	NODE	MOMENT
1	.1875	.5002	1	.3127	2	.1875
2	-.4998	-.1875	2	-.1875	3	.0001
3	-.4998	-.1875	3	-.0001	4	-.1874
4	-.1875	.4998	4	.1874	5	.3123

L O A D   C A S E   2

BER	AXIAL FORCE	SHEAR FORCE	NODE	END MOMENTS MOMENT	NODE	MOMENT
1	-.3125	.2004	1	.2128	2	-.0124
2	-.7996	.3125	2	.0124	3	.3001
3	-.7996	-.6875	3	-.3001	4	-.3874
4	-.6875	.7996	4	.3874	5	.4122

L O A D   C A S E   3

BER	AXIAL FORCE	SHEAR FORCE	NODE	END MOMENTS MOMENT	NODE	MOMENT
1	-.6875	-.3001	1	-.4126	2	-.3875
2	.1999	.6875	2	.3875	3	.3000
3	.1999	-.3125	3	-.3000	4	-.0125
4	-.3125	-.1999	4	.0125	5	-.2124

L O A D   C A S E   4

BER	AXIAL FORCE	SHEAR FORCE	NODE	END MOMENTS MOMENT	NODE	MOMENT
1	-.1875	-.5002	1	-.3127	2	-.1875
2	.4998	.1875	2	.1875	3	-.0001
3	.4998	.1875	3	.0001	4	.1874
4	.1875	-.4998	4	-.1874	5	-.3123

E FOR ELASTIC SOLUTION = 1.0640 SECONDS.

I N T E R M E D I A T E   O U T P U T

TIME OF INVERSION 1 = .011 SEC

ITERATION	PHASE	OBJECTIVE VALUE	SUM OF INFEAS	IN COLUMN	OUT COLUMN	MINIMUM REDUCED COST	PIVOT RATIO	DETERMINANT	ITERATION TIME-MILSEC	ITER. SINCE INVERSION
1	1	.10000000+01	.10+01	4	0	-.32+01	.00	.20+01	7.8	1
2	1	.10000000+01	.10+01	1	0	-.25+01	.00	.20+01	2.8	2
3	1	.10000000+01	.10+01	13	0	-.19+01	.00	.20+01	8.0	3
4	1	.44703489-07	.45-07	15	0	-.13+01	.77+00	.26+01	9.0	4
5	2	.27575557+01	.45-07	2	15	-.23+00	.14+01	.15+01	9.0	5
6	2	.27575557+01	.45-07	16	6	-.41+00	.00	.15+01	7.8	6

SOLUTION IS OPTIMAL. MIN. OBJECTIVE = .27575557+01
THE LAST INVERSION TO CHECK THE INVERSE

TIME OF INVERSION 2 = .018 SEC

SOLUTION IS OPTIMAL. MIN. OBJECTIVE = .27575557+01

F I N A L   O U T P U T   R E P O R T

SOLUTION IS OPTIMAL	
MINIMUM OBJECTIVE	.27575557+01
SUM OF INFEASIBILITIES	.00
SUM OF /AZ-RHS/	.75-08
SUM OF /DJ/ IN BASIS	.45-07
MAXIMUM DJ IN BASIS	.30-07
PIVOT TOLERANCE	.10-04
REDUCED COST TOLERANCE	-.10-02
FEASIBILITY TOLERANCE	.10-02
SOLUTION TIME SECS	.25
A- MATRIX DENSITY	.6563
TOTAL ITERATIONS	6
TOTAL INVERSIONS	2

SOLUTION VECTOR

INDEX	JX	X
1	1	.13787778+01
2	2	.13787778+01
3	13	.00000000
4	16	.00000000

S H A K E D O W N   S O L U T I O N.

SOLUTION IS FEASIBLE AND OPTIMAL.

INCREMENTAL COLLAPSE LOAD FACTOR = 2.7576

ALTERNATING PLASTICITY COLLAPSE LOAD FACTOR = 2.7576

MINIMUM COLLAPSE LOAD FACTOR = 2.7576

INCREMENTAL COLLAPSE LOAD MECHANISMS.

HINGE POSITION MEMBER	JOINT	ROTATION
1	1+	1.0000
	2	
2	2	
	3	
3	3	
	4	
4	4+	.0000
	5-	.0000

ALTERNATING HINGE IN MEMBER: 1
AT JOINT: 1

TOTAL EXECUTION TIME = 1.5566 SECONDS.

APPENDIX IV

Computer Program for the Limit Analysis of Plane Stress Problems

```
S(1).LAPT3
      CALL TIME(TIME1)
      CALL INPUT2
      CALL ASET2
      CALL HOMEN2
      CALL COMPT2
      CALL SOLVE2
      CALL ELSLN2(TIME1)
      CALL LPSET3(TIME1)
      STOP EX END
      END
```

```
DOUGLAS(1).INPUT2
      SUBROUTINE INPUT2
      COMMON/BLOCK1/A(150,150)
     +       /BLOCK2/RF(150),CF(150)
     +       /BLOCK3/NRA1,NRA2,NRA3,NCA1,NCA2,NEL,NNP
     +       /BLOCK6/E,NU,YIELD(3)
     +       /BLOCK7/NPI(50),NPJ(50),NPK(50),X(50),Y(50)
      REAL NU
      INTEGER RF,CF
      DIMENSION TITLE(12)
C**************************************************************
C*             VARIABLE LIST                                  *
C*             =============                                  *
C*                                                            *
C* NNP    = NO OF NODE POINTS                                 *
C* NEL    = NO OF ELEMENTS
C* NBN    = NO OF BOUNDARY NODES
C* NLN    = NO OF LOADED NODES
C* NPI(I) = NODE POINT I OF ELEMENT I
C* NPJ(I) = NODE POINT J OF ELEMENT I
C* NPK(I) = NODE POINT K OF ELEMENT I
C* X(I)   = X COORDINATE OF NODE I
C* Y(I)   = Y COORDINATE OF NODE I
C**************************************************************
C
C*****************************************************************
C*  READ IN THE TITLE AND PROBLEM SIZE.                          *
C*****************************************************************
      READ (8,100) TITLE
      WRITE(5,110) TITLE
      READ (8,101)NNP,NEL,NBN,NLN
      WRITE(5,102)NNP,NEL,NBN,NLN
      NRA1 = 2*NNP
      NRA2 = NRA1
      NCA1 = 3*NEL
      NCA2 = NCA1 + 1
C**************************************************************
C*        READ IN THE BOUNDARY NODE NUMBERS.                  *
C**************************************************************
      WRITE(5,107)
      DO 60 I=1,NBN
      READ (8,101)J,J1,J2
      WRITE(5,108)J,J1,J2
      IF(J1 .EQ. 0 .AND. J2. EQ. 0)GO TO 60
      IF(J1 .EQ. 0) GO TO 55
      IF(J2 .EQ. 0) GO TO 50
      RF(2*J-1) = 1
      RF(2*J  ) = 1
      NRA2 = NRA2 - 2
      GO TO 60
50    RF(2*J-1) = 1
      NRA2 = NRA2 - 1
      GO TO 60
55    RF(2*J)   = 1
      NRA2 = NRA2 - 1
60    CONTINUE
      NRA3 = NCA1 + NRA1 - NRA2
C*****************************************************************
C*     READ IN THE NODE COORDINATES.                              *
C*****************************************************************
```

```
      WRITE(5,103)
      DO 20 I=1,NNP
      READ (8,101)  X(I),Y(I)
      WRITE(5,104)I,X(I),Y(I)
20    CONTINUE
C*****************************************************
C*     READ IN THE ELEMENT NODES.                   *
C*****************************************************
      WRITE(5,105)
      DO 40 I=1,NEL
      READ (8,101)  NPI(I),NPJ(I),NPK(I)
40    WRITE(5,106)I,NPI(I),NPJ(I),NPK(I)
C*****************************************************
C*      READ IN THE LOAD VECTOR.                    *
C*****************************************************
      WRITE(5,109)
      DO 80 I=1,NLN
      READ (8,101)J,T1,T2
      WRITE(5,120)J,T1,T2
      J1 = 2*J-1
      J2 = 2*J
      A(J1,NCA2) = T1
      A(J2,NCA2) = T2
80    CONTINUE
C*****************************************************
C*      READ IN THE MATERIAL PROPERTIES.            *
C*****************************************************
      READ (8,101)E,NU,(YIELD(I),I=1,3)
      WRITE(5,121)E,NU,(YIELD(I),I=1,3)
      WRITE(5,150)NRA1,NRA2,NRA3,NCA1,NCA2
150   FORMAT(' NRA1 =',I3,': NRA2 =',I3,': NRA3 =',I3
     +,': NCA1 =',I3,': NCA2 =',I3)
C*****************************************************
C*      FORMAT STATEMENTS FOLLOW.                   *
C*****************************************************
100   FORMAT(12A6)
110   FORMAT(1H1,12A6)
101   FORMAT()
102   FORMAT(' NUMBER OF NODE POINTS      = ',I4,/,
     +       ' NUMBER OF ELEMENTS         = ',I4,/,
     +       ' NUMBER OF BOUNDARY NODES   = ',I4,/,
     +       ' NUMBER OF LOADED NODES     = ',I4)
103   FORMAT(' NODE COORDINATES:  NODE    X     Y      ')
104   FORMAT(15X,I4,F10.4,F10.4)
105   FORMAT(' ELEMENT NODES:  ELEMENT  NODE I  NODE J  NODE K')
106   FORMAT(20X,I3,5X,I3,5X,I3,5X,I3)
107   FORMAT(' BOUNDARY NODES:  NODE  RESTRAINT X Y (1=RES)')
108   FORMAT(18X,I3,13X,I1,I2)
109   FORMAT(' LOADING     NODE       X       Y')
120   FORMAT(12X,I3,F8.3,F8.3)
121   FORMAT(' MATERIAL PROPERTIES:',/,' E =',E10.6,
     +' MU =',E10.5,' SIGXX = ',E10.5,' SIGYY = ',E10.5,
     +' SIGXY =',E10.5)
      RETURN
      END
```

DOUGLAS(1).ASET2

```
      SUBROUTINE ASET2
      COMMON/BLOCK1/A(150,150)
     +       /BLOCK2/RF(150),CF(150)
     +       /BLOCK3/NRA1,NRA2,NRA3,NCA1,NCA2,NEL,NNP
     +       /BLOCK4/AREA(50)
     +       /BLOCK7/NPI(50),NPJ(50),NPK(50),X(50),Y(50)
      INTEGER RF,CF
C****************************************************************
C*          VARIABLE LIST
C*          =============                                        *
C*                                                               *
C* NNP   = NO OF NODE POINTS                                     *
C* NEL   = NO OF ELEMENTS
C* NBN   = NO OF BOUNDARY NODES
C* NLN   = NO OF LOADED NODES
C****************************************************************
      WRITE(5,200)
      DO 20 I=1,NEL
      NP1 = NPI(I)
      NP2 = NPJ(I)
      NP3 = NPK(I)
      A1 = X(NP3) - X(NP2)
      A2 = X(NP1) - X(NP3)
      A3 = X(NP2) - X(NP1)
      B1 = Y(NP2) - Y(NP3)
      B2 = Y(NP3) - Y(NP1)
      B3 = Y(NP1) - Y(NP2)
C
      T1 = X(NP1)*B1 + X(NP2)*B2 + X(NP3)*B3
      AREA(I) = (A3*B2 - A2*B3)/2.0
      T1 = AREA(I)/T1
      A1 = A1 * T1
      B1 = B1 * T1
      A2 = A2 * T1
      B2 = B2 * T1
      A3 = A3 * T1
      B3 = B3 * T1
C
      WRITE(5,201)I,NP1,NP2,NP3,AREA(I)
C
      A(2*NP1-1,3*I-2) = B1
      A(2*NP2-1,3*I-2) = B2
      A(2*NP3-1,3*I-2) = B3
      A(2*NP1  ,3*I-1) = A1
      A(2*NP2  ,3*I-1) = A2
      A(2*NP3  ,3*I-1) = A3
      A(2*NP1-1,3*I)   = A1
      A(2*NP1  ,3*I)   = B1
      A(2*NP2-1,3*I)   = A2
      A(2*NP2  ,3*I)   = B2
      A(2*NP3-1,3*I)   = A3
      A(2*NP3  ,3*I)   = B3
 20   CONTINUE
C*************************************
C* FORMAT STATEMENTS FOLLOW.
C*************************************
 200  FORMAT(///,' ELEMENT' ,6X,'NODES',10X,'AREA',)
 201  FORMAT(4I6,F12.6,6E8.2)
      RETURN
      END
```

```
DOUGLAS(1),ROWEN2
      C*************************************************************************
      C*   THIS SUBROUTINE REDUCES THE STATICS MATRIX
      C*   (A) TO ITS ROW-ECHELON FORM SO THAT THE
      C*   INDEPENDANT SELF-STRESS SYSTEMS MAY BE IDENTIFIED
      C*************************************************************************
            SUBROUTINE ROWEN2
            COMMON/BLOCK1/A(150,150)
           +      /BLOCK2/RF(150),CF(150)
           +      /BLOCK3/NRA1,NRA2,NRA3,NCA1,NCA2,NEL,NNP
            INTEGER RF,CF
      C*********************************************************
      C*           VARIABLE LIST                              *
      C*           =============                              *
      C*                                                      *
      C* NNP    = NO OF NODE POINTS                           *
      C* NEL    = NO OF ELEMENTS
      C* NBN    = NO OF BOUNDARY NODES
      C* NLN    = NO OF LOADED NODES
      C* RF(I) = ROW FLAG OF ROW I (RF(I)=1 IF ROW I DELETED)
      C* CF(I) = COL FLAG OF COL I (CF(I)=1 IF COL I LIN DEP)
      C* NRA1 = NO OF ROWS IN A (COUNTING DELETED BDRY ROWS)
      C* NRA2 = NO OF ROWS IN A (BOUNDARY ROWS DELETED)
      C* NRA3 = TOTAL NO OF ROWS IN THE AUGMENTED STATICS MATRIX
      C* NCA1 = NO OF COLS IN A
      C* NCA2 = NO OF COLS IN A + NO OF LOAD VECTORS
      C* I, J, K, L ARE COUNTERS
      C*********************************************************
      C
            DO 10 K=1,NCA1
            CF(K) = 1
      10    CONTINUE
            I = 1
            J = 1
      C*************************************************************************
      C*    LOOK FOR A NON-ZERO ELEMENT IN COLUMN
      C*   J BENEATH ROW I.
      C*************************************************************************
      20    IF(RF(I) .NE. 1)GO TO 25
            I = I+1
            IF (I .GE. NRA1)GO TO 100
            GO TO 20
      25    DO 30 K = I,NRA1
            IF(RF(K) .EQ. 1)GO TO 30
            IF(ABS(A(K,J)) .GT. 1.0E-7)GO TO 40
      30    CONTINUE
            IF(J .GE. NCA1)GO TO 100
      C*************************************************************************
      C*   IF ALL THE ELEMENTS IN COLUMN J ARE ZERO BELOW
      C*   ROW J THEN THIS COLUMN IS LINEARLY DEPENDANT
      C*   ON THE PREVIOUS COLUMNS. FLAG THIS COLUMN.
      C*************************************************************************
            CF(J) = 1
            J = J+1
            GO TO 20
      C*************************************************************************
      C*    IF THERE IS ONE NON-ZERO ELEMENT , LOOK FOR
      C*    THE LARGEST ELEMENT IN COLUMN J.
      C*************************************************************************
      40    XMAX = 0.0
```

```
      DO 42 K=I,NRA1
      T = ABS(A(K,J))
      IF(RF(K) .EQ. 1)GO TO 42
      IF( XMAX .GE. T)GO TO 42
      L = K
      XMAX = T
42    CONTINUE
C*********************************************************************
C*     IF THE LARGEST VALUE IN COLUMN J IS NOT IN
C*     ROW I, THEN SWOP ROW I WITH THE ROW IN WHICH
C*     LARGEST ELEMENT WAS FOUND.
C*********************************************************************
      IF(L .EQ. I)GO TO 60
      DO 50 K=I,NCA2
      T = A(L,K)
      A(L,K) = A(I,K)
      A(I,K) = T
50    CONTINUE
C*********************************************************************
C*    GET 1.0 IN POSITION I,J BY DIVIDING
C*    ROW I BY THE VALUE IN A(I,J).
C*********************************************************************
60    T = A(I,J)
      DO 65 K=1,NCA2
      A(I,K) = A(I,K)/T
65    CONTINUE
C*********************************************************************
C*    OPERATE ON ALL THE ROWS (EXCEPT ROW I) SO THAT
C*    THEY HAVE ZEROES IN COLUMN J.
C*********************************************************************
      DO 72 K=1,NRA1
      IF(RF(K) .EQ. 1)GO TO 72
      IF( K .EQ. I)GO TO 72
      T  = A(K,J)
      DO 70 L=1,NCA2
      A(K,L) = A(K,L) - T*A(I,L)
      IF(ABS(A(K,L)) .LT. 1.0E-8) A(K,L) = 0.0
70    CONTINUE
72    CONTINUE
C*********************************************************************
C*    IF THE PROCESS IS COMPLETE, EXIT.
C*********************************************************************
      CF(J) = 0
      IF(I .GE. NRA1 .OR. J .GE. NCA1)GO TO 100
      I = I+1
      J = J+1
      GO TO 20
100   RETURN
      E N D
```

```
..GLAS(1).COMPT2
      SUBROUTINE COMPT2
      COMMON/BLOCK1/A(150,150)
     +       /BLOCK2/RF(150),CF(150)
     +       /BLOCK3/NRA1,NRA2,NRA3,NCA1,NCA2,NEL,NNP
     +       /BLOCK4/AREA(50)
     +       /BLOCK6/E,NU,SIGXX,SIGYY,SIGXY
      INTEGER RF,CF
      REAL NU
C**********************************************************
C*          VARIABLE LIST
C*          =============                               *
C*                                                      *
C* NNP   = NO OF NODE POINTS                            *
C* NEL   = NO OF ELEMENTS
C* NBN   = NO OF BOUNDARY NODES
C* NLN   = NO OF LOADED NODES
C* RF(I) = ROW FLAG OF ROW I (RF(I)=1 IF ROW I DELETED)
C* CF(I) = COL FLAG OF COL I (CF(I)=1 IF COL I LIN DEP)
C* NRA1 = NO OF ROWS IN A (COUNTING DELETED BDRY ROWS)
C* NRA2 = NO OF ROWS IN A (BOUNDARY ROWS DELETED)
C* NRA3 = TOTAL NO OF ROWS IN THE AUGMENTED STATICS MATRIX
C* NCA1 = NO OF COLS IN A
C* NCA2 = NO OF COLS IN A + NO OF LOAD VECTORS
C* I, J, K, L ARE COUNTERS
C**********************************************************
C*****************************************************************
C*     SET UP THE RESIDUAL FORCE SYSTEMS FROM THE LINEARLY       *
C*     DEPENDANT COLUMNS IN (A).                                  *
C*****************************************************************
      KC = 0
      DO 20 I=1,NRA1
      IF(RF(I) .EQ. 1) GO TO 20
      KR = NRA1
      KC = KC+1
10    IF(CF(KC) .EQ.0)GO TO 12
      KC = KC+1
      GO TO 10
12    DO 15 J=1,NCA1
      IF(CF(J) .NE. 1)GO TO 15
      KR = KR+1
      A(KR,KC) = A(I,J)
15    CONTINUE
20    CONTINUE
      KR = NRA1
      DO 40 J=1,NCA1
      IF(CF(J) .NE. 1)GO TO 40
      KR = KR+1
      A(KR,J) = -1.0
40    CONTINUE
C**************************************
C*      PRINT OUT THE STATICS MATRIX.
C**************************************
      READ(9,111) IA
111   FORMAT()
      IF( IA .EQ. 0) GO TO 71
      K1 = NCA1/16 + 1
      DO 70 K=1,K1
      WRITE(5,200) K
      J1 = (K-1)*16 + 1
```

```
      J2 = J1 + 15
      IF(J2 .GT. NCA1) J2 = NCA1
      DO 70 I=1,NRA3
      T1 = ' '
      IF(RF(I) .NE. 1) GO TO 65
      T1 = '*'
65    WRITE(5,210)T1,I,(A(I,J),J=J1,J2)
70    CONTINUE
71    CONTINUE
C***********************************************************************
C*   STORE THE SHAKEDOWN CONSTRAINT EQUATIONS
C***********************************************************************
      K1 = NRA1 + 1
      CALL ERTRAN(6,'QASG,T 12. . ')
      DO 50 I=K1,NRA3
      DO 50 J=1,NCA1
      WRITE(12)A(I,J)
50    CONTINUE
      END FILE 12
      REWIND 12
C***********************************************************************
C*      MULTIPLY THE SELF STRESS SYSTEMS BY THE FLEXIBILITY
C*          MATRIX (F).
C***********************************************************************
      DO 80 I = K1,NRA3
      K = 0
      DO 80 J=1,NCA1,3
      K = K + 1
      T1 = E/AREA(K)
      A1 = A(I,J)
      A2 = A(I,J+1)
      A(I,J)    = (A1-NU*A2)/T1
      A(I,J+1) = (A2-NU*A1)/T1
      A(I,J+2) =  A(I,J+2)*(1.0+NU)*2.0/T1
80    CONTINUE
C
      DO 90 I=K1,NRA3
      T1 = 0.0
      DO 85 J=1,NCA1
      T1 = AMAX1(T1,ABS(A(I,J)))
85    CONTINUE
      T1 = 0.5/T1
      DO 90 J=1,NCA1
      A(I,J) = A(I,J)*T1
90    CONTINUE
C
      IF(IA .EQ. 0) GO TO 101
      K1 = NCA1/16 + 1
      DO 100 K=1,K1
      WRITE(5,205) K
      J1 = (K-1)*16 + 1
      J2 = J1 + 15
      IF(J2 .GT. NCA1) J2 = NCA1
      DO 100 I=1,NRA3
      T1 = ' '
      IF(RF(I) .NE. 1) GO TO 95
      T1 = '*'
95    WRITE(5,210)T1,I,(A(I,J),J=J1,J2)
100    CONTINUE
101   CONTINUE
```

```
C*************************************
C*  FORMAT STATEMENTS FOLLOW.
C*************************************
200   FORMAT(1H1,' A MATRIX AUGMENTED (BEFORE',
     +' MULTIPLYING THE SELF STRESS SYSTEMS',
     +' BY THE FLEXABILITY MATRIX) PAGE',I3)
205   FORMAT(1H1,' A MATRIX (SELF STRESS INCL.) PAGE:',I3)
210   FORMAT(1H ,A1,I2,16E8.2)
      RETURN
      E N D
```

```
DGLAS(1).SOLVE2
      SUBROUTINE SOLVE2
C*************************************************************************
C*        THIS SUBROUTINE SOLVES FOR THE ELASTIC FORCES.
C*        IT USES GAUSS-JORDAN REDUCTION WITH PARTIAL PIVOTING.
C*************************************************************************
      COMMON/BLOCK1/A(150,150)
     +        /BLOCK2/RF(150),CF(150)
     +        /BLOCK3/NRA1,NRA2,NRA3,NCA1,NCA2,NEL,NNP
      INTEGER RF,CF
C********************************************************
C*          VARIABLE LIST                                *
C*          =============                                *
C*                                                       *
C* NNP   = NO OF NODE POINTS                             *
C* NEL   = NO OF ELEMENTS
C* NBN   = NO OF BOUNDARY NODES
C* NLN   = NO OF LOADED NODES
C* RF(I) = ROW FLAG OF ROW I (RF(I)=1 IF ROW I DELETED)
C* CF(I) = COL FLAG OF COL I (CF(I)=1 IF COL I LIN DEP)
C* NRA1 = NO OF ROWS IN A (COUNTING DELETED BDRY ROWS)
C* NRA2 = NO OF ROWS IN A (BOUNDARY ROWS DELETED)
C* NRA3 = TOTAL NO OF ROWS IN THE AUGMENTED STATICS MATRIX
C* NCA1 = NO OF COLS IN A
C* NCA2 = NO OF COLS IN A + NO OF LOAD VECTORS
C* I, J, K, L ARE COUNTERS
C********************************************************
      I = 1
      J = 1
20    IF(RF(I) .NE. 1) GO TO 25
      I = I+1
      GO TO 20
25    DO 30 K = I,NRA3
      IF(RF(K) .EQ. 1) GO TO 30
      IF(ABS(A(K,J)) .GT. 1.0E-7)GO TO 40
30    CONTINUE
C*************************************************************************
C*    IF ALL THE ELEMENTS IN COLUMN I ARE ZERO BELOW I
C*    THEN THE I TH COLUMN IS LINEARLY DEPENDANT ON THE
C*    PREVIOUS COLUMNS, HENCE WE HAVE A MECHANISM.
C*************************************************************************
      N = I/2
      M = J/3
      WRITE(5,600)N,M
      STOP  SOLVE2
40    XMAX = 0.0
      DO 42 K=I,NRA3
      T = ABS(A(K,J))
      IF(RF(K) .EQ. 1) GO TO 42
      IF( XMAX  .GE.  T)GO TO 42
      L = K
      XMAX = T
42    CONTINUE
      IF(L .EQ. I)GO TO 60
      DO 50 K=1,NCA2
      T = A(L,K)
      A(L,K) = A(I,K)
      A(I,K) = T
50    CONTINUE
60    T = A(I,J)
```

```
      DO 65 K=1,NCA2
      A(I,K) = A(I,K)/T
65    CONTINUE
      DO 72 K=1,NRA3
      IF(RF(K) .EQ. 1) GO TO 72
      IF(   K   .EQ. I) GO TO 72
      T   = A(K,J)
      IF(T .EQ. 0.0) GO TO 72
      DO 70 L=1,NCA2
      A(K,L) = A(K,L) - T*A(I,L)
      IF(ABS(A(K,L)) .LT. 1.0E-8) A(K,L) = 0.0
70    CONTINUE
72    CONTINUE
C***********************************************************************
C*     IF MATRIX REDUCTION IS COMPLETE, EXIT.
C***********************************************************************
      IF(I .GE. NRA3 .OR. J .GE. NCA1)GO TO 100
      I = I+1
      J = J+1
      GO TO 20
600   FORMAT(1H1,//,1H ,' THE AUGMENTED STATICS MATRIX IS',
     +' SINGULAR.',//,'  STRUCTURE IS A MECHANISM.',//
     +'   SOLUTION TERMINATED.',//,
     +'   SINGULARITY OCCURS NEAR THE NODE ',I6,/,
     +'                         AND MEMBER  ',I6)
100   RETURN
      E N D
```

```
)GLAS(1).ELSLN2
          SUBROUTINE ELSLN2(TIME1)
          COMMON/BLOCK1/A(150,150)
         +      /BLOCK2/RF(150),CF(150)
         +      /BLOCK3/NRA1,NRA2,NRA3,NCA1,NCA2,NEL,NNP
         +      /BLOCK4/AREA(50)
         +      /BLOCK6/E,NU,SIGXX,SIGYY,SIGXY
         +      /BLOCK7/NPI(50),NPJ(50),NPK(50),X(50),Y(50)
         +      /BLOCK8/ES(150)
          INTEGER RF,CF
          REAL NU
          J = 0
          DO 10 I=1,NRA3
          IF(RF(I) .EQ. 1) GO TO 10
          J = J + 1
          ES(J) = A(I,NCA2)
   10     CONTINUE
          WRITE(5,200)
          DO 20 I=1,NEL
          WRITE(5,210)I,NPI(I),NPJ(I),NPK(I),AREA(I),
         +              ES(3*I-2),ES(3*I-1),ES(3*I)
   20     CONTINUE
   C
          CALL TIME (TIME2)
          TIME2 = TIME2 - TIME1
          WRITE(5,220) TIME2
   200    FORMAT(1H1,' ELASTIC SOLUTION.'///,T10
         +,'ELEMENT',T25,'NODES',T37,'AREA',T49,'SIG XX'
         +,T66,'SIG YY',T82,'SIG XY'/)
   210    FORMAT(T10,I5,T20,3I4,T36,F5.2,T47,F10.4,T64
         +,F10.4,T80,F10.4)
   220    FORMAT(////,11X,' TIME FOR ELASTIC SOLUTION ',
         + F10.4,' SECONDS.')
          RETURN
          E N D
```

```
)ELAS(I).LPSET3
      SUBROUTINE LPSET3(TIME1)
      DIMENSION TOL(4),IFIX(20),JX(160),IOUT(4),X(162),Y(162),S(250)
     +,RHS(60),COST(250),T(160),EC(60,60),PI(160),ERR(4)
      COMMON/BLOCK1/A(60,160)
     +       /BLOCK2/RF(150),CF(150)
     +       /BLOCK3/NRA1,NRA2,NRA3,NCA1,NCA2,NEL,NNP
     +       /BLOCK4/AREA(50)
     +       /BLOCK6/E,NU,YIELD(3)
     +       /BLOCK7/NPI(50),NPJ(50),NPK(50),XJ(50),YJ(50)
     +       /BLOCK8/ES(150)
      INTEGER RF,CF
      REAL NU
C**************************************************************
C*            VARIABLE LIST
C*            =============                                  *
C*                                                           *
C* NNP   = NO OF NODE POINTS                                 *
C* NEL   = NO OF ELEMENTS
C* NBN   = NO OF BOUNDARY NODES
C* NLN   = NO OF LOADED NODES
C* RF(I) = ROW FLAG OF ROW I (RF(I)≠1 IF ROW I DELETED)
C* CF(I) = COL FLAG OF COL I (CF(I)≠1 IF COL I LIN DEP)
C* NRA1 = NO OF ROWS IN A (COUNTING DELETED BDRY ROWS)
C* NRA2 = NO OF ROWS IN A (BOUNDARY ROWS DELETED)
C* NRA3 = TOTAL NO OF ROWS IN THE AUGMENTED STATICS MATRIX
C* NCA1 = NO OF COLS IN A
C* NCA2 = NO OF COLS IN A + NO OF LOAD VECTORS
C* I, J, K, L ARE COUNTERS
C**************************************************************
C
      READ(5,100) IA
100   FORMAT()
      DO 10 I=1,NCA1
      J = 2*I
      L = (I-1)/3 + 1
      A(1,J-1) =  ES(I) * AREA(L)
      A(1,J  ) = -ES(I) * AREA(L)
10    CONTINUE
      NCON = NCA1 - NRA2 + 1
      NVAR = 2*NCA1
      DO 20 I=2,NCON
      RHS(I) = 0.0
      T(I)   = ' '
20    CONTINUE
      RHS(1) = 1.0
      T(1)   = ' '
C
      DO 30 I=2,NCON
      DO 30 J=1,NCA1
      READ(12) T1
      K = 2*J
      L = (J-1)/3 + 1
      A(I,K-1) = -T1 * AREA(L)
      A(I,K  ) = -T1 * AREA(L)
30    CONTINUE
C
      DO 40 I=1,NCA1
      J = 2*I
      K = MOD(I,3)
```

```
          L = (I-1)/3 + 1
          IF(K.EQ.0)K = 3
          COST(J-1) = YIELD(K) * AREA(L)
          COST(J  ) = YIELD(K) * AREA(L)
   40     CONTINUE
C
          IFIX( 1) = 60
          IFIX( 2) = 160
          IFIX( 3) = NCON
          IFIX( 4) = NVAR
          IFIX( 5) = 0
          IFIX( 6) = 5000
          IFIX( 7) = 200
          IFIX( 8) = 4000
          IFIX( 9) = 1
          IFIX(10) = 0
          IFIX(11) = 1
          IFIX(12) = 1
          IFIX(13) = 0
          IFIX(14) = 1
          IFIX(15) = 60
          IFIX(16) = NVAR
          IFIX(17) = 1
          IFIX(18) = ' L.A. '
          IFIX(19) = 0
          IFIX(20) = 0
C
          TOL(1) = 0.0
          TOL(2) = 0.0
          TOL(3) = 0.0
          TOL(4) = 0.0
C
          IF(IA.EQ.0)GO TO 50
C
          K1 = NVAR
          DO 650 I=1,NCON
  650     WRITE(5,651)I,(A(I,J),J=1,K1),RHS(I)
  651     FORMAT(1H ,I2,F8.3,24F5.1,/,1H ,10X,24F5.1)
          WRITE(5,220) IFIX(3),IFIX(4)
  220     FORMAT(///,'  NUMBER OF CONSTRAINTS = ',I6,
     +           /,'  NUMBER OF VARIABLES   = ',I6)
C
   50     CALL SIMPLX(T,RHS,COST,IFIX,TOL,OBJ,X,JX,PI,EE,ERR,IOUT,Y,S)
C
          RETURN
          END
```

Example of data input for limit analysis of plane stress problems

Example 7.1

```
ASD*DOUGLAS(1).TD1
TEST DATA      SET 1
7,5,2,2
1,1,1
6,1,1
0.0,0.0
0.0,1.0
2.0,1.0
3.0,0.0
4.0,1.0
6.0,0.0
6.0,1.0
1,3,2
1,4,3
3,4,5
4,6,5
5,6,7
3,0.0,-10.0
5,0.0,-10.0
200000.0,0.3,10.0,10.0,10.0
0
1
```

Explanation of Data Input

line 1 : Title of example

line 2 : Structural parameters kjm km, kr, $k\ell$

kj = number of node points

km = number of elements

kr = number of restrained nodes

$k\ell$ = number of loaded nodes

lines 3 to 4 : Number of the restrained node and the restraint for the X and Y directions (1 = restrained, 0 = unconstrained)

lines 5 to 11 : Node co-ordinates X, Y

lines 12 to 16 : Element apices (anticlockwise order)

lines 17 to 18 : Loaded node number and the X and Y components of load on the node

line 19 : E (Youngs Modulus), ν(Poisson's Ratio) and the yield stresses σ_{oxx}, σ_{oyy}, σ_{oxy}

line 20 : Statics matrix output control (1 = print [A], 0 = suppress printout)

line 21 : Linear Programming constraint output control (1 = print constraints, 0 = suppress printout)

```
IT DATA      SET 1
BER OF NODE POINTS    =     7
BER OF ELEMENTS       =     5
BER OF BOUNDARY NODES =     2
BER OF LOADED NODES   =     2
NDARY NODES:  NODE  RESTRAINT X Y (1=RES)
                1               1 1
                6               1 1
E COORDINATES:  NODE    X       Y
                1     .0000    .0000
                2     .0000   1.0000
                3    2.0000   1.0000
                4    3.0000    .0000
                5    4.0000   1.0000
                6    6.0000    .0000
                7    6.0000   1.0000
MENT NODES:  ELEMENT  NODE I   NODE J   NODE K
                1        1        3        2
                2        1        4        3
                3        3        4        5
                4        4        6        5
                5        5        6        7
DING     NODE        X       Y
           3     .000 -10.000
           5     .000 -10.000
ERIAL PROPERTIES:
.200000+06 MU = .30000+00 SIGXX =  .10000+02 SIGYY =  .10000+02 SIGXY = .100C
1 = 14: NRA2 = 10: NRA3 = 19: NCA1 = 15: NCA2 = 16


MENT       NODES             AREA
 1      1     3     2     1.000000
 2      1     4     3     1.500000
 3      3     4     5     1.000000
 4      4     6     5     1.500000
 5      5     6     7     1.000000
```

```
ùT DATA      SET 1
BER OF NODE POINTS    =     7
BER OF ELEMENTS       =     5
BER OF BOUNDARY NODES =     2
BER OF LOADED NODES   =     2
DARY NODES:  NODE  RESTRAINT X Y (1=RES)
              1                1 1
              6                1 1
E COORDINATES:  NODE    X       Y
              1      .0000     .0000
              2      .0000    1.0000
              3     2.0000    1.0000
              4     3.0000     .0000
              5     4.0000    1.0000
              6     6.0000     .0000
              7     6.0000    1.0000
MENT NODES:  ELEMENT  NODE I   NODE J    NODE K
                 1       1        3         2
                 2       1        4         3
                 3       3        4         5
                 4       4        6         5
                 5       5        6         7
DING     NODE        X        Y
           3      .000 -10.000
           5      .000 -10.000
ERIAL PROPERTIES:
.200000+06 MU = .30000+00 SIGXX =  .10000+02 SIGYY =  .10000+02 SIGXY = .1000
1 = 14: NRA2 = 10: NRA3 = 19: NCA1 = 15: NCA2 = 16
```

MENT	NODES			AREA
1	1	3	2	1.000000
2	1	4	3	1.500000
3	3	4	5	1.000000
4	4	6	5	1.500000
5	5	6	7	1.000000