

FINITE ELEMENT ANALYSIS OF
AXISYMMETRIC CRACKED SOLIDS

8701 MUL 35

by

IHSAN ALI ALSHARQI, B.Sc., M.Sc.

THE UNIVERSITY OF ASTON
IN BIRMINGHAM

A thesis submitted for the degree
of Doctor of Philosophy in the
University of Aston in Birmingham

JULY 1977

26 JUN 1978

216567

(DC) 620.1126 ALS

FINITE ELEMENT ANALYSIS OF
AXISYMMETRIC CRACKED SOLIDS

by

IHSAN ALI ALSHARQI

A thesis submitted for the degree
of Doctor of Philosophy in the
University of Aston in Birmingham

1977

SUMMARY

This investigation was concerned with the development of a finite element facility for the determination of fracture mechanics and general stress analysis data in axisymmetric solids.

The work was carried out in two stages. In the first stage a finite element program for solving axisymmetric problems subjected to axisymmetric loading was developed. The element used was a six node isoparametric triangular ring element. An automatic mesh generation scheme was developed which reduced the effort of preparing the required input data. A complete program incorporating this scheme is described and examples of its application are presented which show very good agreement with theoretical and other finite element solutions.

In the second stage the general finite element program was augmented to provide a fracture mechanics facility. In order to cater for singularities in the crack tip region, a singular core, over which an analytic solution is used, was embedded into the finite element mesh surrounding the crack tip. The modifications to the standard finite element program are described and several mode I and mixed mode I and II examples were solved to check the influence of mesh design and core parameters. The results obtained demonstrate the usefulness and accuracy of the technique and compare very well with solutions available in the published literature.

A complete listing of all the computer programs is presented together with their input data instructions.

CRACKS AXISYMMETRIC SOLIDS FINITE ELEMENTS

ACKNOWLEDGEMENTS

The author wishes to express his sincere gratitude to his supervisor Mr. T. H. Richards, for his continuous help and encouragement.

The author is grateful to Professor K. Foster, the Head of the Department of Mechanical Engineering, and to Mrs. P. Taber for the preparation of the drawings.

The author wishes to acknowledge with deep appreciation the help and inspiration of his wife Rawa throughout the course of this work.

Finally the author is grateful to the Ministry of Oil of the Republic of Iraq for providing him with the scholarship.

LIST OF SYMBOLS

τ_{ij}	Stress tensor
σ	Direct stress
τ_{mean}	Spherical component
ϵ_{ij}	Strain tensor
E, μ, ν	Elasticity constants
x_i	Cartesian axes
r, θ, z	Polar coordinates
χ	Airy stress function
$[k]_c$	Element stiffness matrix
$\{q\}_e$	Element displacement vector
$\{Q\}_e$	Element force vector
$[K]$	Overall stiffness matrix
$\{q\}$	Overall displacement vector
$\{Q\}$	Overall force vector
ζ, η and L_1, L_2, L_3	Natural coordinates
u, w	Displacement components
$[N]$	Shape functions matrix
$[J]$	Jacobian matrix
$/J/$	Jacobian = det.[J]
κ	= $(3-4\nu)$ for plane strain = $(3-\nu)/(1+\nu)$ for plane stress
$[B]$	Strain displacement matrix
$[C]$	Elasticity matrix

Π	Total potential energy
U	Strain energy
Ω	Potential energy of applied loads
U_c	Core strain energy
R_c	Core radius
W_i	Weight coefficients
L_{1i}, L_{2i}, L_{3i}	Integrating points
S	Strain energy density factor
R_p	Radius of crack tip plastic zone
G	Strain energy release rate
C	Compliance
Γ	Core/finite element interface
\bar{T}	Traction vector
J	Rice's path independent integral
$\psi(z), \phi(z)$	Complex functions
N_1	Number of nodes on core/finite element interface
K_I, K_{II}, K_{III}	Stress intensity factors
$[I]$	Identity matrix
$[L], [L]^t$	lower and upper triangular matrices

CONTENTS

	Page
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: ESSENTIALS OF ELASTICITY THEORY	5
2.1 Introduction	5
2.2 The stress tensor	5
2.3 The strain tensor	7
2.4 Hooke's law	8
2.5 The two-dimensional problem	9
2.6 Axisymmetric stress distribution	11
CHAPTER 3: THE FINITE ELEMENT METHOD	14
3.1 Introduction	14
3.2 Natural coordinates and shape functions	18
3.3 The six node isoparametric triangular ring element	24
3.4 The strain-displacement relations	24
3.5 Constitutive relations	29
3.5.1 Introduction	29
3.5.2 Isotropic material	29
3.5.3 Anisotropic stratified materials	30
3.6 The variational formulation of element stiffness and loads	32
3.7 Assembly of the overall stiffnesses and loads	35
3.8 Automatic mesh generation	36
3.8.1 Introduction	36
3.8.2 Mesh generation for an axisymmetric structure	37
3.8.3 Mesh generation around a core	41

CHAPTER 4:	THEORIES OF BRITTLE FRACTURE	45
	4.1 Introduction	45
	4.2 The Griffith theory	46
	4.3 Irwin's theory	49
	4.4 The strain energy density theory	52
	4.5 The crack tip plastic zone	55
	4.5.1 Introduction	55
	4.5.2 Extension of the Griffith concept	57
	4.5.3 Extension of Irwin's concept	58
CHAPTER 5:	METHODS OF DETERMINING STRESS INTENSITY FACTORS	61
	5.1 Introduction	61
	5.2 Experimental methods	61
	5.2.1 Photoelasticity	61
	5.2.2 Compliance	63
	5.2.3 Crack tip opening displacement measurement	63
	5.3 Analytical methods	64
	5.3.1 Westergaard stress function	66
	5.3.2 Complex stress function	66
	5.4 Numerical method	67
	5.4.1 Boundary collocation	69
	5.4.2 Stress concentrations	71
	5.4.3 Green's function	73
	5.4.4 Integral transforms and dislocation models	76
	5.4.5 Force-displacement matching	79
	5.4.6 Alternating method	79
	5.4.7 Asymptotic approximation	79
	5.4.8 The finite element method	80

CHAPTER 6:	MODIFICATION TO THE FINITE ELEMENT FORMULATION	101
	TO INCORPORATE A SINGULAR CORE	
6.1	Introduction	101
6.2	Modification to the finite element formulation to include the Hilton and Hutchinson type core element.	101
CHAPTER 7:	COMPUTER PROGRAM PROCEDURES	108
7.1	Introduction	108
7.2	Procedure (STRDIS)	111
7.3	Application of boundary conditions	116
	7.3.1 Procedure LOADING	116
	7.3.2 Procedure BOUNCONST	122
7.4	Procedure CONSTREL	125
7.5	Evaluation and assembly of the overall stiffness matrix	128
	7.5.1 Introduction	128
	7.5.2 Procedure ADDARRAY	128
	7.5.3 Procedure ASSEMBLY	131
7.6	Procedure SYMVBSOL	137
7.7	Procedure NODSTR	141
7.8	Procedure ELESTR	144
7.9	Procedure FEINPUT	147
7.10	Procedure CCRINPUT	156
7.11	Procedure MMINPUT	162
7.12	Procedure SBINPUT	172
7.13	Procedure CCRM1	182
7.14	Procedure CCRMM12	196
7.15	Procedure KARBDMMST	208
7.16	Procedure KARM1BND	210
7.17	Procedure RESIDUAL	212
7.18	Procedure E04AAA	216

CHAPTER 8:	NUMERICAL EXAMPLES	218
8.1	Introduction	218
8.2	Applications of the general axisymmetric program	219
8.2.1	Thick cylinder under internal pressure	219
8.2.2	Circular plate under uniform pressure	223
8.3	Mode I fracture problems	234
8.3.1	A round bar with a circumferential normal edge crack	234
8.3.2	A round bar with a normal penny shaped crack	241
8.3.3	A round bar with a crack extending from the base of an external groove	243
8.3.4	A round bar with an internal void and a circumferential normal crack	249
8.4	Mixed mode I and II fracture problems	251
8.4.1	A test problem	251
8.4.2	A round bar with a conical shaped circumferential crack	257
8.4.3	A shouldered bar with a circumferential crack emanating from its fillet	262
8.5	The effect of inclusions on the values of stress intensity factors	268
CHAPTER 9:	DISCUSSIONS AND CONCLUSIONS	274
9.1	Discussions	274
9.2	Possible topics for further research	279
9.2.1	Further developments to the computer programs	279
9.2.2	New areas for future research	280
9.3	Conclusions	285

CHAPTER 10: APPENDICES	288
10.1 Numerical integration formulae for triangles	288
10.2 Near crack tip equations for stresses, displacement and strains	289
10.3 Strain energy of the core region	296
10.3.1 Mode I formulation	296
10.3.2 Mixed mode I and II formulation	297
10.4 Programs Listings and sample problems	299
10.4.1 The general axisymmetric program	299
10.4.2 Mode I fracture program	328
10.4.3 Mixed mode I and II program	347
10.4.4 Mixed mode shouldered bar program	370
CHAPTER 11: REFERENCES	392

CHAPTER 1

INTRODUCTION

1. Introduction

A fundamental requirement of an engineering structure is to sustain the loadings applied to it during its service life without failure. However a small percentage of engineering components, although satisfying conventional strength analysis conditions, do not fulfil this requirement. There are several modes of mechanical failure, [81], among them is the mode of fast fracture which involves the unstable propagation of a crack in a structure. In this mode, once the crack starts to move, the loading system is such that it produces accelerating growth. This fracture is often referred to as "brittle", because the instability first occurred when the applied stress was less than the general yield stress in the uncracked part of the structure. The catastrophic nature of failures due to brittle fracture, examples of which are listed in, [14], imposed the need to consider the reduction in the fracture strength of metals due to the existence of crack-like defects in them. The presence of a crack introduces a singularity in the stress field and there are several approaches to measure its magnitude. Irwin's approach [19], concentrates on a crack tip region and introduces three factors which define this magnitude known as the stress intensity factors. Although a plastic zone is formed around a crack tip and hence stress singularities can not exist, it was found that the tip region is

sufficiently large so that events within a small plastic crack tip zone do not cause significant variations in the system's strain energy and assuming linearity remain a good approximation, [29].

The accurate determination of the stress intensity factors is central to the application of this approach to design work, and a brief description of the important methods available for this purpose is presented in Chapter 5. The geometries and loadings of engineering structures are such that analytic techniques dealing with idealized situations are not adequate and approximate numerical methods must be resorted to. A very powerful numerical method used extensively in structural engineering is the finite element method described in Chapter 3, which was used by many authors to calculate stress intensity factors. It was seen that its application without catering for the singularities at the crack tip required a fine mesh sub-division in the neighbourhood of the crack tip for a reasonable accuracy, and methods which modify the standard finite element formulation to allow for these singularities were more economic and accurate. One such method which requires a small modification to the finite element formulation and results in the direct evaluation of stress intensity factors is the Hilton and Hutchinson method described in section (5.4.8). *ref [73]*.

Many important engineering components are solids of revolution, and the aim of this project was to evaluate the stress intensity factors of axisymmetric cracked solids subjected to axisymmetric loadings using the finite element method incorporating a Hilton and Hutchinson type core element. To achieve this aim, computer programs were

developed to solve single and mixed mode fracture problems. In order to study particular cases of special interest where voids and grooves are present, and to be able to obtain better matching with the circular core shape, an isoparametric element was chosen to map the curved boundaries involved. Due to the existence of stress raisers and cracks, the stress field varies rapidly throughout the structure and its accurate representation by a constant strain element would require fine discretization, while a coarser mesh with a linear strain element will produce higher accuracy and be more economical. Therefore the element chosen as the basis for the present work, was an isoparametric six node linear strain triangular ring element.

A major disadvantage of the finite element technique is the large volume of input data required. Apart from the time consuming task of preparing the correct data, small errors may pass unnoticed and give wrong results. This chance was eliminated and the effort was cut drastically by developing an automatic mesh generation scheme described in section(3.8).

The computer programs developed in this project were found to be very effective in the determination of mode (I) and mixed mode (I) and (II) stress intensity factors for a variety of crack and geometry configurations. The effects of inclusions of different material properties on the values of stress intensity factors were examined, and the angle of crack initiation was evaluated by adopting Sih's strain energy density criterion, [25].

The representative examples considered illustrate that complicated problems can be tackled with a modest size computer and by relatively small modifications to a standard finite element program. However, several modifications to the developed programs can still be made, and due to lack of time interesting possible areas of research had to be left for future work as discussed in Chapter 9.

CHAPTER 2

ESSENTIALS OF ELASTICITY THEORY

2.1 Introduction

The rigorous solutions of the three dimensional problems of elasticity are few. Faced with this, the engineer resorted, in some special cases, to acceptable simplifications which reduced the problem to a one of two dimensions. As the problems became increasingly complicated, approximate numerical methods were sought to solve them. Although the analysis procedure is simplified by the introduction of these methods, the volume of data and mathematical work required limited their application. With the introduction and wide use of electronic digital computers, the limitation was removed, and these methods gained wide acceptance and recognition.

For the convenience of reference, a summary of some of the essential results of the elasticity theory is presented in this chapter.

2.2 The stress tensor

The state of stress at a point is fully described by nine stress components (τ_{ij}) , where $i = j = 1, 2, 3$.

Knowing the components of (τ_{ij}) with respect to some set of cartesian axes (X_i) , they can be found with respect to some other set (X'_i) by means of:

$$\tau'_{\alpha\beta} = l_{\alpha i} l_{\beta j} \tau_{ij} \quad (2.1)$$

or in matrix form:

$$[\tau'] = [L]^t [\tau] [L] \quad (2.2)$$

This equation represents the transformation of a tensor quantity, and it may be shown that a particular choice of ($l_{\alpha i}$'s) may be found for which the stresses are normal only i.e. principal stresses, [1]. For a description of material behaviour (e.g. plasticity, or visco-elasticity) it is sometimes useful to introduce the deviator stress defined as:

$$\hat{\tau}_{ij} = \tau_{ij} - \delta_{ij} \tau_{\text{mean}} \quad (2.3)$$

where

$$\tau_{\text{mean}} = \frac{1}{3} (\tau_1 + \tau_2 + \tau_3) \quad (2.4)$$

$$\tau_i (i = 1, 2, 3) = \text{principal stresses}$$

$$\tau_{\text{mean}} = \text{Spherical component}$$

The stress varies continuously from point to point in a continuum. It may be shown that equilibrium requires, [1]:

$$\tau_{ij,j} + F_i = 0 \quad (2.5)$$

$$i = 1, 2, 3$$

$$\text{and } \tau_{ij} = \tau_{ji} \quad (2.6)$$

$$i = 1, 2, 3$$

where:

F_i = the components of the body force/unit volume.

The situation is statically indeterminate and the solution requires consideration of deformations and constitutive equations as described later.

2.3 The strain tensor

The deformation at a point with respect to three orthogonal cartesian coordinates is completely defined by the components of the strain tensor (ϵ_{ij}).

Similar to the stress tensor, these components can be found for another set of coordinates by:

$$\epsilon'_{\alpha\beta} = l_{\alpha i} l_{\beta j} \epsilon_{ij} \quad (2.7)$$

• A chosen set of orthogonal coordinates which causes the shearing strains to become zero are called principal axes of strain; and the corresponding strains, the principal strains.

Strains vary continuously in a continuum and, if a state of infinitesimal strain is assumed, they are related to displacements by [1]:

$$\epsilon_{ij} = \frac{1}{2} (U_{i,j} + U_{j,i}) \quad (2.8)$$

These equations show that six strain components, which are functions of position, are functions of only three displacement components. Therefore the ϵ_{ij} cannot be chosen arbitrarily, but a relation between them must exist, otherwise the displacements will not be single valued and continuous.

To ensure the compatibility of strains the following relations must be satisfied, [2].

$$\epsilon_{ij,kl} + \epsilon_{kl,ij} - \epsilon_{ik,jl} - \epsilon_{jl,ik} = 0 \quad (2.9)$$

where: $i = j = k = l = 1, 2, 3$

This system consists of eighty one equations, some of which are identically satisfied, leaving only six equations which are known as the equations of compatibility.

2.4 Hooke's law

The relation between stresses and strains have been established experimentally for a number of materials and it has been found that for a practically useful range of materials, the relations are linear. This represents a generalization of Hooke's law.

In the general three dimensional case, it takes the form, [2]:

$$\epsilon_{ij} = \frac{1+\nu}{E} \tau_{ij} - \frac{\nu}{E} \tau_{kk} \delta_{ij} \quad (2.10)$$

The inverse form is often useful

$$\tau_{ij} = \lambda \epsilon_{kk} \delta_{ij} + 2\mu \epsilon_{ij} \quad (2.11)$$

where $\mu = \frac{E}{2(1+\nu)}$ = Modulus of rigidity

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$$

For certain problems, it is convenient to write the relation in terms of the deviator or spherical components:

$$\hat{\tau}_{ij} = 2\mu \hat{\epsilon}_{ij} \quad (2.12)$$

$$\tau_{kk} = 3k \epsilon_{kk} \quad (2.13)$$

where $k = \frac{E}{3(1-2\nu)} = \text{Bulk modulus.}$

2.5 The two-dimensional problem

To obtain a solution to the general elastic problem, three fundamental steps are taken:

1. Equilibrium must be satisfied.
2. The strains implied by the stresses must be compatible.
3. A relation between stresses and strains is required.

Analytic solutions are available to only a limited number of special problems with simple shapes and loadings, many of which are summarised in, [1]. Hence, the need arises to simplify problems so that a mathematical solution is possible, yet the simplified problem remains close enough to representing the true physical situation. Special geometry and loading configurations can reduce the three dimensional problem to one of two dimensions. An example of such special cases are the plane stress, plane strain, and axisymmetric problems.

The equations of equilibrium for a two dimensional problem are reduced to, [1]:

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} = 0 \quad (2.14)$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} = 0 \quad (2.15)$$

And the compatibility equations are reduced to:

$$\nabla^2(\sigma_x + \sigma_y) = 0 \quad (2.16)$$

The traction boundary conditions are given by:

$$T_i^n = \tau_{ij} n_j \quad (2.17)$$

A method for solving equations (2.15, 2.16 and 2.17) is by introducing a function $\chi(x,y)$ which, if body forces are negligible, has the following relations with the stresses:

$$\sigma_x = \frac{\partial^2 \chi}{\partial y^2} \quad (2.18)$$

$$\sigma_y = \frac{\partial^2 \chi}{\partial x^2} \quad (2.19)$$

$$\tau_{xy} = -\frac{\partial^2 \chi}{\partial x \partial y} \quad (2.20)$$

From the definition, stresses derived from χ automatically satisfy equilibrium. The true solution is that which satisfies the compatibility relation (2.20). Therefore, the function χ must satisfy the biharmonic equation:

$$\frac{\partial^4 \chi}{\partial x^4} + \frac{2\partial^4 \chi}{\partial x^2 \partial y^2} + \frac{\partial^4 \chi}{\partial y^4} = 0 \quad (2.21)$$

or $\nabla^4 \chi = 0 \quad (2.22)$

Thus the solution of the two dimensional problem reduces to finding a solution of equation (2.21) which satisfies the boundary conditions (2.17) of the problem. The function (χ) is known as the Airy Stress Function.

2.6 Axisymmetric stress distribution

Many important problems involve solids of revolution deformed symmetrically with respect to the axis of revolution. The deformation is symmetrical with respect to the z-axis (Fig.2.1), hence the stress components are independent of the angular (θ) coordinate. The derivatives with respect to (θ) vanish leaving ($\tau_r, \tau_\theta, \tau_z, \tau_{rz}$) as the only non-zero stress components, [1].

The strain-displacement relations are:

$$\epsilon_r = \frac{\partial u}{\partial r}, \quad \epsilon_\theta = \frac{u}{r}, \quad \epsilon_z = \frac{\partial w}{\partial z}, \quad \gamma_{rz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} \quad (2.23)$$

For axisymmetric situations, the constitutive relations for linear isotropic elasticity can be conveniently expressed in matrix form as:

$$\begin{Bmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_z \\ \tau_{rz} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 \\ & 1-\nu & \nu & 0 \\ & & 1-\nu & 0 \\ \text{Symm.} & & & \frac{1-2\nu}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_r \\ \epsilon_\theta \\ \epsilon_z \\ \gamma_{rz} \end{Bmatrix} \quad (2.24)$$

Several mathematical solutions of axisymmetric problems are available, [1]; but, as for the three dimensional problem, they deal mainly with simple geometries and loading conditions. Real engineering

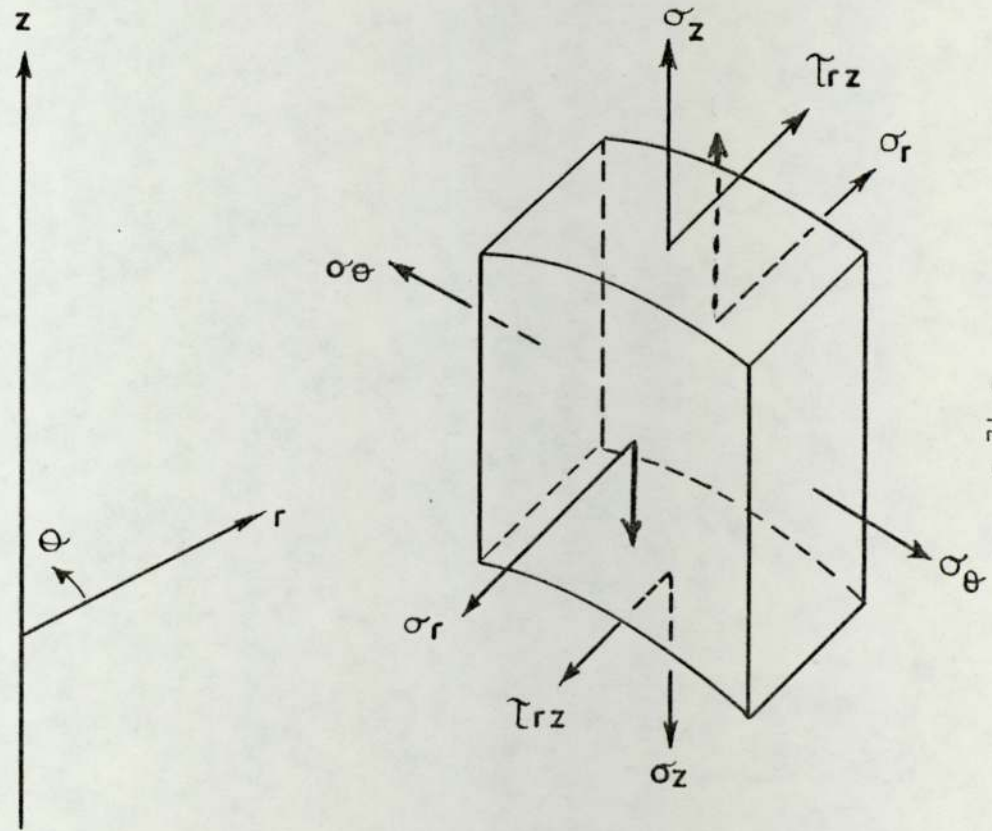
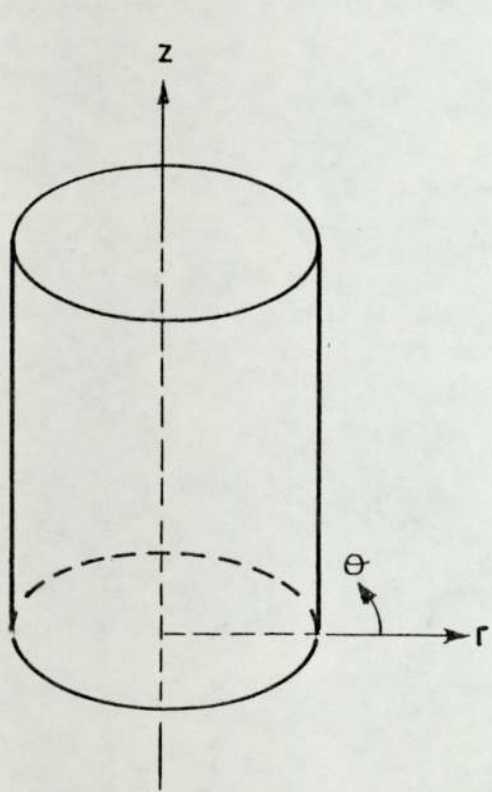


Fig. 2. 1

problems are generally complicated, therefore it became necessary to resort to numerical methods. A very powerful numerical method, which was used throughout the work described in this thesis, is the finite element method described in detail in the following chapter.

CHAPTER 3

THE FINITE ELEMENT METHOD

3.1 Introduction

In the finite element method the body is discretized into an equivalent system of smaller units, and approximate solutions for each unit are combined to obtain an approximate solution for the body.

There are three approaches to the method,[3]:

1. Displacement method: In which the displacements are the unknowns.
2. Equilibrium method: In which the stresses are the unknowns.
3. Mixed Method: In which some displacements and some stresses are the unknowns.

The work presented in this thesis follows the first method and it will be the only one discussed from hereon.

The analysis procedure may be summarized by the following steps:

1. Discretization: The process of dividing the continuum into an equivalent system of smaller subdivisions (finite element), is called discretization. There are some basic guidelines to help in performing this operation, but the final decision on type, shape and number of finite elements remains with the individual. A solid of revolution is

divided into axisymmetric ring elements, which can have various cross section shapes, (Fig. 3.1); such arrangements are of particular interest in the present work.

2. Selection of displacement function: It is difficult to select a displacement function which can represent exactly the variation of displacement in the element. Instead, a function is chosen to represent that variation only approximately. Three factors influence the choice of the displacement function:

- a) The type and degree of the function (e.g. a polynomial of a certain degree).
- b) The displacement magnitudes describing the function (e.g. nodal displacements with or without their derivatives for some or all nodes).
- c) The convergence requirements to ensure that the results approach the real solution.

These requirements are,[3]:

- i) The function must be continuous within the element, and no openings, or overlaps occur between adjacent elements, i.e. continuity of displacements is required across inter element boundaries.
- ii) The rigid body displacements of the element must be included in the function.
- iii) The constant strain states of the element must be included in the displacement function.

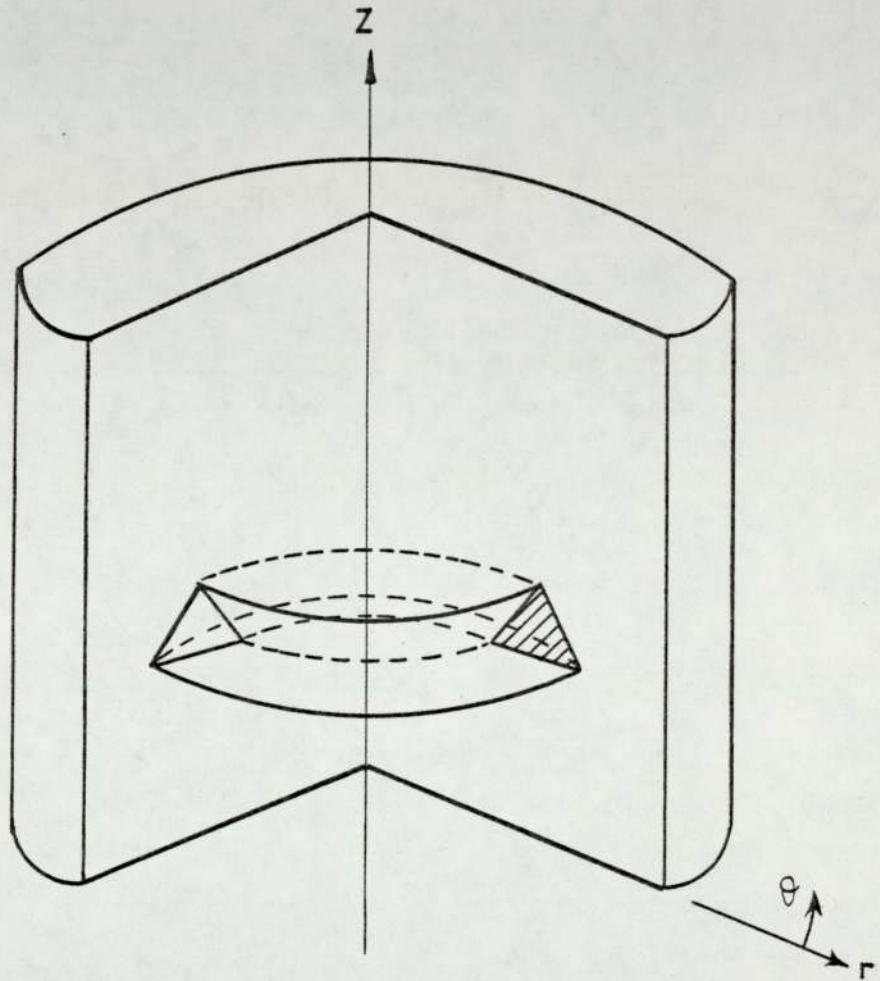


Fig. 3.1.

3. Derivation of element stiffness matrix: The stiffness $[K]_e$ relates the modal displacements $\{q\}_e$ to the nodal forces $\{Q\}_e$ through the element stiffness equilibrium relation:

$$[K]_e \{q\}_e = \{Q\}_e \quad (3.1)$$

The elements of $[K]_e$ are derived from the material and geometric properties of the element. They may be obtained by the use of the principle of minimum potential energy as will be described later.

4. Assembly of the overall algebraic equations of the discretized continuum: The global stiffness matrix $[K]$, and the global load vector $\{Q\}$ are assembled. The global stiffness equilibrium relation is arrived at

$$[K]\{q\} = \{Q\} \quad (3.2)$$

At this stage the equations are modified by the introduction of the appropriate geometric boundary conditions.

5. Solution of the equations: The equations (3.2) assembled in the previous step are solved for the unknown displacements.

6. Calculation of strains and stresses: The strains are computed by using the relations between them and the relevant displacements. By using Hooke's law, the stresses are computed from the strains.

The operations involved in the previous steps for the axisymmetric case will be described in detail in the following articles.

3.2 Natural coordinates and shape functions

A variational principle will be adopted later for the formulation of element stiffness and load and will be described. In general, the potential energy for a linearly elastic body can be expressed as a sum of the strain energy due to internal stresses and the potential of the body forces and surface tractions. By taking the variation of the total potential energy with respect to general coordinates, an equilibrium equation (3.2) is arrived at. Because the integrations involved are awkward, formulation in terms of natural coordinates becomes advantageous.

A local coordinate system is defined for a particular element and not for the entire body whose coordinates system is called the global system. A natural coordinates system is a local one which specifies a point within the element by a set of dimensionless numbers whose values do not exceed unity, and hence the point is specified relative to the nodes of the element alone and is independent of its orientation or position in space. Consequently, element stiffness formulation can be carried out without being directly concerned with the global coordinates or geometry. Finally it is more convenient to carry out the computations with coordinates between zero and one, as simple arithmetic operations with large numbers alongside small ones tend to produce solutions which are not well conditioned.

A natural coordinates system of a triangular ring element is shown in (Fig. 3.2). The three coordinates are L_1 , L_2 and L_3 ; but only two of them are independent. Their relation to the global cylindrical

coordinates system is given by:

$$\begin{Bmatrix} 1 \\ r \\ z \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ r_1 & r_2 & r_3 \\ z_1 & z_2 & z_3 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} \quad (3.3)$$

This set of natural coordinates can be considered as one of area coordinates. Thus the coordinates of point (P) Fig.(3.3) are,[4]:

$$L_1 = \frac{A_1}{A}, \quad L_2 = \frac{A_2}{A}, \quad L_3 = \frac{A_3}{A} \quad (3.4)$$

It can be seen that L_1 , L_2 and L_3 are not independent but related by:

$$\sum_{i=1}^3 L_i = 1 \quad (3.5)$$

Equation (3.3) can be inverted to yield:

$$\begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} 2A_{23} & b_1 & a_1 \\ 2A_{31} & b_2 & a_2 \\ 2A_{12} & b_3 & a_3 \end{bmatrix} \begin{Bmatrix} 1 \\ r \\ z \end{Bmatrix} \quad (3.6)$$

Here, A_{ij} = Areas of triangles whose vertices are the origin of the global coordinates system 0 and node i and j Fig.(3.3) and, [3]:

$$\begin{aligned} a_1 &= r_3 - r_2 & b_1 &= z_2 - z_3 \\ a_2 &= r_1 - r_3 & b_2 &= z_3 - z_1 \\ a_3 &= r_2 - r_1 & b_3 &= z_1 - z_2 \end{aligned} \quad (3.7)$$

$$2A = a_3 b_2 - a_2 b_3 = a_1 b_3 - a_3 b_1 = a_2 b_1 - a_1 b_2 \quad (3.8)$$

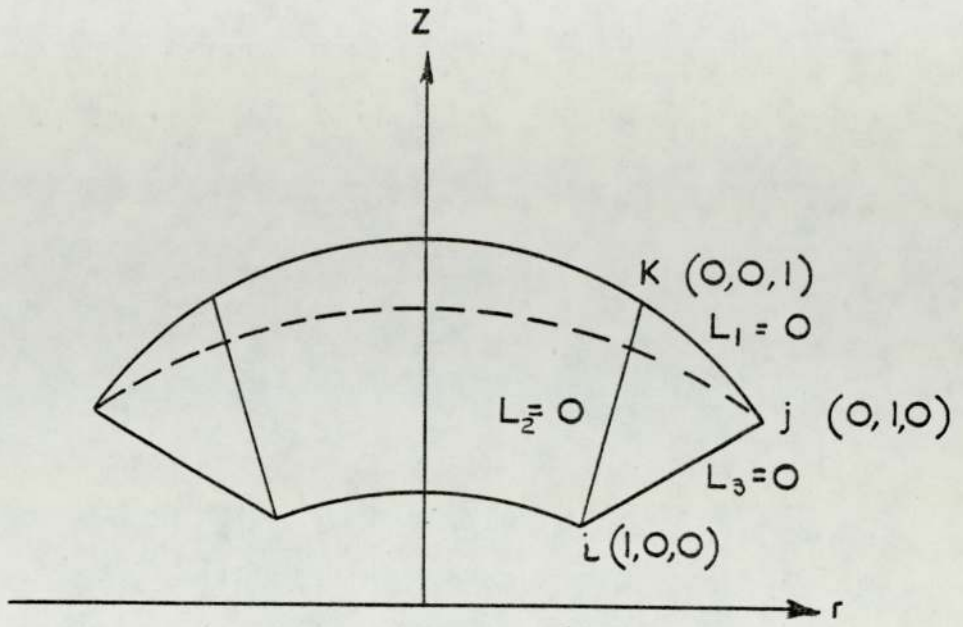


Fig. 3. 2

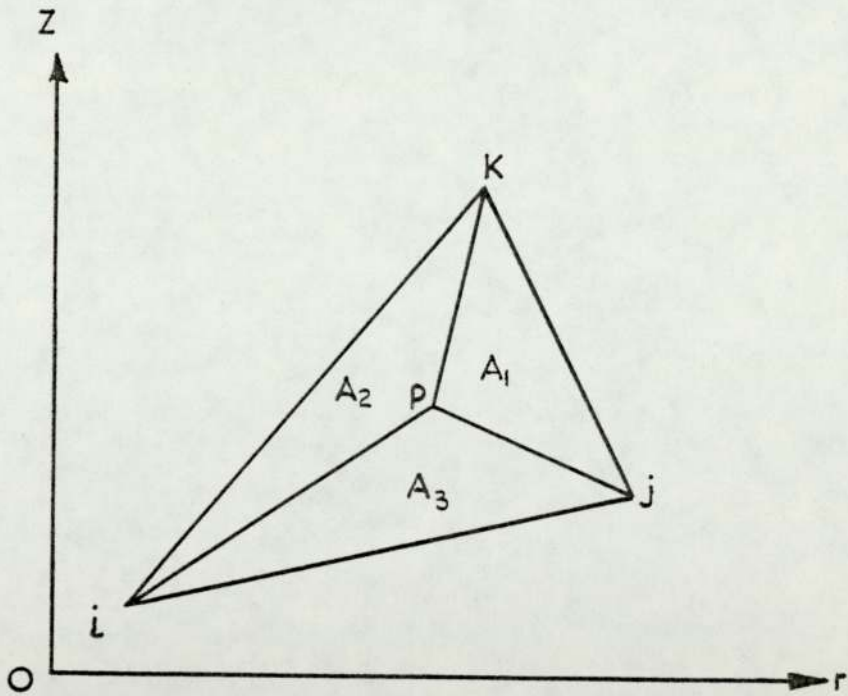


Fig. 3. 3

It is required now to assume a function which represents approximately the variation of displacements in the element and satisfies the requirements described in section (3.1). These functions are called displacement functions and a linear model in a polynomial form in the two dimensional case is, [3]:

$$\begin{aligned} u &= \alpha_1 + \alpha_2 r + \alpha_3 z \\ w &= \alpha_4 + \alpha_5 r + \alpha_6 z \end{aligned} \tag{3.9}$$

And a quadratic model is:

$$\begin{aligned} u &= \alpha_1 + \alpha_2 r + \alpha_3 z + \alpha_4 r^2 + \alpha_5 rz + \alpha_6 z^2 \\ w &= \alpha_7 + \alpha_8 r + \alpha_9 z + \alpha_{10} r^2 + \alpha_{11} rz + \alpha_{12} z^2 \end{aligned} \tag{3.10}$$

Where the (α 's) are known as generalized coordinates. Writing equation (3.10) in matrix form:

$$\{u\} = [\phi]\{\alpha\} \tag{3.11}$$

Where:

$$\{u\} = \begin{Bmatrix} u(r,z) \\ w(r,z) \end{Bmatrix} \tag{3.12}$$

$$[\phi] = \begin{bmatrix} \{\phi_1\}^t & \{0\}^t \\ \{0\}^t & \{\phi_1\}^t \end{bmatrix} \tag{3.13}$$

$$\{\phi_1\}^t = [1 \quad r \quad z \quad r^2 \quad rz \quad z^2] \tag{3.14}$$

$$\{\alpha\}^t = [\alpha_1 \quad \alpha_2 \quad \alpha_3 \dots \alpha_{12}] \tag{3.15}$$

The nodal displacements can be evaluated by substituting the nodal coordinates, and the vector of nodal displacements of the nodes for the element under consideration may be written as:

$$\{q\} = [A]\{\alpha\} \quad (3.16)$$

where: $\{q\}^t = [u_1 w_1 \quad u_2 w_2 \dots]$

Equation (3.16) is inverted to give:

$$\{\alpha\} = [A]^{-1}\{q\} \quad (3.17)$$

Substituting equation (3.17) into (3.11) gives:

$$\{u\} = [\phi][A]^{-1}\{q\} = [N]\{q\} \quad (3.18)$$

Examining $[N]$ shows that it has the form of an interpolation function matrix, and another direct and more elegant way of generating it would be the use of interpolation theory, and thus eliminating matrix $[A]$ from the formulation.

An interpolation function or a shape function is a function which has unit value at one nodal point and zero value at the other nodal points, therefore it is convenient to express it in terms of natural coordinates.

Beginning with the simple case of linear interpolation over the triangular element, prescribed values at three nodes are required, so the vertices become the nodal points. Any linear displacement function $u(r,z)$ over the element with prescribed nodal values u_1 , u_2 and u_3 , is a linear combination of the three Lagrangian polynomials $N_1(L_1, L_2, L_3)$,

$N_2(L_1, L_2, L_3)$, and $N_3(L_1, L_2, L_3)$ with scalar coefficients u_1 , u_2 , and u_3 , [5].

For a three node triangle:

$$u(L_1, L_2, L_3) = [N_1(L_1, L_2, L_3) \quad N_2(L_1, L_2, L_3) \quad N_3(L_1, L_2, L_3)] \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} \quad (3.19)$$

For linear interpolation the shape functions are the natural coordinates i.e.:

$$N_i(L_1, L_2, L_3) = L_i \quad (i = 1, 2, 3) \quad (3.20)$$

If $u(L_1, L_2, L_3)$ is interpreted as the radial displacement function, a similar interpolating function $w(L_1, L_2, L_3)$ can be written for the axial displacement function in terms of nodal values w_1, w_2 , and w_3 .

For a quadratic displacement function $u(r, z)$, six function values must be prescribed. Selecting the triangular element with nodes at the vertices and side mid points, the corresponding shape functions $N_i(L_1, L_2, L_3)$ are:

$$\begin{aligned} N_1 &= L_1(2L_1 - 1) \\ N_2 &= L_2(2L_2 - 1) \\ N_3 &= L_3(2L_3 - 1) \\ N_4 &= 4L_1L_2 \\ N_5 &= 4L_2L_3 \\ N_6 &= 4L_3L_1 \end{aligned} \quad (3.21)$$

3.3 The six node isoparametric triangular ring element

An element is said to be isoparametric if both the geometry and the displacements of the element are described in terms of the same parameters and are of the same order. The simplest finite element to give displacements that converge to the analytic solution must contain constant strain terms. As the strains are first derivatives of displacements, a linear function may be chosen for the displacements. However better results can be expected from fewer elements if the strains were allowed to vary linearly within the element, [6]. As seen from the previous article, selecting the triangle vertices and midside points as nodal points and the shape functions of (3.21) yields a quadratic displacement function. From the definition of the isoparametric element, a coordinate transformation of the form:

$$\{X\} = [N]\{X_n\} \quad (3.22)$$

is obtained where, $[N]$ is given by (3.21) and $\{X_n\}$ are the element natural coordinates. Hence, an element which is straight sided in local coordinates terms becomes curved in global coordinates terms. This property enables the isoparametric element to represent curved boundaries. The mapping from local coordinates to global ones for the six node isoparametric triangular element is shown in (Fig.3.4).

3.4 The strain-displacement relations

The relation between the global and local sets of coordinates for the six node isoparametric ring element is given by:

$$\begin{Bmatrix} r \\ z \end{Bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 \\ z_1 & z_2 & z_3 & z_4 & z_5 & z_6 \end{bmatrix} \begin{Bmatrix} N_1 \\ N_2 \\ \cdot \\ N_6 \end{Bmatrix} \quad (3.23)$$

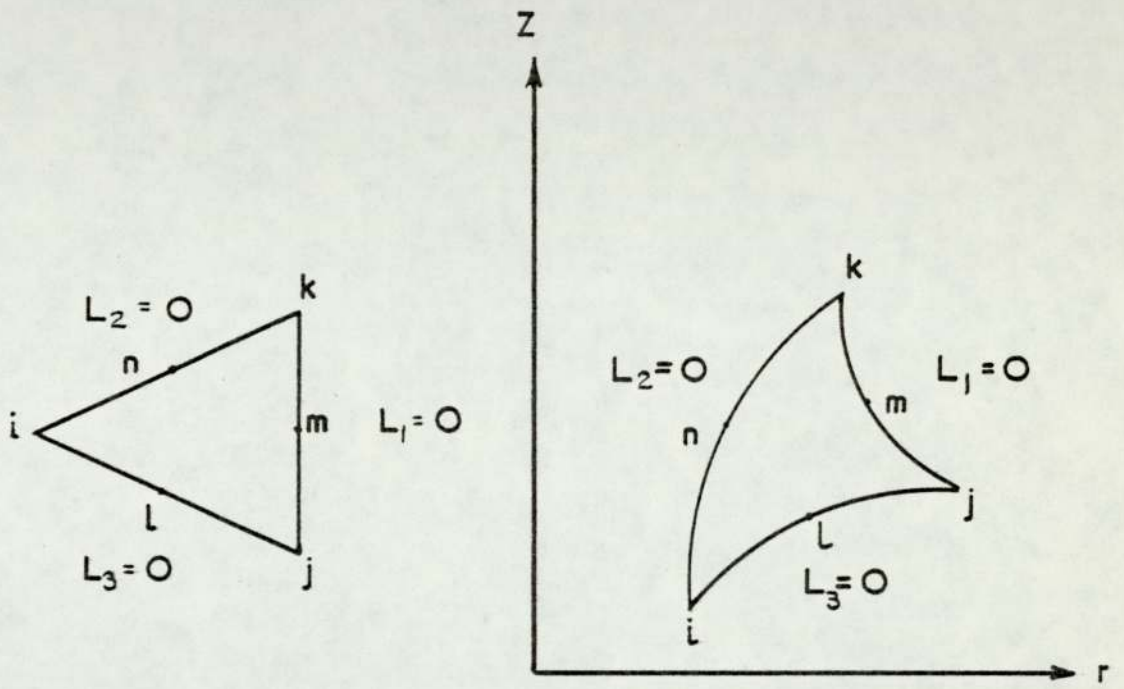


Fig. 3.4.

And the relation between the displacements is:

$$\begin{Bmatrix} u \\ w \end{Bmatrix} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{bmatrix} \begin{Bmatrix} N_1 \\ N_2 \\ \vdots \\ N_6 \end{Bmatrix} \quad (3.24)$$

where $\{N_i\}$ $i = 1, 2, \dots, 6$ is given by equation (3.21).

In the axisymmetric case the strain-displacement relations are given by equations (2.25) which require derivatives of the form

$\frac{\partial u}{\partial r}$ and $\frac{\partial w}{\partial z}$... etc.

$$\begin{aligned} \text{Where} \quad \frac{\partial u}{\partial r} &= \sum_{i=1}^6 \frac{\partial N_i}{\partial r} \frac{\partial u}{\partial N_i} \\ \frac{\partial w}{\partial z} &= \sum_{i=1}^6 \frac{\partial N_i}{\partial z} \frac{\partial w}{\partial N_i} \\ &\vdots \\ &\vdots \\ &\text{etc.} \end{aligned} \quad (3.25)$$

Hence equations (3.23) need to be inverted to give terms like

$\frac{\partial N_i}{\partial r}$ and $\frac{\partial N_i}{\partial z}$.

To be able to do this, two new variables (s) and (t) are introduced so that:

$$L_1 = s, \quad L_2 = t, \quad L_3 = 1-s-t \quad (3.26)$$

Now

$$\begin{Bmatrix} \frac{\partial N_i}{\partial s} \\ \frac{\partial N_i}{\partial t} \end{Bmatrix} = \begin{bmatrix} \frac{\partial r}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial r}{\partial t} & \frac{\partial z}{\partial t} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} \quad (3.27)$$

or

$$\begin{Bmatrix} \frac{\partial N_i}{\partial s} \\ \frac{\partial N_i}{\partial t} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} \quad (3.28)$$

where $[J]$ = Jacobean matrix

By inverting equation (3.28) we obtain:

$$\begin{Bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial s} \\ \frac{\partial N_i}{\partial t} \end{Bmatrix} \quad (3.29)$$

where

$$[J]^{-1} = \frac{1}{/J/} \begin{bmatrix} \frac{\partial z}{\partial t} & -\frac{\partial z}{\partial s} \\ -\frac{\partial r}{\partial t} & \frac{\partial r}{\partial s} \end{bmatrix} \quad (3.30)$$

$/J/$ = The determinant of the Jacobean matrix and is called simply the Jacobian, that is

$$/J/ = \frac{\partial r}{\partial s} \frac{\partial z}{\partial t} - \frac{\partial r}{\partial t} \frac{\partial z}{\partial s} \quad (3.31)$$

To evaluate the Jacobian, the following relations are used:

$$\frac{\partial r}{\partial s} = \sum_{i=1}^6 \frac{\partial N_i}{\partial s} \frac{\partial r}{\partial N_i} \quad (3.32)$$

$$\frac{\partial r}{\partial t} = \sum_{i=1}^6 \frac{\partial N_i}{\partial t} \frac{\partial r}{\partial N_i}$$

Similarly for $\frac{\partial z}{\partial s}$ and $\frac{\partial z}{\partial t}$

$$\frac{\partial N_i}{\partial s} \text{ and } \frac{\partial N_i}{\partial t} \text{ are obtained from equations (3.26) and}$$

$\frac{\partial r}{\partial N_i}$ and $\frac{\partial z}{\partial N_i}$ are obtained from equations (3.23) and hence by substituting

in equations (3.29), $\frac{\partial N_i}{\partial r}$ and $\frac{\partial N_i}{\partial z}$ are obtained, and they are multiplied

by $\frac{\partial u}{\partial N_i}$ and $\frac{\partial w}{\partial N_i}$ which are obtained from equations (3.24) to yield the

required strains as shown in equations (3.25). Written in matrix form,

the strain displacement relation is:

$$\begin{Bmatrix} \epsilon_r \\ \epsilon_\theta \\ \epsilon_z \\ \gamma_{rz} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{u}{r} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} \end{Bmatrix} \begin{bmatrix} B_{1,1} & 0 & B_{1,3} & 0 & \dots & B_{1,11} & 0 \\ B_{2,1} & 0 & B_{2,3} & 0 & \dots & B_{2,11} & 0 \\ 0 & B_{3,2} & 0 & B_{3,4} & \dots & 0 & B_{3,12} \\ B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & \dots & B_{4,11} & B_{4,12} \end{bmatrix} \begin{Bmatrix} u_1 \\ w_1 \\ u_2 \\ w_2 \\ \dots \\ u_6 \\ w_6 \end{Bmatrix}$$

(3.33)

or $\{\epsilon\} = [B]\{q\}$

(3.34)

3.5 Constitutive relations

3.5.1 Introduction

The six components of strain at every point in the general elastic problem are functions of the six stress components at that point.

This relation in matrix form is:

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & \cdot & a_{16} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & \cdot & a_{26} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{61} & a_{62} & \cdot & \cdot & \cdot & \cdot & a_{66} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{xz} \\ \tau_{xy} \end{Bmatrix} \quad (3.35)$$

By using the principle of conservation of energy, it may be shown that the matrix $[a_{ij}]$ where $i = j = 1 - 6$ is symmetric and hence only (21) of the (36) coefficients are sufficient to describe any material, [7]. The number of elastic constants for any material may be reduced further by considering planes or axes of elastic symmetry. Since the work presented in this thesis deals with axisymmetric problems, the choice of materials is limited to isotropic and anisotropic stratified materials only.

3.5.2 Isotropic material

In the finite element assumed displacement method, stresses are calculated from strains, hence the inverse form of equation (3.35)

is used. For an axisymmetric problem with linear isotropic material, the constitutive relations are given in equation (2.24) which has the general form:

$$\{\sigma\} = [c]\{\epsilon\} \quad (3.36)$$

3.5.3 Anisotropic stratified materials

In this case the constants E_1 and ν_1 are associated with the behaviour in the plane strata which is parallel to $r-\theta$ axes, and E_2, μ_2, ν_2 with the direction normal to them which is the z axis (Fig. 3.5).

$$\text{writing } \frac{E_1}{E_2} = n \quad \text{and} \quad \frac{\mu_2}{E_2} = m \quad (3.37)$$

The constitutive relations are

$$\begin{Bmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_z \\ \tau_{rz} \end{Bmatrix} = \frac{E_2}{(1+\nu_1)(1-\nu_1-2\nu_2^2)} \begin{bmatrix} n(1-\nu_2^2) & (\nu_1+\nu_2^2)n & \nu_2(1+\nu_1) & 0 \\ & n(1-\nu_2^2) & \nu_2(1+\nu_1) & 0 \\ \text{Symm.} & & 1-\nu_1^2 & 0 \\ & & & m(1+\nu_1) \\ & & & x(1-\nu_1-2\nu_2^2) \end{bmatrix} \begin{Bmatrix} \epsilon_r \\ \epsilon_\theta \\ \epsilon_z \\ \gamma_{rz} \end{Bmatrix} \quad (3.38)$$

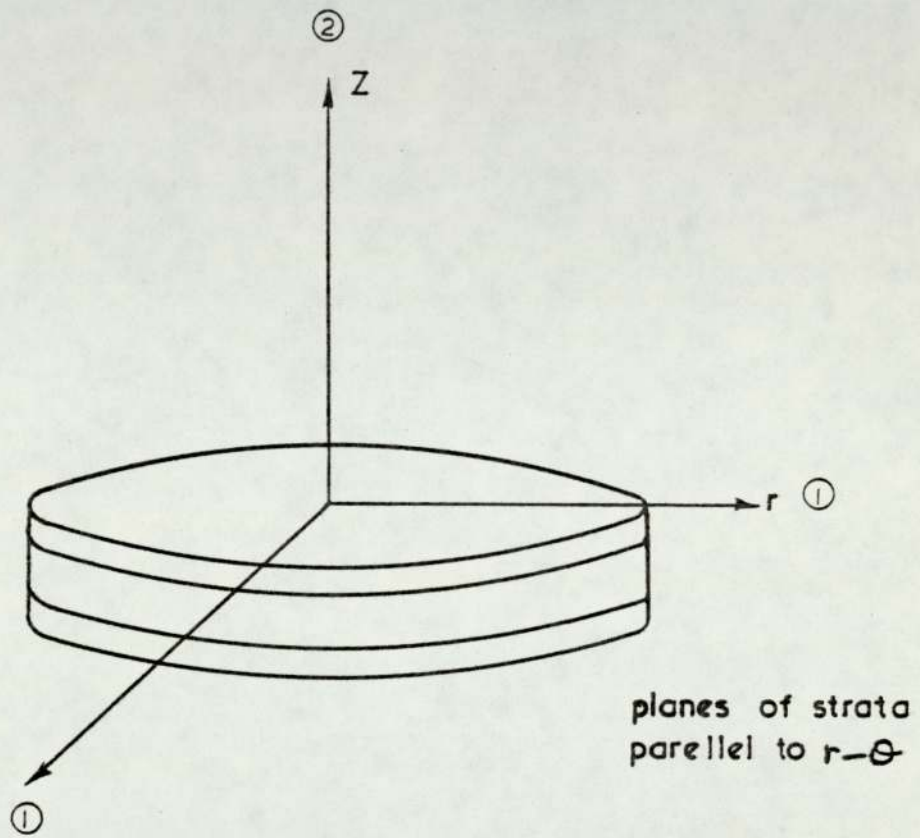


Fig. 3.5

3.6 The variational formulation of element stiffness and loads

The total potential energy of an elastic body is defined as:

$$\Pi = U + \Omega \quad (3.39)$$

where U = Strain energy

Ω = Potential of applied loads

The principle of minimum potential energy can be stated as follows: of all the possible displacement configurations a body can assume which satisfy compatibility and the constraints or kinematic boundary conditions, the configuration satisfying equilibrium makes the potential energy assume a minimum value, [3]. Therefore

$$\delta\Pi = \delta U - \delta W = 0 \quad (3.40)$$

where W = work done by the loads.

The potential energy for a linear elastic body can be expressed as:

$$\begin{aligned} \Pi = \int_{vol} U_0 dvol - \int_{vol} (\bar{X} + \bar{Y} + \bar{Z}) dvol - \int_S (\bar{T}_x U + \bar{T}_y V \\ + \bar{T}_z w) dS \end{aligned} \quad (3.41)$$

where U_0 = Strain energy/unit volume

$\bar{X}, \bar{Y}, \bar{Z}$ = body forces

$\bar{T}_x, \bar{T}_y, \bar{T}_z$ = Surface tractions

The strain energy density for a linear elastic body can be written in matrix form as:

$$\frac{1}{2} \{\epsilon\}^t \{\sigma\} dvol = \frac{1}{2} \{\epsilon\}^t [c] \{\epsilon\} dvol \quad (3.42)$$

Substituting equation (3.42) into equation (3.41) and writing it in matrix form gives:

$$\Pi = \frac{1}{2} \int_{vol} (\{\epsilon\}^t [C] \{\epsilon\} - 2\{u\}^t \{\bar{X}\}) dvol - \int_S \{u\}^t \{\bar{T}\} dS \quad (3.43)$$

where:

$$\{u\}^t = [u \ v \ w]$$

$$\{\bar{X}\}^t = [\bar{X} \ \bar{Y} \ \bar{Z}]$$

$$\{\bar{T}\}^t = [\bar{T}_x \ \bar{T}_y \ \bar{T}_z]$$

V = volume of element

S = part of the element surface over which tractions are specified.

Substituting equations (3.18, 3.34, and 3.36) into equation (3.43) gives:

$$\Pi = \frac{1}{2} \int_{vol} (\{q\}^t [B]^t [C] [B] \{q\} - 2\{q\}^t [N]^t \{\bar{X}\}) dvol - \int_S \{q\}^t [N]^t \{\bar{T}\} dS \quad (3.44)$$

Applying the variational principle:

$$\{\delta q\}^t \left(\int_{vol} [B]^t [C] [B] dvol \{q\} - \int_{vol} [N] \{\bar{X}\} dvol - \int_S [N]^t \{\bar{T}\} dS_1 \right) = 0 \quad (3.45)$$

Since $\{\delta q\}^t$ is arbitrary:

$$\int_{vol} [B]^t [C] [B] dVol \{q\} = \int_{Vol} [N] \{\bar{X}\} dVol + \int_S [N]^t \{\bar{T}\} dS \quad (3.46)$$

or

$$[K]_e \{q\}_e = \{Q\}_e \quad (3.47)$$

where

$$[K]_e = \int_{vol} [B]^t [C] [B] dVol \quad (3.48)$$

$$\{Q\}_e = \int_{vol} [N]^t \{\bar{X}\} dVol + \int_S [N]^t \{\bar{T}\} dS \quad (3.49)$$

And dVol for the axisymmetric ring element is:

$$dVol = 2\pi R dA \quad (3.50)$$

where R = the perpendicular distance from the axis
of revolution to the centre of dA

$$dA = dr dz \quad (3.51)$$

The evaluation of the integrals in equations (3.48, 3.49) in closed form for the six node isoparametric ring element requires evaluating (21) separate integrals listed in, [6]. As this is a cumbersome operation, it becomes more feasible to resort to numerical integration which should be performed with respect to global coordinates. But, as the [B] matrix is a function of local natural coordinates, it becomes necessary to transform the integral into them by applying a standard process using the Jacobian (det[J]), [8]:

$$dA = dr dz = \det[J] dL_1 dL_2 \quad (3.52)$$

Hence equation (3.48) becomes:

$$[K]_e = \int_0^1 \int_{1-L_2}^1 [B]^t [C] [B] 2\pi R \det[J] dL_1 dL_2 \quad (3.53)$$

To calculate an integral of the form:

$$I = \int \int f(L_1, L_2, L_3) dL_1 dL_2, \quad (3.54)$$

it is replaced in all methods of numerical integration by the sum:

$$I = \sum_{i=1}^n W_i f(L_{1i}, L_{2i}, L_{3i}) \quad (3.55)$$

where n = integration points = number of points at which the function is to be evaluated

W_i = weight coefficients

$f(L_{1i}, L_{2i}, L_{3i})$ = the function value at the integrating point L_{1i}, L_{2i}, L_{3i} .

There are several quadrature formulae, [4], but the one chosen is the Gauss-Legendre one. Appendix (10.1) gives the details of the weighting coefficients and the triangular coordinates. The order of integration which effects the computing time and accuracy of the solution, was chosen to be quadratic. This choice was guided by the experience obtained by Robertson, [9], in solving several two dimensional crack problems, and was later justified by the accuracy of the results obtained.

Equation (3.54) in terms of the quadrature formula (3.56) becomes:

$$[K]_e = 2\pi R \sum_{i=1}^n W_i ([B]^t [C] [B] \det[J])_{L_{1i}, L_{2i}, L_{3i}} \quad (3.56)$$

where $([B]^t [C] [B] \det[J])_{L_{1i}, L_{2i}, L_{3i}}$ means the evaluation of the product in parenthesis at integrating point i .

3.7 Assembly of the overall stiffnesses and loads

If the continuum under consideration is divided into (E) number of elements interconnected at (N) number of nodes, with two degrees of freedom per node; then the total number of undetermined displacements is $(2N)$ and the order of the overall stiffness matrix is

[2NX2N]. The assembly of the overall stiffness matrix and load vector can be achieved by expanding the element stiffness matrix $[K]_e$ to a size of [2NX2N] and the element load vector $\{Q\}_e$ to a size of {2N}, changing local numbers to global numbers, and locating the coefficients in their appropriate positions in the new enlarged matrices. Summing up the contributions of all the elements, the overall stiffness matrix $[K]$ and the overall load vector $\{Q\}$ are obtained:

$$[K]\{q\} = \{Q\} \quad (3.57)$$

where

$$[K] = \sum_{e=1}^E [K]_e \quad (3.58)$$

$$\{Q\} = \sum_{e=1}^E \{Q\}_e \quad (3.59)$$

3.8 Automatic mesh generation

3.8.1 Introduction

The preparation of an error-free input data for a complicated problem is very tedious and time consuming. To minimize this effort many authors include some facility for numbering the nodes and elements, and computing nodal coordinates. Becker and Brisband,[10], automated the input data for special cases of simple geometries. Frederick, Wong, and Edge,[11], developed a partially automated method of discretizing irregular and non-homogeneous two dimensional continua. A more general approach utilizing the concept of natural coordinates in mapping curved boundaries was developed also [12,13]. However, all these methods are not fully automatic and require some manual instructions through which engineering judgement is exercised. An automatic mesh generation scheme was developed to help in reducing the input data required for the various problems tackled in this thesis. The six node isoparametric ring element was chosen for the work and the scheme was based on it.

3.8.2 Mesh generation for an axisymmetric structure

By definition, the structure is symmetric with respect to the axis of revolution, hence only half of its longitudinal section need be discretized. If there is further symmetry with respect to a diametral section, then only one quarter of the section is discretized. The method will be described for a simple case shown in Fig. (3.6). Lines AB, DC, and all lines parallel to them are called nodal rows, while lines AD, BC, and all lines parallel to them are called nodal columns. It could be shown that

$$\text{The number of nodal rows} = (\text{No. of element rows} \times 2) + 1 \quad (3.60)$$

$$\text{The number of nodal columns} = (\text{No. of element columns} \times 2) + 1$$

$$\text{The number of nodes/row} = \text{nodal columns} \quad (3.62)$$

$$\text{The number of nodes/column} = \text{nodal rows} \quad (3.63)$$

It is seen that equations (3.60) to (3.63) are functions of element rows and columns, hence they are the only two factors that need be specified to generate the mesh in Fig. (3.6).

The fact that each nodal point is an intersection of a nodal row and column is used in numbering the nodes as:

$$\text{Node number} = (M \times N) + J \quad (3.64)$$

where

$$N = \text{Nodal rows}$$

$$M = I - 1$$

$$I = \text{Nodal column number}$$

$$J = \text{Nodal row number}$$

○ node number
□ element number

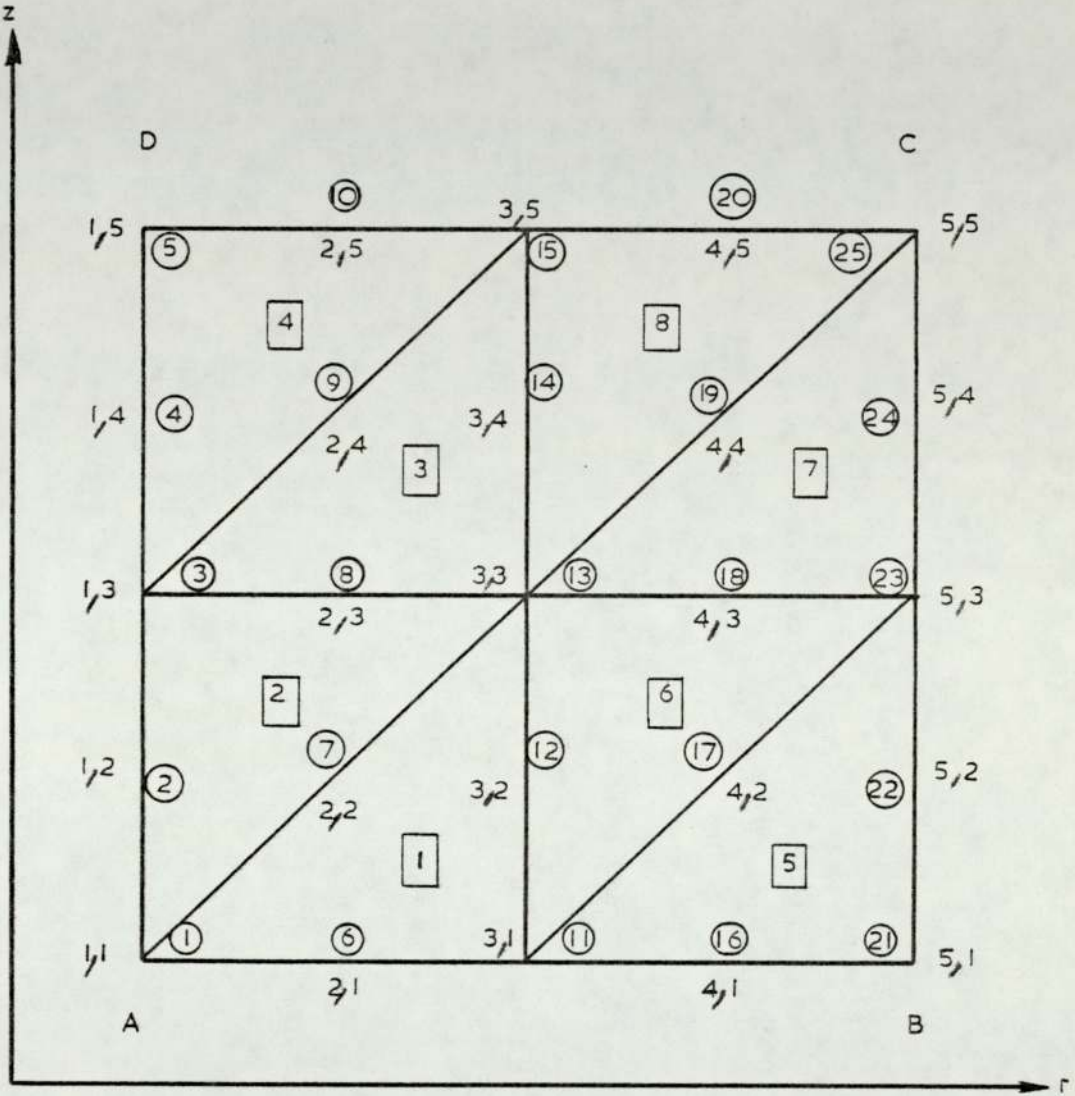


Fig. 3.6

The coordinates of the nodes, for the special case of a rectangular section, are obtained by specifying the starting and finishing global r and z coordinates of the structure, i.e. points A, B, and D in Fig. (3.6).

$$\Delta r = \frac{r_{\text{finish}} - r_{\text{start}}}{\text{element columns} \times 2} \quad (3.65)$$

$$\Delta z = \frac{z_{\text{finish}} - z_{\text{start}}}{\text{element rows} \times 2} \quad (3.66)$$

Therefore for a point (N)

$$r(N) = r_{\text{start}} + (I-1)\Delta r \quad (3.67)$$

$$z(N) = z_{\text{start}} + (J-1)\Delta z \quad (3.68)$$

If the section has curved boundaries so that the element rows and/or columns do not have the same length, then the starting and finishing r and z global coordinates must be specified for all nodal rows and columns.

To work out the nodal connections in the sequence (i,j,k,l,m,n) shown in Fig. (3.7), it was noted that odd number elements and even number elements, each follow a similar pattern. In each element column, the corresponding nodal numbers increase by two when moving from one odd element to the next one, and similarly for even elements. When moving from one element column to the next, the numbers of corresponding nodes increase by a factor of (2XNodal rows). A computer code performing these operations will be described in section (7.9). The nodal connections of elements can be specified in such a manner that symmetry with respect to a diametral section is obtained Fig.(3.8).

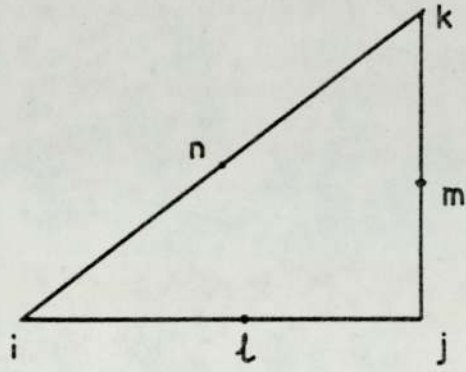


Fig. 3.7

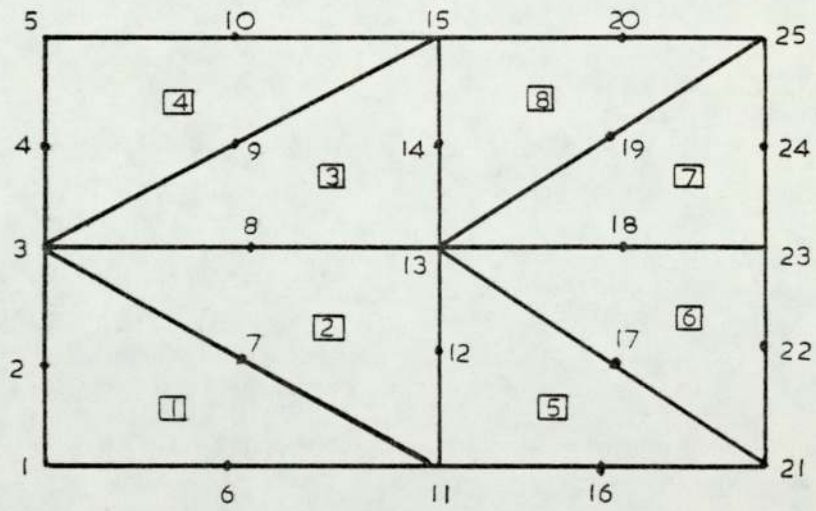


Fig. 3.8

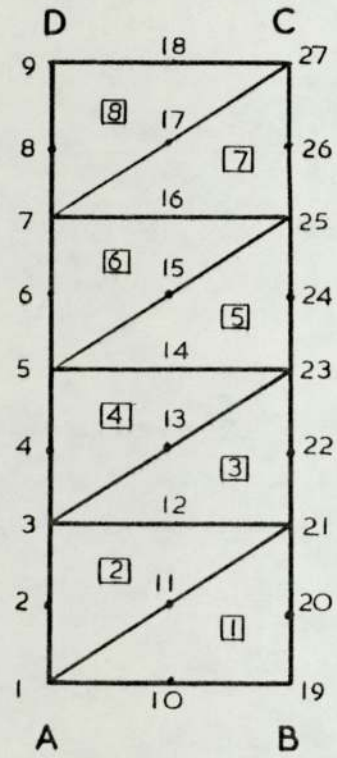
This will prove very useful in certain fracture problems as will be seen in Chapter 8. The computer code for it will be described in section (7.9) also.

3.8.3 Mesh generation around a core

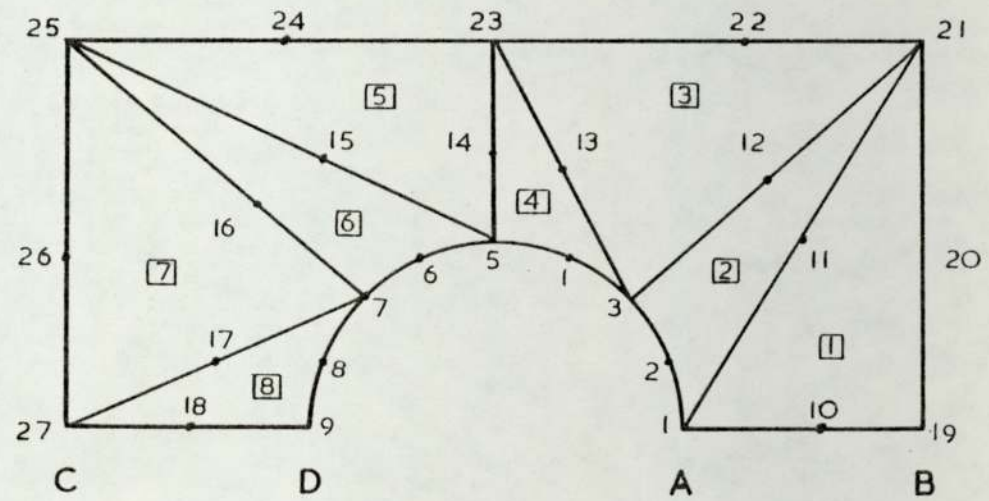
It will be shown later, how in fracture problems, a mesh is generated around a circular or semicircular core. To achieve this the nodal rows and columns, which are straight lines, are bent into semicircular, circular, or rectangular shapes as shown in Fig. (3.9). It must be noted that even number nodal columns, which represent mid-side nodes, cannot be bent arbitrarily, but their shape will be a function of that of the columns immediately before and after them.

The radius of a circular or semicircular nodal column, and the coordinates of the vertices and the number of nodal rows intersecting each side need to be specified. However, the core size in fracture is a function of the crack length, and the number of nodes on the core and hence the number of nodal rows is the maximum allowed by the computer storage which works out to be the same for a large range of problems. Therefore these parameters were built into the scheme and the only things which need to be specified were the crack length and the dimensions of the discretized section.

When bending nodal rows and columns, care must be taken not to severely distort or overlap elements Fig. (3.10), as this will cause the computer program to fail.



underformed mesh



deformed mesh

Fig. 3.9

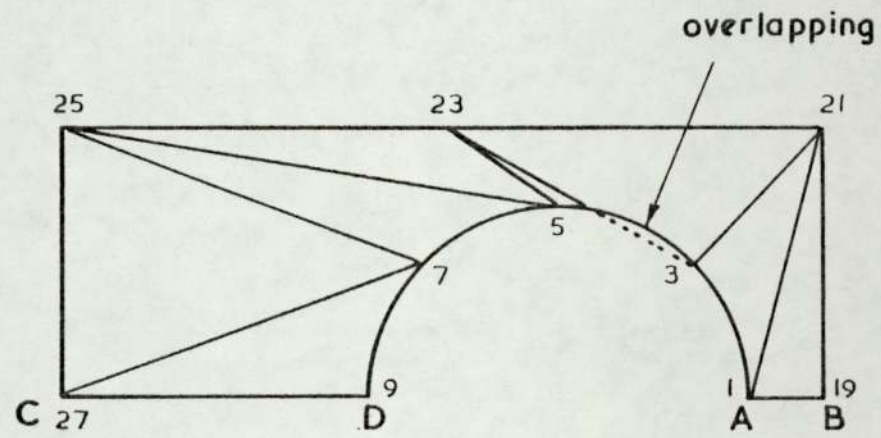


Fig 3.10

The nodal connections for this case remain exactly as those described in the previous section (3.8.2).

The computer code which performs the operations described in this section will be presented in sections (7.10), (7.11) and (7.12).

CHAPTER 4

THEORIES OF BRITTLE FRACTURE

4.1 Introduction

It has been seen in practice, that although safety requirements were apparently satisfied in the design of some engineering components, they failed during initial loading or after several load cycles, leading in some cases to disastrous results, [14]. Their failure was attributed to the initial existence of small cracks or flaws in them. It became apparent that designing cross section areas to keep the gross stresses below the yield point of the material is not adequate for high strength materials which are sensitive to the presence of flaws or cracks, [15]. The theory of fracture mechanics was developed to provide an analysis which includes the reduction of the strength of the material due to that presence.

The Griffith theory, which was proposed over half a century ago, is the starting point. It defined the conditions which make a small crack in a solid unstable by adopting an energy balance approach.

An alternative approach, developed by Irwin, focused attention on the mechanical environment near the crack tip and is known as the stress intensity factor approach. He noted that the stresses in the vicinity

of a crack tip are characterized by a special factor (K) called the stress intensity factor and used the critical intensity of these stresses as a material constant.

These approaches and the various modes of crack propagation will be described subsequently.

4.2 The Griffith theory

Griffith stated that unstable crack propagation takes place if a small crack growth released more stored energy than that which the newly created crack surface has absorbed, [16]. The evaluation of energies required by this theory is difficult if considerable plastic deformation is associated with the crack extension. To overcome this, Griffith directed his attention to hypothetical materials which behave in a purely elastic manner prior to crack propagation. The specific problem considered was that of a crack of length (2a) embedded in an infinite elastic body subjected to direct uniform stresses perpendicular and parallel to the crack surface (σ_x and σ_y) Fig. (4.1). It could be shown that this system is equivalent to a crack of length (2a) subjected to a uniform stress ($\sigma_0 = \sigma_y$) perpendicular to its surface Fig.(4.2) and that (σ_x) has no influence on crack stability, [17].

The strain energy of the body in Fig. (4.2) is equal to the work done in deforming the crack surface and is given by, [18],

$$U = \frac{[\kappa+1]}{8\mu} \pi a^2 t \sigma_0^2 \quad (4.1)$$

where

$$\begin{aligned} t &= \text{uniform thickness.} \\ \kappa &= 3-4\nu \quad , \text{ plane strain} \\ &= (3-\nu)/(1+\nu), \text{ plane stress} \end{aligned}$$

Applying this result to the original system of Fig. (4.1):

$$U_{\text{total}} = \frac{(\kappa+1)\pi a^2 t \sigma_y^2}{8\mu} + \bar{U} \quad (4.2)$$

where

\bar{U} = The energy component independent of crack presence.

The surface energy of the crack is:

$$U_{\text{surface}} = 4 \text{ at } \gamma \quad (4.3)$$

where

γ = Specific surface energy

For a crack increment of $(2\delta a)$:

$$\delta U_{\text{total}} = \frac{(\kappa+1)\pi a t \sigma_y^2}{4\mu} \delta a \quad (4.4)$$

And

$$\delta U_{\text{surface}} = 4t\gamma\delta a \quad (4.5)$$

According to Griffith, the crack growth will be unstable if:

$$\frac{(\kappa+1)\pi a t \sigma_y^2}{4\mu} \delta a > 4t\gamma\delta a \quad (4.6)$$

or

$$\sigma_{\text{critical}} \geq 4 \sqrt{\frac{\mu\gamma}{\pi a(\kappa+1)}} \quad (4.7)$$

where

σ_{critical} is that σ_y which causes instability

The significance of this theory was the demonstration of the existence of a relationship between crack length and failure stress.

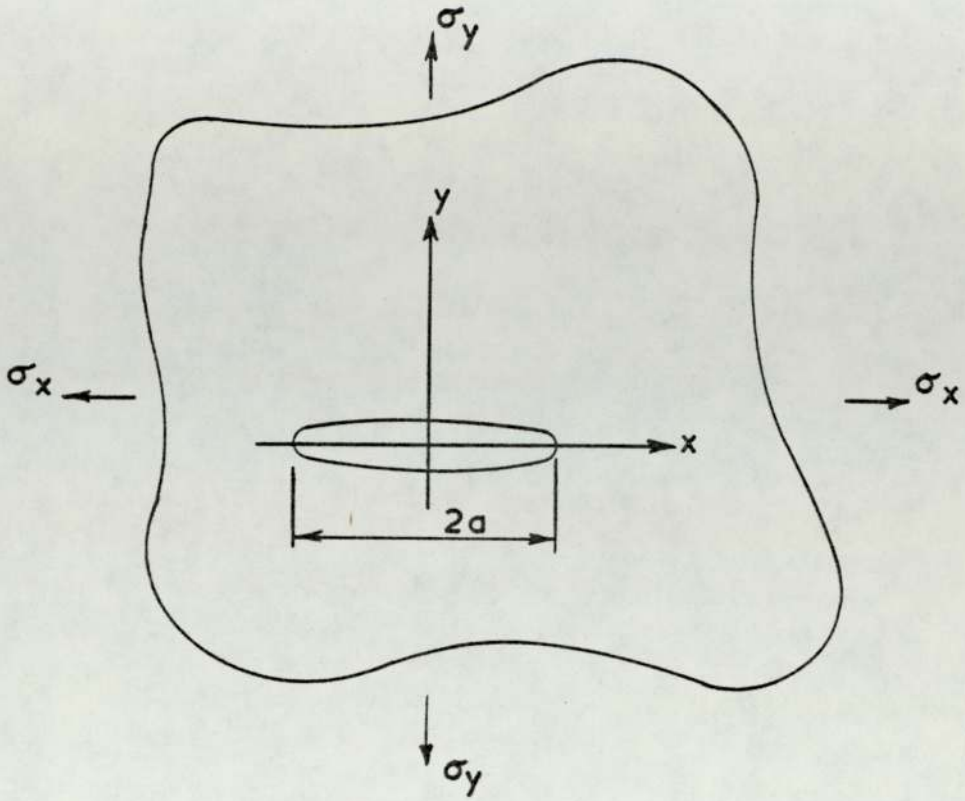


Fig. 4.1

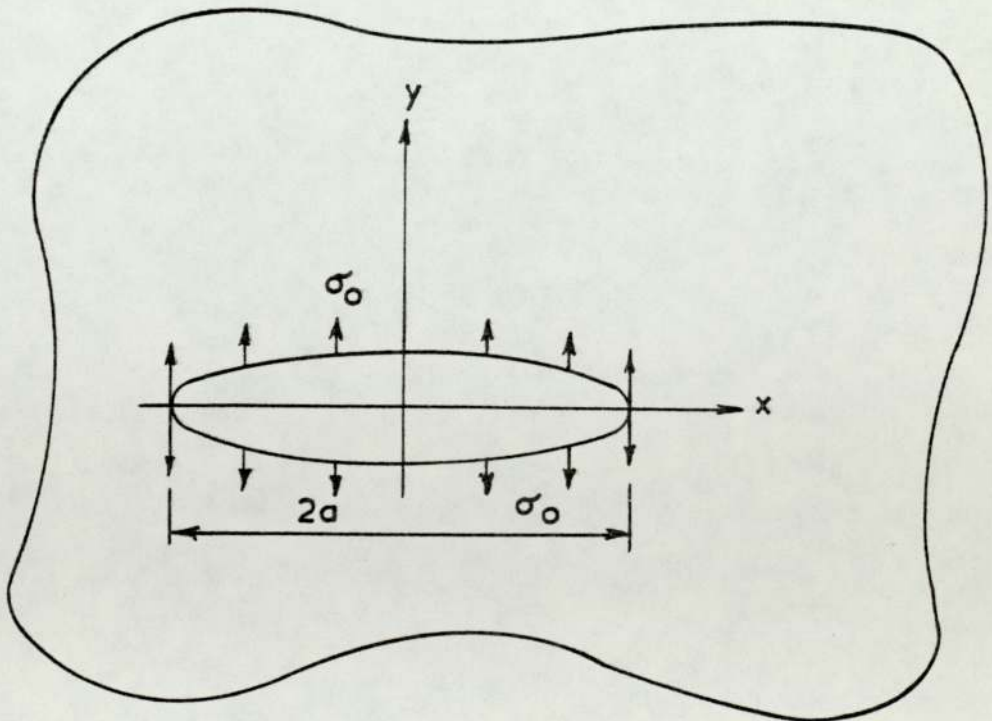


Fig. 4.2

4.3 Irwin's theory

Irwin observed three independent basic deformations which are sufficient to describe all modes of crack behaviour in the general state of elastic stress, [19]. Each one of these movements is associated with a particular stress field in the vicinity of the crack, and each stress field is characterized by a special factor. The three factors are called the stress intensity factors and they define the magnitude of the local stress field.

The three modes of fracture are the opening mode, the sliding mode, and the tearing mode; and the stress intensity factors associated with them are K_I , K_{II} , and K_{III} respectively. The movement of the upper and lower crack surfaces with respect to each other for each mode are shown in Fig. (4.3).

It could be shown that the mechanical energy released during incremental crack extension is independent of loading configuration and that the strain energy release rate is given by, [20]:

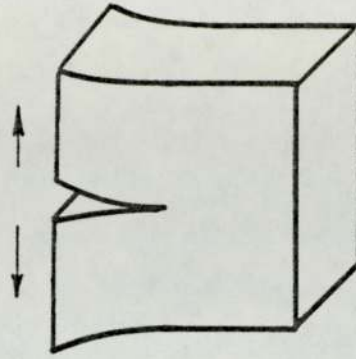
$$G = -\frac{\partial U}{\partial A} \quad (4.8)$$

where

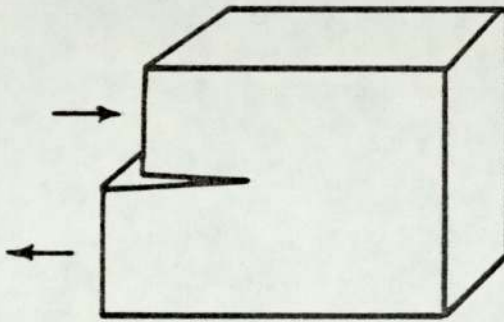
U = Strain potential energy stored
in the elastic medium.

A = Crack area.

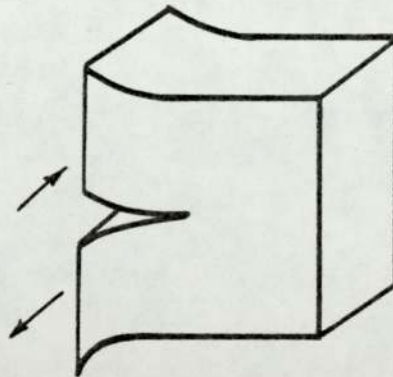
The stresses and displacements expressions near the crack tip in the plane problem are given by, [21]:



Opening



Sliding



Tearing

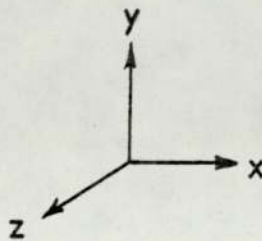


Fig. 4.3

$$\begin{aligned} {}_m\sigma_{ij} &= K_m r^{-\frac{1}{2}} {}_m f_{ij}(\theta) \quad i, j = 1, 2 \\ & \quad m = I, II, III \end{aligned} \quad (4.9)$$

$$\begin{aligned} {}_m u_i &= (K_m r^{\frac{1}{2}}/\mu) {}_m g_i(\theta) \quad i = 1, 2 \\ & \quad m = I, II, III \end{aligned} \quad (4.10)$$

where

K_m = stress intensity factor of the m^{th} mode
 ${}_m f_{ij}(\theta)$ & ${}_m g_i(\theta)$ = known functions given
for all modes and cases in Appendix (10.2)

Irwin, by using virtual work arguments arrived at the relationship between the stress intensity factor (K) and the strain energy release rate (G), [19].

$$G = \frac{\kappa+1}{8\mu} K^2 \quad (4.11)$$

Formally this relation may be generalized to cover the three modes, [20]:

$$G_I = \frac{\kappa+1}{8\mu} K_I^2 \quad (4.12)$$

$$G_{II} = \frac{\kappa+1}{8} K_{II}^2 \quad (4.13)$$

and

$$G_{III} = K_{III}^2/2\mu \quad (4.14)$$

However, under the second and third modes, the crack tends to extend in a non-planar fashion, hence a criterion for fracture based on critical values of (G_{II} & G_{III}) becomes difficult to justify, [22]. Therefore, the cases governed by this criterion are only those of Mode I.

Experiments on suitable specimens determine the stress intensity factor at the point of fracture (K_{IC}), which is the critical value for K_I , and is then regarded as a material property. A means for predicting the fracture behaviour of structures is then provided.

4.4 The strain energy density theory

In the general engineering problem, loading direction is not always perpendicular to cracks or flaws, and hence a mixed mode problem is present. A strain energy density criterion for fracture may be considered, according to which crack propagation occurs when a function of K_I , K_{II} , and K_{III} reaches a critical value, [23].

$$f(K_I, K_{II}, K_{III}) = f_{cr} \quad (4.15)$$

A single mode problem will be a special case of the general problem.

With reference to Fig. (4.4), the stress and displacement expressions in the immediate neighbourhood of the crack tip were given in equations (4.9 and 4.10), these expressions define the stresses and displacements in a small region surrounding the crack tip which is considered circular for convenience. As the radius of this region (R_c) is very small compared to (a) and (R), it can be assumed that the conditions of plane strain prevail within it, [24].

The cross sectional area of the element shown in Fig. (4.5) is:

$$dA = r \, d\theta \, dr \quad (4.16)$$

The volume of the toroid of cross section (dA) is:

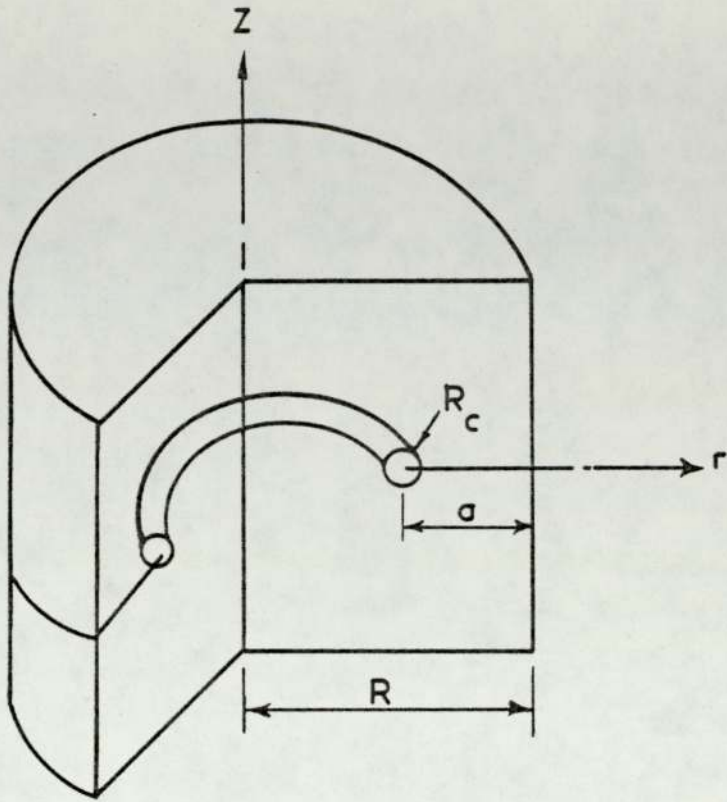


Fig. 4.4

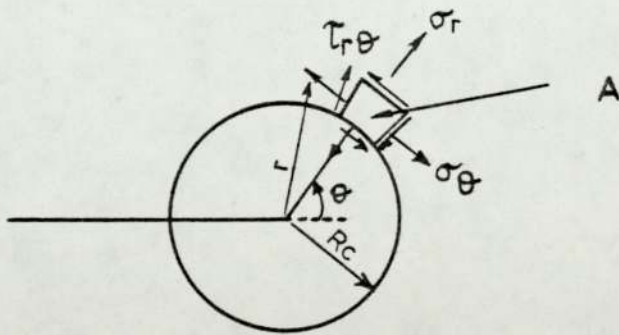


Fig. 4.5

$$dVol = 2\pi[(R-a)+rcos\theta]dA \quad \dagger \quad (4.17)$$

The strain energy stored in this volume is given by, [24]:

$$\begin{aligned} dW = & \frac{1}{2} \left[\sigma_r \frac{\partial u}{\partial r} + \sigma_\theta \left(\frac{u}{r} + \frac{1}{r} \frac{\partial v}{\partial \theta} \right) + \right. \\ & \left. \tau_{r\theta} \left(\frac{1}{r} \frac{\partial u}{\partial \theta} + \frac{\partial v}{\partial r} - \frac{v}{r} \right) \right] \\ & \times [2\pi[(R-a)+ rcos\theta]dA] \end{aligned} \quad (4.18)$$

The local strain energy density field can be expressed from equation (4.18) in quadratic form as:

$$\frac{dW}{dA} = \frac{1}{r} (a_{11}K_I^2 + 2a_{12}K_I K_{II} + a_{22}K_{II}^2 + a_{33}K_{III}^2) \quad (4.19)$$

where

$$a_{11} = \frac{2\pi[(R-a)+ rcos\theta]}{16\mu} \times [(3-4\nu-cos\theta)(1+cos\theta)] \quad (4.20)$$

$$a_{12} = \frac{2\pi[(R-a)+ rcos\theta]}{16\mu} \times [(2sin\theta)[cos\theta-(1-2\nu)]] \quad (4.21)$$

$$a_{22} = \frac{2\pi[(R-a)+ rcos\theta]}{16\mu} \times [4(1-\nu)(1-cos\theta)+(1+cos\theta)(3cos\theta-1)] \quad (4.22)$$

$$a_{33} = \frac{2\pi[(R-a)+ rcos\theta]}{4\mu} \quad (4.23)$$

(dW/dA) becomes larger as (r) is made smaller, reaching a limit at the boundary of the core region (r = R_c). In equations (4.20) to (4.23) the term[(R-a)+ rcosθ]will become[R-a + R_ccosθ]and as R_c is considered very small compared with R and a, this term can be approximated to(R-a)only. The intensity of the varying energy field along

† For a penny-shaped crack:

$$dVol = 2\pi[a+rcos\theta]dA$$

the periphery of the core is referred to as the strain energy density factor (S) and is given by:

$$S = a_{11}K_I^2 + 2a_{12}K_I K_{II} + a_{22}K_{II}^2 + a_{33}K_{III}^2 \quad (4.24)$$

where

$$a_{11} = [2\pi(R-a)/16\mu] \times [(3-4\nu-\cos\theta)(1+\cos\theta)] \quad (4.25)$$

$$a_{12} = [2\pi(R-a)/16\mu] \times [2\sin\theta[\cos\theta-(1-2\nu)]] \quad (4.26)$$

$$a_{22} = [2\pi(R-a)/16\mu] \times [4(1-\nu)(1-\cos\theta)+(1+\cos\theta)(3\cos\theta-1)] \quad (4.27)$$

$$a_{33} = [2\pi(R-a)/4\mu] \quad (4.28)$$

According to the strain energy density criterion for fracture, initial crack growth takes place in the direction along which the strain energy density factor possesses a minimum value ($\frac{\partial S}{\partial \theta} = 0$, at which $\theta = \theta_0$); and crack initiation occurs when it reaches a critical value ($S = S_{cr}$, for $\theta = \theta_0$), [25].

4.5 The crack tip plastic zone

4.5.1 Introduction

The material around the crack tip has been assumed so far to behave in a linear elastic manner. In practice most of the structural materials (especially metals) tend to exhibit a yield stress above which they deform plastically. This means that a plastic zone exists around the crack tip, and hence stress singularities cannot exist. A rough estimate of the size of the plastic zone is simple to make. With reference to Fig. (4.6), the magnitude of the elastic stress distribution ahead of the crack is plotted. Until a distance (r_p) the

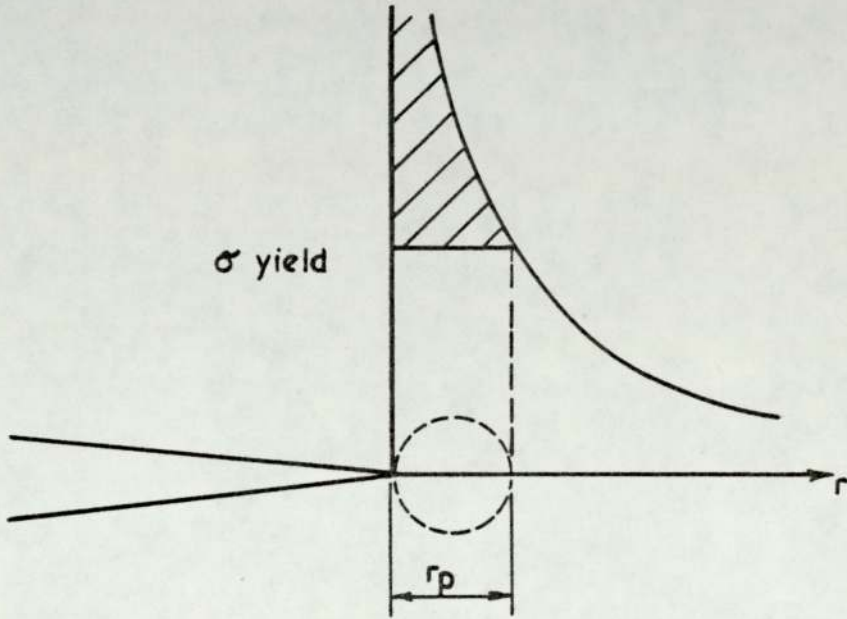


Fig.4.6

stress is higher than (σ_{yield}), and hence to first approximation, (r_p) is the size of the plastic zone and is given by, [26]:

$$r_p = \frac{\sigma_a^2}{2\pi \sigma_{yield}^2} \quad (4.29)$$

However, the actual plastic zone is larger than (r_p) and is not circular. A more accurate impression of its shape can be obtained by examining the yield condition for different angles at the crack tip, [27,28].

4.5.2 Extension of the Griffith concept

Irwin, [29] and Orwan, [30] working independently noted that the energy required for crack growth in metals is larger than the surface energy to create the new free surface. This was due to plastic deformation in front of the crack and hence energy is expended in the formation of a new plastic zone at the tip of the advancing crack. They suggested that the total potential energy becomes:

$$\Pi = U + \Omega + U_p \quad (4.30)$$

where

U = Elastic strain energy

Ω = Potential of applied loads

U_p = Plastic work done

And that the surface energy is modified to:

$$\gamma = \gamma_e + \gamma_p \quad (4.31)$$

where

γ_p = work done in plastic deformation.

They concluded however, that because of the moderately long range ($1/\sqrt{r}$) dependence of the stress field (equation 4.9), the elastic strain energy density is not highly localized about the crack tip. Hence, events within a small plastic crack tip zone are unlikely to cause significant variations in the system's strain energy; and the values of the strain energy release rate obtained by assuming linearity remain a good approximation.

4.5.3 Extension of Irwin's concept

Irwin argued that the occurrence of the plastic zone makes the crack behave as if it were longer than its physical size, [31]. The effective crack size is given by:

$$a_{\text{eff}} = a + \delta \quad (4.32)$$

where

$$\delta = \text{correction due to plasticity}$$

With reference to Fig. (4.7), the crack (a) is replaced by a longer one ($a + \delta$) and the elastic stress distribution is given at the tip of the effective crack length. (δ) must be large enough to carry the load lost by cutting the area (A) from the elastic stress distribution, hence:

$$\text{Area}(A) = \text{Area}(B) \quad (4.33)$$

By assuming (δ) to be very small compared to the crack size (a), and similar to (r_p) in equation (4.29), the distance (λ) is given by:

$$\lambda = \frac{\sigma^2(a+\delta)}{2\sigma_{\text{yield}}^2} \approx r_p \quad (4.34)$$

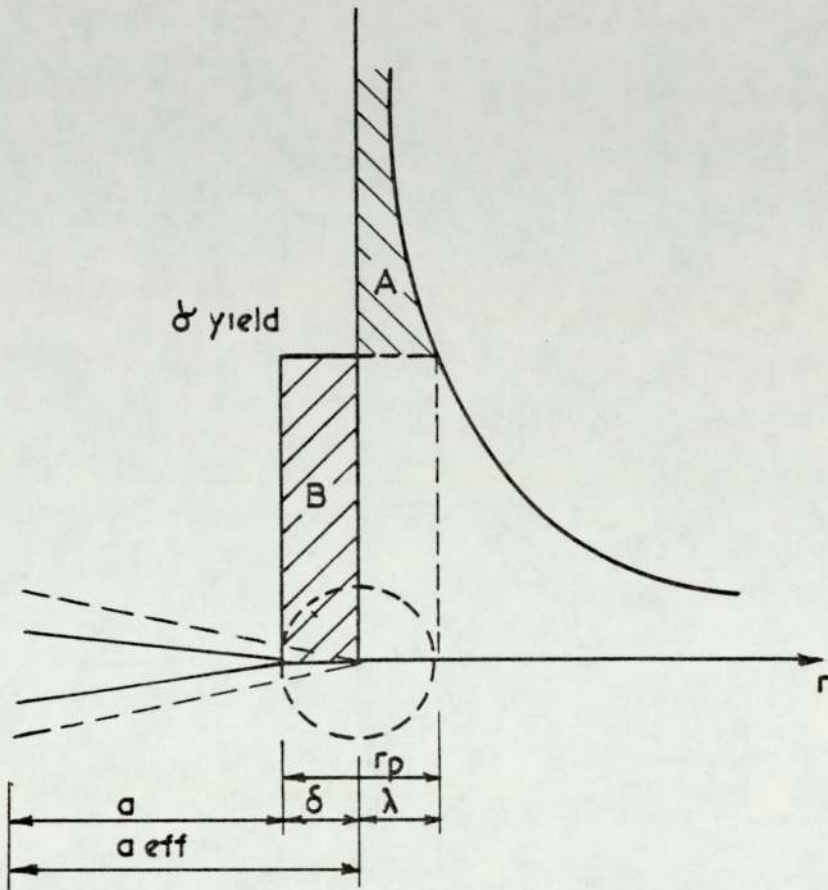


Fig. 4.7

The area(B) is given by:

$$B = \sigma_{\text{yield}} \delta \quad (4.35)$$

Therefore, the requirement $B=A$ will yield, [26]:

$$\delta = r_p \quad (4.36)$$

And

$$R_p = \lambda + \delta = 2r_p \quad (4.37)$$

which means that the size of the plastic zone (R_p) is twice the first estimate (r_p). Since $\delta = r_p$ and according to Irwin's argument, it follows that the crack behaves as if its length is $(a + r_p)$. The quantity (r_p) is known as Irwin's plastic zone correction.

CHAPTER 5

METHODS OF DETERMINING STRESS INTENSITY FACTORS

5.1 Introduction

The determination of the relevant stress intensity factors is required when adopting the stress intensity factor approach to crack problems. The prediction of failure loads and dangerous flaw sizes will depend ultimately on the accuracy of computing these factors. The methods of determining them may be broadly divided into experimental, analytical, and numerical classes. Following is a brief description of the important methods in each field.

5.2 Experimental methods

These methods may be divided into two main groups. The first involves direct measurements on a model, and the second uses a known relationship between the stress intensity factors and a measurable quantity.

5.2.1 Photoelasticity

This is an example of the first group of methods, and its advantage is that it is a well established method in stress analysis, especially in calculating stress concentration factors, hence its equipment and materials are readily available. However the representation of the crack must be done by a slit of a finite root radius, therefore equivalent crack lengths must be defined, [32].

The relationship between the maximum shear stress and the stress intensity factors K_I and K_{II} is given by

$$\tau_{\max} = \frac{1}{2\sqrt{2\pi r}} [(K_I \sin\theta + 2K_{II} \cos\theta)^2 + (K_{II} \sin\theta)^2]^{\frac{1}{2}} \quad (5.1)$$

where (r) and (θ) are polar coordinates centered at the crack tip.

This relationship is used in two different ways to determine the stress intensity factors.

The first is by measuring (τ_{\max}) on lines perpendicular to and through the crack tip, [33]:

$$\tau_{\max} = \frac{(K_I^2 + K_{II}^2)^{\frac{1}{2}}}{2\sqrt{2\pi r}} \quad (5.2)$$

and on a line parallel to the crack:

$$\tau_{\max} = \frac{K_{II}}{\sqrt{2\pi r}} \quad (5.3)$$

K_I , and K_{II} can then be determined from equations (5.2) and (5.3).

The second measures the angle (θ) , at which a tangent to the isochromatic fringes is perpendicular to the radius (r) , from the isochromatic fringe pattern near the tip [34, 35]:

$$\frac{\partial \tau_{\max}}{\partial \theta} = 0 \quad (5.4)$$

$$\left(\frac{K_{II}}{K_I}\right)^2 - \frac{4}{3} \left(\frac{K_{II}}{K_I}\right) \cot 2\theta - \frac{1}{3} = 0 \quad (5.5)$$

This relation together with equation (5.1) are used to solve for K_I and K_{II} .

5.2.2 Compliance

This method is an example of the second group of experimental techniques. The strain energy release rate (G) could be written in terms of the load (Q) and the rate of change of compliance (C), which is the reciprocal of stiffness, with respect to crack area (A) as,[36]:

$$G = \frac{Q^2}{2} \frac{dC}{dA} \quad (5.6)$$

For mode I fracture (equation 4.12)

$$\frac{\kappa+1}{8\mu} K_I^2 = \frac{Q^2}{2} \frac{dC}{dA} \quad (5.7)$$

therefore

$$K_I^2 = Q^2 \frac{4\mu}{\kappa+1} \frac{dC}{dA} \quad (5.8)$$

or

$$K_I = 2Q \left[\frac{\mu}{\kappa+1} \frac{dC}{dA} \right]^{\frac{1}{2}} \quad (5.9)$$

The compliance is measured for different crack lengths and a graph of compliance verses crack area is constructed. The slope of the graph is substituted in equation (5.9) to determine K_I . As discussed in section (4.3), this method is not suitable for calculating K_{II} or K_{III} .

5.2.3 Crack tip opening displacement measurement

The experimental methods discussed previously are related to purely elastic situations. If however, the plastic zone is large compared to the crack size, linear elastic fracture mechanics do not apply any longer. Under conditions of general yield, plastic flow is no longer contained, but the plastic zone spreads through the entire cracked section. Assuming negligible strain hardening, the stress at the crack tip hardly increases after general yield and the fracture

condition is reached upon the occurrence of a sufficiently large strain. A measure of the plastic strain at the crack tip is the crack tip opening displacement (CTOD). A criterion first proposed by Wells, [37] stated that fracture takes place when critical(CTOD) is exceeded.

The direct measurement of(CTOD)is difficult. With reference to Fig. (5.1) the(CTOD)can be determined indirectly from measuring the (COD)by, [26]:

$$COD = \frac{4\sigma}{E} \sqrt{a^2 - x^2 + \frac{E^2}{16\sigma^2} (CTOD)^2} \quad (5.10)$$

The equations for crack tip opening which are given in, [38], state that:

$$CTOD = \frac{G_I}{\lambda\sigma_{yield}} = \frac{K_I^2(1-\nu^2)}{E\lambda\sigma_{yield}} \text{ (plane strain)} \quad (5.11)$$

$$= \frac{K_I^2}{E\lambda\sigma_{yield}} \text{ (plane stress)} \quad (5.12)$$

Various values for (λ) have been reported in the literature and a review of them is available in, [26].

5.3 Analytical methods

These methods have the advantage of giving explicit expressions for the stress intensity factors. They have been applied only to a number of ideal (special) situations because of the difficulty of tackling the stress analysis boundary value problem for complex geometry and/or loading.

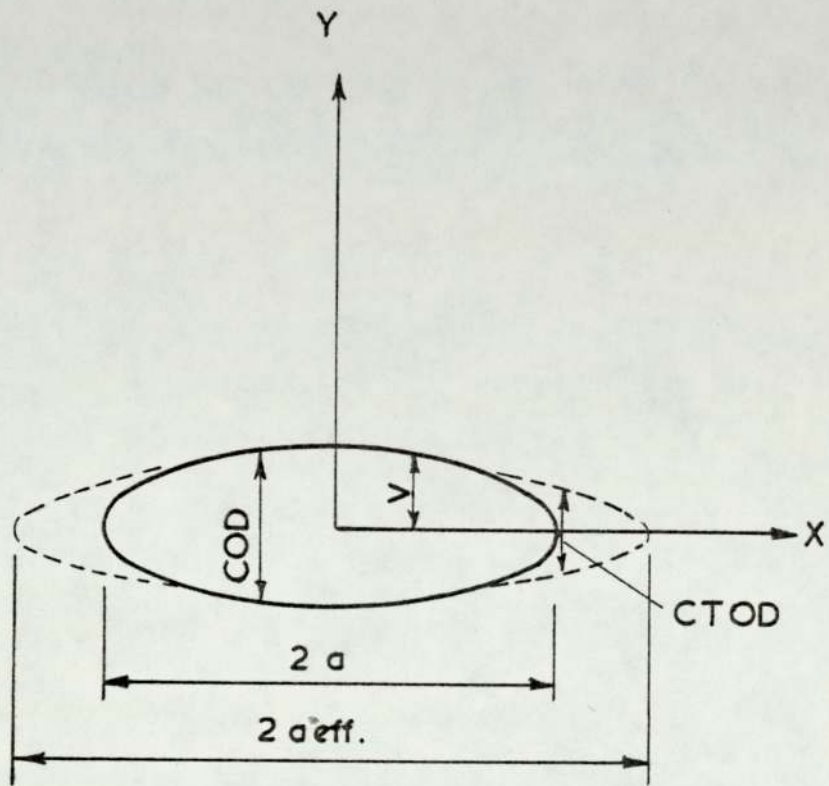


Fig. 5.1.

5.3.1 Westergaard stress function

Several complex forms of the Airy stress function (section 2.5) may be used for the solution of crack problems. Westergaard proposed the function, [39]:

$$\psi = R_e \bar{\bar{\phi}}(z) + \chi_2 I_m \bar{\phi}(z) \quad (5.13)$$

where $\bar{\phi}(z) = \int \phi(z) dz \quad (5.14)$

$$\bar{\bar{\phi}}(z) = \int \bar{\phi}(z) dz \quad (5.15)$$

It may be shown that the stress obtained from this stress function satisfy equilibrium, compatibility, and stress-strain relationships. By examining the boundary conditions of each specific problem and selecting the appropriate stress function ($\phi(z)$), the stress intensity factors of various but somewhat simple crack configurations can be obtained, [40,41].

5.3.2 Complex stress function

According to Muskhelishvili's approach, the Airy stress function (F) may be written in terms of two complex functions $\phi(z)$ and $\psi(z)$ as, [42]:

$$F = R_e [\bar{z}\phi(z) + \int \psi(z) dz] \quad (5.16)$$

Applying equations (2.20, 2.21 and 2.22) to equation (5.16) yields:

$$\sigma_x + \sigma_y = 4R_e [\phi'(z)] \quad (5.17)$$

And

$$\sigma_x - \sigma_y + 2i\tau_{xy} = 2[\bar{z}\phi''(z) + \psi'(z)] \quad (5.18)$$

From these relations it may be shown that:

$$K_I - iK_{II} = 2\sqrt{2\pi} \lim_{Z \rightarrow Z_1} (\sqrt{Z-Z_1} \phi'(z)) \quad (5.19)$$

This approach differs from Westergaard's since cracks may be mapped into holes using a mapping function of the form:

$$Z = \omega(\zeta) \quad (5.20)$$

The crack tip z in the z -plane will correspond to a point ζ in the ζ -plane in Fig. (5.2).

Equation (5.20) becomes

$$K_I - iK_{II} = 2\sqrt{2\pi} \lim_{\zeta \rightarrow \xi_1} \sqrt{\omega(\zeta) - \omega(\zeta_1)} \frac{\phi'(\zeta)}{\omega'(\zeta)} \quad (5.21)$$

Where

$$\begin{aligned} \phi(\zeta) &= \phi(\omega(\zeta)) \\ \phi'(\zeta) &= d\phi(\zeta)/d\zeta \end{aligned} \quad (5.22)$$

This method, which is known as conformal mapping, is discussed in detail in, [43].

5.4 Numerical methods

These methods, which involve certain approximations, are unavoidable for the solution of problems with complicated shapes and loadings since solutions by analytical methods are not possible.

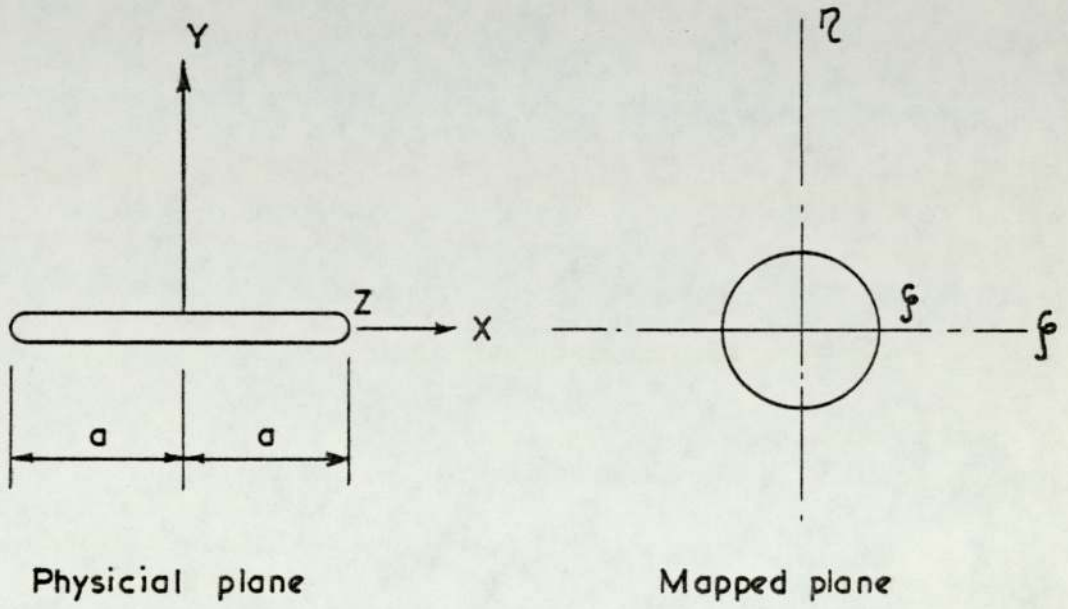


Fig. 5.2

5.4.1 Boundary collocation

This method consists in finding certain coefficients of the elastic crack solution by satisfying the boundary conditions at only a finite number of points along the boundary of the body. The process yields a set of equations

$$\sum_{j=1}^m a_{ij} X_j = R_i, \quad i=1,2,\dots,n \quad (5.23)$$

where

$$m \leq n$$

The problem is reduced to solving n linear simultaneous equations for the m unknowns (X_j). The stress function may be expressed in a series form or as a series representation of a complex function. It must be noted that there is no general proof that collocation solutions will converge by increasing the number of boundary points as that may result in making the set of linear simultaneous equations ill-conditioned. The accuracy of the solution must depend largely on the experience and judgement of the analyst, [44].

A) The Williams' stress function:

This is an Airy stress function which satisfies the conditions that the normal and shearing stresses are zero along the crack surface, and it is convenient to write it as, [45]:

$$\chi = \chi_e + \chi_o \quad (5.24)$$

where

$$\chi_e = \text{even terms}$$

$$\chi_o = \text{odd terms}$$

$$\begin{aligned} \chi_e = & \sum_{n=1}^{\infty} \{ (-1)^{n-1} A_{2n-1} r^{n+\frac{1}{2}} [-\cos(n - \frac{3}{2})\theta + \\ & \frac{2n-3}{2n+1} \cos(n + \frac{1}{2})\theta] \\ & + (-1)^n A_{2n} r^{n+1} [-\cos(n-1)\theta + \cos(n+1)\theta] \} \end{aligned} \quad (5.25)$$

$$\begin{aligned} \chi_o = & \sum_{n=1}^{\infty} \{ (-1)^{n-1} B_{2n-1} r^{n+\frac{1}{2}} [\sin(n - \frac{3}{2})\theta - \sin(n+\frac{1}{2})\theta] \\ & + (-1)^n B_{2n} r^{n+1} [-\sin(n-1)\theta + \frac{n-1}{n+1} \sin(n+1)\theta] \} \end{aligned} \quad (5.26)$$

It may be shown that the stress intensity factors are, [46]:

$$K_I = -A_1 \sqrt{2\pi} \quad (5.27)$$

$$K_{II} = B_1 \sqrt{2\pi} \quad (5.28)$$

The solution of the problem is obtained by taking the following steps:

- a) The determination of the stress function (F) for the uncracked configuration.
- b) Evaluating (F) and $(\frac{\partial F}{\partial n})$ at a number of points around the boundary.
- c) Substituting the value of (F) and $(\frac{\partial F}{\partial n})$ at each point into (χ) and $(\frac{\partial \chi}{\partial n})$ to yield a set of linear simultaneous equations whose solution gives (A_1) and (B_1)

The stress intensity factors for a number of edge crack specimen have been obtained using this method, [47, 48].

B) Complex stress function.

It has been shown how Muskhelishvili writes a stress function in terms of two complex functions $\phi(z)$ and $\psi(z)$ (section 5.3.2). The $\phi'(z)$ and $\psi(z)$ may be expanded in series which automatically satisfies the boundary conditions on the crack:

$$\phi'(z) = \frac{1}{\sqrt{z^2 - a^2}} \sum_{n=0}^{\infty} C_n z^n + \sum_{n=0}^{\infty} D_n z^n \quad (5.29)$$

$$\psi(z) = \frac{1}{\sqrt{z^2 - a^2}} \sum_{n=0}^{\infty} C_n z^n - \sum_{n=0}^{\infty} D_n z^n \quad (5.30)$$

A finite number of coefficients C_n and D_n are obtained by matching the prescribed stresses at discrete points around the remaining part of the boundary, [46].

Results of many problems using this method involving mode I, II, and III deformation, interaction between cracks, and cracks in stiffened sheets are reported in, [49].

5.4.2 Stress concentrations

With reference to Fig. (5.3), the stress at the apex of the major axis of the ellipse (σ_m) is given by:

$$\sigma_m = \sigma \left(1 + 2\frac{a}{b}\right) \quad (5.31)$$

The crack configuration may be approximated to a narrow elliptical cavity having a radius of curvature:

$$\rho = \frac{b^2}{a} \quad , \quad \rho \ll a \quad (5.32)$$

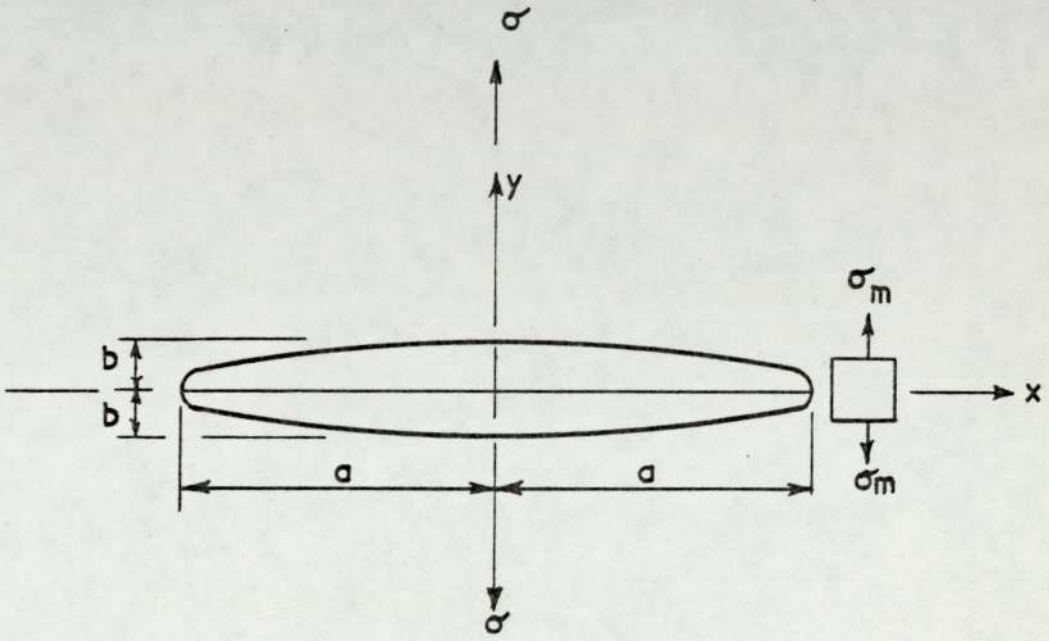


Fig. 5.3

Equation (5.31) may be rewritten as:

$$\sigma_m = \sigma(1 + 2\sqrt{\frac{a}{\rho}}) \approx 2\sigma\sqrt{\frac{a}{\rho}} \quad (5.33)$$

or

$$\sigma\sqrt{a} = \frac{1}{2} \sigma_m \sqrt{\rho} \quad (5.34)$$

The stress intensity factors may be obtained from the limiting values of maximum stresses at the base of a notch whose radius of curvature is allowed to vanish

$$K_I = \frac{\sqrt{\pi}}{2} \lim_{\rho \rightarrow 0} (\sigma_m \sqrt{\rho}) \quad (5.35)$$

If τ_m is the maximum value of τ_{xy} then:

$$K_{II} = \sqrt{\pi} \lim_{\rho \rightarrow 0} (\tau_m \sqrt{\rho}) \quad (5.36)$$

And if τ'_m is the maximum value of τ_{zy} then:

$$K_{III} = \sqrt{\pi} \lim_{\rho \rightarrow 0} (\tau'_m \sqrt{\rho}) \quad (5.37)$$

Although relations (5.35), (5.36) and (5.37) are exact, those used for σ_m and τ_m themselves are approximate.

The values of σ_m for a variety of notches can be found in the work of Neuber, [50].

5.4.3 Green's function

The stress intensity factors for crack problems involving arbitrary surface tractions may be found from those of a known problem.

With reference to Fig. (5.4) the stress intensity factors for a crack having concentrated normal and tangential forces (P) and (Q) is given by, [51]:

$$K = K_I - i K_{II} = \frac{Q+iP}{2\pi\sqrt{a}} \left[\frac{\kappa-1}{\kappa+1} + \sqrt{\frac{b+a}{b-a}} \right] \quad (5.38)$$

Equations (5.38) may be used as the fundamental Green's function for generating a solution to a crack problem having surface tractions $\sigma_y(x,0)$, and $\tau_{xy}(x,0)$. Letting:

$$P = \sigma_y dx \quad (5.39)$$

$$Q = \tau_{xy} dx \quad (5.40)$$

and integrating from ($X = -a$ to a), the K_I and K_{II} expressions for a crack subject to arbitrary loads on the upper surface* are:

$$K_I = \frac{1}{2\pi\sqrt{a}} \int_{-a}^a \sigma_y \sqrt{\frac{a+x}{a-x}} dx + \frac{1}{2\pi\sqrt{a}} \int_{-a}^a \tau_{xy} dx \quad (5.41)$$

$$K_{II} = \frac{1}{2\pi\sqrt{a}} \left(\frac{\kappa-1}{\kappa+1} \right) \int_{-a}^a \sigma_y dx + \frac{1}{2\pi\sqrt{a}} \int_{-a}^a \tau_{xy} \sqrt{\frac{a+x}{a-x}} dx \quad (5.42)$$

If the loads are balanced on the upper and lower crack surfaces, then equations (5.41) and (5.42) reduce to:

$$K = K_I - iK_{II} = \frac{1}{\pi\sqrt{a}} \int_{-a}^a [\sigma_y - i\tau_{xy}] \sqrt{\frac{a+x}{a-x}} dx \quad (5.43)$$

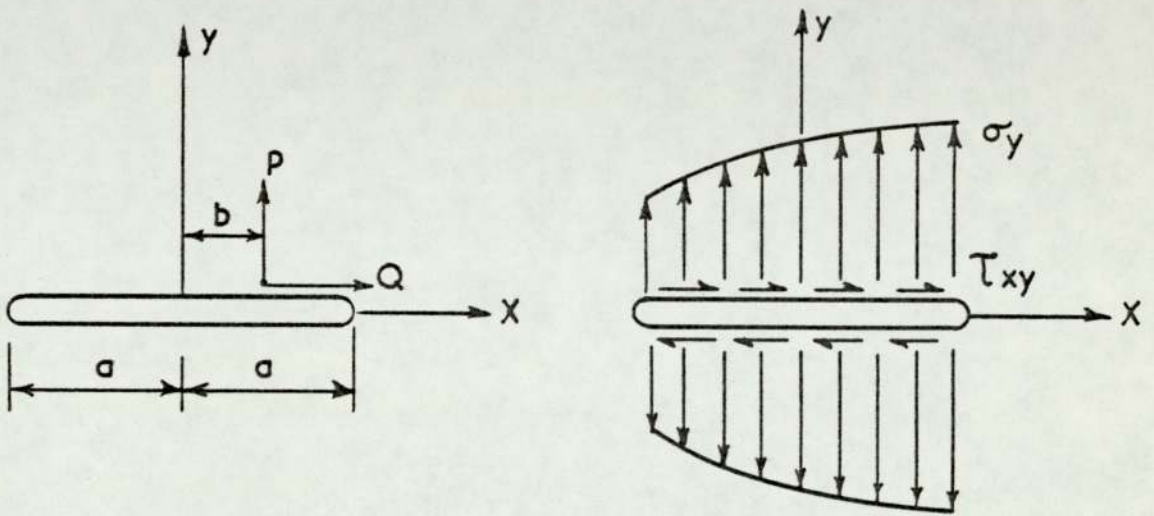


Fig. 5.4

Sih, [43], and Erdogan, [52], applied this approach to obtain a solution to the problem of a plate with a finite crack with concentrated loads (P) and (Q) and a concentrated couple (M) applied at an arbitrary point.

5.4.4 Integral transforms and dislocation models

The elastic problem is considered as a mixed boundary value problem and solved using standard transform techniques. These methods reduce to the solution of an integral equation of the form:

$$\int_{-a}^a K(S,X)q(S)dS = L(x) \quad (5.44)$$

where

$L(x)$ = The known stress along the crack site
in the uncracked body.

$K(S,X)$ = known Kernel.

$q(S)$ = unknown function

The function $q(S)$ is related to the derivatives of the relative displacement $v(X,0)$ of the crack face by, [46]:

$$\frac{E}{2(1-\nu^2)} \frac{\partial v(X,0)}{\partial x} = q(x) \quad (5.45)$$

Thus, knowing $q(x)$ enables the stress intensity factor to be found by:

$$K_a = \lim_{x \rightarrow a} \left[\sqrt{2\pi(a-x)} \frac{E}{2(1-\nu^2)} \frac{\partial v(X,0)}{\partial x} \right] \quad (5.46)$$

The problem of a star-shaped crack subjected to internal pressure was solved by this method using a series expansion of (q), [53]. Radial cracks in discs and cylinders were also solved, [54].

Another class of problems in which the crack is represented by a continuous distribution of dislocation singularities, Fig. (5.5), leads to an integral equation similar to equation (5.44).

The density of this distribution ($D_i(S)$) is related to the Berger's vector (b_i) as:

$$b_i = \int_S D_i(S) dS \quad (5.47)$$

where

S = path around dislocation distribution.

The crack configuration in Fig. (5.5a) can be modelled by a continuous array of dislocations of density ($D_y(S)$) lying along ($y = 0$), $|X| < a$, having Berger's vector:

$$b_y = \int_S D_y(S) dS \quad (5.48)$$

These dislocations cause stresses along ($y = 0$) given by, [46]:

$$\sigma_y = -\sigma_x = \frac{2\mu}{\pi(1+\kappa)} \int_{-a}^a \frac{D_y(S) dS}{X - S} \quad (5.49)$$

And

$$\tau_{xy} = 0 \quad (5.50)$$

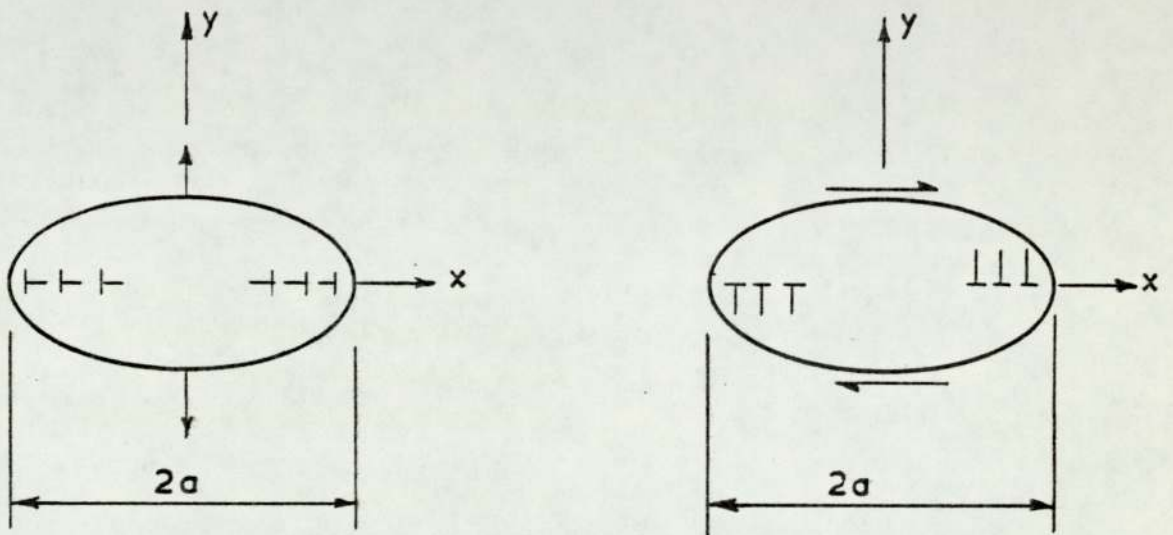


Fig. 5.5.

Applying the boundary conditions on ($y = 0$), and in order that the displacements be single valued in the crack problem, the net Berger's vector must be zero, i.e.

$$\int_{-a}^a D_y(S) dS = 0 \quad (5.51)$$

It may be shown that the stress intensity factor can be determined from, [46]:

$$K_I = \frac{2\sqrt{2\pi\mu}}{(1+\kappa)} \lim_{x \rightarrow a} [(a-x)^{\frac{1}{2}} D_y(x)] \quad (5.52)$$

5.4.5 Force-displacement matching

This method is used in configurations having different materials. The forces and displacements are matched along the boundaries joining the materials. It has been used to determine the stress intensity factors for cracks in stiffened sheets, [55].

5.4.6 Alternating method

This method uses existing solutions of simple crack problems to construct approximate ones to a more complicated range. This is done by superposing the known solutions of component problems, each satisfying boundary conditions on a portion of the boundary. Its application to two dimensional problems has been made by Irwin, [56].

5.4.7 Assymptotic approximation

In some crack problems, solutions can not be found for arbitrary values of the physical parameters, although they are available for specific values of them. This method relies on solutions

at both ends of the range of certain parameters such that a satisfactory interpolation can be made for intermediate values of them. This method has been used to find the stress intensity factors for finite width sheets with cracks under tension, [57].

5.4.8 The finite element method

All the methods for computing stress intensity factors previously described are appropriate for problems with simple geometries and loadings. The finite element method, described in chapter (3), is capable of handling the more complicated problems and can easily model arbitrary crack configurations. It is convenient to divide the application of the method to fracture problems into two groups. The first relies on standard finite element programs and makes no special provision for the crack tip singularity. In the second, special crack tip finite elements are used, which incorporate the solutions for the singular stresses and displacements at the tip of the crack.

A) Standard finite element computer methods

i) The stress method: As this method does not make provisions for the crack tip singularity, a fine mesh is needed to represent the stress field near the crack tip. The appropriate nodal point stresses obtained by the finite element method are substituted into equations (4.9), and the relevant stress intensity factor is obtained by the proper choice of the stress components (σ_{ij}) and the angle (θ). For example the mode I stress intensity factor may be determined from σ_y at a small distance r from the crack tip and an angle $\theta = 0$. However, results obtained by this method were found to be unsatisfactory, [24].

ii) The displacement method: As the primary unknowns in the finite element displacement method are the displacements, while the stresses are obtained by differentiating them and applying the relevant elasticity relations; it seemed that better results may be achieved by using the displacements obtained from the finite element method. The relevant stress intensity factor may be obtained by substituting the appropriate displacement components (u_j) and angle (θ) in equations (4.10). If these equations and the displacement components (u_j) are exact, a unique value of the stress intensity factor is obtained. However, neither condition is fulfilled, especially that regarding the values of the displacement components, which is due to the fact that no provision was made for the crack tip singularity. Therefore, instead of attempting to find single exact values for the stress intensity factors, several values are evaluated at a number of points close to the crack tip and are plotted versus their distance from the tip of the crack. Such a curve becomes linear some distance away from the crack tip, [24], and by extrapolating the linear portion to the tip point, a better estimate of the stress intensity factor is obtained.

It was shown that the distance from the crack tip at which the curve becomes linear decreases as the fineness of the mesh is increased; and in the specific problem considered by Chan, [64], a (K_I) value (9%) below the theoretical obtained from a coarse mesh was improved to (5%) below with a fine one. This method was used to solve mode I problems by Chan et al., [57], mixed mode I and II problems by Kobayashi et al., [58], and three dimensional problems by Miamoto et al., [59].

The method's accuracy was improved further by adding a second term to the leading one in the series expansion of the crack tip displacement formulae (4.10). This led to the extension of the range of the validity of the formulae, and by calculating (u_r) and (u_θ) displacements at a pair of adjacent points (Appendix 10.2) values of K_I , K_{II} , α_1 , and α_2 were obtained. The values for the stress intensity factors obtained for several points are plotted versus their distance from the crack tip, and were found to exhibit a maximum. This maximum value was found to be a better estimate of the stress intensity factor than that obtained by extrapolating the linear portion of the curve. In the specific problem solved by Oglesby and Lomacky, [24], results were of the order of (1 to 5%) below the theoretical value.

iii) The energy methods: It may prove to be more advantageous to calculate the stress intensity factors from energy estimates near the crack tip. This way the accuracy of the solution is less dependent on the values of displacement or stresses obtained near the tip of the crack and a coarser mesh will yield equivalent accuracy. Also there will be no need to extrapolate (K_I) curves for which there is no theoretical justification. The energy approach may be implemented by using the following methods:

a) Compliance calibration: It has been shown that the energy release rate (G) is related to the compliance (C) by equation (5.6). The compliance of cracked structure which is the reciprocal of the stiffness is calculated by finite elements for different crack lengths, [60], the energy release rate dU/dA is then calculated, from which (K_I) is evaluated using equation

(5.7). The (K_I) values for a centre crack strip problem was evaluated using this method, [61] and its value was found to be within (5%) of that obtained by boundary collocation.

b) Crack closure work: This method calculates the amount of work required to close successive nodal intervals along a crack,[62]. The displacement of the first nodal point along the crack face from the tip is calculated with all the boundary conditions applied. By applying unit loads to this node, the work required to close the crack over nodal intervals can be calculated. A graph of the closure work done versus crack area is plotted and the strain energy release rate (G) is determined from its slope: the (K_I) value is then determined from (G) as in the method described previously in (a).

c) The J-integral method: Rice defined a line integral (J), along an arbitrary contour (r) surrounding a crack tip, Fig.(5.6) by, [63]

$$J = \int_r (U dy - T \frac{\partial u}{\partial x} dS) \quad (5.53)$$

where

U = Strain energy density

T = Traction vector along the outward normal to the contour.

u = Displacement vector on an arc element (dS) along arc (S).

Having determined U, T, and $\frac{\partial u}{\partial x}$ along the chosen contour by finite element, the integral is evaluated in a counter-clockwise direction

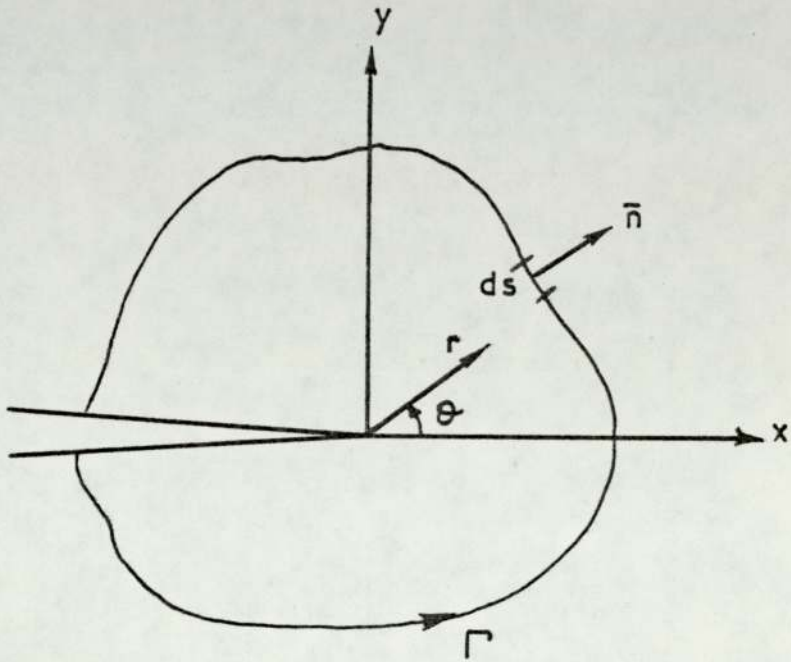


Fig. 5.6

starting at the lower crack surface. He has shown also that (J) is equal to the strain energy release rate (G), hence (K_I) can be evaluated from it. In a mixed mode situation, it was also shown that:

$$J = \frac{1-\nu^2}{E} (K_I^2 + K_{II}^2), \text{ plane strain} \quad (5.54)$$

$$J = \frac{1}{E} (K_I^2 + K_{II}^2), \text{ plane stress} \quad (5.55)$$

Therefore the sums of the squares of (K_I) and (K_{II}) can be determined, but not K_I and K_{II} separately.

The (K_I) value for a compact tension specimen was evaluated by this method, [64], and was found to be (35%) lower than that obtained by boundary collocation.

B) Special crack tip finite element methods

All the methods discussed previously use a standard finite element program, with a fine mesh near the crack tip, to determine the stress intensity factors; and with the energy methods requiring less mesh fineness. However, the major disadvantage to these methods, is that their monotonic convergence to the true solution is not assured. It has been shown that errors adjacent to points of singularity are of the same order as those of the remainder of the elements, hence their effects extend over a finite region around these singularities, [65].

The large computer storage required, together with the lack of accuracy due to extrapolation, plotting, and bad convergence justifies the use of another approach with specially constructed elements at the crack tip in which the singularity condition is built into their displacement functions.

i) Byskov's element: Byskov proposed a polygonal element containing a linear crack, [66]. The specific shape applied was triangular with a straight crack extending from one vertex to the interior, Fig.(5.7). The displacement terms used for the element were obtained from Muskhelishvili's complex stress function in terms of a power series, [42]. The number of terms of the series retained is equal to the number of degrees of freedom of the element. Although equilibrium conditions are satisfied within the element and the crack face is traction free, incompatibility along the interface with neighbouring elements exists. There can be no control over the influence of this incompatibility, as only three conventional elements can surround the Byskov's one. The stiffness matrix of the element is determined numerically and values of K_I , K_{II} and nodal points displacements are obtained directly.

ii) Tracey's element: In Tracey's approach, the near tip displacement field is represented with the same accuracy as that away from the tip, by using quadrilateral isoparametric elements, [67]. The crack tip is surrounded by triangular elements, Fig. (5.8), and to achieve this two of the nodes of the quadrilateral element are made to coincide Fig.(5.9). The elements in the (x,y) coordinates are mapped to a square in the (ζ,η) auxiliary coordinates system, Fig.(5.10), by the transformation:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} X_A & X_B & X_C & X_D \\ Y_A & Y_B & Y_C & Y_D \end{bmatrix} \begin{Bmatrix} \zeta(1-\eta) \\ \zeta\eta \\ (1-\zeta)\eta \\ (1-\zeta)(1-\eta) \end{Bmatrix} \quad (5.56)$$

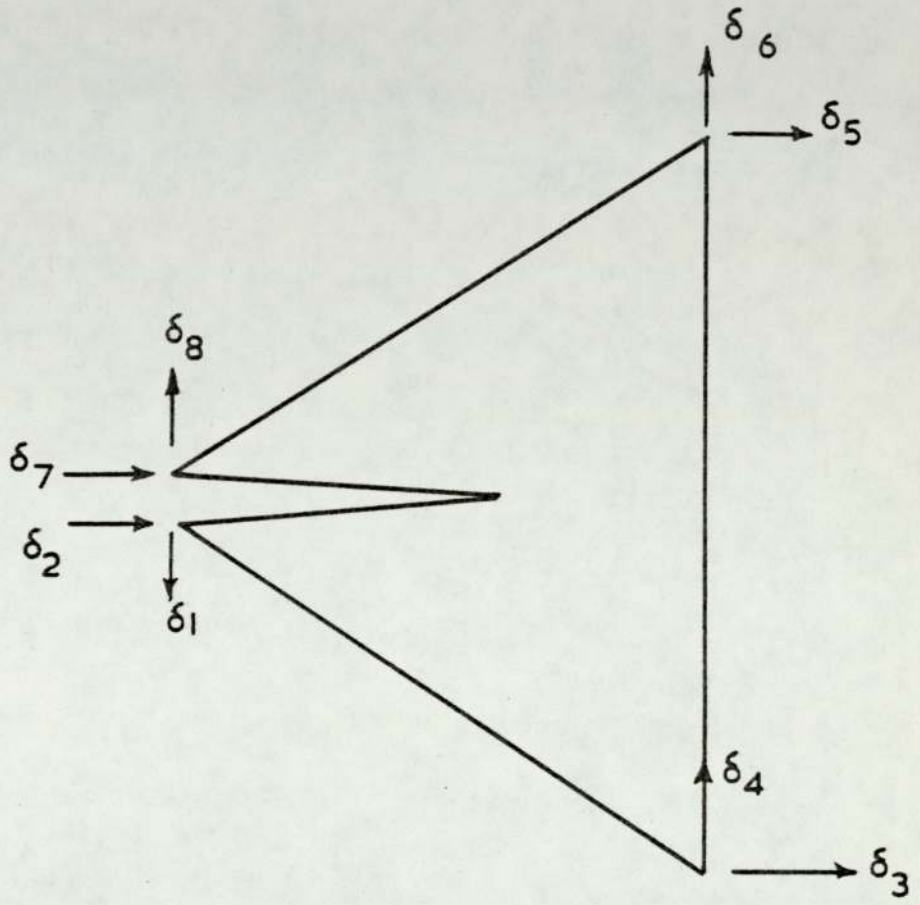


Fig . 5.7.

Thus along each edge of the element one of the auxiliary coordinates is constant and the other varies linearly with respect to (X) and (Y). The displacement functions within an element have the form:

$$u_x(\zeta, \eta) = \alpha_{11} + \alpha_{12}\zeta + \alpha_{13}\eta + \alpha_{14}\zeta\eta \quad (5.57)$$

$$v_y(\zeta, \eta) = \alpha_{21} + \alpha_{22}\zeta + \alpha_{23}\eta + \alpha_{24}\zeta\eta \quad (5.58)$$

Where the constants (α_{ij}) are expressible in terms of nodal displacements.

These displacement functions satisfy the compatibility conditions and are continuous across element boundaries.

The quadrilateral elements are used everywhere except very near the crack tip where the special triangular element is employed and whose displacement functions are of the form:

$$u_x(\zeta, \eta) = \beta_{11} + \beta_{12}\zeta^{\frac{1}{2}} + \beta_{13}\zeta^{\frac{1}{2}}\eta \quad (5.59)$$

$$v_y(\zeta, \eta) = \beta_{21} + \beta_{22}\zeta^{\frac{1}{2}} + \beta_{23}\zeta^{\frac{1}{2}}\eta \quad (5.60)$$

Where (β_{ij}) are determined from nodal values of displacement components. The line segment, $\zeta = 0$, in the auxiliary plane corresponds to the crack tip.

The displacement functions given in equations (5.59) and (5.60) result in compatible displacements between adjacent triangular elements and between them and the quadrilateral ones also.

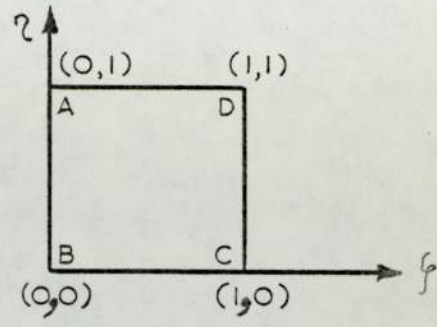
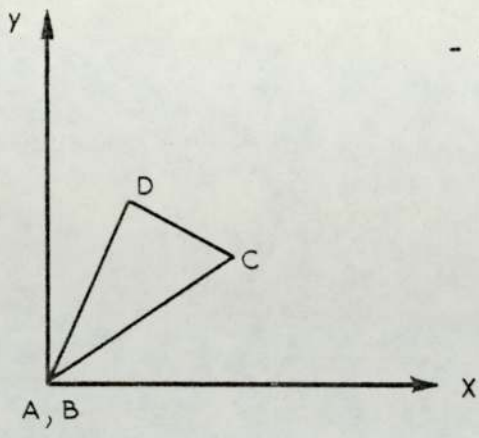


Fig. 5.8

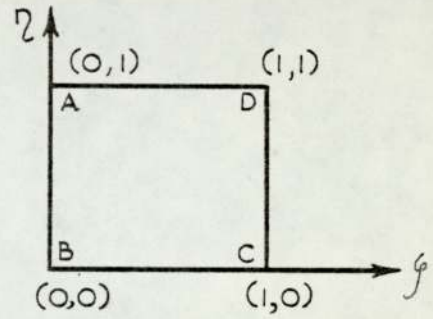
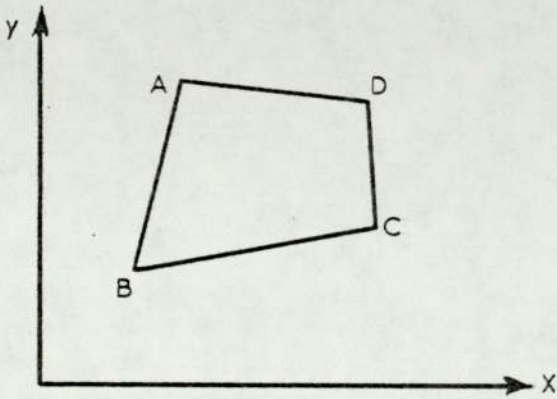


Fig. 5.9.

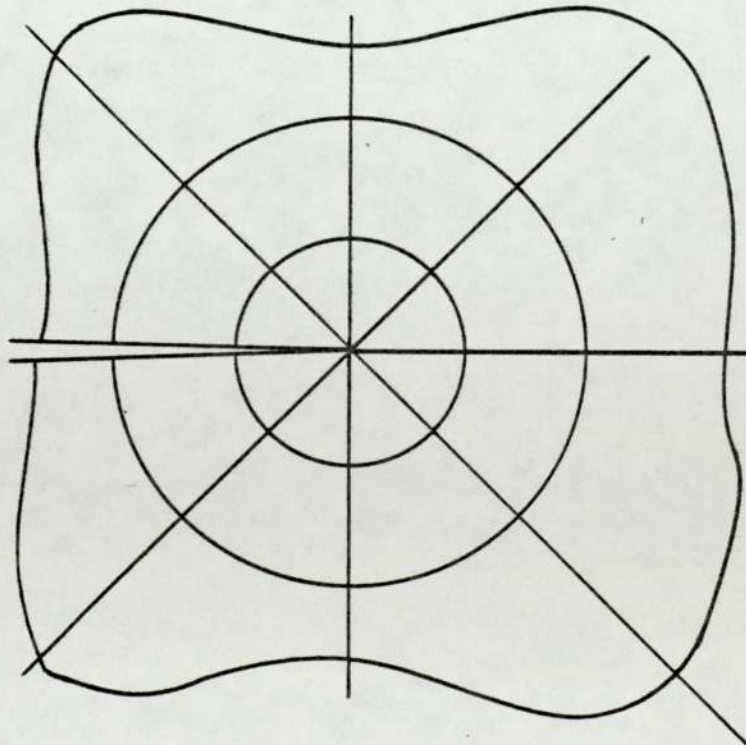


Fig. 5.10

The nodal point displacements and stresses are obtained from the finite element solution and from them the values of K_I and K_{II} are evaluated as described previously in the displacement and stress methods. The problem of a double notched edge plate under tension was solved by Tracey using this method. He averaged the values obtained for (K_I) from the displacements of the first ring of nodes surrounding the crack tip, and the results were within (4%) of the exact value for (248) degrees of freedom and (2.9%) for (548) degrees of freedom.

iii) The singular isoparametric element: Elements which embody the inverse square root singularity were developed independently by Henshell, [68], and Barsoum, [69]. Their shape functions $N_i(\zeta, \eta)$ are polynomials and hence, $(\frac{\partial N_i}{\partial \zeta})$ and $(\frac{\partial N_i}{\partial \eta})$ are non-singular. The strains may be written as:

$$\{\epsilon\} = [J]^{-1} [B(\zeta, \eta)] \{u_i\} \quad (5.61)$$

where

$[B(\zeta, \eta)]$ = The strain displacement matrix.

$$\{u_i\} = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} = \text{nodal displacements}$$

$[J]$ = Jacobean matrix defined in section (3.4)

The singularity can be achieved by making $[J]$ singular at the crack tip, this implies that $(\det.[J])$ vanishes at the crack tip:

$$\det[J] = \frac{\partial(x, y)}{\partial(\zeta, \eta)} \quad (5.62)$$

For a six node isoparametric triangular element, Fig.(5.11), the singularity is investigated along the x-axis, ($\eta = 0$).

Choosing $X_1 = 0$, $X_2 = X_5 = X_3 = L$, $X_4 = X_6 = \frac{L}{4}$,

It can be shown that, [69]:

$$X = (\zeta^2 + 2\zeta + 1) \frac{L}{4} \quad (5.63)$$

Therefore

$$\zeta = [-1 + 2\sqrt{\frac{X}{L}}] \quad (5.64)$$

The term $\frac{\partial X}{\partial \zeta}$ in the Jacobean is:

$$\frac{\partial X}{\partial \zeta} = \frac{L}{2} (1 + \zeta) = \sqrt{\frac{X}{L}} \quad (5.65)$$

which makes the Jacobean singular at $(X = 0, \zeta = -1)$.

Although the strains and stresses in these elements are singular, their total strain energy is finite and hence their stiffness. These elements satisfy compatibility, continuity, and convergence requirements. They also satisfy constant strain and rigid body motion conditions.

A ring of these elements surrounds the crack tip and the stress intensity factors are obtained in the same manner as that followed by Tracey in the previous section. The same problem of a double edge notched plate under tension was solved by this method, [69], and the results for (K_I) were within 1% of the exact value for (700) degrees of freedom.

iv) The hybrid element: Tong, Pian and Lasry, used the hybrid element concept and the complex variable technique to construct a special super-element at the crack tip, Fig. (5.12). This element

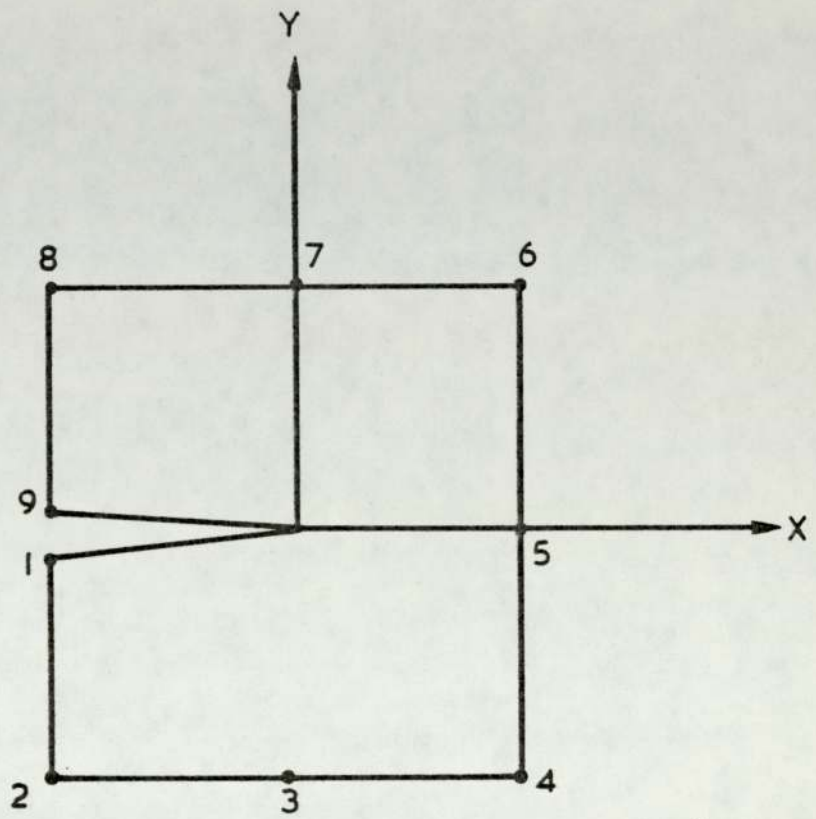


Fig . 5. 12

is used jointly with ordinary elements elsewhere. The super-element incorporates the crack tip singularity, and good results were obtained from a coarse mesh near the tip, [70].

A hybrid functional was defined for a plane problem with tractions (\bar{T}_i) over boundary (S_σ) and displacement (\bar{u}_i) over boundary (S_u) as:

$$\Pi_m = \sum_m \pi_m \tag{5.66}$$

where

$$\begin{aligned} \pi_m = & \int_{\partial A_m} (\tilde{u}_i - u_i) T_i ds - \int_{(S_\sigma)_m} \tilde{u}_i \bar{T}_i dS \\ & + \int_{A_m} [\sigma_{ij}(u_{i,j} + u_{j,i}) - S_{ijkl} \sigma_{ij} \sigma_{kl}] dA \end{aligned} \tag{5.67}$$

where

A_m = area of the m^{th} element.

∂A_m = boundary of A_m

S_{ijkl} = compliance coefficients

u_i and σ_{ij} are defined in A_m

T_i and \tilde{u}_i are defined over ∂A_m

$\tilde{u}_i = \bar{u}_i$ over (S_u)

$(S_\sigma)_m$ element surface where tractions prescribed

Using Muskhelishvili's complex stress function, and to account for singularities by choosing proper displacements and stresses for the crack element, the following mapping function is introduced:

$$Z = W(\zeta) = \zeta^2 \tag{5.68}$$

or

$$\zeta = Z^{\frac{1}{2}} \tag{5.69}$$

Writing the two complex functions required for Mushkelishvili's method as:

$$\phi(\zeta) = \sum_{j=1}^N b_j \zeta^j \quad (5.70)$$

$$\psi(\zeta) = \sum_{j=1}^N [\overline{b_j}(-1)^j + \frac{j}{2} b_j] \zeta^j \quad (5.71)$$

It may be shown that, [70]:

$$K_I = \sqrt{2(\beta_1)} \quad (5.72)$$

$$K_{II} = \sqrt{2(\beta_{N+1})} \quad (5.73)$$

This method has been used to solve the problem of plate with edge crack and central crack, and the results for the stress intensity factors obtained directly by the relation between the displacements and (β) stated in, [70], gave good accuracy for a small number of degrees of freedom.

v) Blackburn element: Blackburn developed a triangular element with the (\sqrt{r}) singularity built into its displacement function which is given by, [71]:

$$u(\zeta, \eta) = b_1 + \frac{(b_2\zeta + b_3\eta)}{\sqrt{\zeta + \eta}} \quad (5.74)$$

Which is equivalent to a constant strain triangular element and:

$$u(\zeta, \eta) = c_1 + \frac{(c_2\zeta + c_3\eta + c_4\zeta\eta)}{\sqrt{\zeta + \eta}} \quad (5.75)$$

which is equivalent to a linear strain one.

where ζ & η = Area coordinates.

These elements were inserted into the C.E.G.B. stress analysis system BERSAFE and the problem of a rectangular centre cracked strip in tension was solved by them. With (4) elements surrounding the crack tip, the stress intensity factors were obtained from the displacements similar to Tracey's method. The results were found to be very accurate compared to those obtained by Paris and Sih, [72]. It was noted that these results obtained by using linear strain elements having (1786) degrees of freedom were more accurate than those using constant strain elements having (480) degrees of freedom, [32].

vi) Hilton and Hutchinson method: This method was developed by Hilton and Hutchinson to evaluate elastic plastic stress intensity factors K_I and K_{III} , [73]. Wilson used this method also to evaluate the elastic stress intensity factor K_{III} , [74]. The philosophy behind this method is that the asymptotic expansion becomes more accurate as the singularity is approached, while finite element approximation is accurate everywhere else. Thus a combination of these two tools, each in the region where it is most accurate, will yield a good solution to the problem. The present work is based on the development of this method to solve single and mixed mode fracture problems of axisymmetric solid, therefore the method will be illustrated by using the axisymmetric problem of a round solid bar of radius (R) containing a circumferential crack of depth (C) and subjected to uniform axial tension. The specimen geometry is shown in Fig. (5.13). A core of radius (R_c) is constructed around the crack tip defining a boundary (Γ), and in view of symmetry, only a quadrant of the upper half of the cylinder is shown in Fig. (5.14).

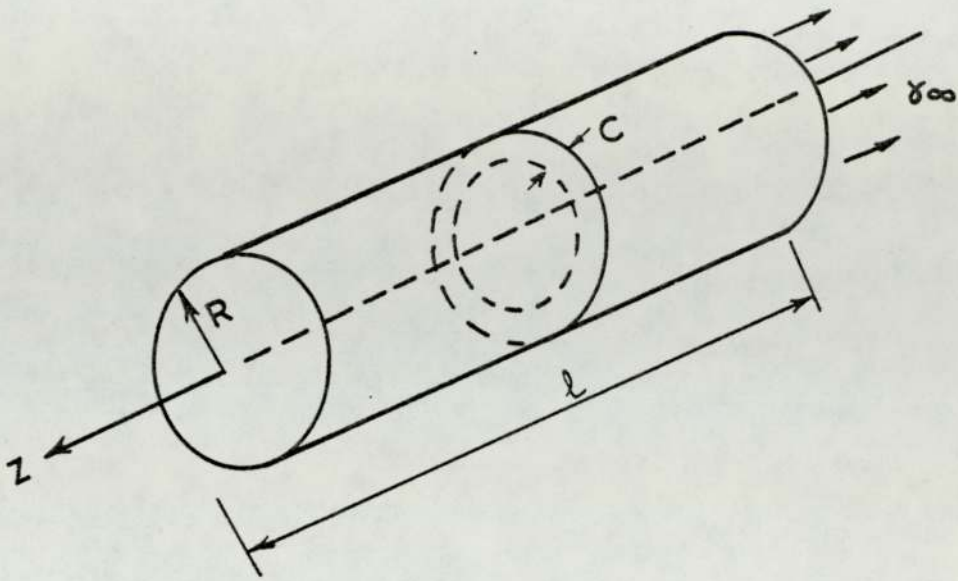


Fig. 5.13

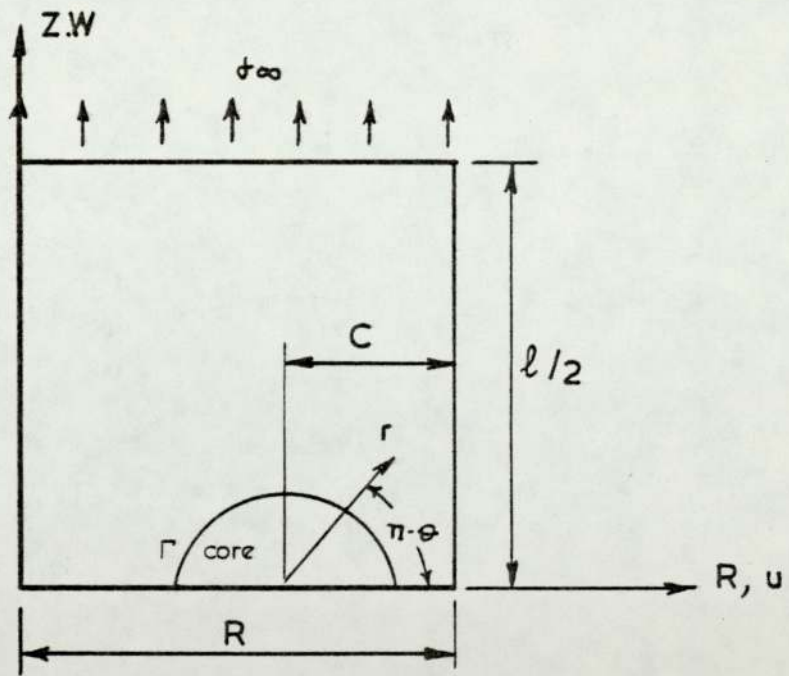


Fig. 5.14

It is assumed that when the ratio of (R_c/C) is very small compared to the crack depth (C) , the conditions of plane strain prevail within the core region, [24]. This assumption may be justified by inference from the existing analytical solutions for embedded flaws in an infinite medium, [75], for the through cracks in finite thickness plates in the region close to the crack border on the interior of the plate, [76], and from the numerical solutions for surface cracks in finite thickness plates, [77]. Another justification can be sought from the numerical results of strain components of elastic plastic analysis of a round notched bar obtained by Tracey, [78].

The displacement components in the core region are given by equations (4.10). These equations represent constraints on the nodal points of the first ring of triangular elements which fall on (Γ) .

The potential energy of the upper half of the cylinder is given by:

$$PE = SE_{core} + \sum_{EL} SE - \int T \cdot u dS \quad (5.76)$$

where:

SE_{core} = Strain energy of the core

$\sum_{EL} SE$ = strain energy of all elements outside the core

$\int T \cdot u dS$ = Work done by traction vector (T) on the surface S
displacement vector (u) .

The core strain energy is given by, [24]:

$$SE_{core} = \frac{1}{2} \int_V \sigma_{ij} \epsilon_{ij} dV \quad (5.77)$$

$$= \pi \frac{(1+\nu)}{2E} R_c [(R-C)(\frac{5}{4}-2\nu)+r(\frac{1}{4}-\frac{\nu}{2})] K_I^2 \quad (5.78)$$

Hence, the stress intensity factor (K_I), the rigid body displacements (δr), and the nodal displacements vector $\{u_i\}$ become generalized coordinates. The governing linear algebraic equations to be solved for K_I , δr and $\{u_i\}$ are obtained from the minimization of the potential energy equation (5.76). It must be observed that the problem is not a free variation one because of the constraints implied on the nodes on (Γ) by equations (4.10). This aspect will be discussed in more detail in Chapter 6. Now:

$$\frac{\partial P_E}{\partial K_I} = 0 \quad (5.79)$$

$$\frac{\partial P_E}{\partial \delta r} = 0 \quad (5.80)$$

$$\frac{\partial P_E}{\partial u_i} = 0 \quad , i = 1, 2, \dots, N \quad (5.81)$$

where: N = total number of unconstrained degrees of freedom.

The continuity of displacement at the interface (Γ) between the core and the neighbouring triangular elements is not fully satisfied, compatibility is assumed only at the nodes, and hence monotonic convergence to the exact solution cannot be expected. However, the number of nodes on the core, and therefore the number of elements surrounding it, can be increased to reduce the discontinuity of displacements to an acceptable level. A further improvement may be achieved if linear strain triangular elements are used outside the core, as in the present work, because their quadratic displacement function is closer to the core's than the linear one of the constant-strain elements.

The advantages of this method are:

1. The simplicity of incorporating the core treatment within a standard finite element program.
2. Separation of K_I , K_{II} , and K_{III} for combined mode analysis.
3. The stress intensity factors are obtained directly from the program.
4. Possibility of extension to elastic-plastic, plastic, and different material properties.

This method will be used to solve all the fracture problems considered in this thesis and is discussed in more detail in the following chapters.

CHAPTER 6

MODIFICATION TO THE FINITE ELEMENT FORMULATION
TO INCORPORATE A SINGULAR CORE

6.1 Introduction

In has been shown in Chapter 5 that, even with a fine mesh, the application of a standard finite element program to fracture problems does not yield accurate results. Incorporating a special crack tip element or a core subregion, with the crack tip singularity embedded in their displacements functions, will improve the results and reduce the number of elements required to idealize the structure. The Hilton and Hutchinson method, which uses a special core element around the crack tip, was discussed briefly and its advantages over other methods were shown in section (5.4.8). This chapter will be devoted to the description of the modifications required to a standard finite element program to incorporate the Hilton and Hutchinson type of core element. There are other methods than the one described for synthesising the singular core and finite element mesh: Richards method utilizing Lagrange Multipliers is an example, [79].

6.2 Modification to the finite element formulation to include the Hilton and Hutchinson type core element

Considering a solid circular bar with a circumferential crack inclined at an angle (θ), Fig.(6.1). The crack tip is surrounded by

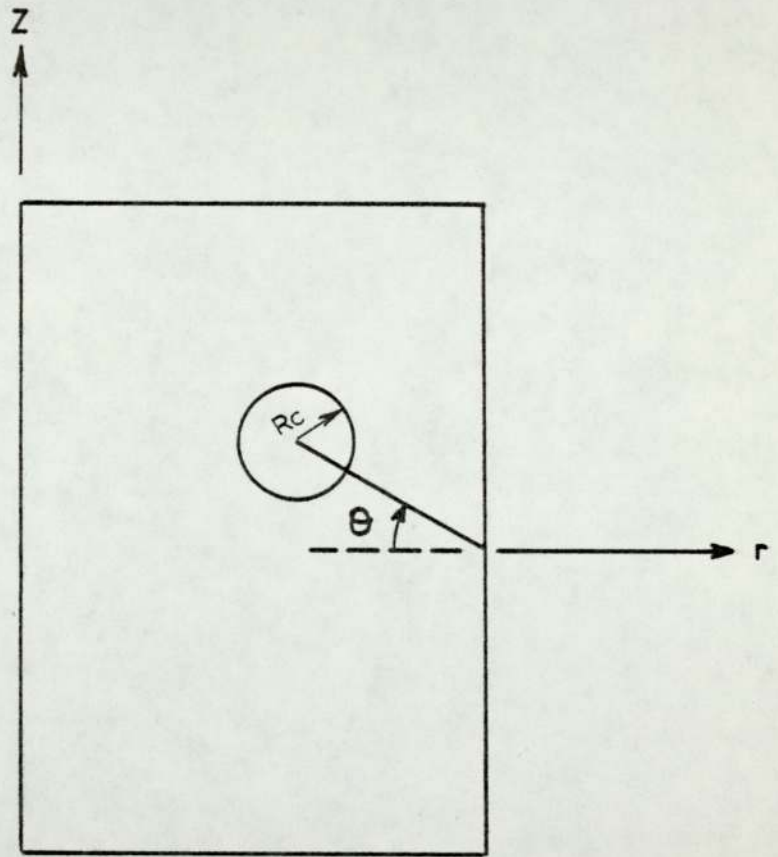
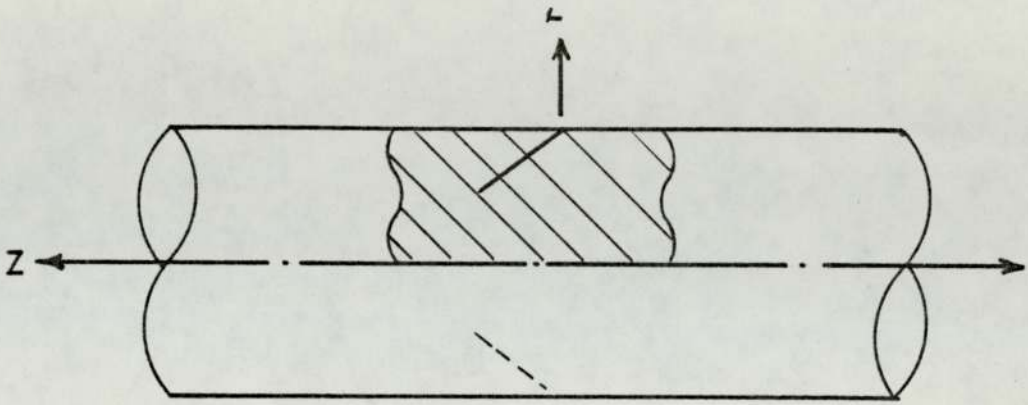


Fig.6.1

a circular core of radius (R_c). The displacement components in the core region are expressed as a series expansion, (Appendix 10.2), in the form:

$$u_r = K_I f(r,\theta) + K_{II} g(r,\theta) + \alpha_1 h(r,\theta) \dots \quad (6.1)$$

$$u_z = K_I \ell(r,\theta) + K_{II} m(r,\theta) + \alpha_2 n(r,\theta) \dots \quad (6.2)$$

where: $f(r,\theta) \dots$ etc. are given in (Appendix 10.2)

Writing equations (6.1) and (6.2) in matrix form:

$$\{u\}_c = [N]_c \{\alpha\}_c \quad (6.3)$$

where:

$$[N]_c = \text{matrix of functions } f(r,\theta) \dots \text{ etc.} \quad (6.4)$$

$\{\alpha\}_c = \text{Vector of unknown parameters,}$

$$K_I, K_{II}, \alpha, \dots \text{ etc.}$$

Within the core, the strains are related to the vector $\{\alpha\}_c$ (Appendix 10.2) by:

$$\{\epsilon\}_c = [B]_c \{\alpha\}_c \quad (6.5)$$

Using the stress-strain relationship

$$\{\sigma\}_c = [C] \{\epsilon\}_c, \quad (6.6)$$

the strain energy of the core is then given by:

$$U_c = \frac{1}{2} \int_{V_c} \{\sigma\}_c^t \{\epsilon\}_c dV \quad (6.7)$$

Therefore

$$U_c = \frac{1}{2} \{\alpha\}_c^t \left(\int_{V_c} [B]_c^t [C] [B]_c dV \right) \{\alpha\}_c, \quad (6.8)$$

or

$$U_c = \frac{1}{2} \{\alpha\}_c^t [K]_c \{\alpha\}_c \quad (6.9)$$

The shape of the core is arbitrary, but a circular core is chosen to simplify the integration required to form $[K]_c$.

The total potential energy of the solid is given by:

$$PE = U_c + \sum_{EL} U_E - \Omega \quad (6.10)$$

where

U_E = Strain energy of elements outside the core.

Ω = Potential energy by the traction vector on the surface displacement vector.

Equation (6.10) may be written as:

$$PE = \frac{1}{2} \{\alpha\}_c^t [K]_c \{\alpha\}_c + \frac{1}{2} \{q\}^t [K] \{q\} - \{q\}^t \{Q\} \quad (6.11)$$

The $(\alpha$'s) and $(q$'s) are not independent because the $(q$'s) on the interface (Γ) are related to the $(\alpha$'s). Therefore, it proves convenient to partition $\{q\}$ into $\{q_1\}$ which corresponds to the nodal displacements on (Γ) and $\{q_2\}$ which corresponds to the nodal displacements of the remainder of the mesh. The stiffness matrix $[K]$ and load vector $\{Q\}$ are partitioned similarly, i.e.

$$\{q\} = \begin{Bmatrix} \{q_1\} \\ \{q_2\} \end{Bmatrix}, \quad \{Q\} = \begin{Bmatrix} \{Q_1\} \\ \{Q_2\} \end{Bmatrix}, \quad [K] = \begin{bmatrix} [K_{11}] & [K_{12}] \\ [K_{21}] & [K_{22}] \end{bmatrix} \quad (6.12)$$

From equation (3.16), the vector of nodal displacements on (Γ) may be expressed as a function of (α) as:

$$\{q_1\} = [A]\{\alpha\}_c, \quad (6.13)$$

where

$[A]$ = matrix of function $f(r,\theta)$...etc. evaluated at the node on (Γ) .

and

$$\{q_1\}^t = \{\alpha\}_c^t [A]^t \quad (6.14)$$

Substituting back in equation (6.11) yields:

$$\begin{aligned} PE &= \frac{1}{2} \{\alpha\}_c^t [K]_c \{\alpha\}_c + \frac{1}{2} \{\alpha\}_c^t [A]^t [K_{11}] [A] \{\alpha\}_c \\ &+ \frac{1}{2} \{\alpha\}_c^t [A]^t [K_{12}] \{q_2\} + \frac{1}{2} \{q_2\}^t [K_{21}] [A] \{\alpha\}_c \\ &+ \frac{1}{2} \{q_2\}^t [K_{22}] \{q_2\} - \{\alpha\}_c^t [A]^t \{Q_1\} \\ &- \{q_2\}^t \{Q_2\} \end{aligned} \quad (6.15)$$

The variation $(\delta(PE))$ is found by treating the $(\alpha$'s) and $(q_2$'s) as free variables which may receive arbitrary independent increments, therefore, for equilibrium:

$$\begin{aligned}
 \delta(\text{PE}) = 0 = & \{\delta\alpha\}_c^t [K]_c \{\alpha\}_c + \{\delta\alpha\}_c^t ([A]^t [K_{11}] [A]) \{\alpha\}_c \\
 & + \frac{1}{2} \{\delta\alpha\}_c^t [A]^t [K_{12}] \{q_2\} + \frac{1}{2} \{\alpha\}_c^t [A]^t [K_{12}] \{\delta q_2\} \\
 & + \frac{1}{2} \{\delta q_2\} [K_{21}] [A] \{\alpha\}_c + \frac{1}{2} \{q_2\} [K_{21}] [A] \{\delta\alpha\}_c \\
 & + \{\delta q_2\}^t [K_{22}] \{q_2\} - \{\delta\alpha\}_c^t [A]^t \{Q_1\} \\
 & - \{\delta q_2\}^t \{Q_2\}
 \end{aligned} \tag{6.16}$$

As $[K]$ is symmetric so that:

$$[K_{12}] = [K_{21}]^t, \tag{6.17}$$

and matrix products of the type $\frac{1}{2} \{\alpha\}_c^t [A]^t [K_{12}] \{\delta q_2\}$ are scalar so that:

$$\begin{aligned}
 \frac{1}{2} \{\alpha\}_c^t [A]^t [K_{12}] \{\delta q_2\} &= \left(\frac{1}{2} \{\alpha\}_c^t [A]^t [K_{12}] \{\delta q_2\} \right)^t \\
 &= \frac{1}{2} \{\delta q_2\}^t [K_{21}] [A] \{\alpha\}_c,
 \end{aligned} \tag{6.18}$$

then,

$$\begin{aligned}
 \delta(\text{PE}) = 0 = & \{\delta\alpha\}_c^t \left[[K]_c \{\alpha\}_c + ([A]^t [K_{11}] [A]) \{\alpha\}_c \right. \\
 & \left. + [A]^t [K_{12}] \{q_2\} - [A]^t \{Q_1\} \right] \\
 & + \{\delta q_2\}^t \left[[K_{22}] \{q_2\} + [K_{21}] [A] \{\alpha\}_c \right. \\
 & \left. - \{Q_2\} \right]
 \end{aligned} \tag{6.19}$$

Since $\{\delta\alpha\}_c$ and $\{\delta q_2\}$ are arbitrary, equation (6.19) yields two simultaneous matrix equations of equilibrium:

$$[K_{22}]\{q_2\} + [K_{21}][A]\{\alpha\}_c - \{Q_2\} = 0 \quad (6.20)$$

$$[K]_c \{\alpha\}_c + ([A]^t [K_{11}] [A]) \{\alpha\}_c + [A]^t [K_{12}] \{q_2\} - [A]^t \{Q_1\} = 0 \quad (6.21)$$

These equations may be written as one re-formed stiffness matrix and load vector of the form,

$$[K]^* \{q\}^* = \{Q\}^* \quad (6.22)$$

as

$$\begin{bmatrix} ([K_{22}]) & ([K_{21}][A]) \\ ([A]^t [K_{12}]) & ([K]_c + [A]^t [K_{11}] [A]) \end{bmatrix} \begin{Bmatrix} \{q_2\} \\ \{\alpha\}_c \end{Bmatrix} = \begin{Bmatrix} \{Q_2\} \\ ([A]^t \{Q_1\}) \end{Bmatrix} \quad (6.23)$$

Equation (6.23) shows how easy it is to modify a standard finite element program to include a "core" element. The overall stiffness matrix and load vector are altered as described in the previous steps and the solution proceeds normally after that. The numerical implementation of the operations described in this chapter will be discussed in Chapter 7.

CHAPTER 7

COMPUTER PROGRAM PROCEDURES

7.1 Introduction

The development of a computer program solving fracture problems of axisymmetric solids using the finite element method and incorporating the special core element of Hilton and Hutchinson was carried out in two stages. The first stage was to develop a general finite element program using a six node isoparametric ring element to solve problems of axisymmetric solids subjected to axisymmetric loading. To simplify the structure of the program, it was divided into substructures (procedures) each performing a stage of the solution. Procedure (STRDIS) formulates the strain displacement relations, and the computer code for the constitutive relations is given in procedure (CONSTREL). The boundary conditions are applied by means of two procedures: (LOADING) which deals with the applied loads, and (BOUNCONST) which deals with the prescribed displacements. It is noted that a large number of elements is required to yield a reasonably accurate solution, hence the computer storage needed for the overall stiffness matrix is large also. Jennings and Tuff, [82], developed a scheme to store this matrix as a one dimensional array, retaining the coefficients between the first non zero term in any row and the leading diagonal, as the matrix is symmetric. To achieve this a special address sequence is used which allocates the appropriate place for each element of the overall stiffness matrix in its two

dimensional form in the one dimensional form. This sequence was programed and used by Robertson and a detailed description of it is available in his Ph.D. thesis, [9]. Robertson's procedure (ADDARRAY) was used together with procedure (ASSEMBLY) to evaluate the element stiffness matrix by performing the integration required numerically, and assembling the overall stiffness matrix as a one dimensional array. The set of simultaneous linear equations $[K]\{q\} = \{Q\}$ is solved for the unknown displacement vector $\{q\}$ by procedure (SYMVBSOL). The output procedures (NODSTR) and (ELESTR) evaluates the strains and stresses of nodal points and element centroids respectively. The computer code developed for the automatic mesh generation and the specification of the prescribed nodal loads and displacements is given in procedure (FEINPUT).

The second stage of the development of the computer program involved augmenting the general program obtained from the first stage to incorporate a Hilton and Hutchinson type of core element. This implied changing the input procedure to generate a core around the crack tip with a specified number of nodes on it and subdividing the remainder of the structure in the normal manner. It is noted that in mode (I) fracture problems, the structure and loading are symmetric with respect to the crack plane, hence only a semicircular core is required; while a full circular one is needed for the mixed mode (I) and (II) problem. The computer code for the mode (I) input requirements are given in procedure (CCRINPUT) and those for mixed mode (I) and (II) in procedure (MMINPUT). The shouldered bar problem has a purpose-built input procedure similar to that of mixed mode (I) and (II) but accommodating the variation in fillet radii required, and is given in procedure (SBINPUT).

The other change implied by incorporating a core element is the modification of the overall stiffness matrix and load vector described in Chapter (6). A computer code for these modifications was developed by Robertson for two dimensional plane strain and plane stress problems and is discussed in,[9]. Two procedures somewhat similar to Robertson's were developed for mode (I) and mixed mode (I) and (II) axisymmetric problems which are (CCRMI) and (CCRMM12) respectively.

Procedures (KARM1BND) and (KARBDMMST) check the array $\underset{\chi}{u}$ bonds and expands them if necessary to allow for the modifications of the overall stiffness matrix and load vector for mode (I) and mixed mode (I) and (II) problems respectively. It will be seen later that the displacements obtained from solving the set of simultaneous linear equations may be used to check the conditioning of the equations by resubstituting them back into the equilibrium equations and obtaining a new force vector $\{Q\}^*$. The computer code to perform this is given in procedure (RESIDUAL). Sih's strain energy density criterion will be used later to determine the direction of crack initiation. It requires finding the angle which minimizes the strain energy density function given by equation (4.24). The standard procedure (E04AAA) from the Nottingham algorithm group Library is used for this purpose.

A detailed description of these procedures will be presented in the following sections of this chapter, and the programs incorporating them together with input instructions and sample problems will be listed in Appendix (10.4).

7.2 Procedure (STRDIS)

This procedure evaluates the coefficients of the [B] matrix relating the strains to the displacements, and the Jacobean matrix [J] and its determinant described in section (3.4), at various points within each element. The number and location of these points is specified by the numerical integration technique and its order chosen to solve the problem. In solving the axisymmetric problem by finite elements, computational problems arise from the calculation of the hoop strain (u/r) for elements with one side coinciding with the axis of revolution. The relation between the natural local coordinates and the global ones is given by:

$$\begin{Bmatrix} 1 \\ r \\ z \end{Bmatrix} = \begin{bmatrix} \{1_i\}^t \\ \{r_i\}^t \\ \{z_i\}^t \end{bmatrix} \left\{ N_i(L_1, L_2, L_3) \right\} \quad i = 1, 2, \dots, 6 \quad (7.1)$$

where

$N_i(L_1, L_2, L_3)$, $i = 1, 2, \dots, 6$ are given in equations (3.2.1).

Therefore

$$r = r_1 N_1 + r_2 N_2 + r_3 N_3 + r_4 N_4 + r_5 N_5 + r_6 N_6 \quad (7.2)$$

For an element with one side coinciding with the axis of revolution, the (r_i) values of the three nodes on that side are zeroes. The remaining three terms defining (r) are all functions of one of the natural coordinates L_1, L_2 or L_3 depending on the node numbering sequence of that particular element. When the value of this natural coordinate is zero for a particular point of integration, the whole value of (r) is reduced to zero. To overcome this, the (r) value for such elements is taken as a constant and is equal to the average of the six (r_i) values of the nodes.

The following steps are taken to formulate the [B] matrix:

1. The coefficients of the Jacobean matrix [J] are determined from equation (3.27), the determinant and inverse of [J] are evaluated, and the coefficients of [J] are replaced by those of $[J]^{-1}$.
2. The shape functions given in equation (3.21) are stated.
3. A code is introduced to distinguish between hollow and solid structures.
4. The radii values required for computing hoop strains and numerical integration are evaluated as described previously in this section.
5. The coefficients of the [B] matrix are evaluated from equation (3.33).

Procedure STRDIS Flowchart

1.

START

$$J[1,1] := X[N[Z,1]]*(4*L1-1) + \\ X[N[Z,3]]*(4*L1+4*L2-3) + \\ 4*L2*X[N[Z,4]] - 4*L2*X[N[Z,5]] \\ + 4*X[N[Z,6]]*1 - 2*L1*L2)$$

$$J[1,2] := Y[N[Z,1]]*(4*L1-1) + Y[N[Z,3]] \\ *(4*L1+4*L2-3) + 4*L2*Y[N[Z,4]]. \\ 4*L2*Y[N[Z,5]] + 4*Y[N[Z,6]] \\ *(1-2*L1-L2)$$

$$J[2,1] := X[N[Z,2]]*(4*L2-1)*X[N[Z,3]] \\ *(4*L1+4*L2-3) + 4*L1*X[N[Z,4]] \\ + 4*X[N[Z,5]]*(1-L1-2*L2) \\ - 4*L1*X[N[Z,6]]$$

$$J[2,2] := Y[N[Z,2]]*(4*L2-1) + Y[N[Z,3]]* \\ (4*L1+4*L2-3) + 4*L1*Y[N[Z,4]] + \\ 4*Y[N[Z,5]]*(1-L1-2*L2) \\ - 4*L1 + Y[N[Z,6]]$$

$$U := J[1,1]*J[2,2] - J[1,2]*J[2,1]$$

$$\text{CHANGE} := J[1,1]$$

$$J[1,1] := J[2,2]/U$$

$$J[1,2] := -J[1,2]/U$$

$$J[2,1] := -J[2,1]/U$$

$$J[2,2] := \text{CHANGE}/U$$

2.

$$NL[1] := L1*(2*L1-1)$$

$$NL[2] := L2*(2*L2-1)$$

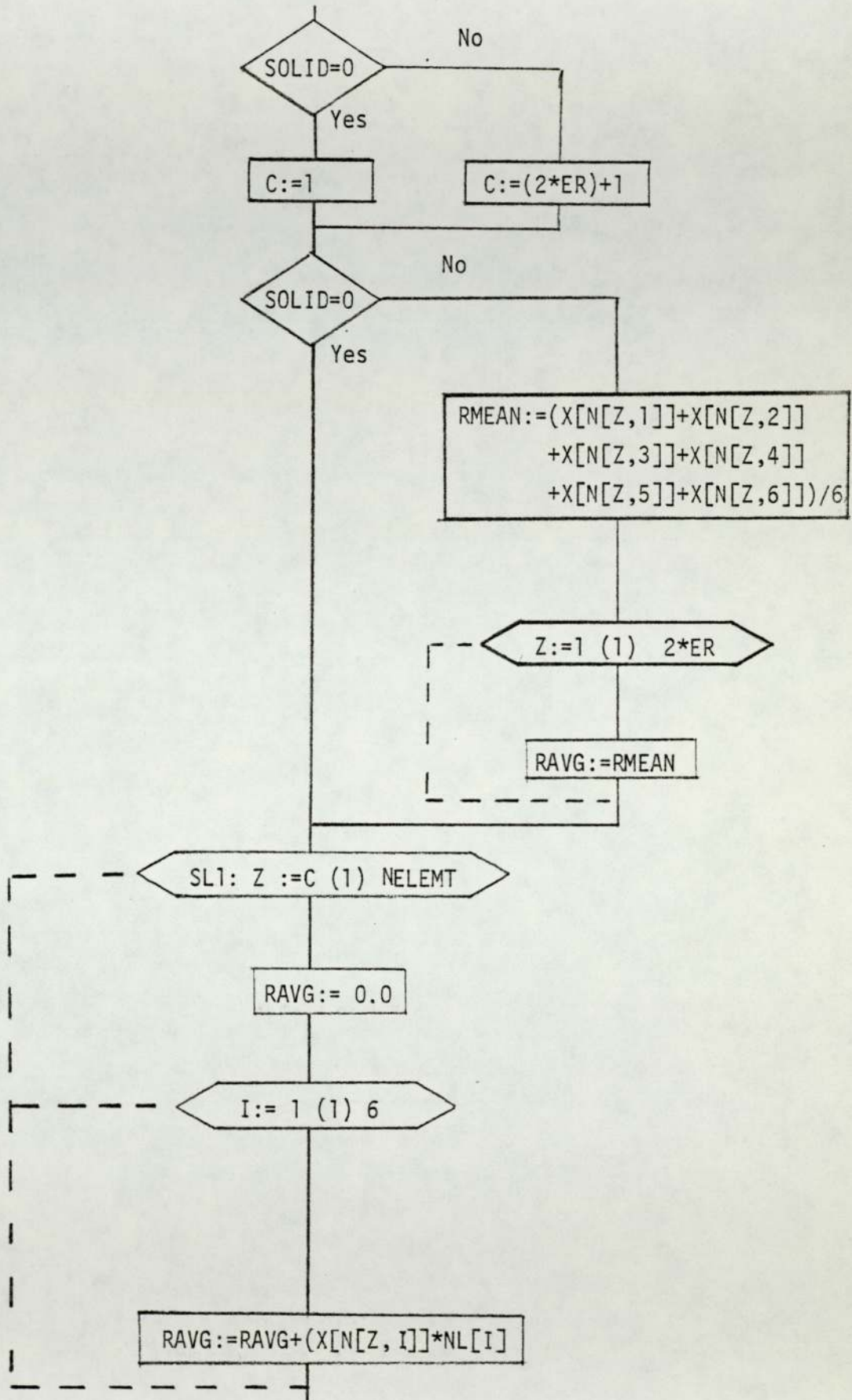
$$NL[3] := L3*(2*L3-1)$$

$$NL[4] := 4*L1*L2$$

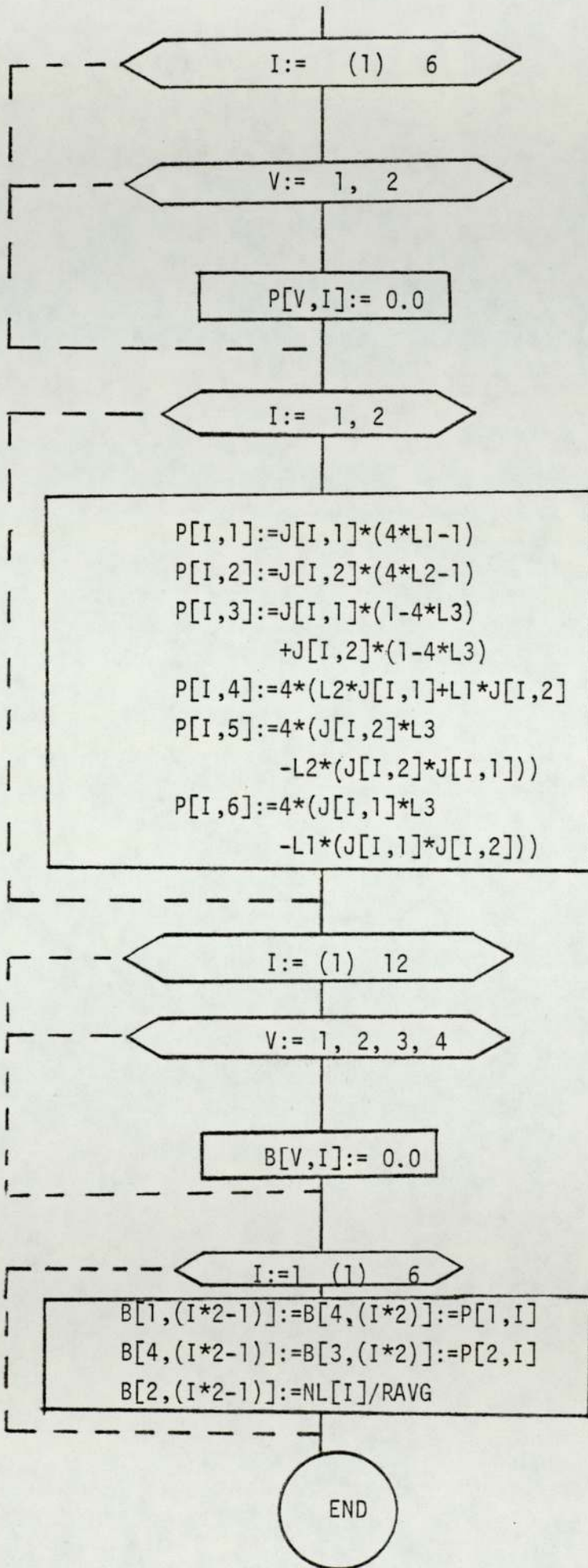
$$NL[5] := 4*L2*L3$$

$$NL[6] := 4*L3*L1$$

3.



5.



7.3 Application of boundary conditions

The two types of boundary conditions considered are prescribed loads and/or displacements. Their directions should coincide with the global axes, and hence an inclination in the direction with respect to them must be resolved manually and the components in their direction are inputted. Non zero as well as zero displacements can be imposed.

7.3.1 Procedure LOADING

This procedure deals with the applied load which may either be nodal point-loads or distributed loads. In the case of nodal point-loads, a code is introduced to distinguish between loads in the (z) and/or (r) directions. The value of the prescribed nodal load on a node (N) is assigned to row (2N-1) or (2N) in the {Q} vector depending on its direction.

A distributed load is best represented by nodal loads which produce equal work on the structure to that by the distributed load. Considering a linearly varying force intensity $P(\xi_B)$ acting on face (B) of the element shown in Fig.(7.1). The force per unit length is $(2\pi r P(\xi_B))$. If (R_p) are the nodal loads statically equivalent to $P(\xi_B)$, then when virtual node displacements (r_B) are produced, the work done by these nodal loads must be equal to that done by the distributed load $P(\xi_B)$.

$$[R_p]^t \{r_B\} = \int_B \{P\}^t \{r_B\} dS \quad (7.3)$$

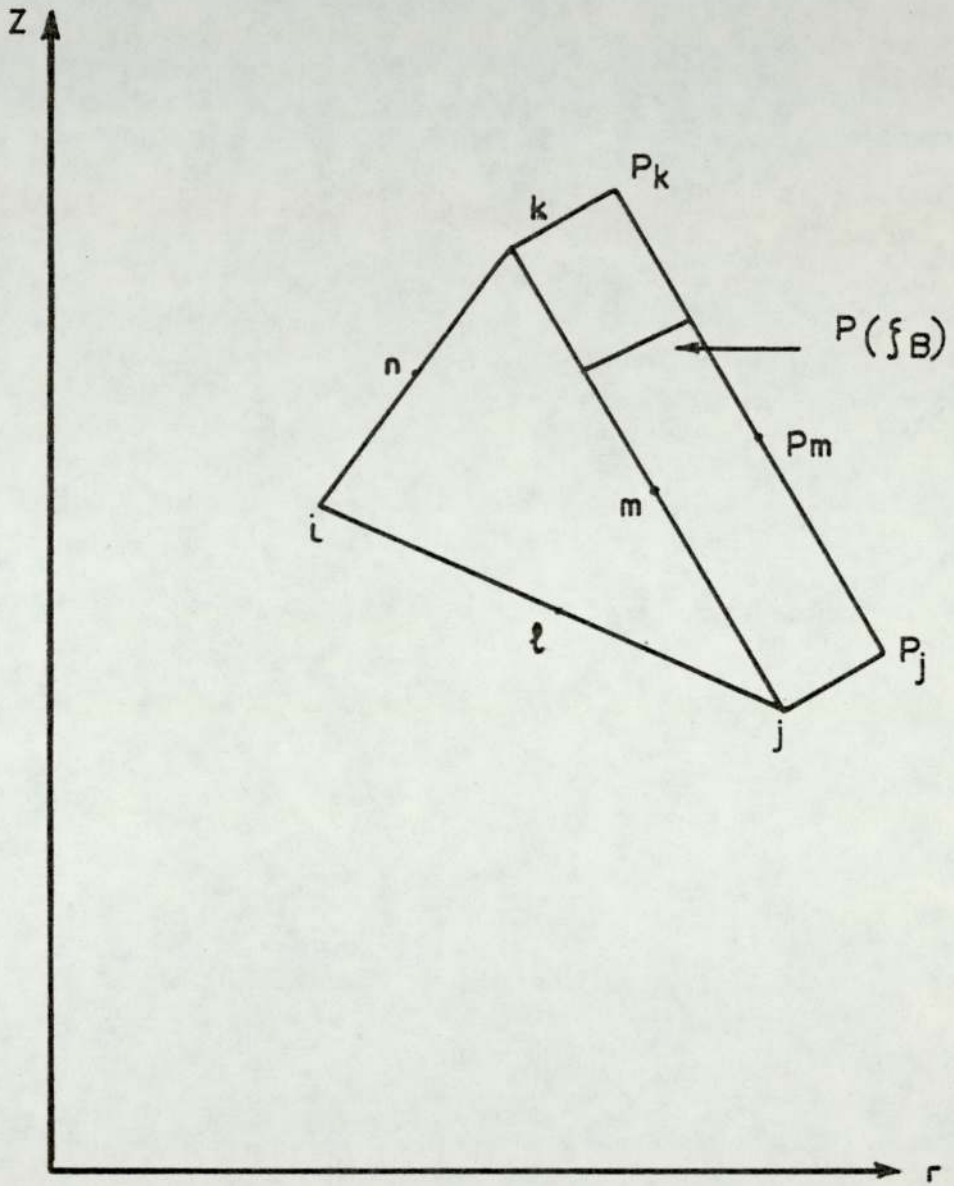


Fig .7.1

The distributed load may be resolved into two components in the (r) and (Z) directions $P_r(\xi_B)$ and $P_z(\xi_B)$ vary as a polynomial of order (n), then the values needed to specify it will be (n+1), hence for convenience the variation will be assumed parabolic. Therefore, the values specified at the (n+1) nodal points (P_r) and (P_z) are related to $P(\xi_B)$ by:

$$\begin{aligned}
 P(\xi_B) &= \begin{bmatrix} P_r(\xi_B) \\ P_z(\xi_B) \end{bmatrix} = \begin{bmatrix} \{\phi_p\}^t & 0 \\ 0 & \{\phi_p\}^t \end{bmatrix} \begin{Bmatrix} P_r \\ P_z \end{Bmatrix} \\
 &= \{\psi_p\}^t \{P\} \qquad (7.4)
 \end{aligned}$$

The displacements on face (B) depends only on the nodal points on it, hence

$$\begin{aligned}
 r_B(\xi_B) &= \begin{bmatrix} u(\xi_B) \\ w(\xi_B) \end{bmatrix} = \begin{bmatrix} \{\phi_B\}^t & 0 \\ 0 & \{\phi_B\}^t \end{bmatrix} \begin{Bmatrix} u_B \\ w_B \end{Bmatrix} \\
 &= \{\psi_B\}^t \{r_B\} \qquad (7.5)
 \end{aligned}$$

From equations (7.4) and (7.5):

$$\{R_p\} = \int_B \{\psi_B\} \{\psi_p\}^t ds \{P\} \qquad (7.6)$$

Hence:

$$\begin{Bmatrix} R_{pr} \\ R_{pz} \end{Bmatrix} = \int_B \begin{bmatrix} \{\phi_B\} \{\phi_p\}^t & 0 \\ 0 & \{\phi_B\} \{\phi_p\}^t \end{bmatrix} ds \begin{Bmatrix} P_r \\ P_z \end{Bmatrix} \qquad (7.7)$$

and

$$\{R_{pr}\} = \int_B \{\phi_B\} \{\phi_p\}^t ds P_r \quad (7.8)$$

$$\{R_{pz}\} = \int_B \{\phi_B\} \{\phi_p\}^t ds P_z \quad (7.9)$$

For the side (j-k) in the element shown in Fig. (7.1)

$$\{R_{pr}\} = 2\pi \int_j^k r \{\phi_B\} \{\phi_p\}^t ds P_r \quad (7.10)$$

If a linear variation of $P(\xi_B)$ is assumed on this side of the element:

$$\{\phi_p\}^t = [L_2(1-L_2)] \quad (7.11)$$

$$\{\phi_B\}^t = [L_2(2L_2-1), (1-L_2)(1-2L_2), 4L_2(1-L_2)] \quad (7.12)$$

$$r = [L_2(1-L_2)] \begin{Bmatrix} r_j \\ r_k \end{Bmatrix} \quad (7.13)$$

and is a scalar multiplier of every term of $\{\phi_B\} \{\phi_p\}^t$.

Hence,

$$\{R_{pr}\} = 2\pi l_1 \int_0^1 r \begin{bmatrix} L_2(2L_2-1) \\ (1-L_2)(1-2L_2) \\ 4L_2(1-L_2) \end{bmatrix} [L_2(1-L_2)] dL_2 P_r \quad (7.14)$$

$$\{R_{pr}\} = \frac{\pi l_1}{30} \begin{bmatrix} r_j \begin{bmatrix} 9 & 1 \\ -1 & 1 \\ 12 & 8 \end{bmatrix} + r_k \begin{bmatrix} 1 & -1 \\ 1 & 9 \\ 8 & 12 \end{bmatrix} \end{bmatrix} \begin{bmatrix} P_{rj} \\ P_{rk} \end{bmatrix} \quad (7.15)$$

For the case of a uniform pressure on the face parallel to the z-axis:

$$r_j = r_k \text{ and } P_{rj} = P_{rk} = P_r \quad (7.16)$$

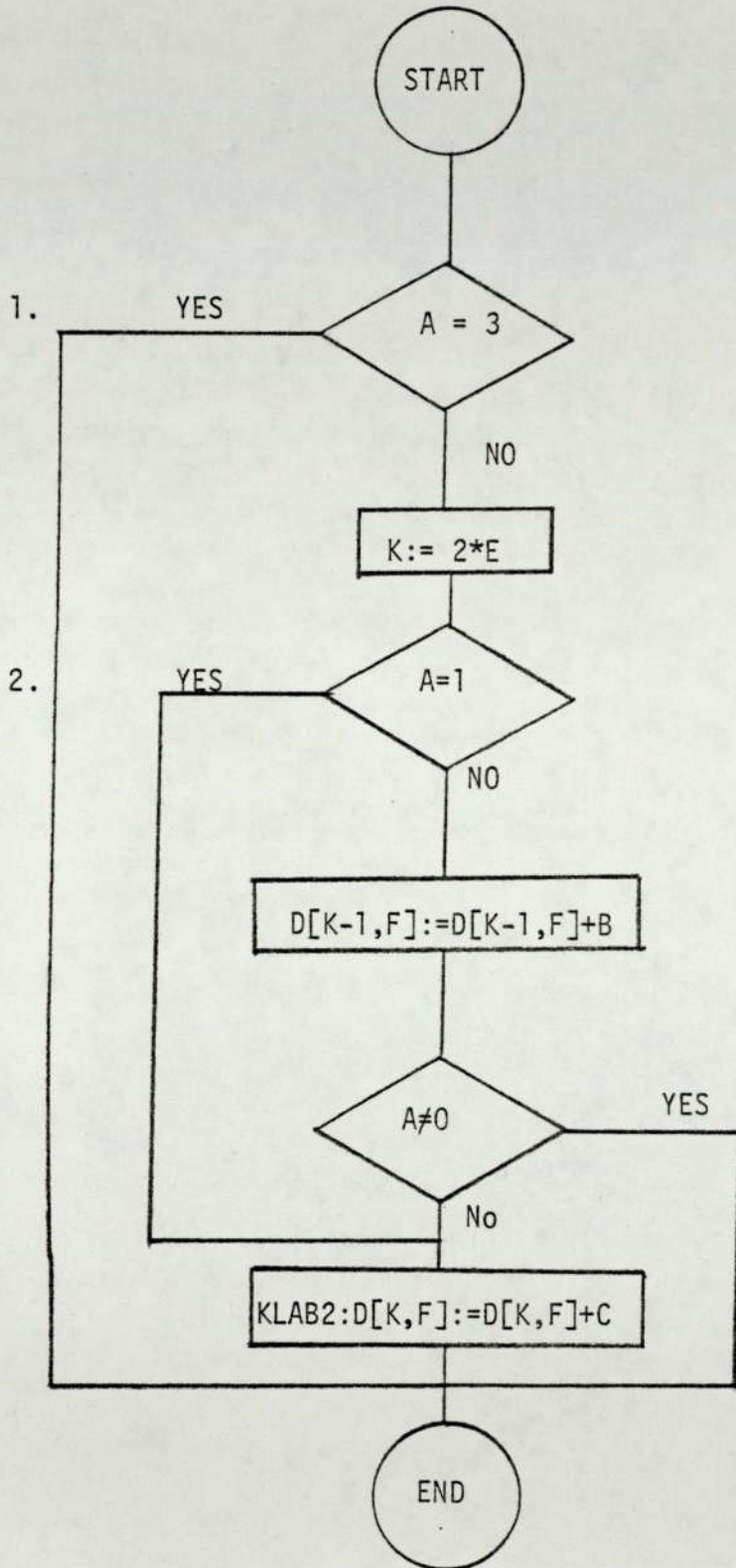
Therefore equation (7.15) becomes:

$$\begin{aligned} \{R_{pr}\} &= \begin{bmatrix} R_{rj} \\ R_{rk} \\ R_{rm} \end{bmatrix} = \frac{\pi \ell_i r_j P_r}{30} \begin{bmatrix} 10 \\ 10 \\ 40 \end{bmatrix} \\ &= \frac{2\pi \ell_i r_j P_r}{6} \begin{bmatrix} 1 \\ 1 \\ 4 \end{bmatrix} \end{aligned} \quad (7.17)$$

The calculation of equivalent nodal loads is done manually, and assigning each load to its appropriate location in the force vector is done by the following steps:

1. Nodes with prescribed displacements in both the (r) and (z) directions are bypassed.
2. A load in the (r) direction on node (N) is allocated to row (2N-1) in {Q} and the one with (z) direction to row (2N).

Procedure LOADING Flowchart



7.3.2 Procedure BOUNCONST

This procedure deals with the prescribed nodal displacements which can either be zeroes i.e. the nodal point is fixed in the (r) and/or (z) directions, or a specified value. In the special case of zero displacements, the corresponding row and column of the appropriate degree of freedom are deleted from the system of equations.

For convenience, the nodal displacement vector {q} may be rearranged and partitioned so that

$$\{q\} = \begin{Bmatrix} \{q_1\} \\ \{q_2\} \end{Bmatrix} \quad (7.18)$$

where

{q₁} = vector of unconstrained nodal displacements

{q₂} = vector of constrained nodal displacements

The system of linear simultaneous equations [K]{q} = {Q}, is rearranged and partitioned accordingly so that:

$$\begin{bmatrix} [K_{11}] & [K_{21}]^t \\ [K_{21}] & [K_{22}] \end{bmatrix} \begin{Bmatrix} \{q_1\} \\ \{q_2\} \end{Bmatrix} = \begin{Bmatrix} \{Q_1\} \\ \{Q_2\} \end{Bmatrix} \quad (7.19)$$

Equation (7.19) can be rewritten as:

$$\begin{bmatrix} [K_{11}] & [0] \\ [0] & [I] \end{bmatrix} \begin{Bmatrix} \{q_1\} \\ \{q_2\} \end{Bmatrix} = \begin{Bmatrix} \{Q_1\} - [K_{21}]^t \{q_2\} \\ \{q_2\} \end{Bmatrix} \quad (7.20)$$

The prescribed displacements $\{q_2\}$ are automatically inserted into the load vector and the system is solved for the unknown displacements $\{q_1\}$.

The procedure inserts the prescribed displacement $\{q_N\}$ at the Nth degree of freedom into the load vector by:

1. The column number (CJ) of the first non-zero coefficient of row (N) in the stiffness matrix is determined.
2. The load vector is modified as:

$$Q_i = Q_i - K_{iN} q_N \quad i = 1, 2, \dots, N \quad (7.21)$$

and the rows of the stiffness matrix are modified from column (CJ) to the leading diagonal as:

$$K_{NM} = K_{MN} = 0$$

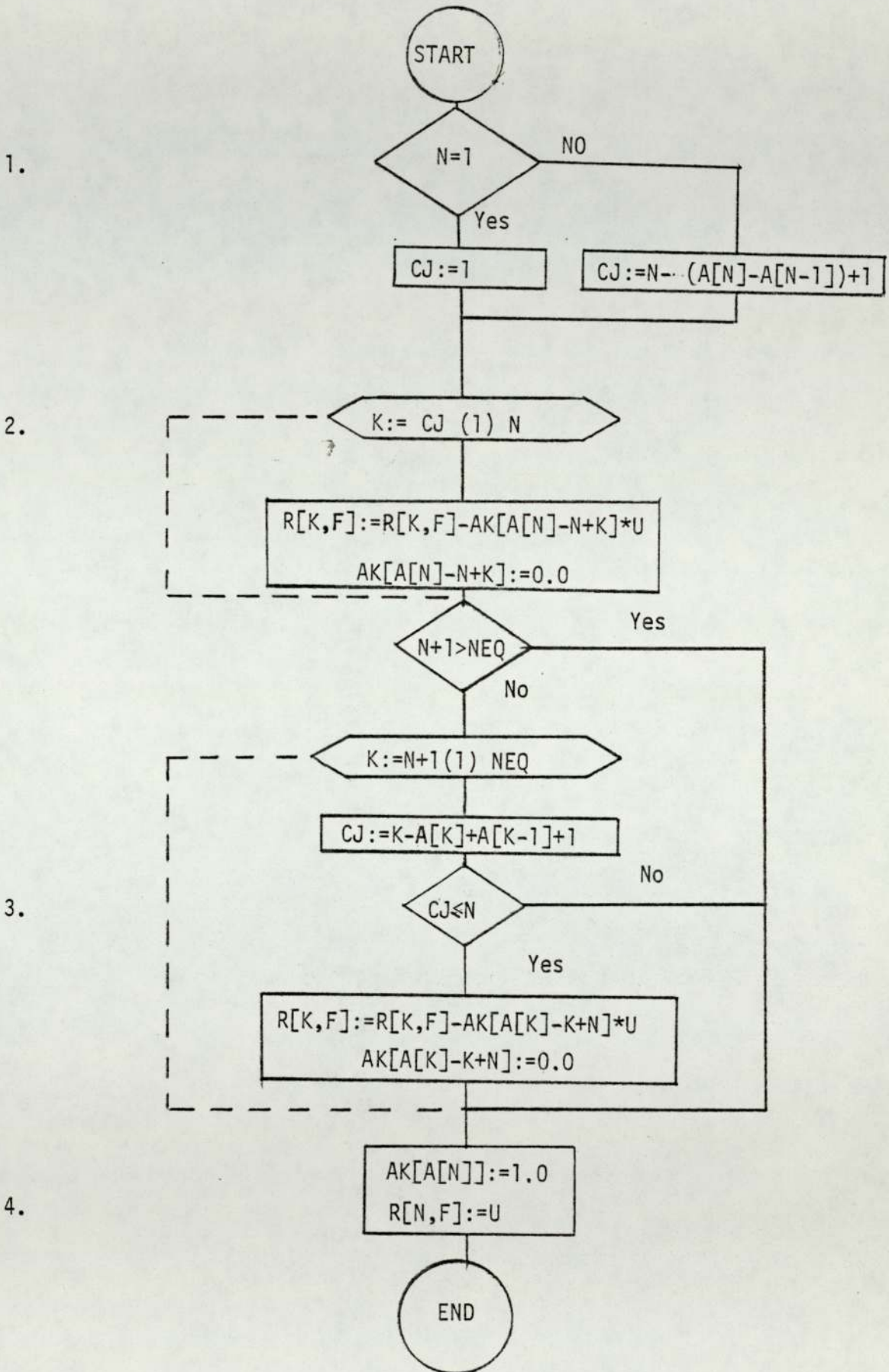
3. A check is carried out to ensure that (CJ) is not greater than the degree of freedom being considered (N).
4. The diagonal term of the stiffness matrix is made unity,

$$K_{NN} = 1 \quad (7.22)$$

and the prescribed displacement is inserted automatically into the load vector.

$$Q_N = q_N \quad (7.23)$$

Procedure BOUNCONST Flowchart



7.4 Procedure CONSTREL

This procedure generates the elasticity matrix [C] which relates the stresses and the strains for the two cases of isotropic and anisotropic stratified materials discussed in section (3.5). It is called independently for each element so that inclusions of different material properties can be included anywhere in the structure. The following steps show how this procedure achieves this objective for any element:

1. A code is read to distinguish between an isotropic or an anisotropic stratified material.
2. The elasticity constants for an isotropic material are read as:

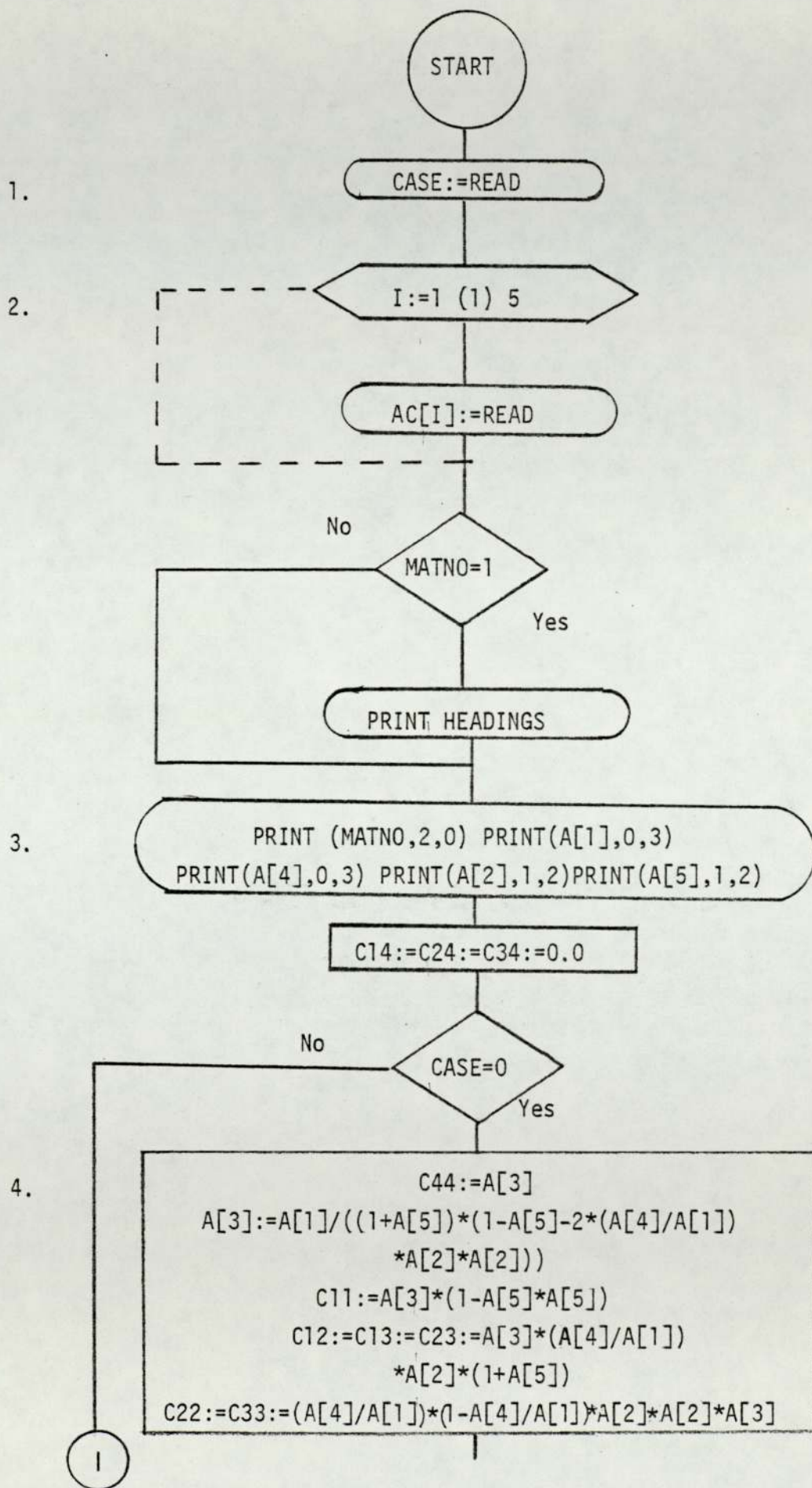
$$E, \nu, \mu, E, \nu$$

the E and ν are read twice to give a regularity of presentation with the anisotropic case, and for an anisotropic stratified material they are read as:

$$E_1, \nu_1, \mu_2, E_2, \nu_2$$

3. The elastic constants are outputed.
4. The [C] matrix for an isotropic material is formulated according to equation (2.24).
5. The [C] matrix for an anisotropic stratified material is formulated according to equation (3.38).
6. Due to symmetry of the [C] matrix only ten coefficients are stored in an array Z[1:NMAT,1:10] in which these coefficients for any given material are stored in the row corresponding to that material's number.

Procedure CONSTREL Flowchart



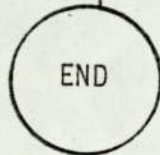


5.

```
P:=(1+A[2])*(1-A[2]-  
2*(A[2]*A[5]))  
C1:=A[1]/P  
C11:=C12:=C*(1-(A[2]*A[5]))  
C12:=C*(A[2]*(A[2]*A[5]))  
C13:=C23:=(C*A[2])*(1+A[2])  
C33:=C*(A[1]/A[4]*(1-A[2]  
*A[2]))  
C44:=A[4]/(2*(1-A[5]))
```

6.

```
Z[MATNO,1]:=C11    Z[MATNO,2]:=C12  
Z[MATNO,3]:=C13    Z[MATNO,4]:=C14  
Z[MATNO,5]:=C22    Z[MATNO,6]:=C23  
Z[MATNO,7]:=C24    Z[MATNO,8]:=C33  
Z[MATNO,9]:=C34    Z[MATNO,10]:=C44
```



7.5 Evaluation and assembly of the overall stiffness matrix

7.5.1 Introduction

The overall stiffness matrix is sparsely populated and symmetric, hence a big saving in computer space can be achieved by storing only the terms between the first non-zero term in each row and the leading diagonal as a one dimensional array. This scheme is also convenient because the method of solving the set of linear simultaneous equations used requires these terms only, as will be seen in section (7.6). To achieve this an address sequence was developed,[82], and will be used in conjunction with procedure ASSEMBLY which evaluates the element stiffness matrices, assembles the overall stiffness matrix, and stores it as a one dimensional array.

7.5.2 Procedure ADDARRAY

The purpose of this procedure is to determine the coefficients of an address array (ADD) which relates the overall stiffness matrix coefficients in two dimensional form to their one dimensional form as:

$$K[i,j] = K[ADD(i)-i + j] \quad (7.24)$$

The size of (ADD) is from zero to the number of degrees of freedom of the structure being considered, i.e. to the number of rows of the overall stiffness matrix. Each coefficient of (ADD) is the accumulative number of terms between the first non-zero term and the leading diagonal in the overall stiffness matrix in all rows up to and including the row corresponding to this coefficient. For a node (A), the corresponding (ADD) coefficient is found by scanning the nodal connections of the first element containing node (A), and if the lowest node

number in it is (B), the number of terms between the first non-zero and the leading diagonal of that particular row is:

$$\text{Number} = A - B + 1 \quad (7.25)$$

If in a later element a larger number was found for this row, it will replace the smaller one so that the maximum number is found. These numbers are then accumulated to give the coefficients of (ADD).

The following steps show how this task is accomplished:

1. For the first element, the nodal connections are compared with each other to find the smallest node number, and the number of terms between the first non zero one and the major diagonal for each node in this element i.e. in each row of the overall stiffness matrix, is calculated from equation (7.25).
2. Step (1) is repeated for the rest of the elements, and if the number of terms for a node was found to be greater than the previously stored one, it will be stored instead.
3. The coefficients of (ADD) are determined from the numbers evaluated by step (2).

7.5.3 Procedure ASSEMBLY

This procedure evaluates the element stiffness matrices by performing the integration involved numerically using Gauss quadrature, then assembles the overall stiffness matrix as a one dimensional array using procedure (ADDARRAY). It was shown in section (3.6) that the element stiffness matrix is given by:

$$[K]_e = \int_0^1 \int_0^{1-L_2} [B]^t [C] [B] 2\pi r \det[J] dL_1 dL_2 \quad (7.26)$$

and that

$$I = \frac{1}{A} \int_A f(L_1, L_2, L_3) dA \approx \sum_{i=1}^n W_i f(L_{1i}, L_{2i}, L_{3i}) \quad (7.27)$$

The order of integration, which affects the computing time and accuracy of the results, was chosen to be quadratic. This choice was guided by the experience obtained by Robertson, [9], in solving several two dimensional crack problems.

Applying the numerical integration to the element stiffness matrix formula yields:

$$[K]_e = 2\pi r \sum_{i=1}^n W_i (([B]^t [C] [B]) \det[J])_{L_{1i}, L_{2i}, L_{3i}} \quad (7.28)$$

which implies the evaluation of $([B]^t [C] [B])$ at the integration point (i) and multiplying it by the weight coefficient (W_i).

As each node of the element has two degrees of freedom, each coefficient of the element stiffness matrix is a (2x2) submatrix and the displacement and force coefficients are (2x1) vectors i.e. for a node(N):

$$\{q_N\} = \begin{Bmatrix} q_{Nr} \\ q_{Nz} \end{Bmatrix} \quad \text{and} \quad \{Q_N\} = \begin{Bmatrix} Q_{Nr} \\ Q_{Nz} \end{Bmatrix} \quad (7.29)$$

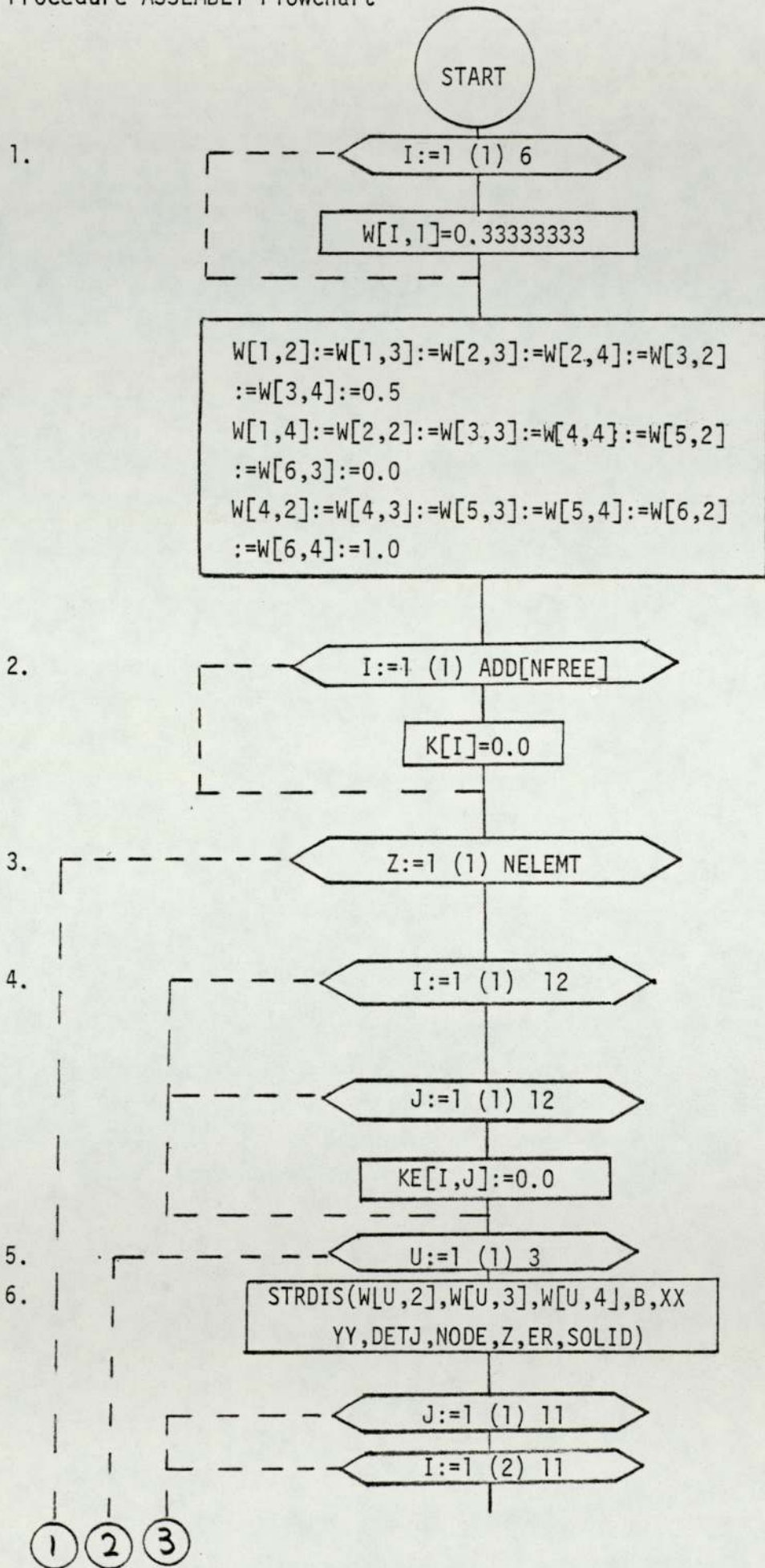
The overall stiffness matrix is assembled by scanning the nodal connections of each element and relating the local node numbers, which are from one to six, to the global numbers which are fed in via the nodal connections. The coefficients of each element stiffness matrix are allocated to their appropriate place in the overall stiffness matrix determined by their nodal connections. At this stage the address array is introduced to store the coefficients of the overall stiffness matrix as a one dimensional array.

These operations are performed by the procedure as follows:

1. An array $W[6 \times 4]$ is introduced, whose first column coefficients are the weight coefficients (W_i) and the remainder of its coefficients are the integration points.
2. Coefficients of the one dimensional array are set to zero.
3. A loop is constructed to repeat all the remaining operations of the procedure for all elements.
4. Coefficients of the element stiffness matrix are set to zero.

5. A loop for the three integration points is constructed.
6. Procedure (STRDIS) is called to evaluate $[B]$, $\det[J]$, and the radii required to generate the element stiffness matrix.
7. $[K]_e$ is evaluated from equation (7.28).
8. Using the address array (ADD) generated from procedure (ADDARRAY) and the nodal connections; the overall stiffness matrix is stored as a one dimensional array.

Procedure ASSEMBLY Flowchart



7. ① ② ③

```

KE[J,I]:=KE[I,J]:=KE[I,J]+W[U,1]
*(B[1,J]*(C[NODE[Z,7],1]*
B[1,I]+C[NODE[Z,7],2]*B[2,I]
+C[NODE[Z,7],4]*B[4,I])+B[2,J]*
(C[NODE[Z,7],2]*B[1,I]+C[NODE[Z,7],5]
*B[2,I]+C[NODE[Z,7],7]*B[4,I])+B[4,J]*
(C[NODE[Z,7],4]*B[1,I]+C[NODE[Z,7],7]
*B[2,I]+C[NODE[Z,7],10]*B[4,I]))
*2*3.1415926*RAVG*0.5*DETJ
KE[J,I+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]*
(B[1,J]*(C[NODE[Z,7],3]*B[3,I+1]
+C[NODE[Z,7],4]*B[4,I+1])+B[2,J]
*(C[NODE[Z,7],6]*B[3,I+1]+C[NODE[Z,7],7]
*B[4,I+1])+B[4,J]*(C[NODE[Z,7],9]*
B[3,I+1]+C[NODE[Z,7],10]*B[4,I+1]))
*2*3.1415926*RAVG*0.5*DETJ

```

J:=2 (2) 12

I:=2 (2) 12

```

KE[J,I]:=KE[I,J]:=KE[I,J]+W[U,1]*(B[3,J]
*(C[NODE[Z,7],8]*B[3,I]+
C[NODE[Z,7],9]*B[4,I]+B[4,J]*
(C[NODE[Z,7],9]*B[3,I]+C[NODE[Z,7],10]
*B[4,I]))*2*3.1415926*RAVG*0.5*DETJ

```

I=12

Yes

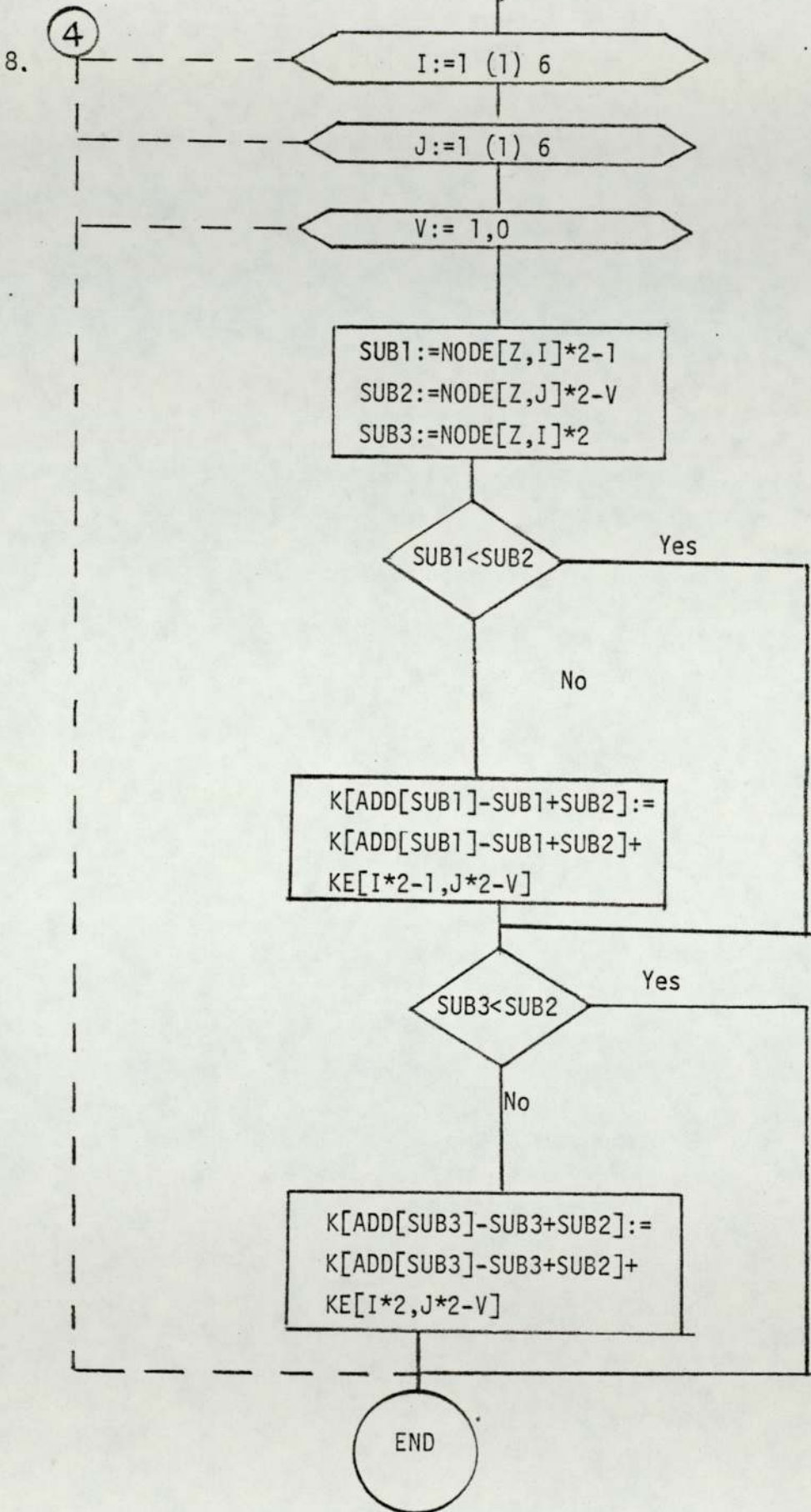
No

```

KE[J,I+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]
*(B[3,J]*(C[NODE[Z,7],3]*B[1,I+1]
+C[NODE[Z,7],6]*B[2,I+1]+C[NODE[Z,7],9]
*B[4,I+1]+B[4,J]*(C[NODE[Z,7],4]
*B[1,I+1]+C[NODE[Z,7],7]*B[2,I+1]
+C[NODE[Z,7],10]*B[4,I+1]))*2*
3.1415926*RAVG*0.5*DETJ

```

④



7.6 Procedure SYMVBSOL

Choleski's method of triangular factorization is used to solve the set of variable band width equations stored by the scheme of Jennings and Tuff as a one dimensional array,[82]. The method makes use of the fact that the zeroes before the first non zero term in any row remain zeroes if no row or column interchange is made during reduction by Gaussian Elimination.

Choleski's triangular factorization yields the following equations:

$$[L][L]^t = [K] \quad (7.30)$$

where

[L] = lower triangular matrix with
positive diagonal terms.

The system of equations to be solved becomes:

$$[L][L]^t\{q\} = \{Q\} \quad (7.31)$$

Assuming that

$$[Y] = [L]^t\{q\} \quad (7.32)$$

and substituting back in (7.31) yields

$$[L][Y] = \{Q\} \quad (7.33)$$

The variables in [Y] are modified right hand side coefficients after elimination. The back substitution process to complete the solution is finding {q} from:

$$[L]^t\{q\} = [Y] \quad (7.34)$$

The matrix [L] overwrites [K] in the store by using the following relations:

$$L_{ii} = \{K_{ii} - \sum_{k=1}^{i-1} L_{ik}^2\}^{1/2} \quad (7.35)$$

for diagonal terms, and

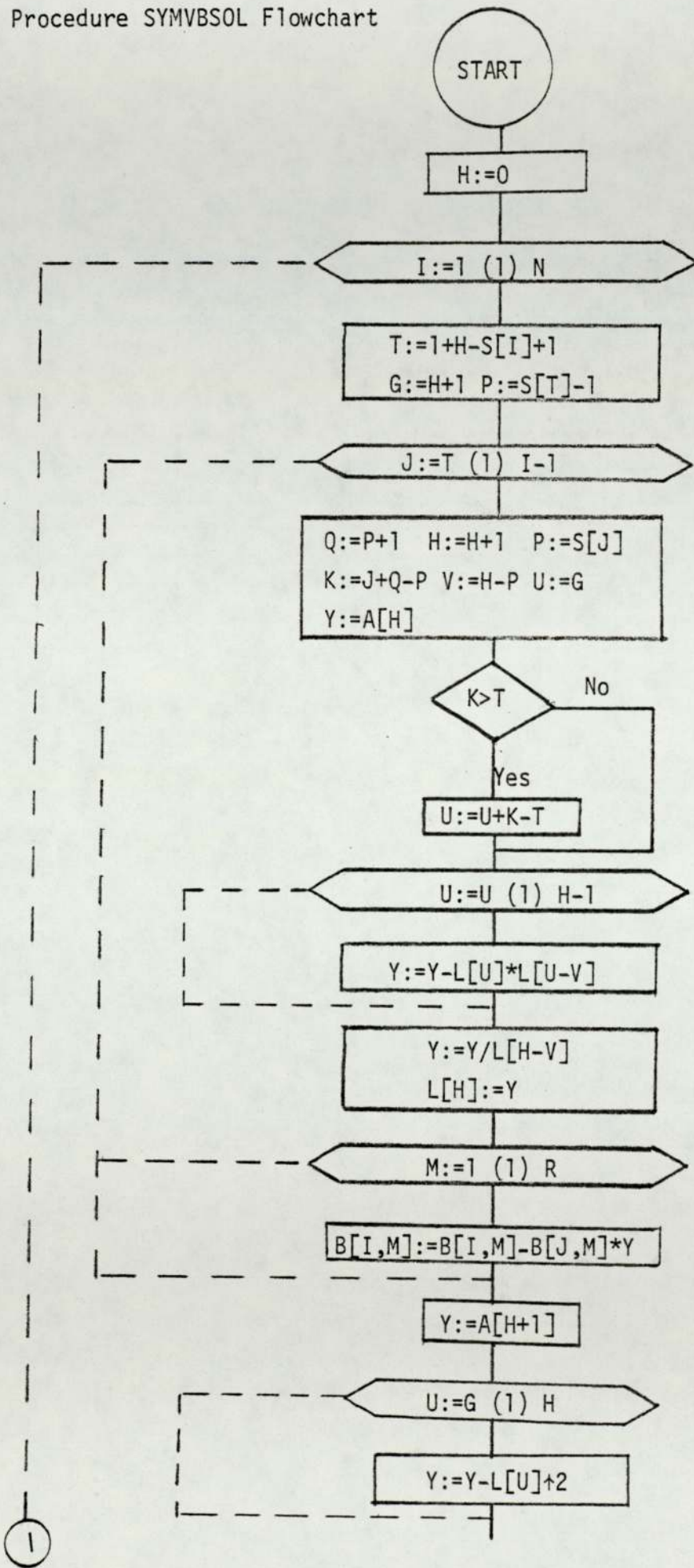
$$L_{ij} = \{K_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{jk}\}/L_{jj} \quad (7.36)$$

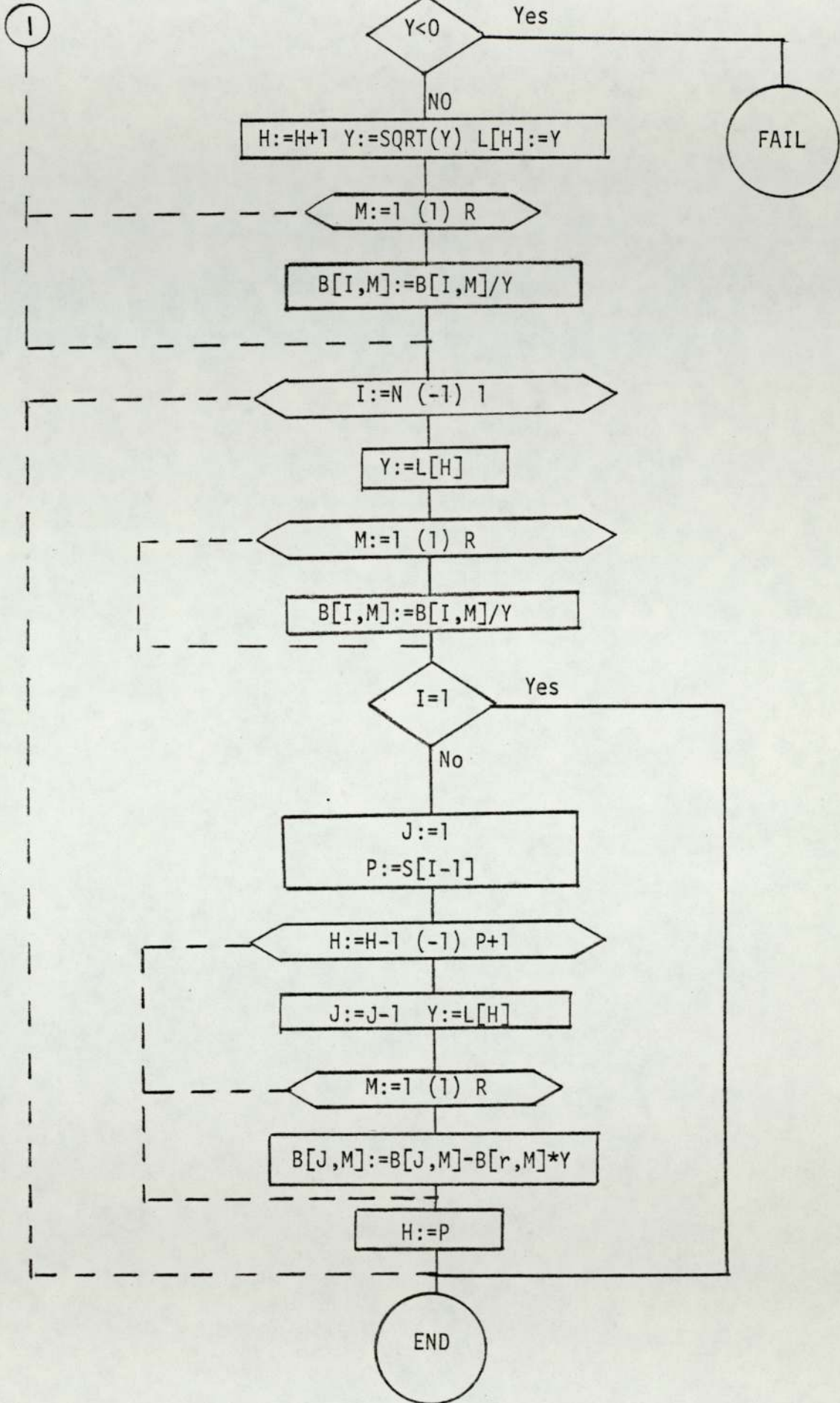
for off diagonal terms.

Equations (7.35) and (7.36) show that only row (i) and column (j) need be stored at a time for any coefficient (L_{ij}) and hence computer back store may be used for a large system of equations.

No steps to explain the flowchart of the procedure will be presented as it was taken as a package from reference, [82].

Procedure SYMVBSOL Flowchart





2

7.7 Procedure NODSTR

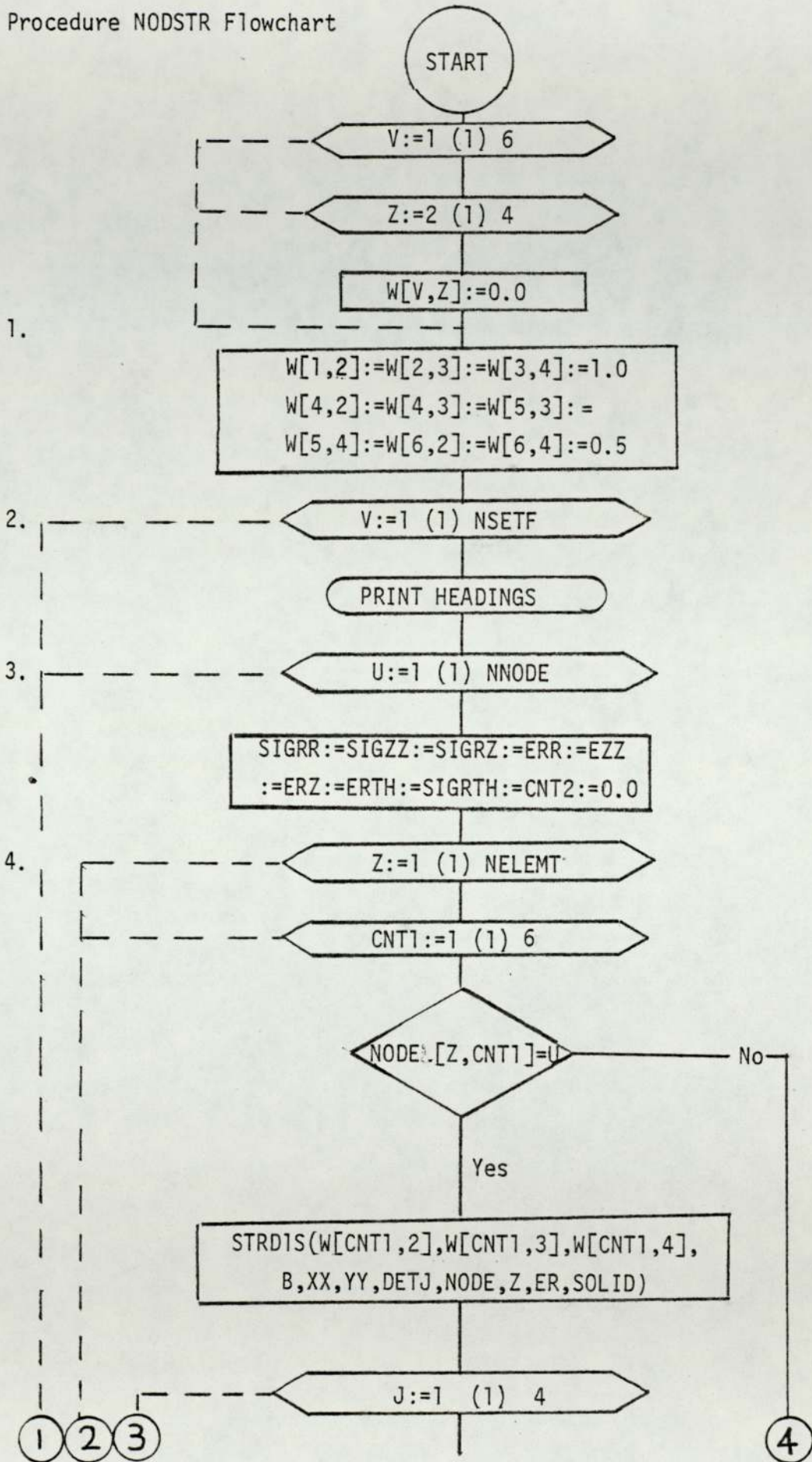
This procedure evaluates the strains and stresses at each node of the discretized structure. The strains at nodes which are shared by more than one element are evaluated by averaging the contributions from the elements sharing the node.

The coefficients of matrix $[B]$, which relates the strains to displacements, are determined at nodal points by calling procedure (STRDIS). Having obtained the nodal displacements $\{q\}$ earlier, the strains are evaluated using equation (3.33). The stresses are obtained from the strains using matrix $[C]$ of section (7.4) and equation (3.36).

These operations are performed by the following steps:

1. Coefficients of array $[W]$ (6×4) are set to the natural coordinates of nodal points.
2. A loop around the sets of forces i.e. for the number of load cases, is constructed.
3. A loop around the number of nodes is constructed.
4. A loop around the number of elements, takes each one in turn and checks if the node considered in step (3) exists and if so the strains and stresses are evaluated as described earlier. The contribution of other elements to the strains and stresses of that node are added and the result is averaged.
5. The averaged result of the strains and stresses is outputted.

Procedure NODSTR Flowchart



1

2

3

4

```

STR[J]:=(B[J,1]*Q[NODE[Z,1]*2-1,V]
+B[J,2]*Q[NODE[Z,1]*2,V]
+B[J,3]*Q[NODE[Z,2]*2-1,V]
+B[J,4]*Q[NODE[Z,2]*2,V]
+B[J,5]*Q[NODE[Z,3]*2-1,V]
+B[J,6]*Q[NODE[Z,3]*2,V]
+B[J,7]*Q[NODE[Z,4]*2-1,V]
+B[J,8]*Q[NODE[Z,4]*2,V]
+B[J,9]*Q[NODE[Z,5]*2-1,V]
+B[J,10]*Q[NODE[Z,5]*2,V]
+B[J,11]*Q[NODE[Z,6]*2-1,V]
+B[J,12]*Q[NODE[Z,6]*2,V])

```

```

J:=NODE[Z,7]
SIGRR:=SIGRR+(C[J,1]*STR[1]+C[J,2]*STR[2]
+C[J,3]*STR[3]+C[J,4]*STR[4])
SIGRTH:=SIGRTH+(C[J,2]*STR[1]+C[J,5]*STR[2]
+C[J,6]*STR[3]+C[J,7]*STR[4])
SIGZZ:=SIGZZ+(C[J,3]*STR[1]+C[J,6]*STR[2]
+C[J,8]*STR[3]+C[J,9]*STR[4])
SIGRZ:=SIGRZ+(C[J,4]*STR[1]*C[J,7]*STR[2]
+C[J,9]*STR[3]+C[J,10]*STR[4])
ERR:=ERR*STR[1] ERTH:=ERTH+STR[2]
EZZ:=EZZ+STR[3] ERZ:=ERZ+STR[4]
CNT2=CNT2+1

```

```

SIGRTH:=SIGRTH/CNT2 SIGRR:=SIGRR/CNT2
SIGZZ:=SIGZZ/CNT2 SIGRZ:=SIGRZ/CNT2
ERTH:=ERTH/CNT2 ERR:=ERR/CNT2
EZZ:=EZZ/CNT2 ERZ:=ERZ/CNT2

```

```

PRINT(U,2,0) PRINT(ERR,0,4)
PRINT(ERTH,0,4) PRINT(EZZ,0,4)
PRINT(ERZ,0,4)
PRINT(SUGRR,0,4)PRINT(SIGRTH,0,4)
PRINT(SIGZZ,0,4)PRINT(SIGRZ,0,4)

```

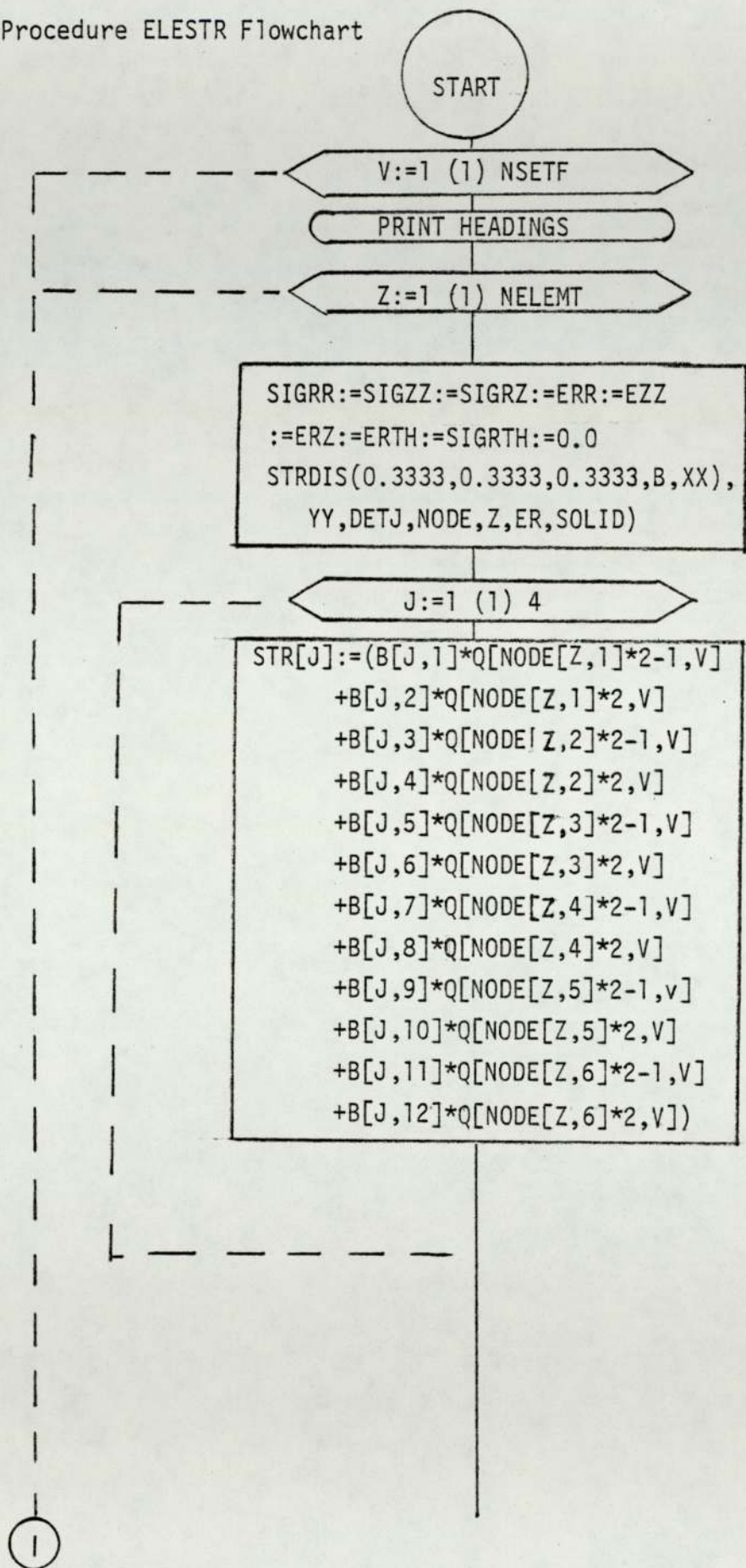
END

5.

7.8 Procedure ELESTR

This procedure evaluates the strains and stresses at the centroid of each element. The steps performed in it are identical to those of procedure (NODSTR) except that the coefficients of array [W] are those of the element centroids rather than nodal points, and therefore there is no averaging.

Procedure ELESTR Flowchart



1

```
J:=NODE[Z,7]
SIGRR:=SIGRR+(C[J,1]*STR[1]+C[J,2]*STR[2]
+C[J,3]*STR[3]+C[J,4]*STR[4])
SIGRTH:=SIGRTH+(C[J,2]*STR[1]+C[J,5]*STR[2]
+C[J,6]*STR[3]+C[J,7]*STR[4])
SIGZZ:=SIGZZ+(C[J,3]*STR[1]
+C[J,6]*STR[2]+C[J,8]*STR[3]
+C[J,9]*STR[4])
SIGRZ:=SIGRZ+(C[J,4]*STR[1]+C[J,7]*STR[2]
+C[J,9]*STR[3]+C[J,10]*STR[4])
ERR:=ERR+STR[1]  ERTH:=ERTH+STR[2]
EZZ:=EZZ+STR[3]  ERZ:=ERZ+STR[4]
```

```
PRINT(Z,2,0) PRINT(ERR,0,4)
PRINT(ERTH,0,4) PRINT(EZZ,0,4)
PRINT(ERZ,0,4)
PRINT(SUGRR,0,4) PRINT(SIGRTH,0,4)
PRINT(SIGZZ,0,4) PRINT(SIGRZ,0,4)
```

END

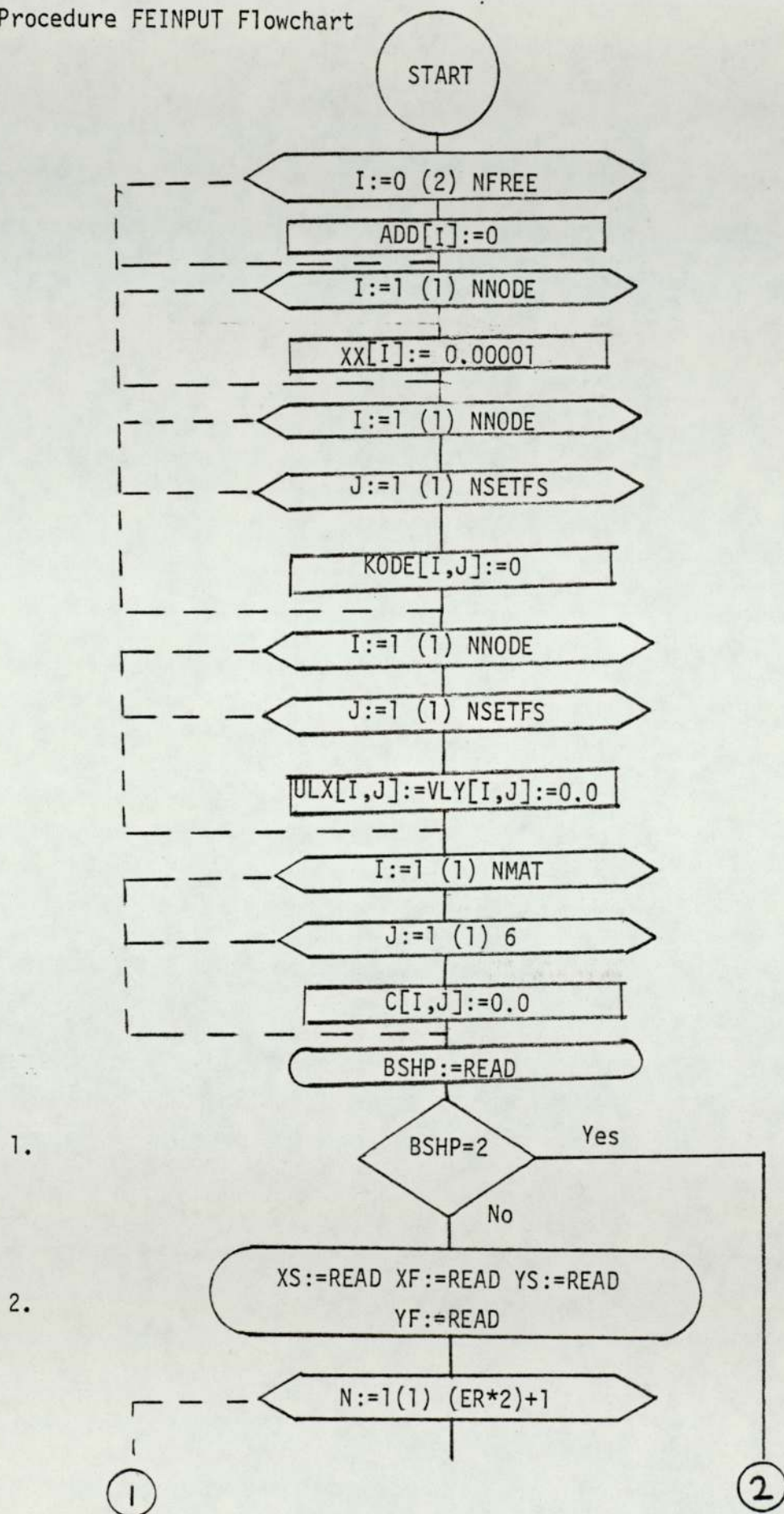
7.9 Procedure FEINPUT

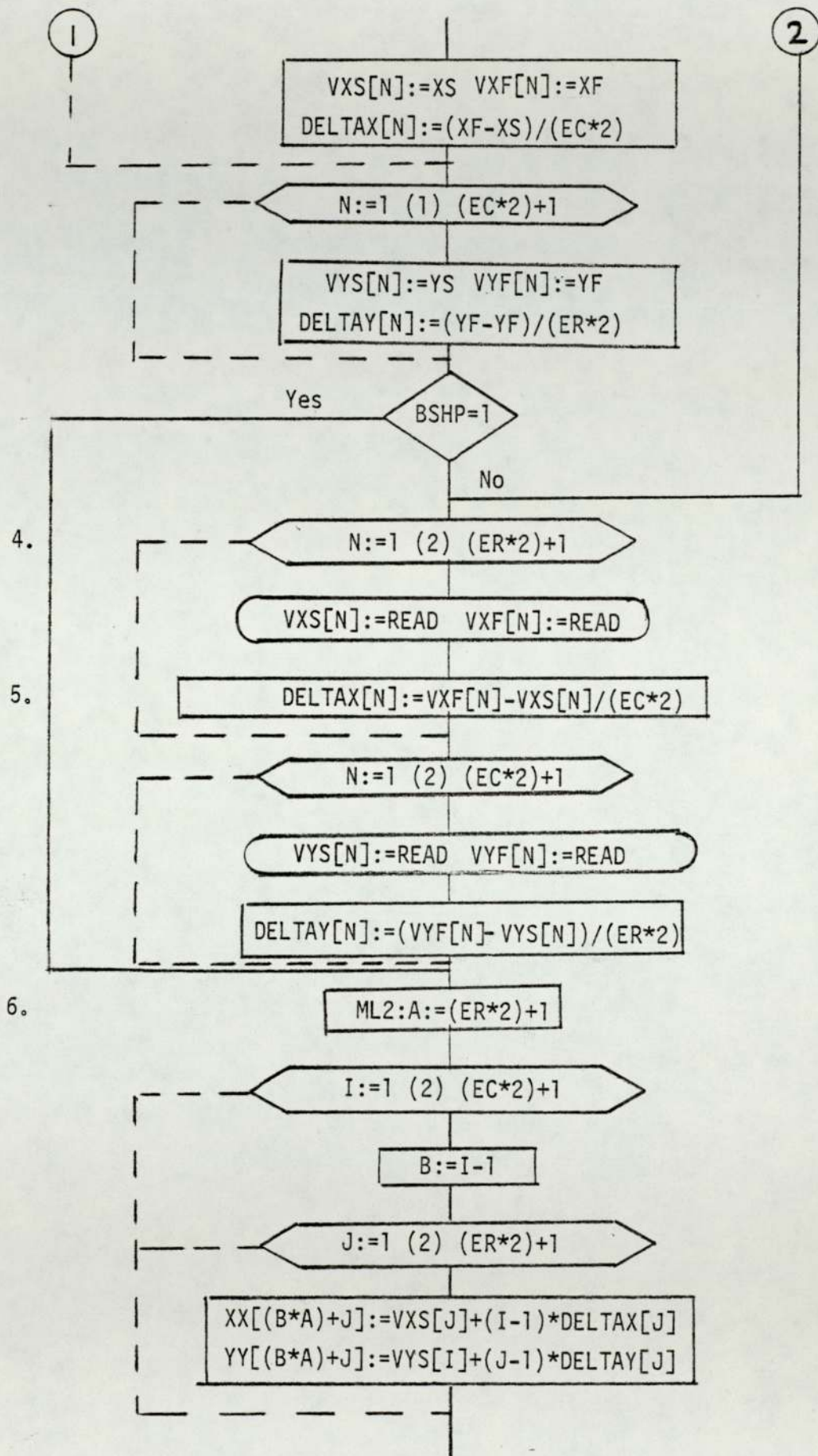
This procedure generates the mesh required for the general axisymmetric program and evaluates the nodal connections of each element as described in section (3.8). The nodes with prescribed loads or displacements are read in as data together with the values of the components of loads or displacements in the global r and/or z directions. The following steps show how the computer code for the automatic mesh generation is developed:

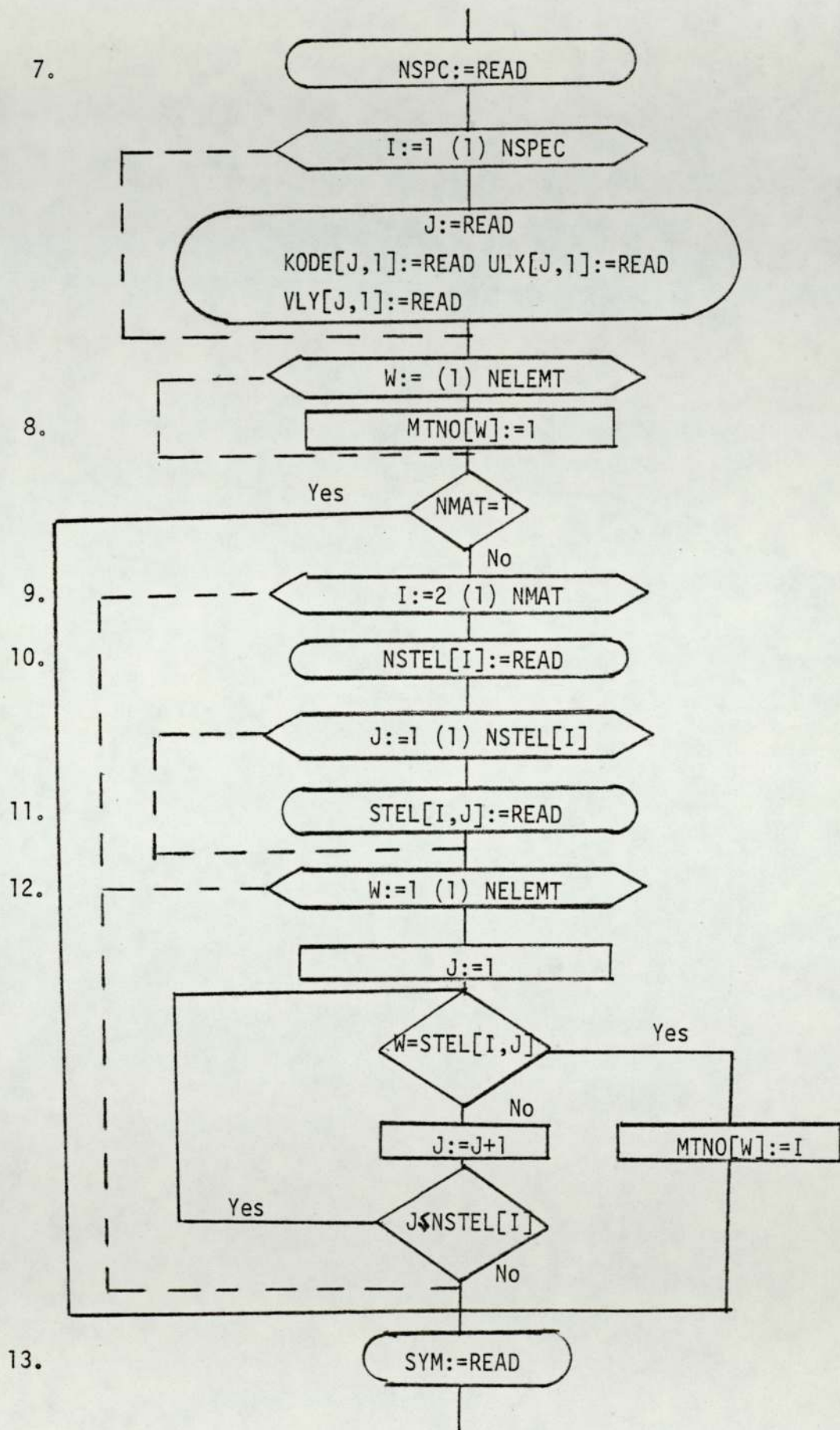
1. A code (BSHP) is introduced to specify whether the longitudinal section is rectangular or not.
2. If the section is rectangular, the length of its sides is determined by reading the r and z coordinates of two perpendicular sides.
3. The distances between the nodes are evaluated from equations (3.66) and (3.62).
4. If the section is not rectangular the r and z coordinates of all the main nodal rows and columns are read.
5. The distances between the nodes are evaluated similarly to step (3).
6. The nodal coordinates of the vertices of the triangular section elements are evaluated.
7. Nodes with prescribed loads or displacements are read together with the values of components of the loads and/or displacements and a code to distinguish between them and their direction.

8. All the material numbers are set to 1.
9. If the structure consists of more than one material then a loop I is constructed from 2 to the number of materials.
10. The number of elements in each material portion are read for each material.
11. The element numbers of these elements are read.
12. A loop is constructed around the element numbers which specifies the appropriate material number to each one as the seventh coefficient of the nodal connections array as discussed in section (3.8.2).
13. A code (SYM) is read to see if symmetry with respect to a diametral plane is required.
14. The nodal connections are assembled according to the code of step (13).
15. The midside nodal coordinates are evaluated from the coordinates of the vertices corresponding to each side according to the connections obtained from step (14).
16. Nodal coordinates together with the code and prescribed loads and/or displacements of step (7) and the nodal connection of step (14) are outputted.

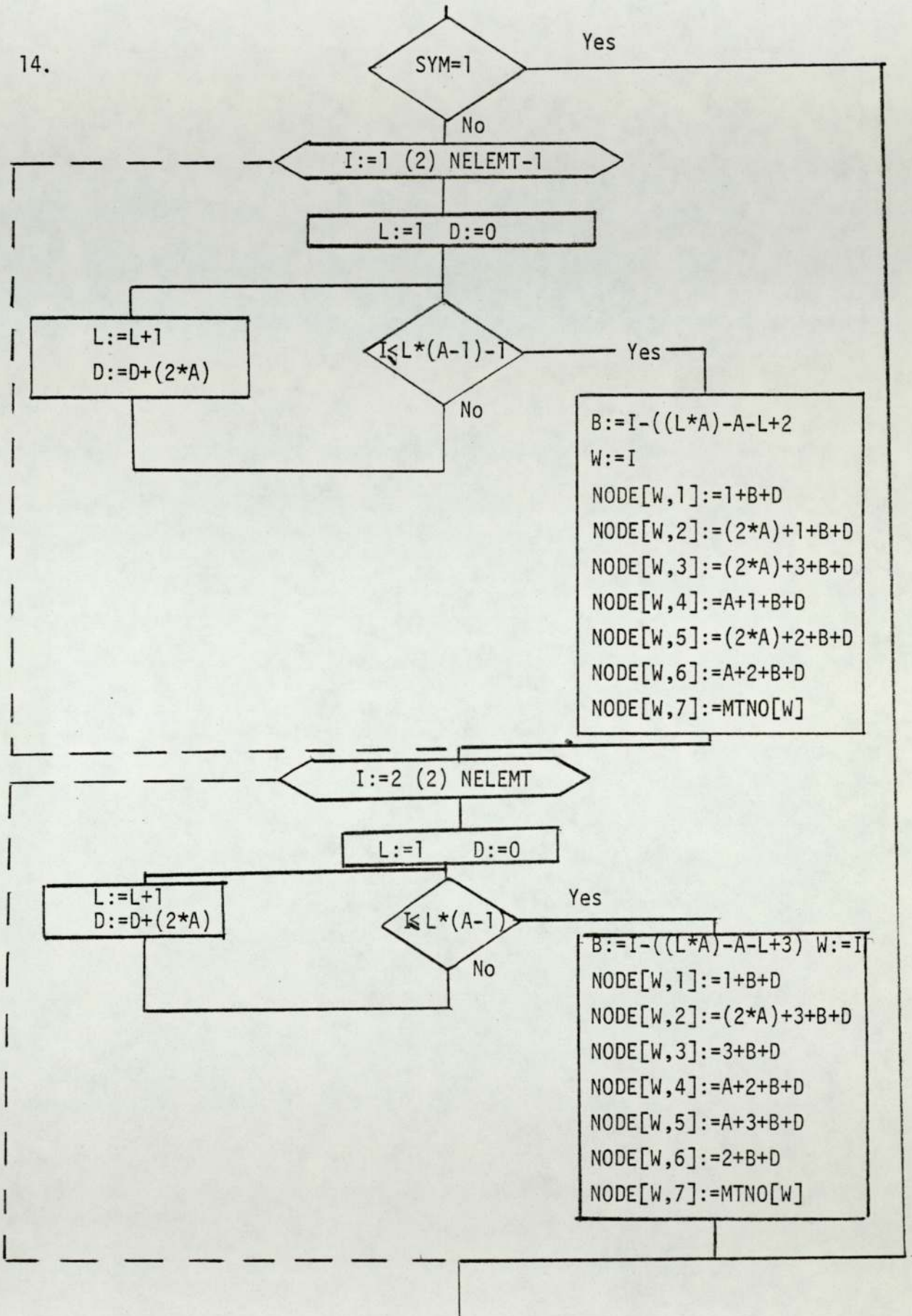
Procedure FEINPUT Flowchart

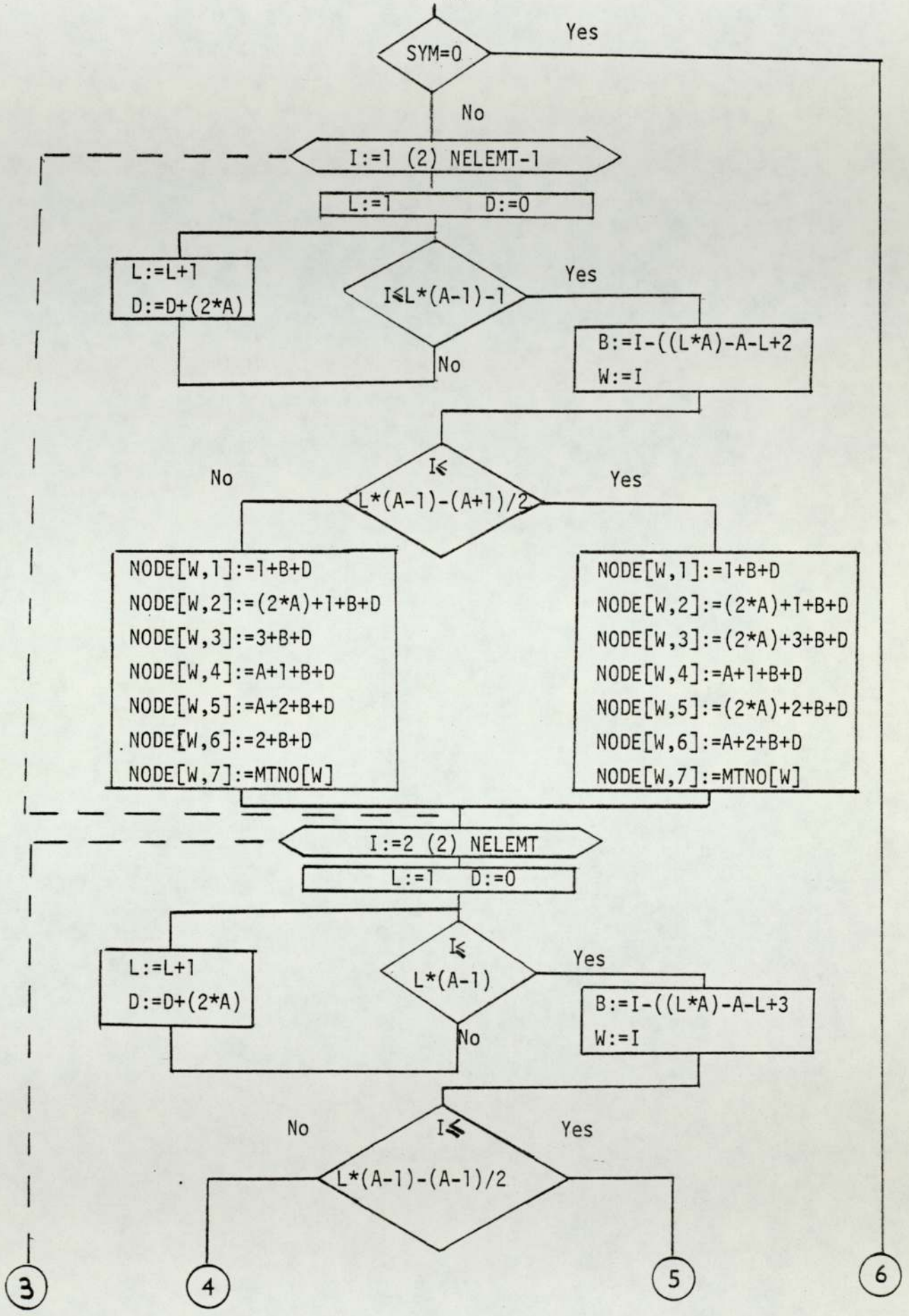


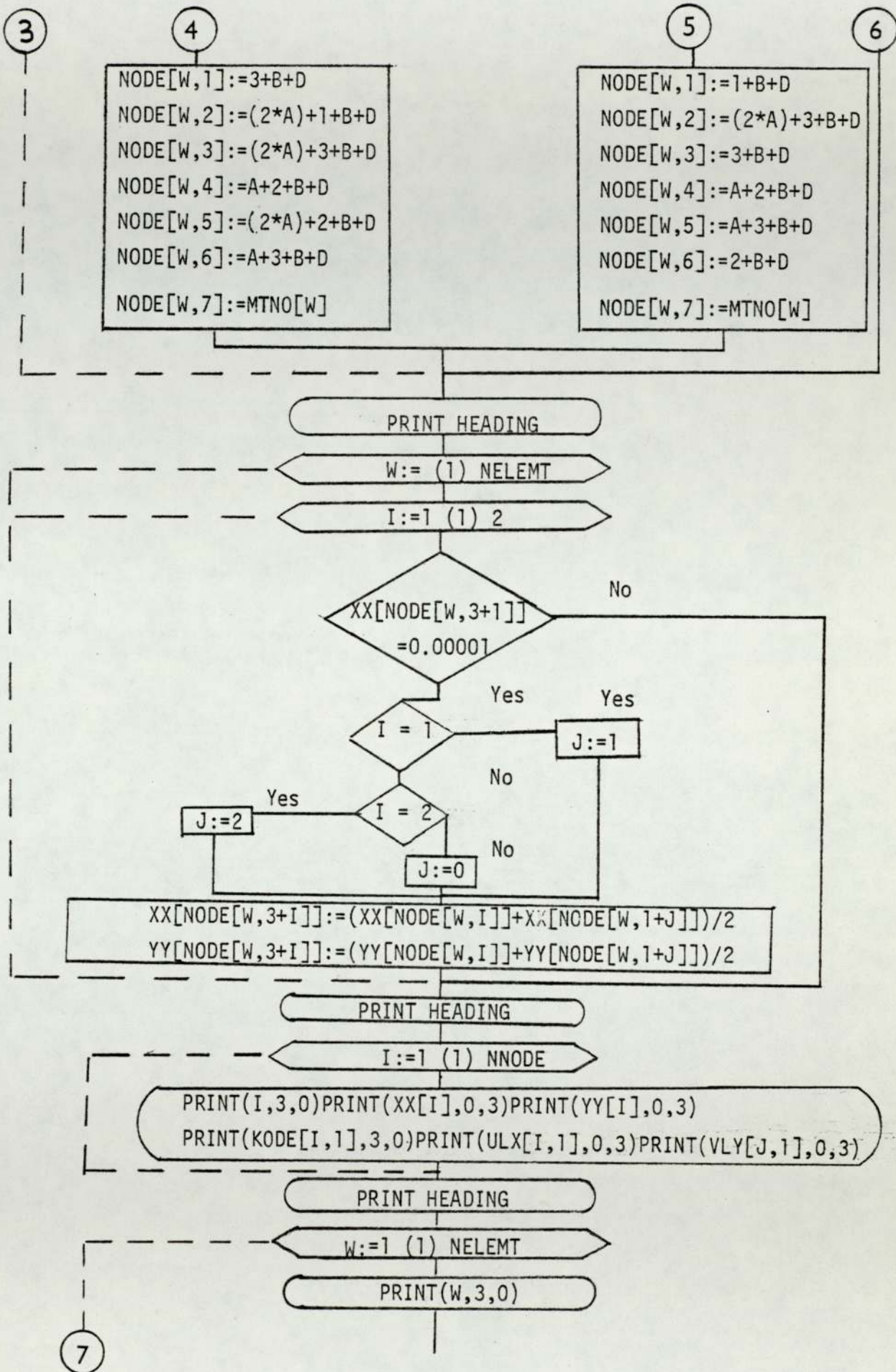


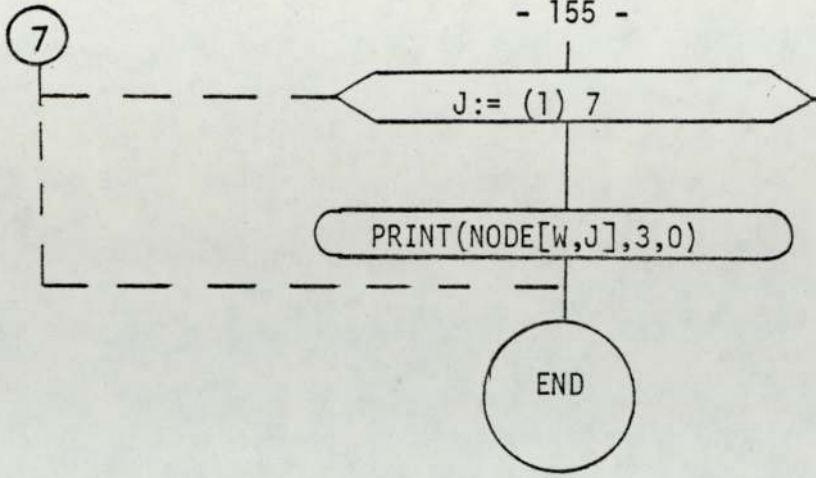


14.







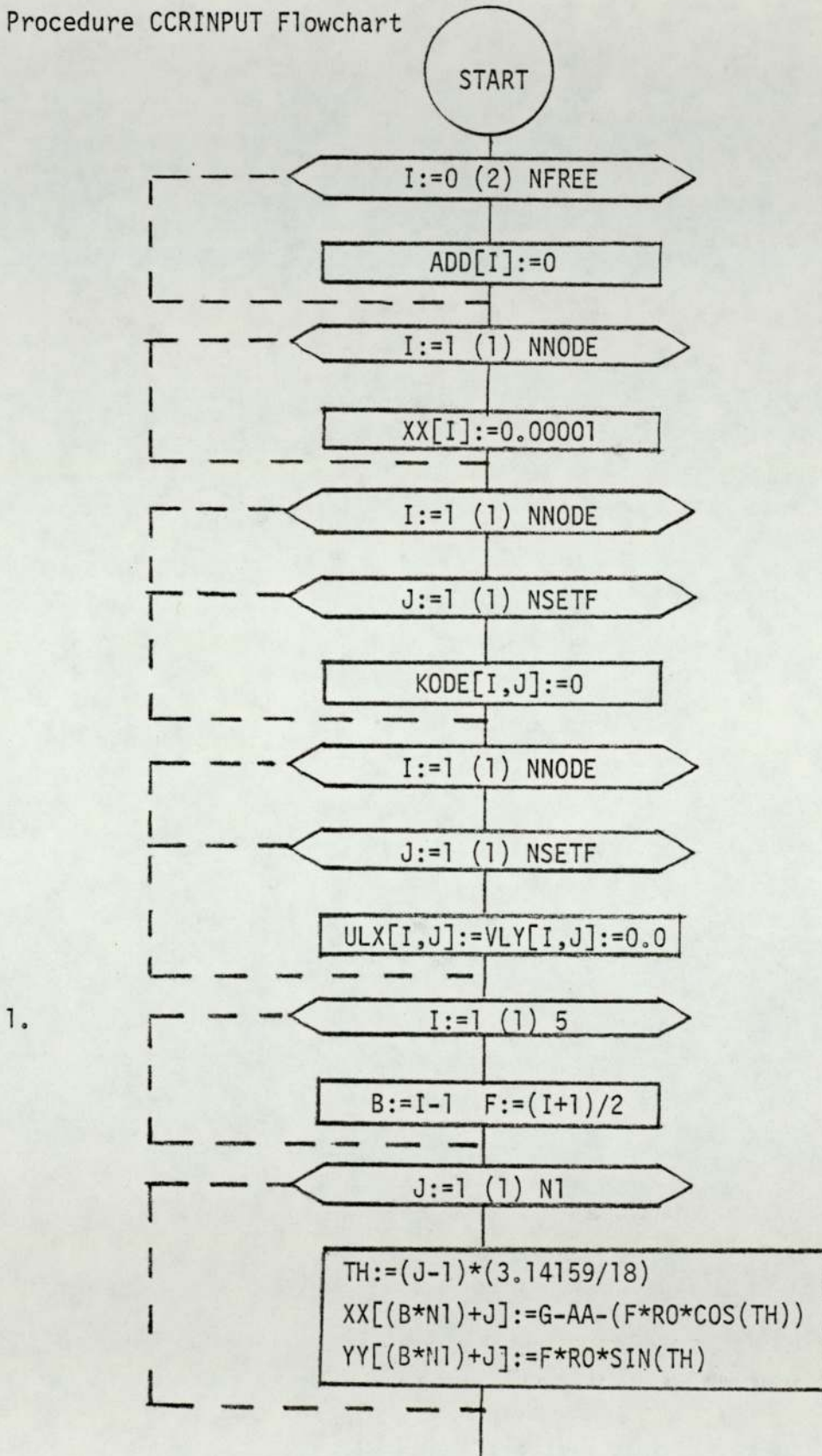


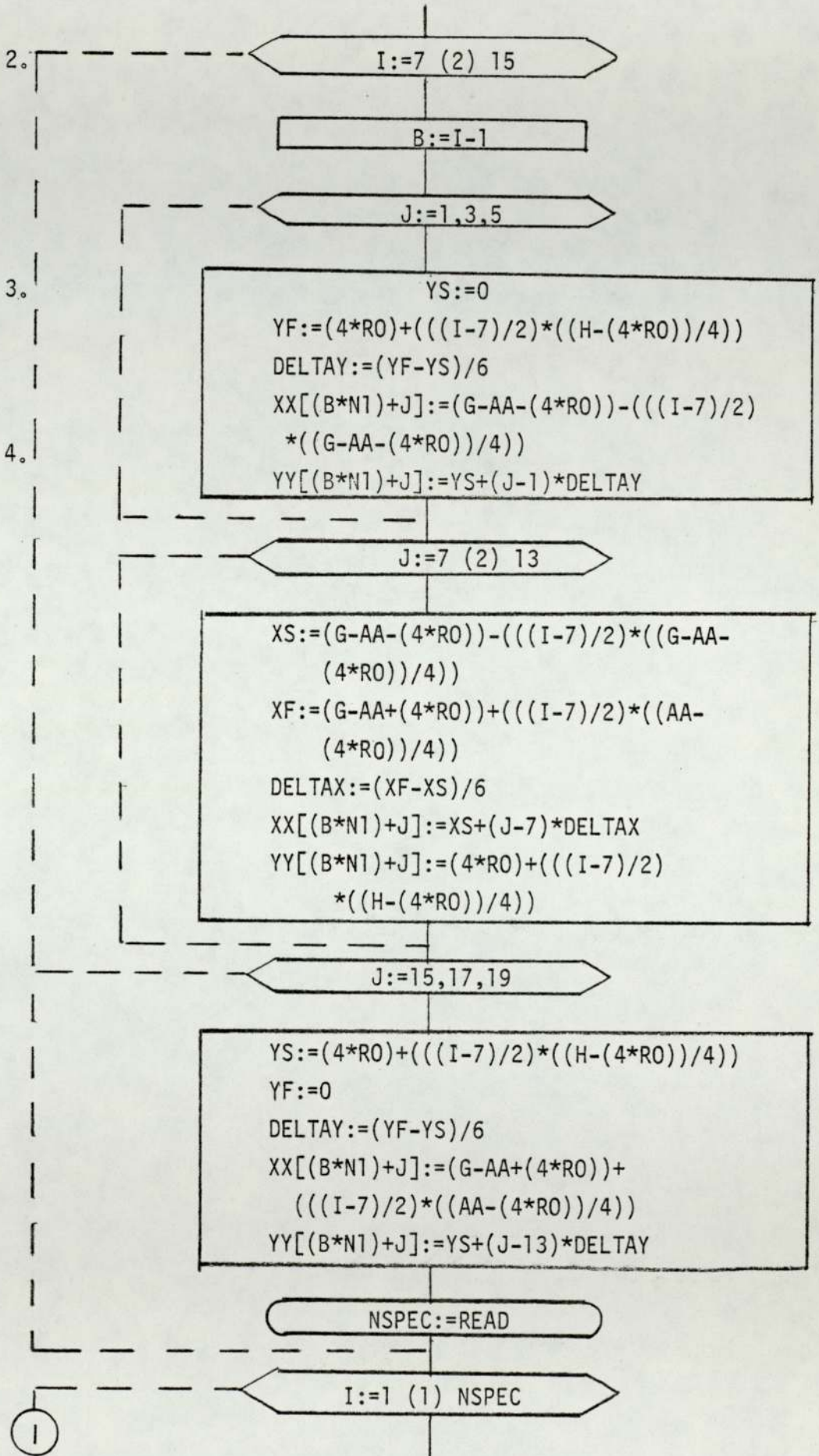
7.10 Procedure CCRINPUT

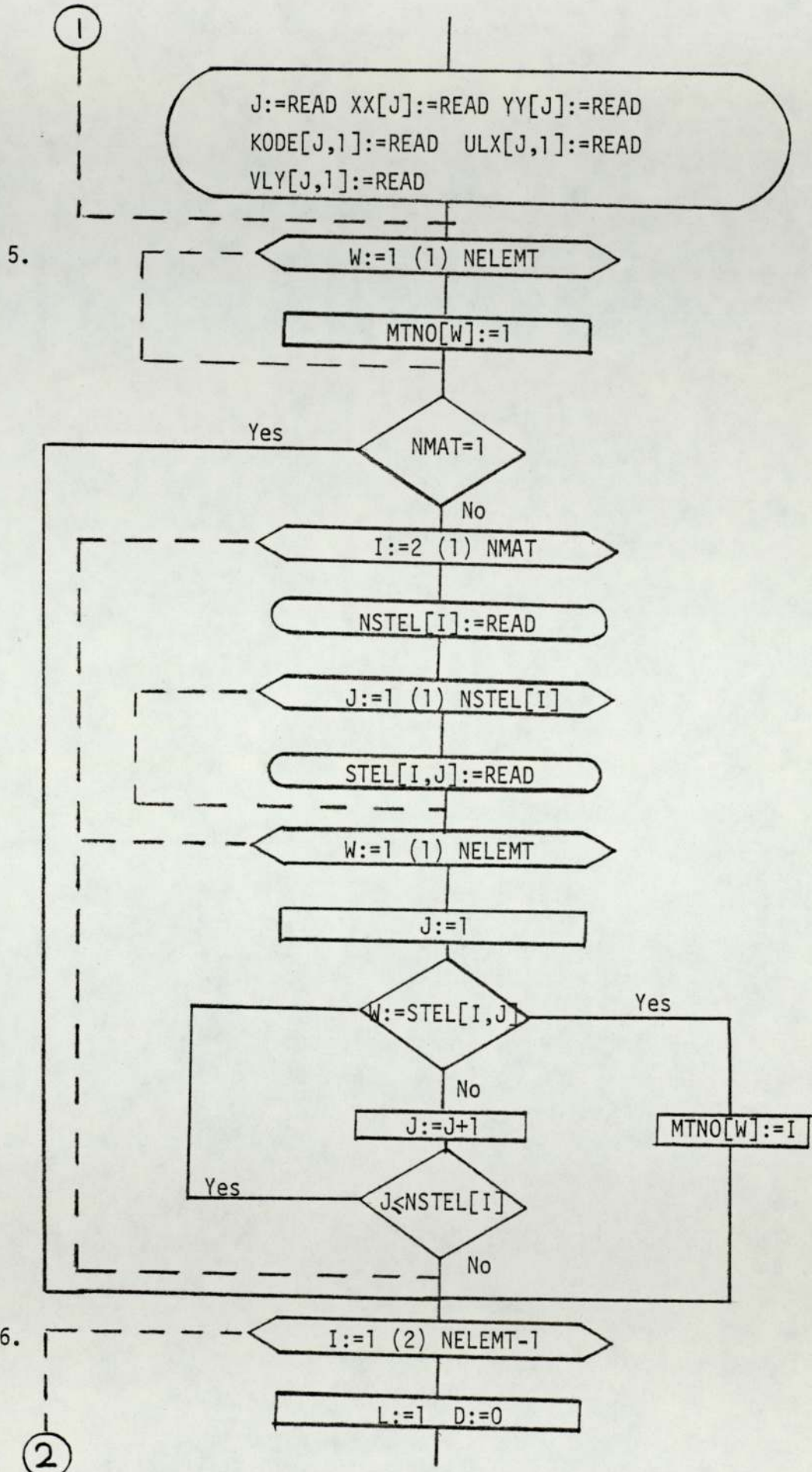
This procedure generates the mesh required for solving mode (I) fracture problems by incorporating the core element. Two rings of elements surrounding a semicircular core are constructed by bending element columns into a semicircular shape as described in section (3.8.3). The remainder of the element columns are bent into rectangular shapes to match the boundaries of the structure. The steps are as follows:

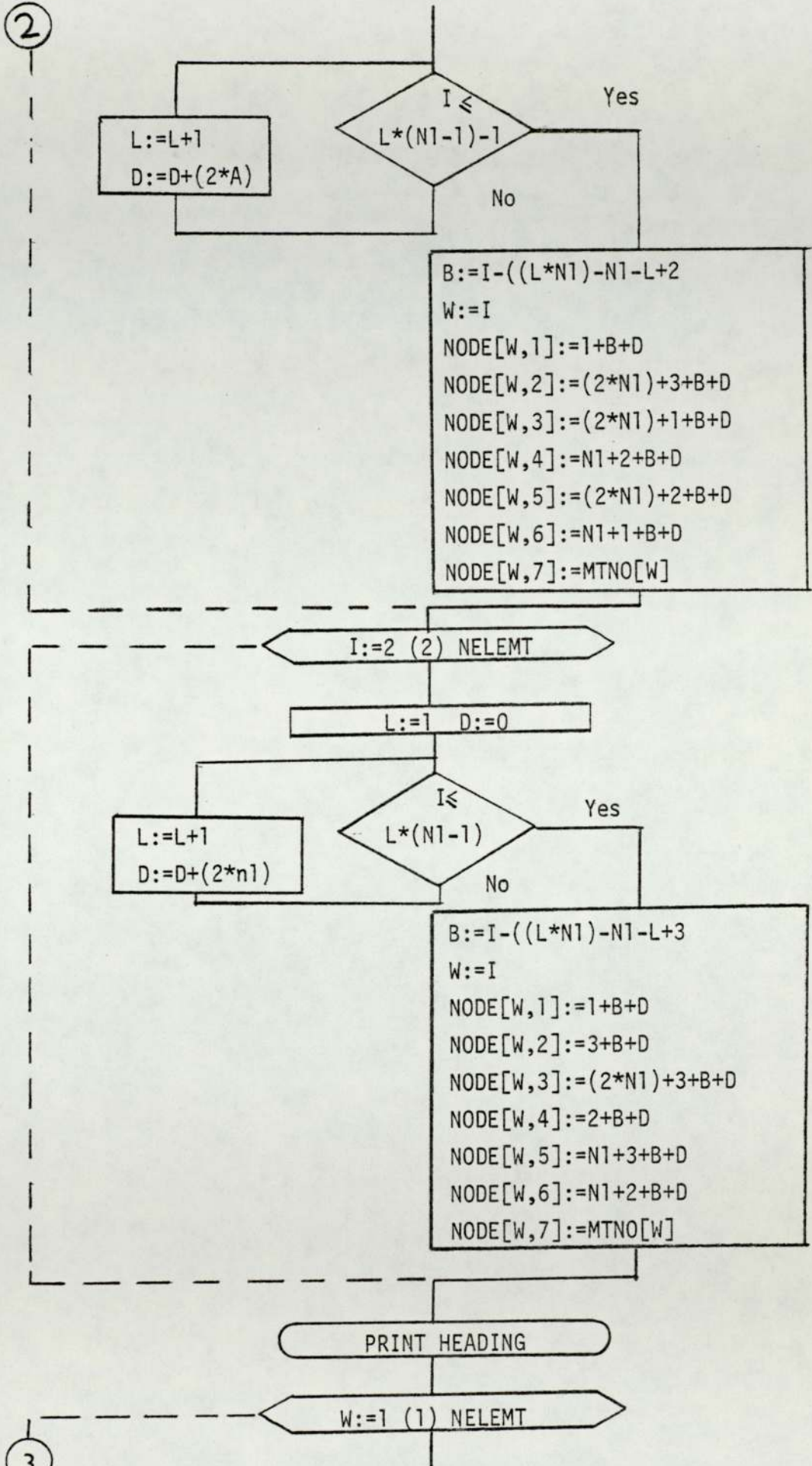
1. For the first five nodal columns, i.e. the first two element columns, the angle (θ) of each nodal point measured from the r-axis with the crack tip as origin is calculated and hence the nodal points global coordinates are evaluated.
2. For the remainder of the nodal columns, the number of main nodes on each side of the new rectangular shape is specified, and the position of each nodal columns with respect to the structure as a whole is fixed.
3. The distance between the nodes for each side of each rectangle is evaluated.
4. The main nodes global coordinates are evaluated.
5. To allow for different material properties, the same steps as (7) to (12) of section (7.9) are followed.
6. In this case only a quadrant of the solid is discretized and hence symmetry with respect to the crack plane is implied, therefore the nodal connections are assembled similarly to step (14) of section (7.9) without the code (SYM).
7. The same steps as (15) and (16) of section (7.9) are followed.

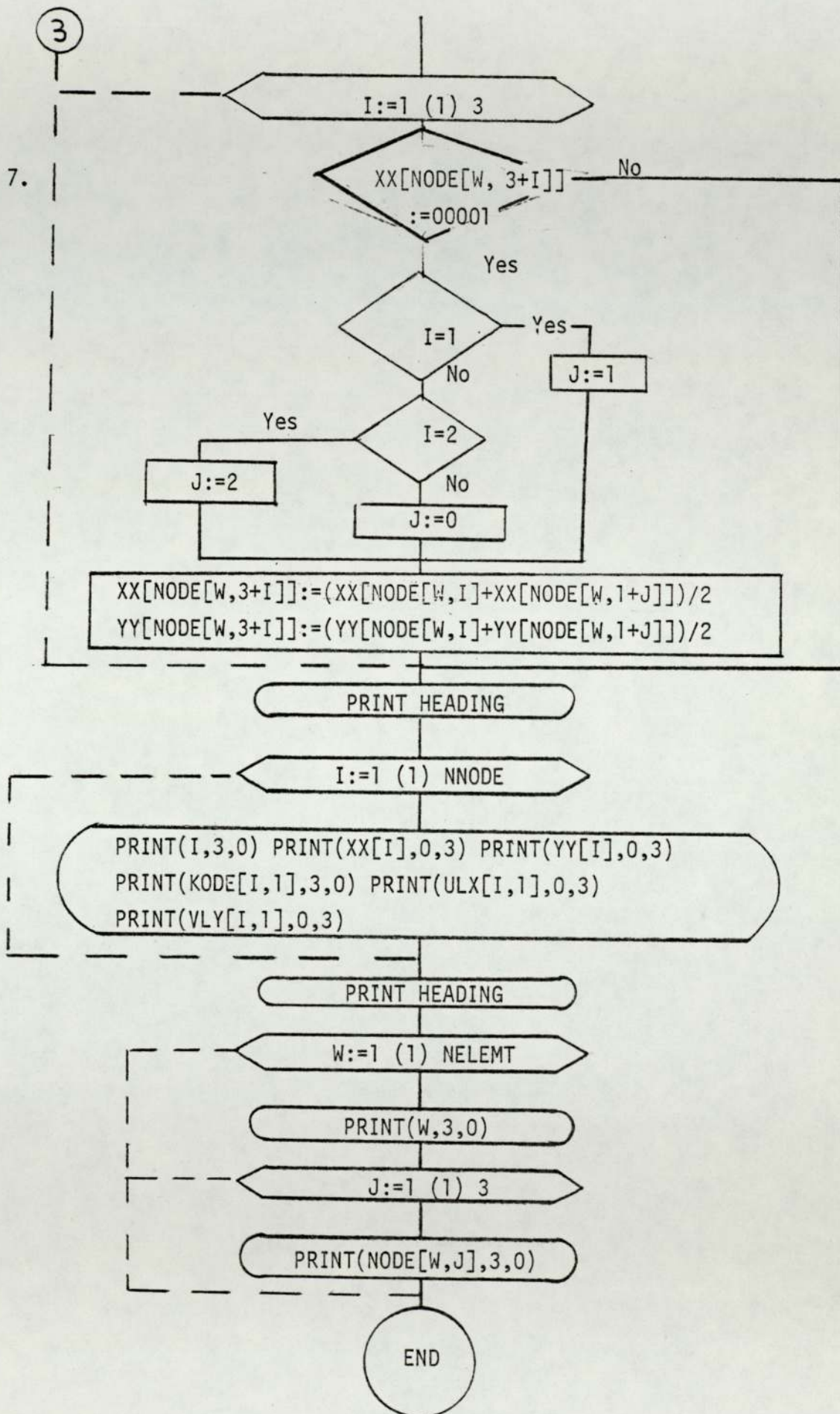
Procedure CCRINPUT Flowchart











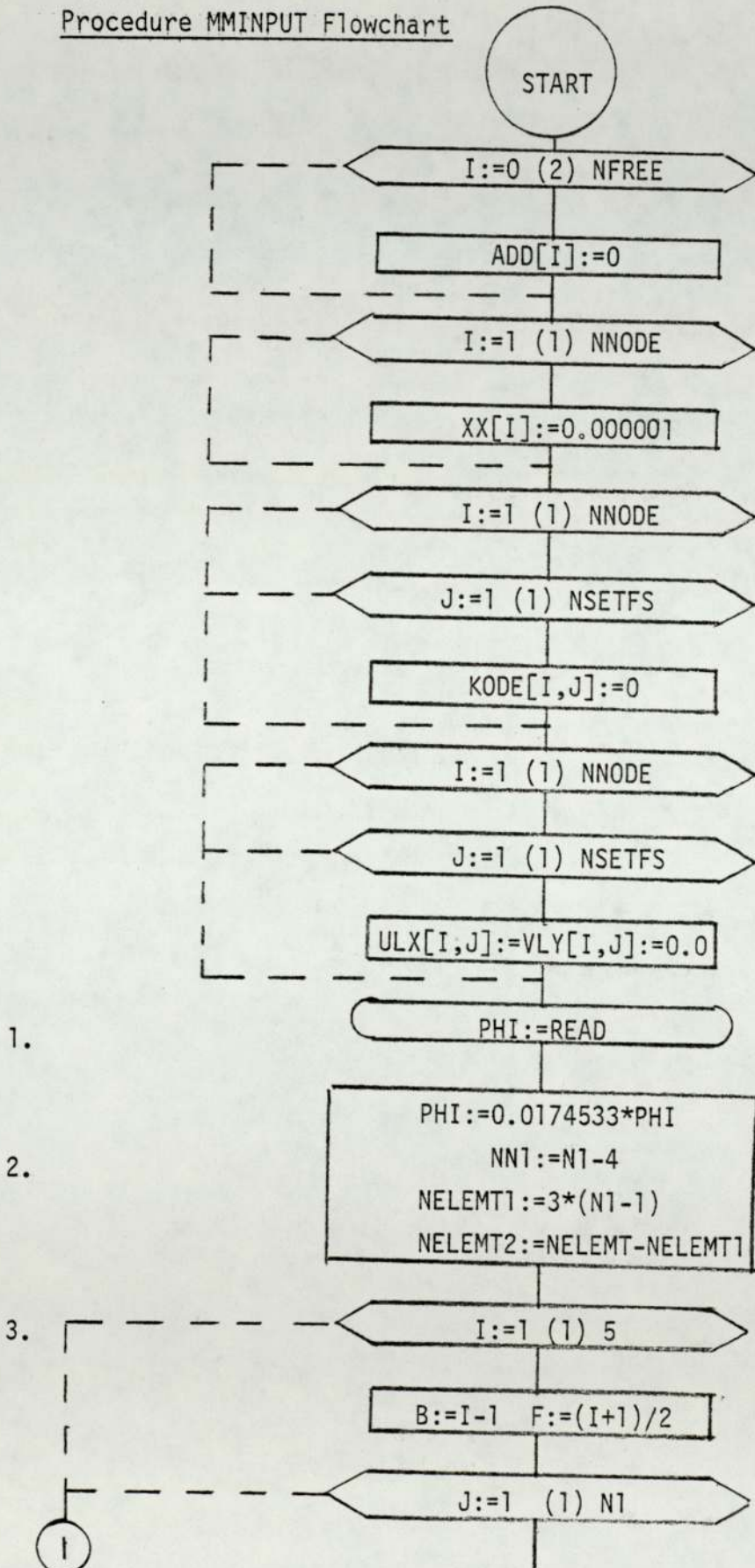
7.11 Procedure MMINPUT

This procedure generates the mesh required for a mixed mode I and II problem with the core having a circular shape and the crack is inclined with various angles to the r-axis. The first five nodal columns are bent into circular shapes around the core and the seventh one is bent into a rectangular shape enclosing the circles. The remaining columns surround three sides of the rectangle only by truncating the nodes corresponding to the fourth side to save on computer storage and still obtain a rectangular shape. It will be shown later that mesh symmetry with respect to the crack plane is required near the crack tip, therefore it will be generated for the first three element columns. The procedure steps are as follows:

1. The angle of crack inclination with the r-axis is read.
2. The number of nodes of the truncated nodal columns is worked out.
3. The nodal coordinates of the first five nodal columns are evaluated.
4. The nodal coordinates for the main nodes of the seventh nodal columns, which is a rectangle enclosing the circular ones, are evaluated.
5. The nodal coordinates for the main nodes of the truncated columns are evaluated.
6. Nodes with prescribed loads and/or displacements are read and allowance for different material properties is made similar to section (7.9)

7. The nodal connections are generated with symmetry with respect to the crack plane for the first three element columns.
8. The remaining steps are similar to section (7.9).

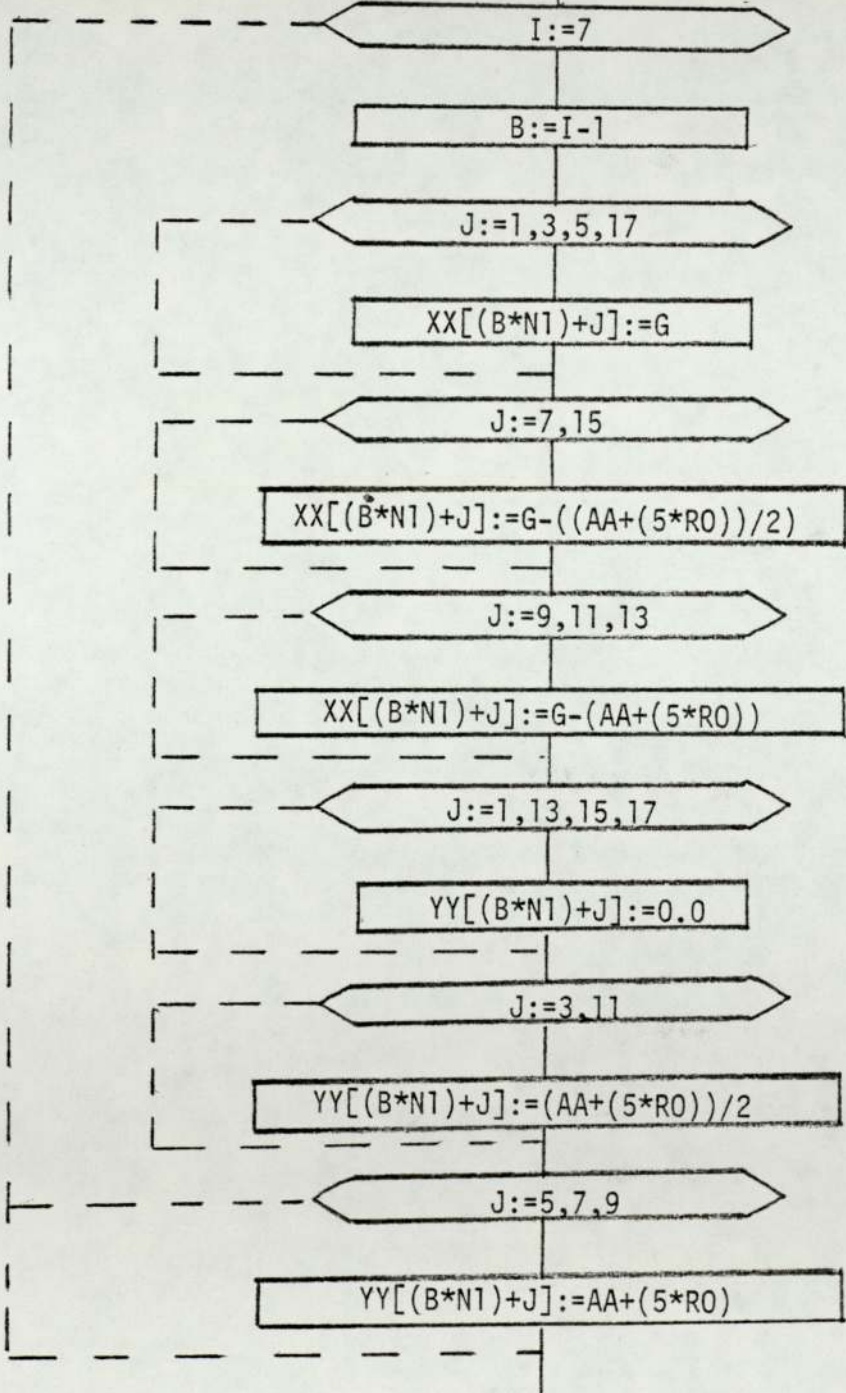
Procedure MMINPUT Flowchart

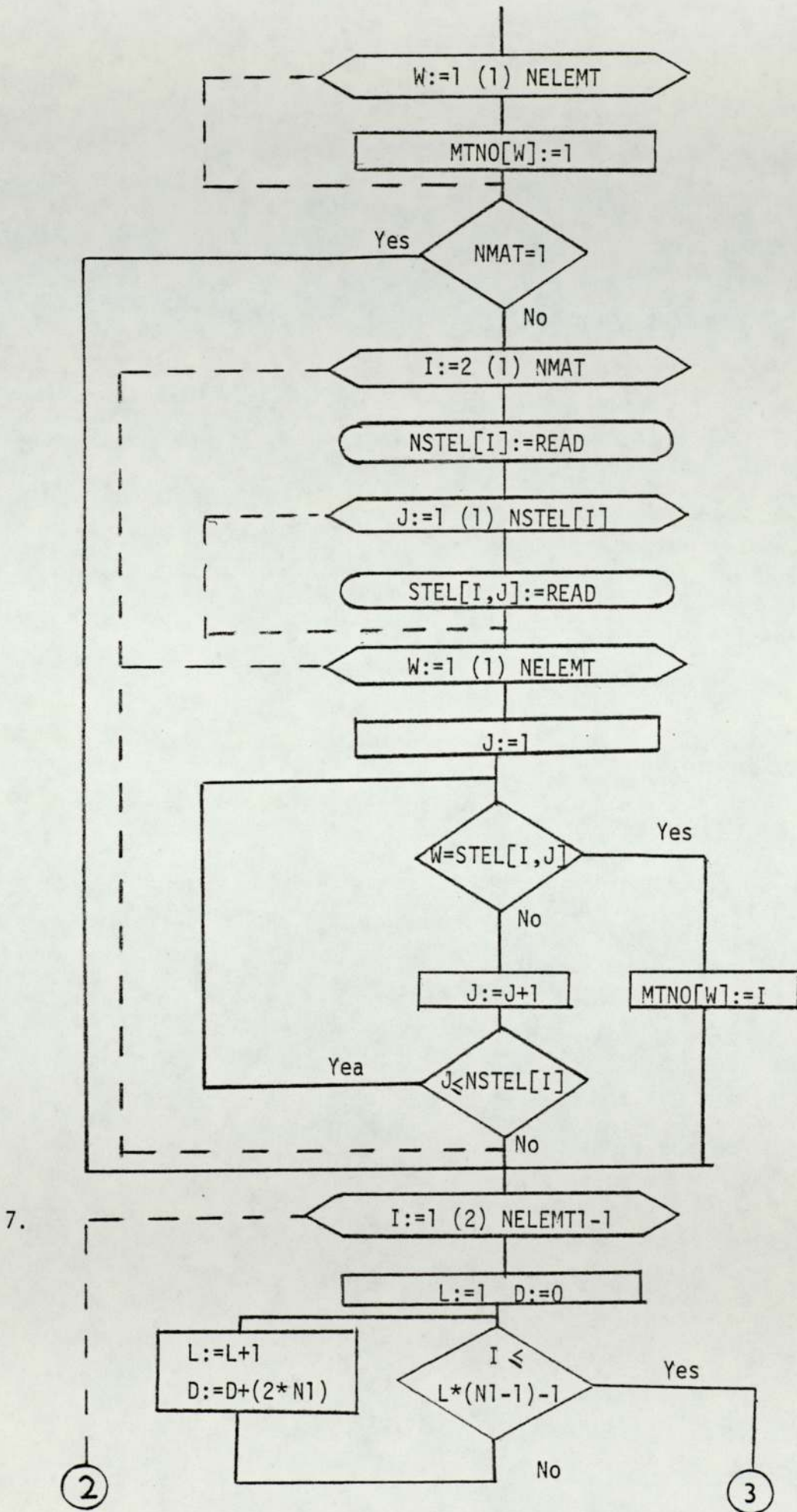


①

```
TH:=(2*(J-1)*3.14159)/(N1-1)
XX[(B*N1)+J]:=(G-(AA*COS(PHI)))+(((F*RO)
  *COS(TH))*COS(PHI))+(((F*RO)
  *SIN(TH))*SIN(PHI))
YY[(B*N1)+J]:=((AA-(F*RO))*SIN(PHI))
  +(((F*RO)*(1-COS(TH)))*SIN(PHI))
  +(((F*RO)*SIN(TH))*COS(PHI))
```

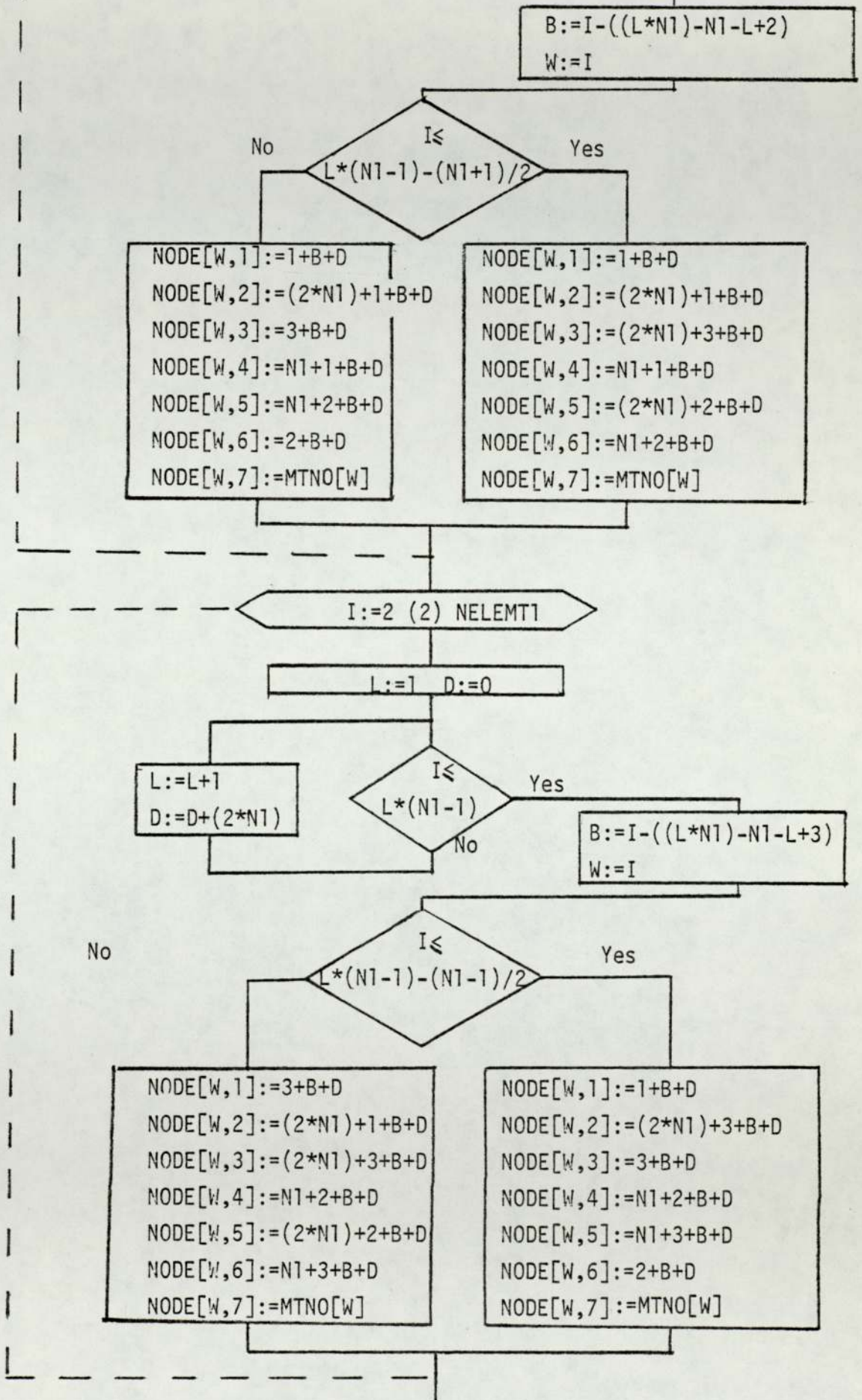
4.

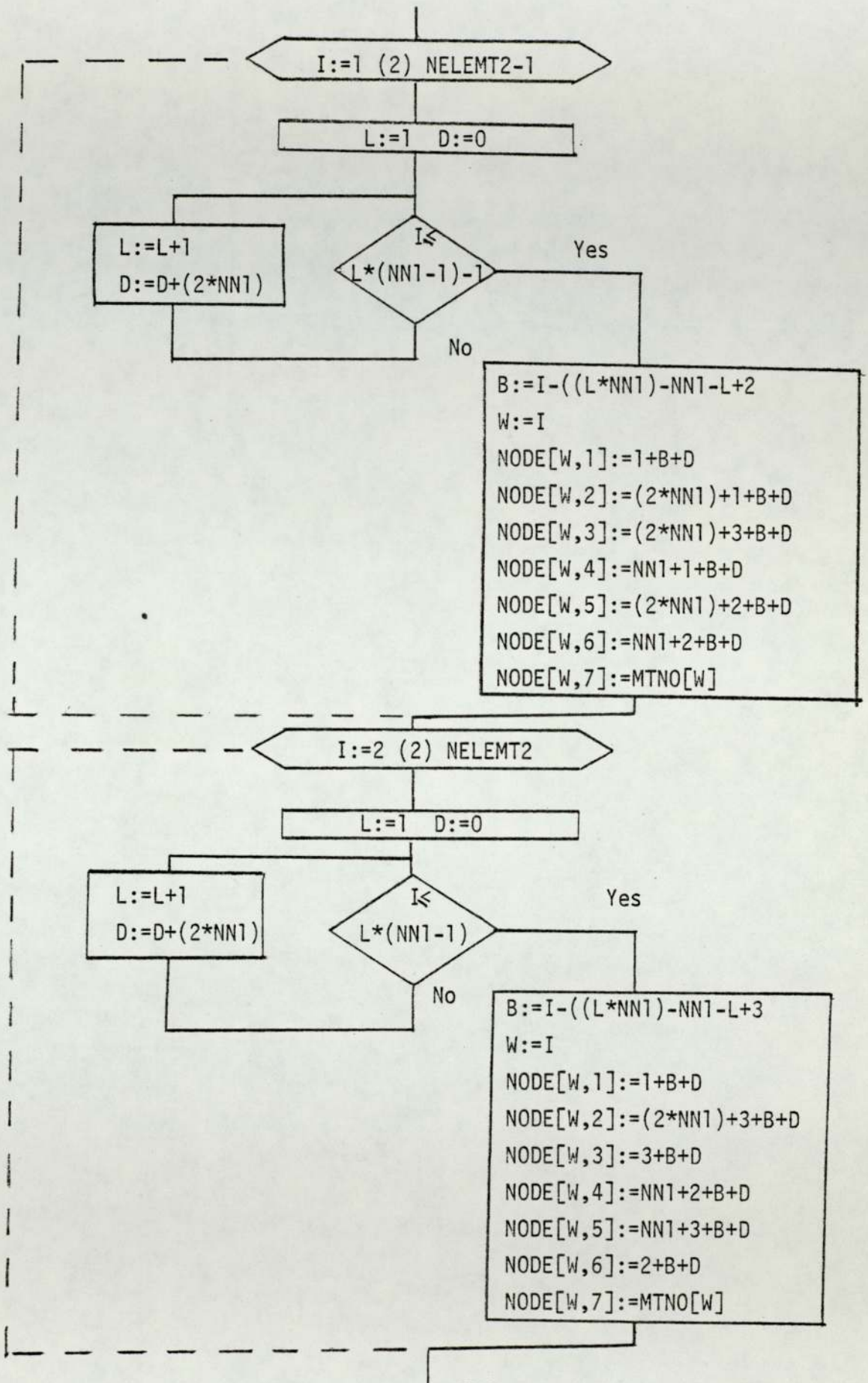


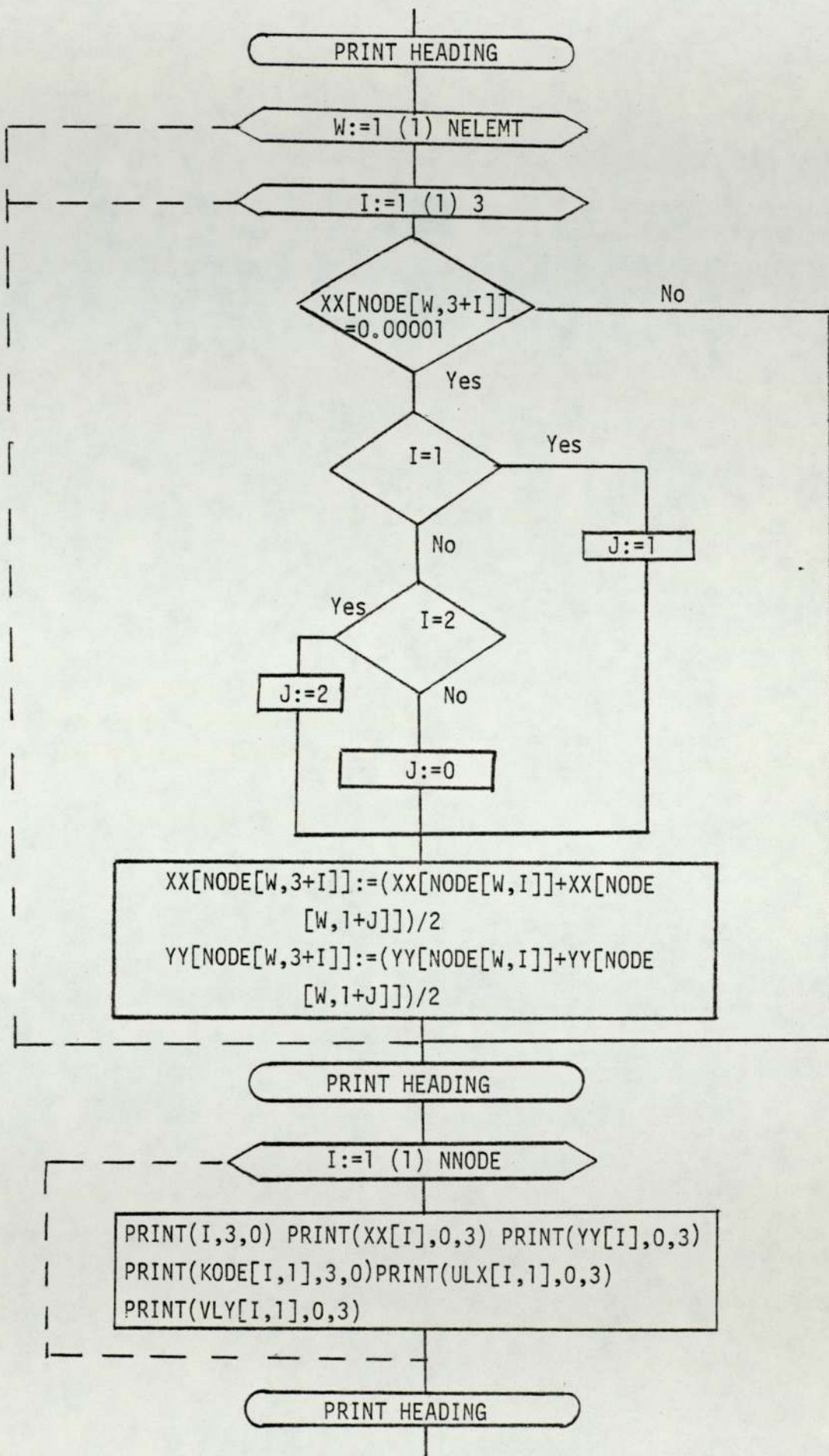


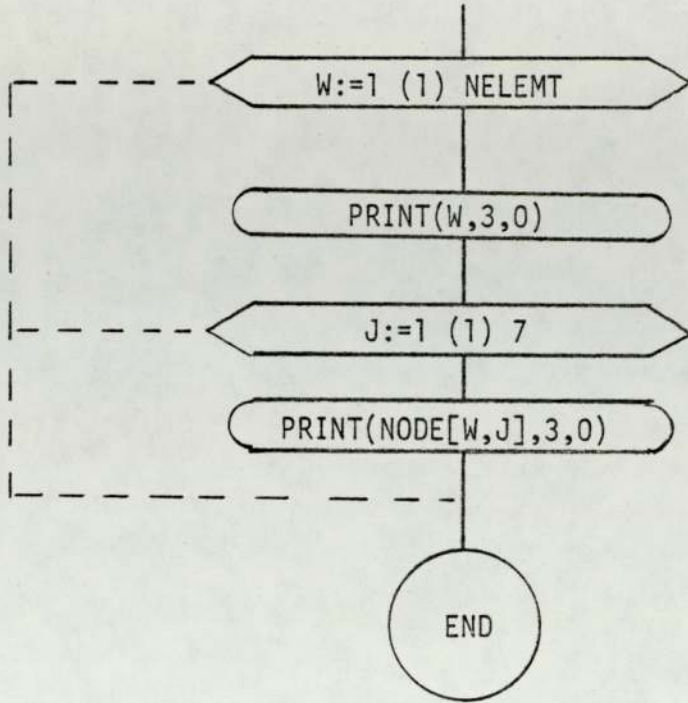
2

3









7.12 Procedure SBINPUT

It was noted that if procedure (MMINPUT) is to be used for solving the shouldered bar problem, its elements are not enough to represent the fillet radii with sufficient accuracy and will be severely distorted in order to match the boundary shape. A purpose built procedure to generate the mesh for this problem was developed, and as it only solves this particular problem, it was not left in the main mixed mode program but was put on a program of its own.

It is identical to procedure (MMINPUT) described in section (7.11) except for generating the truncated nodal columns. Instead of surrounding the three sides of the rectangular mesh, they extend outwards to generate the shoulder with the facility of having variable fillet radii. All the steps are similar to those of section (7.11) but the ratio of the shouldered bar diameters and the fillet radii required for any example are read and the mesh is generated accordingly.

①

```
TH:=(2*(J-1)*3.14159)/(N1-1)
XX[(B*N1)+J]:=(G-(AA*COS(PHI)))+
  (((F*RO)*COS(TH))*COS(PHI))+
  (((F*RO)*SIN(TH))*SIN(PHI))
YY[(B*N1)+J]:=((AA-(F*RO))*SIN(PHI))+
  (((F*RO)*(1-COS(TH)))*SIN(PHI))+
  (((F*RO)*SIN(TH))*COS(PHI))
```

I:=7

B:=I-1

J:=1,3,5,17

XX[(B*N1)+J]:=G

J:=7,15

XX[B*N1+J]:=G-((AA+(5*RO))/2)

J:=9,11,13

XX[(B*N1)+J]:=G-(AA+(5*RO))

J:=1,13,15,17

YY[(B*N1)+J]:=0.0

J:=3,11

YY[(B*N1)+J]:=(AA+(5*RO))/2

J:=5,7,9

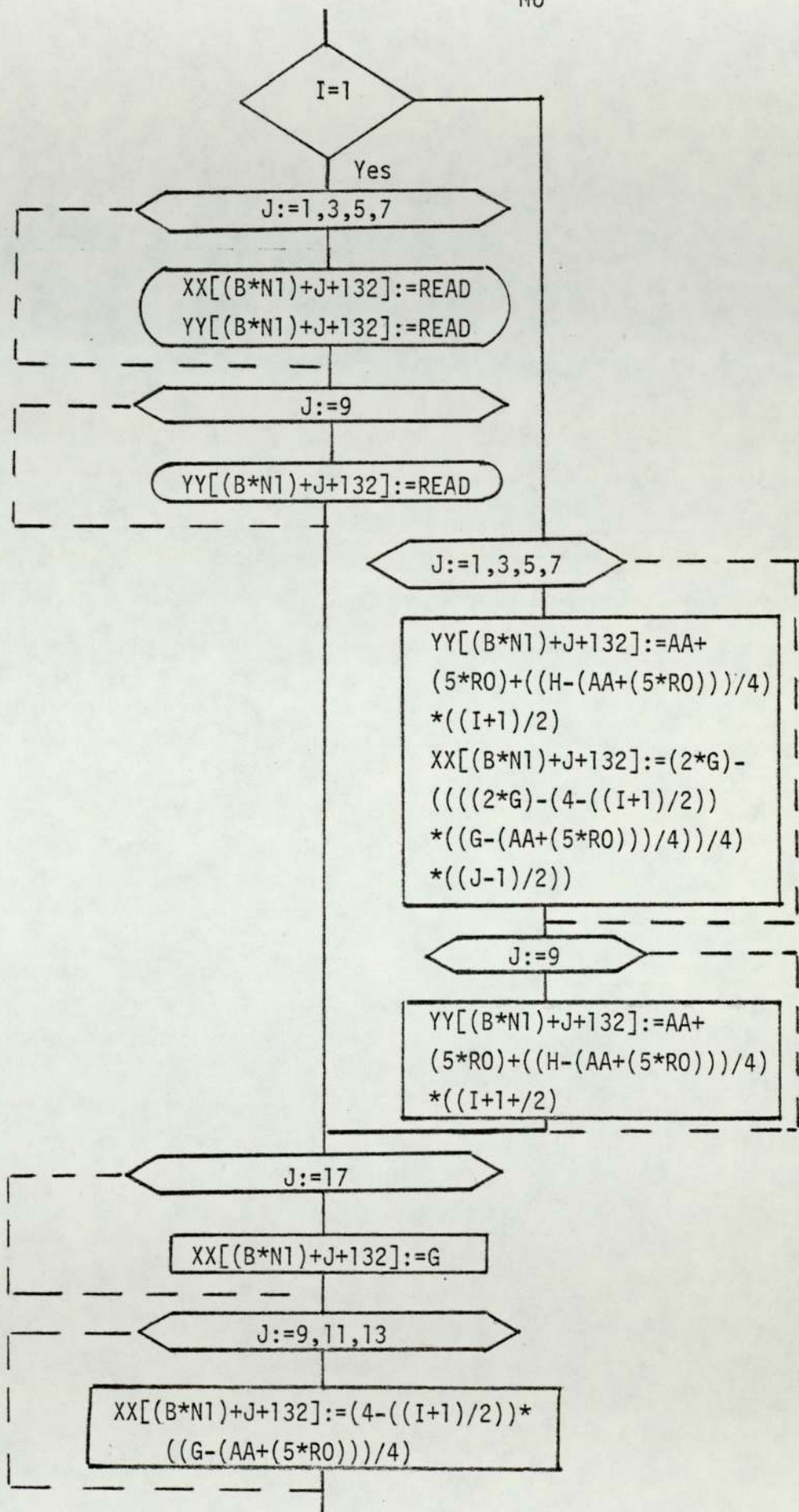
YY[(B*N1)+J]:=AA+(5*RO)

I:=2 (2) 7

B:=I-1

②

2



2

2

$XX[(B*N1)+15+132]:=XX[(B*N1)+13+132]$
 $+((G-XX[(B*N1)+13+132]))/2$

J:=13,15,17

$YY[(B*N1)+J+132]:=-((I+1/2)*(H/4))$

$YY[(B*N1)+11+132]:=(YY[(B*N1)+9+132]$
 $+YY[(B*N1)+13+132])/2$

NSPEC:=READ

I:=1 (1) NSPEC

J:=READ KODE[J,1]:=READ
ULX[J,1]:=READ VLY[J,1]:=READ

W:=1 (1) NELEMT

MTNO[W]:=1

Yes
No
NMAT=1

I:=2 (1) NMAT

NSTEL[1]:=READ

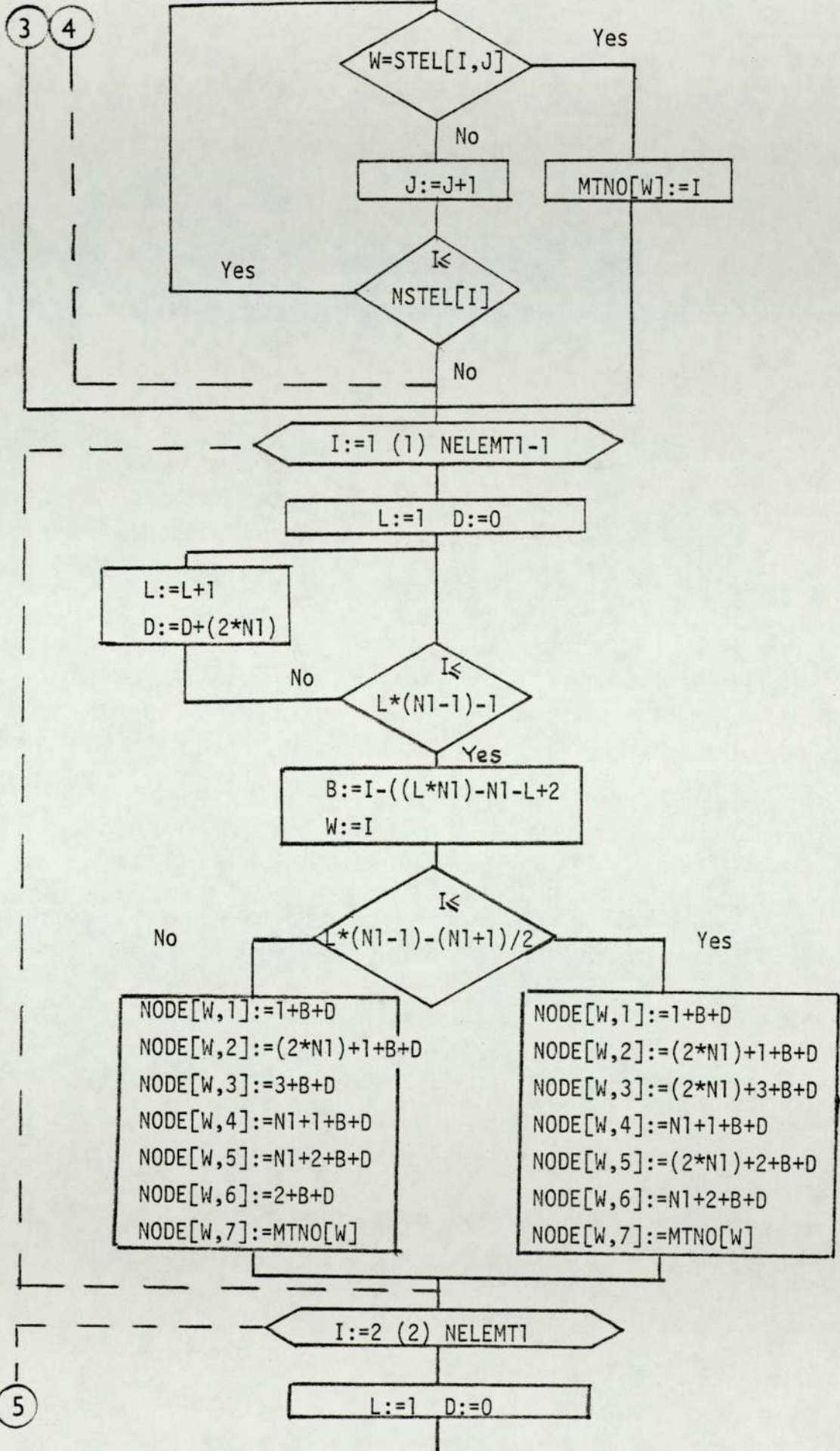
J:=1 (1) NSTEL[I]

STEL[I,J]:=READ

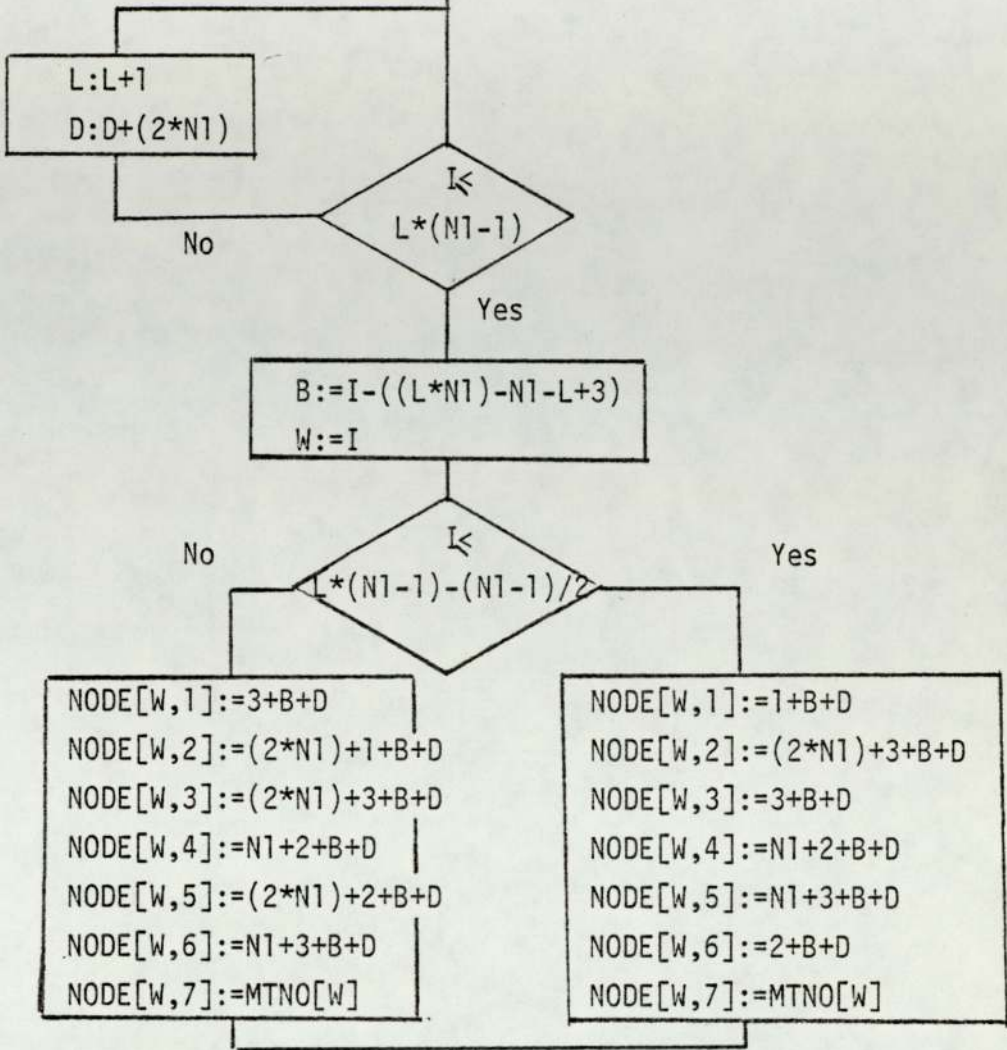
W:= (1) NELEMT

J:=1

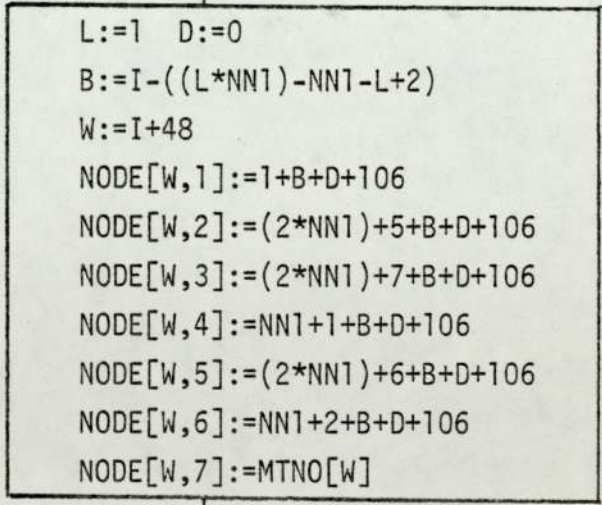
3 4



5



I:=1 (2) NN1-2



I:=2 (2) NN1-1

6

6

```
L:=1 D:=0
B:=I-((L*NN1)-NN1-L+3)
W:=I+48
NODE[W,1]:=1+B+D+106
NODE[W,2]:=(2*NN1)+7+B+D+106
NODE[W,3]:=3+B+D+106
NODE[W,4]:=NN1+2+B+D+106
NODE[W,5]:=NN1+3+B+D+106
NODE[W,6]:=2+B+D+106
NODE[W,7]:=MTNO[W]
```

I:=1 (2) NELEMT2-1

L:=1 D:=0

L:=L+1
D:=D+(2*N1)

I <= L*(N1-1)-1

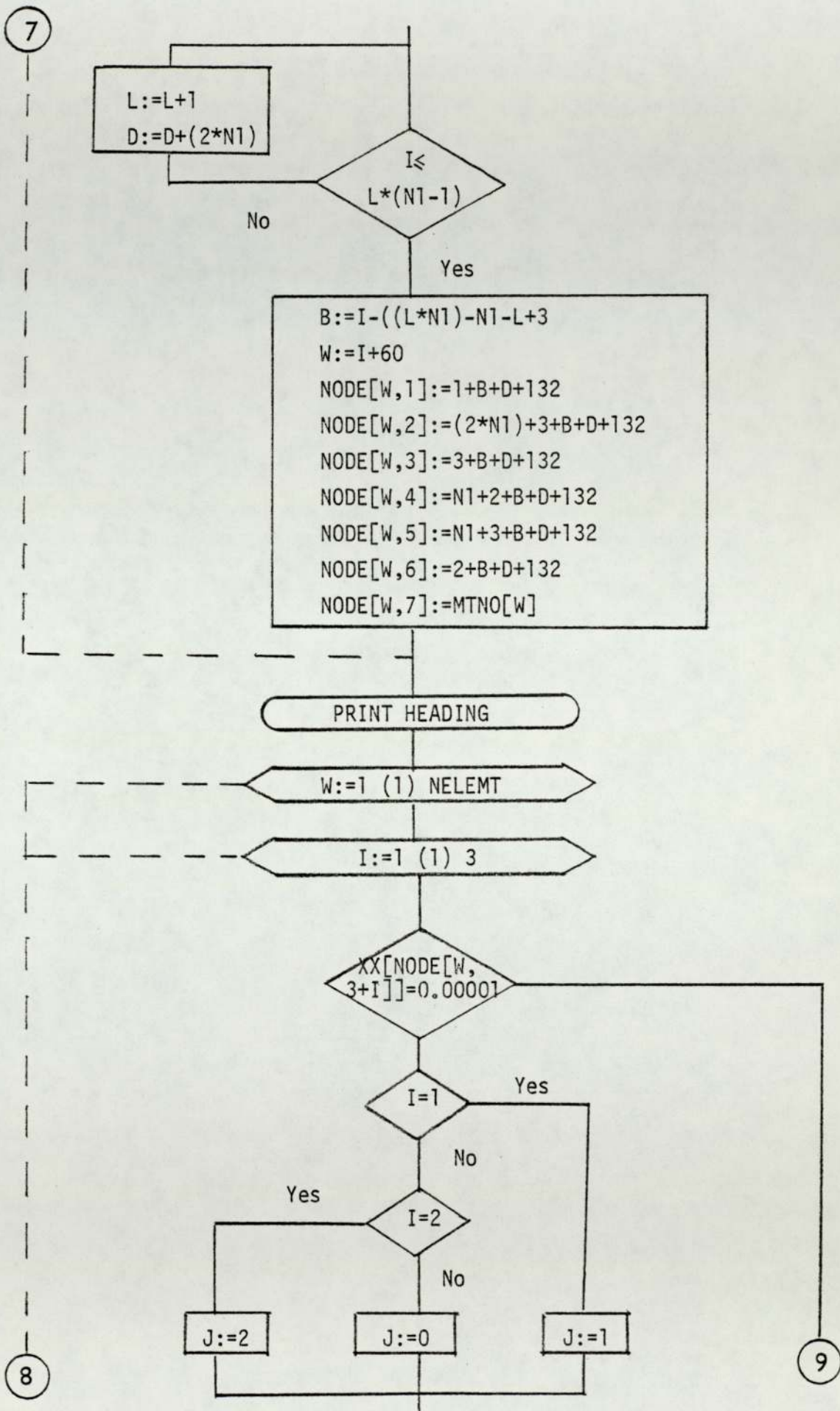
Yes

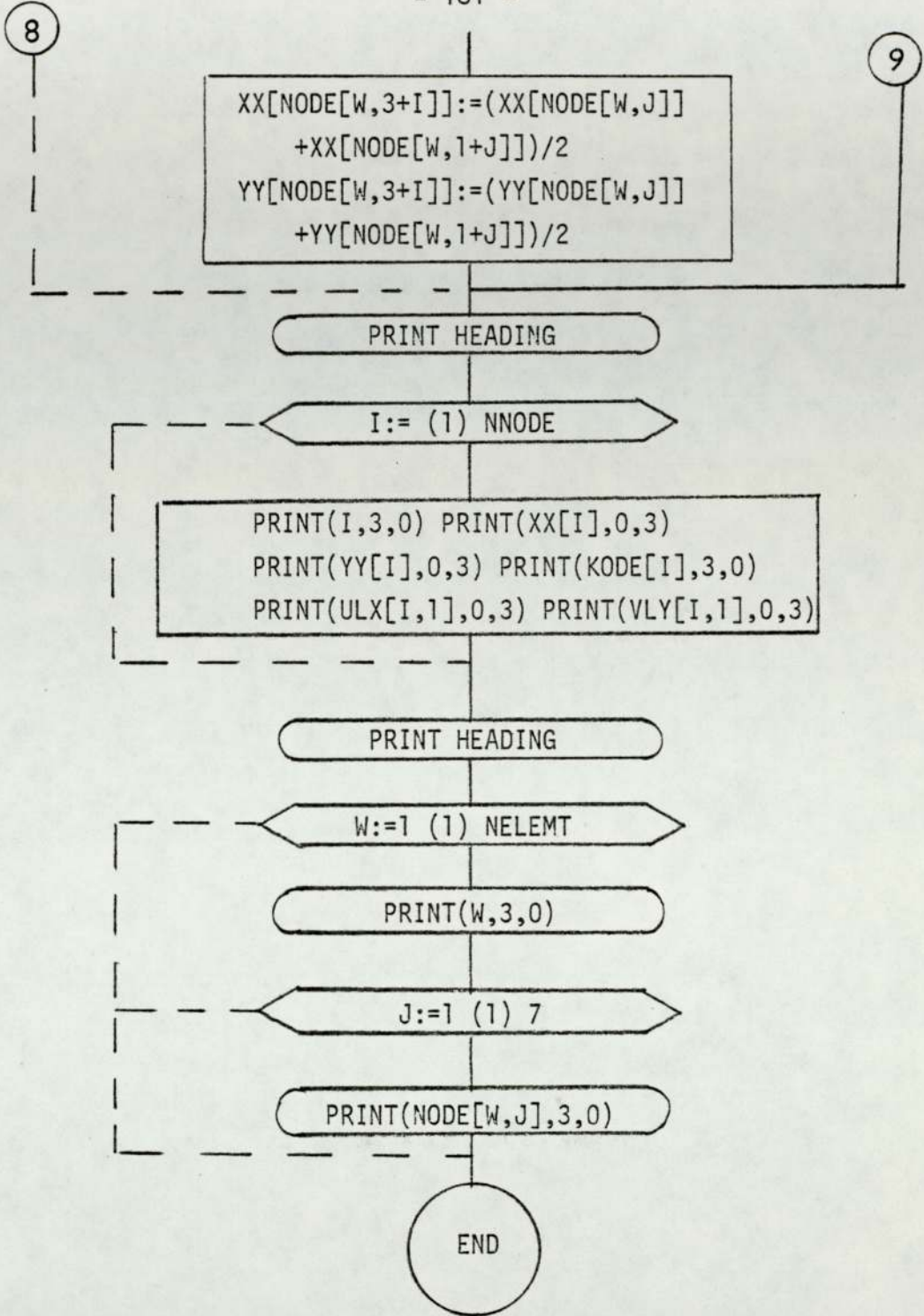
```
B:=I-((L*N1)-N1-L+2)
W:=I+60
NODE[W,1]:=1+B+D+132
NODE[W,2]:=(2*N1)+1+B+D+132
NODE[W,3]:=(2*N1)+3+B+D+132
NODE[W,4]:=N1+1+B+D+132
NODE[W,5]:=(2*N1)+2+B+D+132
NODE[W,6]:=N1+2+B+D+132
NODE[W,7]:=MTNO[W]
```

I:=2 (2) NELEMT2

L:=1 D:=0

7





7.13 Procedure CCRM1

This procedure performs the modifications described in Chapter 6 to the overall stiffness matrix and load vector to include the special core element for a mode (I) fracture problem. In this case there is only one rigid body displacement of the crack tip (δ_r) in the r-direction. The near tip displacement components may be written as:

$$U_r = \delta_r + K_I f(r, \theta) \quad (7.37)$$

$$W_z = K_I g(r, \theta) \quad (7.38)$$

where $f(r, \theta)$ and $g(r, \theta)$ are given in Appendix (10.2).

It has been shown in Chapter 6 how the overall stiffness matrix $[K]$ is modified to $[K]^*$ and the potential energy of the system is minimized with respect to K_I , δ_r and u_i which are the unconstrained nodal displacements.

It is convenient to number the nodes on the core/finite element interface from (1) to (N1) as such a numbering sequence will result in making the nodes of the first ring of elements surrounding the core contribute stiffness matrix coefficients connecting nodes on the interface with the remainder of the nodes only. By performing the partitioning suggested in equation (6.12), the set of equations will have the form shown in Fig. (7.2). The reassembled stiffness matrix $[K]^*$ will have the form shown in Fig. (7.3)

It was shown in equation (6.13) that:

$$\{q_1\} = [A]\{\alpha\}_c \quad (7.39)$$

$$\begin{bmatrix}
 K_{11} & K_{12} & 0 \\
 K_{21} & & \\
 & K_{22} & \\
 0 & &
 \end{bmatrix}
 \begin{bmatrix}
 u_1 \\
 w_1 \\
 \\
 w_{NG}
 \end{bmatrix}
 =
 \begin{bmatrix}
 Q_{r1} \\
 Q_{z1} \\
 \\
 Q_{zNG}
 \end{bmatrix}$$

Fig. 7.2.

$$\begin{bmatrix}
 & & & K_{12}^* \\
 & K_{11}^* & & \\
 & & & 0 \\
 NG-2N_1 & K_{21}^* & 0 & K_{22}^*
 \end{bmatrix}
 \begin{bmatrix}
 \\
 \\
 \\
 \delta r \\
 K_1
 \end{bmatrix}
 =
 \begin{bmatrix}
 \\
 \\
 \\
 0 \\
 0
 \end{bmatrix}$$

$4N_1$

Fig. 7.3

where

$$\{q_1\} = \begin{bmatrix} \begin{Bmatrix} u_1 \\ w_1 \end{Bmatrix} \\ \begin{Bmatrix} u_2 \\ w_2 \end{Bmatrix} \\ \vdots \\ \begin{Bmatrix} u_{N1} \\ w_{N1} \end{Bmatrix} \end{bmatrix} \quad (7.40)$$

$$\{\alpha\}_c = \begin{Bmatrix} \delta_r \\ K_I \end{Bmatrix} \quad (7.41)$$

$$[A] = \begin{bmatrix} \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} & \begin{Bmatrix} f(R_c, \theta_1) \\ g(R_c, \theta_1) \end{Bmatrix} \\ \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} & \begin{Bmatrix} f(R_c, \theta_2) \\ g(R_c, \theta_2) \end{Bmatrix} \\ \vdots & \vdots \\ \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} & \begin{Bmatrix} f(R_c, \theta_{N1}) \\ g(R_c, \theta_{N1}) \end{Bmatrix} \end{bmatrix} \quad (7.42)$$

and the functions $f(R_c, \theta)$ and $g(R_c, \theta)$ are given in Appendix (10.2) and with $r = R_c$ (core radius) and $\theta =$ the angle between the positive r -axis and the particular node on the interface.

From Appendix (10.3)

$$[K]_c = \begin{bmatrix} 0 & 0 \\ 0 & \frac{rR_c \pi^2}{8\mu} (2\kappa - 1) \end{bmatrix} \quad (7.43)$$

It is now necessary to write the matrix operations of equation (6.23) in a form suitable for programming:

$$K_{22}^*[1,1] = K_c[1,1] + \sum_{i=1,3,5}^{2N_1-1} \sum_{j=1,3,5}^{2N_1-1} K[i,j] \quad (7.44)$$

$$\begin{aligned} K_{22}^*[1,2] &= K_{22}^*[2,1] \\ &= K_c[2,1] + \sum_{i=1,3,5}^{2N_1-1} \sum_{j=1,2,3}^{2N_1} K[i,j]u_j \end{aligned} \quad (7.45)$$

$$K_{22}^*[2,2] = K_c[2,2] + \sum_{i=1,2,3}^{2N_1} \sum_{j=1,2,3}^{2N_1} K[i,j]u_i u_j \quad (7.46)$$

$$K_{12}^*[1,j] = \sum_{i=1,3,5}^{2N_1-1} K[i,j + 2N] \quad (7.47)$$

or from the symmetry of [K] equation (7.47) may be written as:

$$K_{12}^*[1,j] = \sum_{i=1,3,5}^{2N_1-1} K[j + 2N_1, i] \quad j = 1,2,3 \dots 4N_1 \quad (7.48)$$

$$K_{12}^*[2,j] = \sum_{i=1,2,3}^{2N_1} K[j + 2N_1, i]u_i \quad j=1,2,3 \dots 4N_1 \quad (7.49)$$

$$K_{11}^*[i,j] = K[i + 2N_1, j + 2N_1] \quad i, j=1,2, \dots, NG-2N_1 \quad (7.50)$$

$$\{Q_2\}^* = \{0\} \quad (7.51)$$

And

$$Q_1^*[i] = Q[i + 2N_1] \quad i, j=1, 2, \dots, NG-2N_1 \quad (7.52)$$

where

$$u_j = f(R_c, \theta(\frac{j+1}{2})) \quad j\text{-odd} \quad (7.53)$$

$$u_j = g(R_c, \theta(\frac{j}{2})) \quad j\text{-even} \quad (7.54)$$

In calculating the values of the coefficients of the $[K]^*$ matrix according to equations (7.44) to (7.50), it must be noted that only the lower half of the original stiffness matrix $[K]$ was stored. Therefore by using symmetry, for a given row (i) the terms $K[i, i+1]$ to $K[i, 2N]$ are replaced by $K[i+1, i]$ to $K[2N, i]$ respectively. As the stiffness matrix coefficients stored were those between the first non-zero coefficients and the major diagonal only, the first non-zero coefficient (j) in row (i) must be found. This will reduce the terms $K[i, 1]$ to $K[i, i]$ to be only $K[i, j]$ to $K[i, i]$. Similarly some of the coefficients $K[i+1, i]$ to $K[2N, i]$ may be zeroes before the first non-zero coefficient of their row and hence not stored. Therefore the first non-zero coefficient of these rows must be calculated also and if any term is smaller than its corresponding row first non-zero coefficient, it must be dropped from the summation. For ease of programming each summation is divided into two as:

$$\Sigma = \Sigma_1 + \Sigma_2 \quad (7.55)$$

where

$$\Sigma = \sum_{\ell=1}^{2N_1} K[i, \ell] \quad (7.56)$$

$$\Sigma_1 = \sum_{\ell=j}^i K[i, \ell] \quad (7.57)$$

and

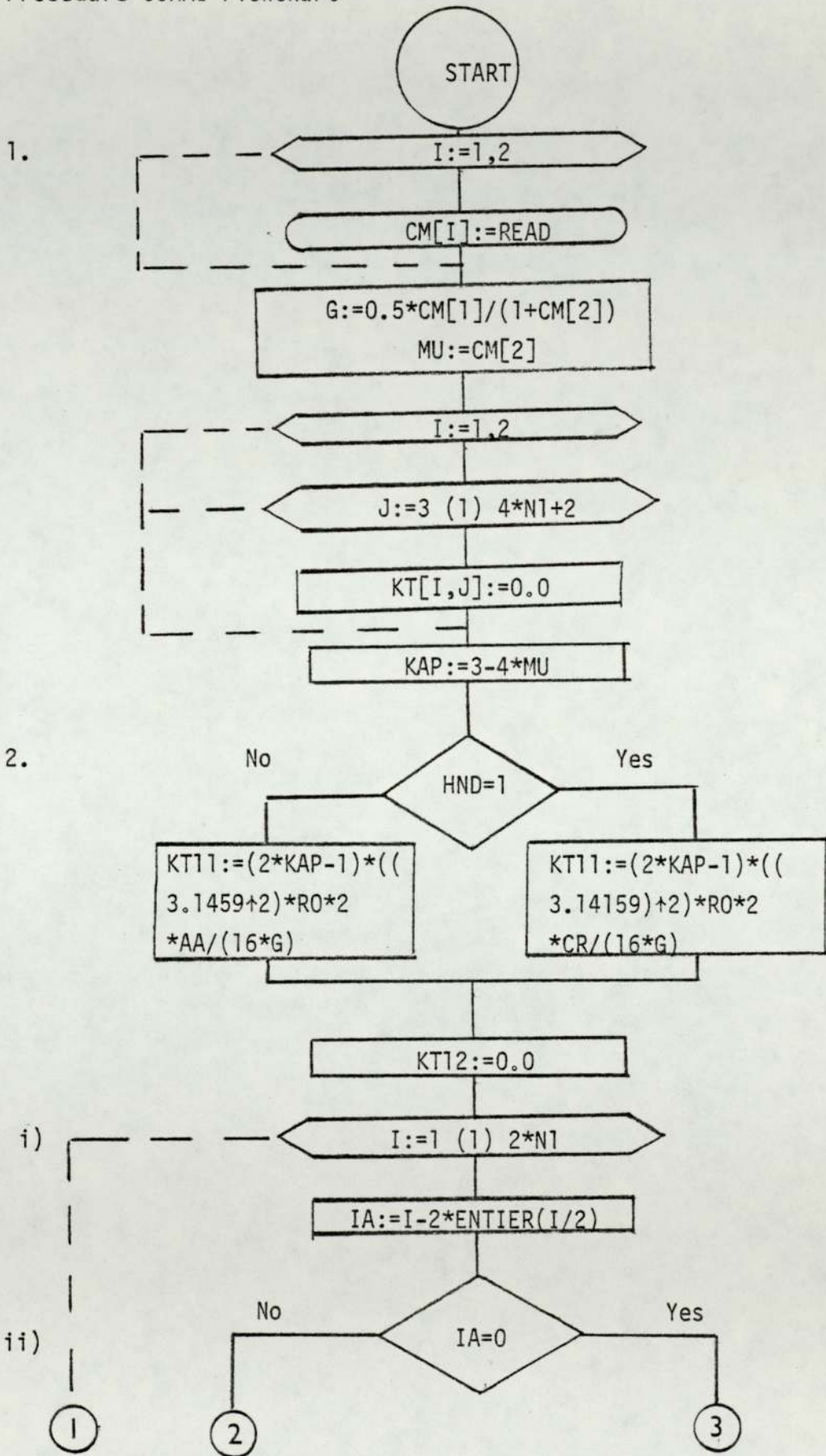
$$\Sigma_2 = \sum_{\ell=i}^{2N_1} K[\ell, i] \quad (7.58)$$

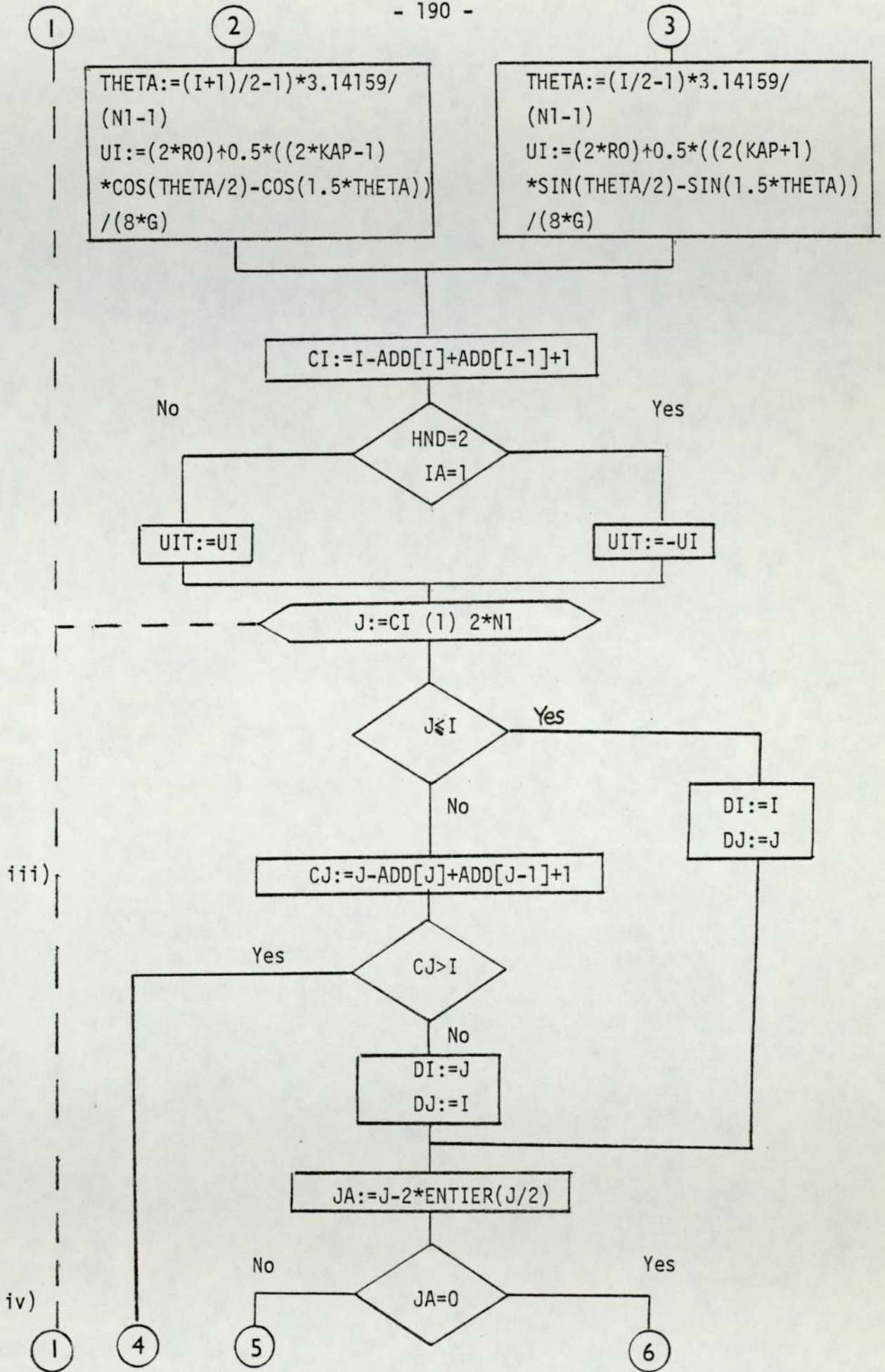
The steps taken to translate the operations described in this section to a computer code are as follows:

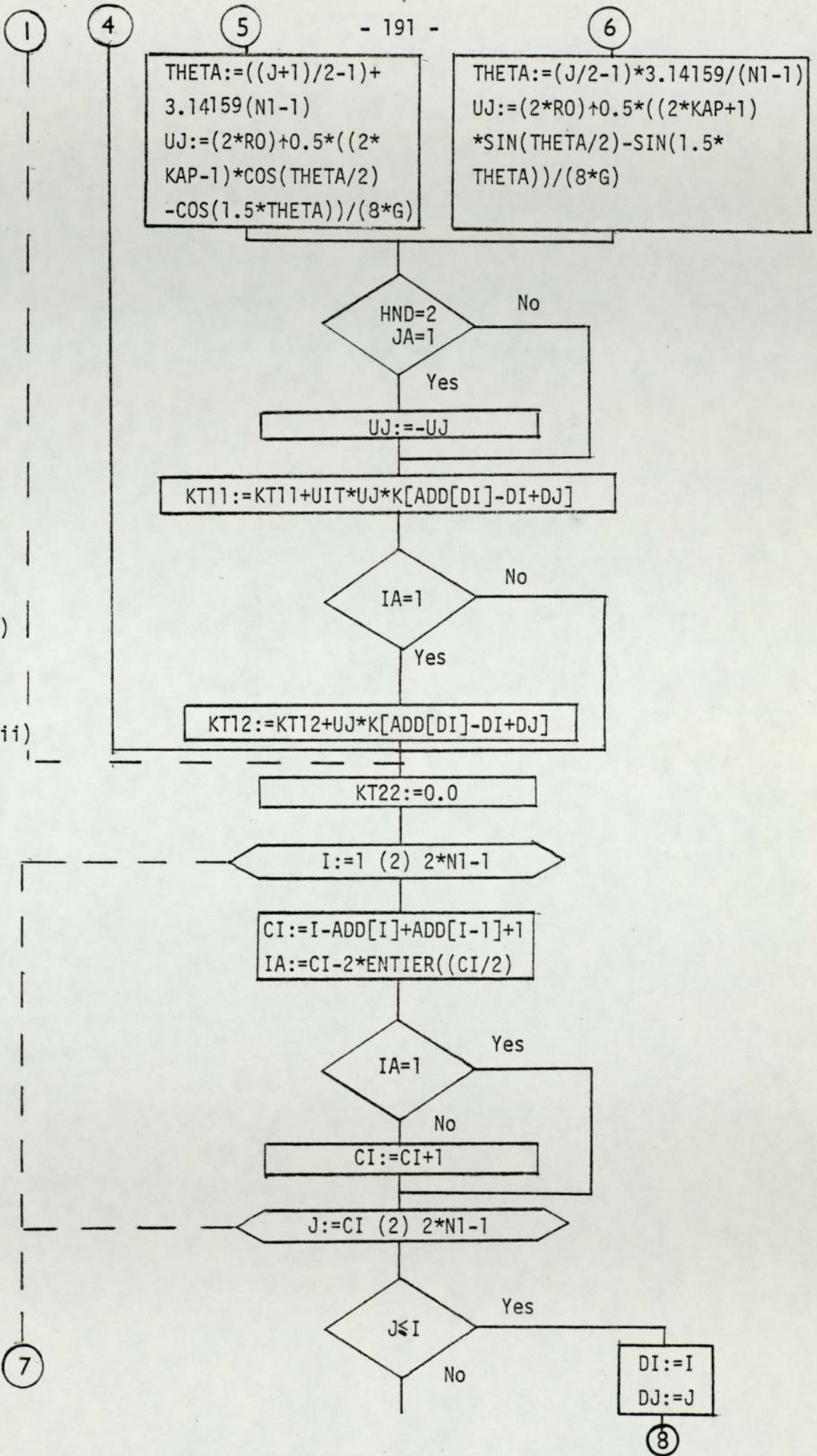
1. The coefficients of array CM[1x2] which are the elastic constants E and ν for the material in the immediate neighbourhood of the crack tip are read and G is calculated from them.
2. Equation (7.46) is evaluated by:
 - i) A loop I from (1) to $(2N_1)$ is constructed.
 - ii) Integer IA distinguishes between odd and even I values which correspond to r and z displacement components respectively, so that the appropriate θ_i and $f(R_c, \theta_i)$ or $g(R_c, \theta_i)$ is calculated as shown in equation (7.42).
 - iii) The first non-zero column number (J) of row (I) is determined.
 - iv) The sign of the radial displacement component is checked and changed if necessary to match the global r-direction.
 - v) A loop (J) is constructed from the first non-zero coefficient in row (I) to $(2N_1)$ and the summations in equation (7.46) are performed as shown in equations (7.57) and (7.58).
 - vi) Steps (ii) to (iv) are repeated with (J) instead of (I).

- vii) The summations of equation (7.45) are performed similarly to (v).
3. Equations (7.44), (7.48), and (7.49) are evaluated similarly.
 4. Equation (7.50) is performed which implies the elimination of rows and columns (1) to $(2N_1)$ from $[K]$ and replacing then by K_{22} , and the address sequence is modified accordingly.
 5. Equation (7.52) which reassembles the load vector is carried out.
 6. Equation (7.51) is implemented and address coefficients for the last two rows of $[K]^*$ are calculated.
 7. The coefficients of $[K]^*$ which have been calculated are allocated to their appropriate positions.

Procedure CCRMI Flowchart

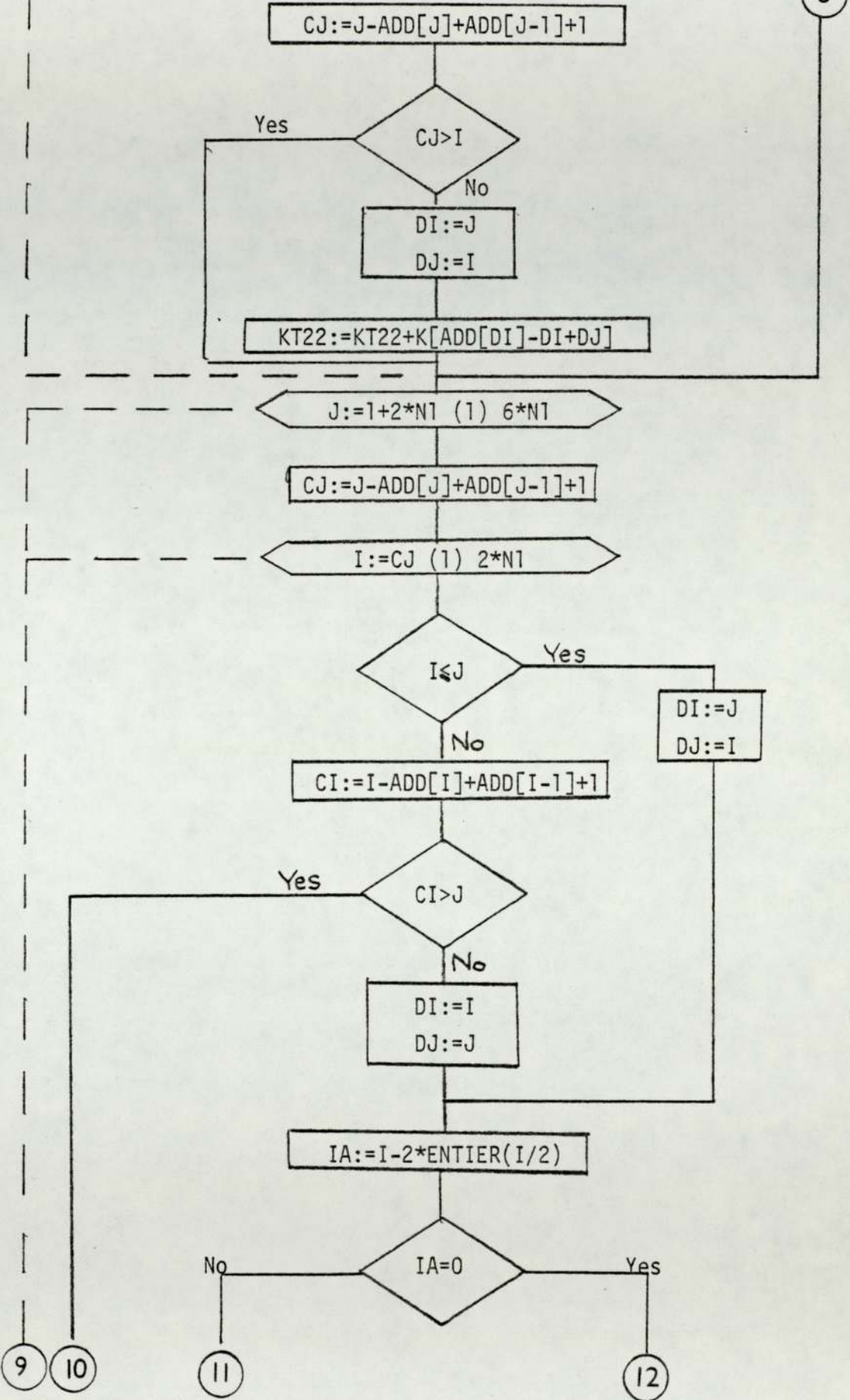


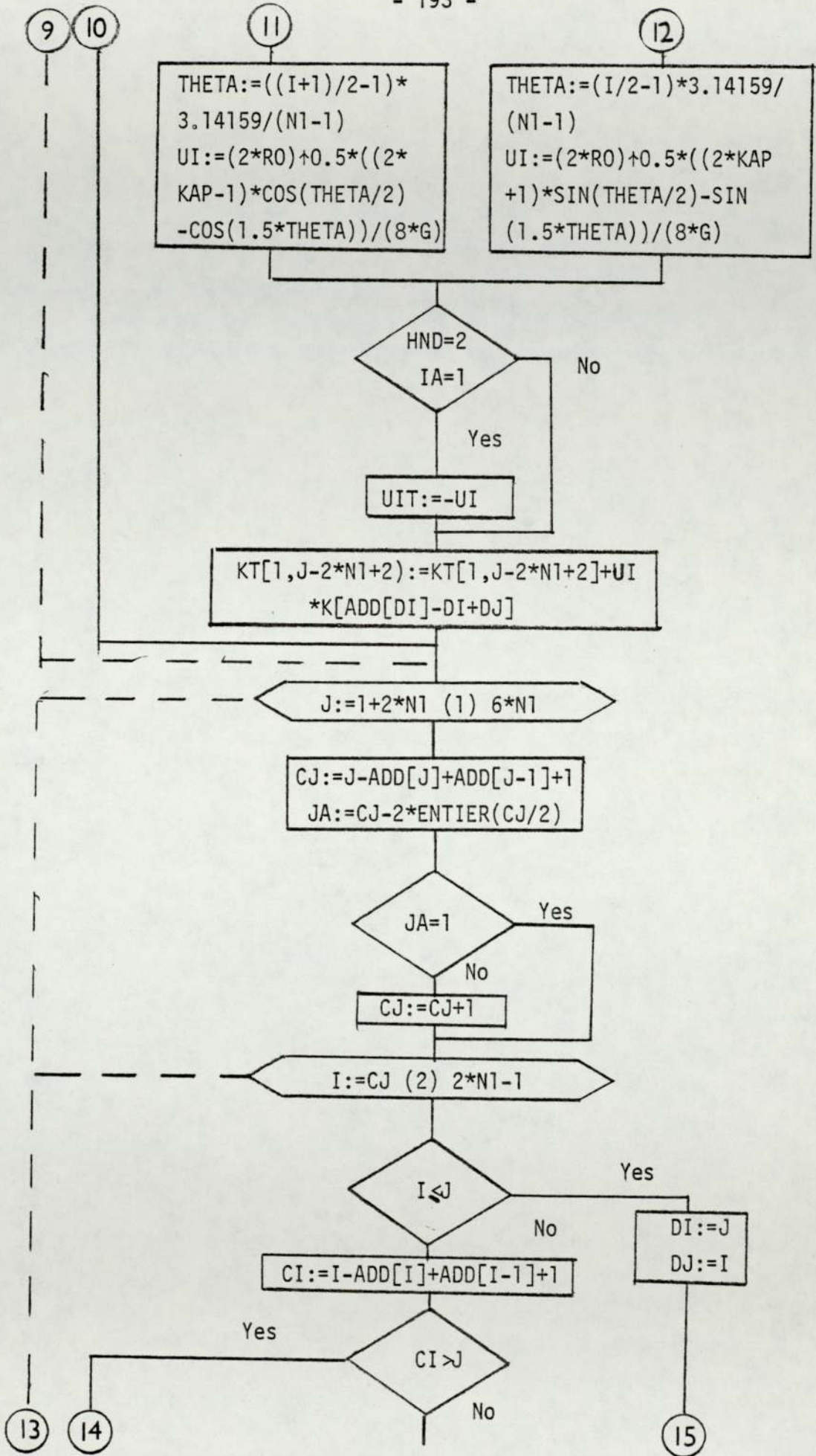


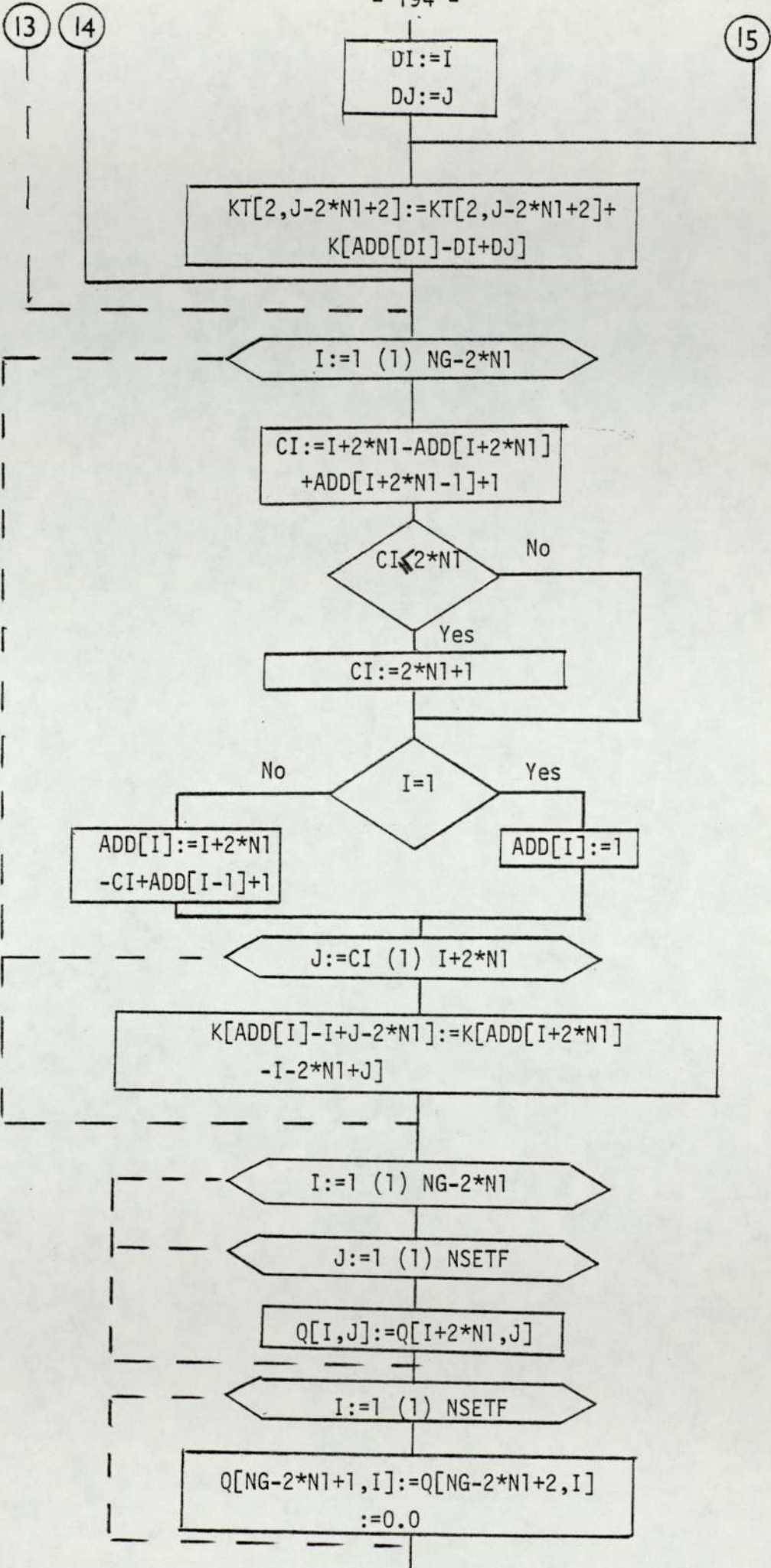


7

8








```
ADD[NG-2*N1+1]:=ADD[NG-2*N1]
+NG-2*N1+1
ADD[NG-2*N1+2]:=ADD[NG-2*N1-1]
+NG-2*N1+2
```

I:=1 (1) NG-2*N1

```
K[ADD[NG-2*N1+1]-NG+2*N1-1+I]
:=0.0
K[ADD[NG-2*N1+2]-NG+2*N1-2+I]
:=0.0
```

7.

```
K[ADD[NG-2*N1+2]-1]:=KT12
K[ADD[NG-2*N1+1]]:=KT22
K[ADD[NG-2*N1+2]]:=KT11
```

I:=3 (1) 4*N1+2

```
K[ADD[NG-2*N1+1)-NG+2*N1
+I-3]:=KT[2,I]
K[ADD[NG-2*N1+2]-NG+2*N1
+I-4]:=KT[1,I]
```

END

7.14 Procedure CCRMM12

This procedure is similar to (CCRM1) but deals with mixed mode I and II problems. In this case the rigid body modes of the core element are the two rigid body displacement components δ_r and δ_z and a rigid body rotation term. Failure to include the rotation term will result in imposing constraints on the core, hence the physical situation will not be accurately represented. With reference to Fig.(7.4), the displacement components obtained from a rotation of ω° are:

$$u_r = -\omega R \sin\theta \quad (7.59)$$

$$w_z = \omega R \cos\theta \quad (7.60)$$

The near tip displacement field may be written as:

$$u_r = f_1(R_c, \theta)K_I + g_1(R_c, \theta)K_{II} + \delta_r + h_1(R_c, \theta)\omega \quad (7.61)$$

$$w_z = f_2(R_c, \theta)K_I + g_2(R_c, \theta)K_{II} + \delta_z + h_2(R_c, \theta)\omega \quad (7.62)$$

Where F_1 , f_2 , g_1 and g_2 are given in Appendix (10.2) and,

$$h_1(R_c, \theta) = -R_c \sin\theta \quad (7.63)$$

$$h_2(R_c, \theta) = R_c \cos\theta \quad (7.64)$$

In this case the potential energy is minimized with respect to δ_r , δ_z , ω , K_I , K_{II} , and u_i . As the crack plane is inclined to the global axes the components of equations (7.61) and (7.62) must be resolved to correspond to them, hence,

$$u_r = u_r \cos\alpha - w_z \sin\alpha \quad (7.65)$$

$$w_z = u_r \sin\alpha + w_z \cos\alpha \quad (7.66)$$

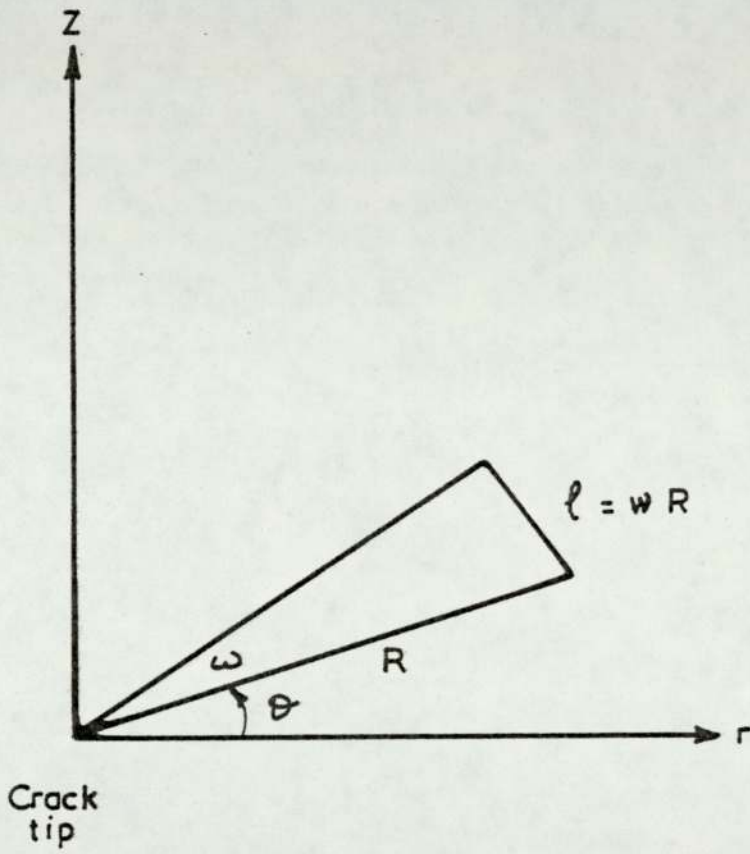


Fig. 7.4

Therefore equations (7.61) and (7.62) become:

$$u_r = F_1(R_c, \theta)K_I + G_1(R_c, \theta)K_{II} + \delta_r \cos\alpha - \delta_z \sin\alpha + H_1(R_c, \theta)\omega \quad (7.67)$$

$$w_z = F_2(R_c, \theta)K_I + G_2(R_c, \theta)K_{II} + \delta_r \sin\alpha + \delta_z \cos\alpha + H_2(R_c, \theta)\omega \quad (7.68)$$

where

$$F_1 = f_1 \cos\alpha - f_2 \sin\alpha \quad (7.69)$$

$$F_2 = f_1 \sin\alpha + f_2 \cos\alpha \quad (7.70)$$

$$G_1 = g_1 \cos\alpha - g_2 \sin\alpha \quad (7.71)$$

$$G_2 = g_1 \sin\alpha + g_2 \cos\alpha \quad (7.72)$$

$$H_1 = h_1 \cos\alpha - h_2 \sin\alpha \quad (7.73)$$

$$H_2 = h_1 \sin\alpha + h_2 \cos\alpha \quad (7.74)$$

The same principle in node numbering on the interface as that of the mode I case is used with the only difference being that the core is a full circle.

The $\{q_1\}$ vector of equation (6.13) is the same as (7.40) but the $\{\alpha\}_c$ is now:

$$\{\alpha\}_c = \begin{Bmatrix} \omega \\ \delta_z \\ \delta_r \\ K_{II} \\ K_I \end{Bmatrix} \quad (7.75)$$

and [A] is:

$$[A] = \left[\begin{array}{ccccc} \left\{ \begin{array}{l} H_1(R_c, \theta_1) \\ H_2(R_c, \theta_1) \end{array} \right\} & \left\{ \begin{array}{l} -\sin\alpha \\ \cos\alpha \end{array} \right\} & \left\{ \begin{array}{l} \cos\alpha \\ \sin\alpha \end{array} \right\} & \left\{ \begin{array}{l} G_1(R_c, \theta_1) \\ G_2(R_c, \theta_1) \end{array} \right\} & \left\{ \begin{array}{l} F_1(R_c, \theta_1) \\ F_2(R_c, \theta_1) \end{array} \right\} \\ \left\{ \begin{array}{l} H_1(R_c, \theta_2) \\ H_2(R_c, \theta_2) \end{array} \right\} & \left\{ \begin{array}{l} -\sin\alpha \\ \cos\alpha \end{array} \right\} & \left\{ \begin{array}{l} \cos\alpha \\ \sin\alpha \end{array} \right\} & \left\{ \begin{array}{l} G_1(R_c, \theta_2) \\ G_2(R_c, \theta_2) \end{array} \right\} & \left\{ \begin{array}{l} F_1(R_c, \theta_2) \\ F_2(R_c, \theta_2) \end{array} \right\} \\ \vdots & & & & \\ \left\{ \begin{array}{l} H_1(R_c, \theta_{N1}) \\ H_2(R_c, \theta_{N1}) \end{array} \right\} & \left\{ \begin{array}{l} -\sin\alpha \\ \cos\alpha \end{array} \right\} & \left\{ \begin{array}{l} \cos\alpha \\ \sin\alpha \end{array} \right\} & \left\{ \begin{array}{l} G_1(R_c, \theta_{N1}) \\ G_2(R_c, \theta_{N1}) \end{array} \right\} & \left\{ \begin{array}{l} F_1(R_c, \theta_{N1}) \\ F_2(R_c, \theta_{N1}) \end{array} \right\} \end{array} \right] \quad (7.76)$$

From Appendix (10.3) the core stiffness matrix may be written as:

$$[K]_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{rR_c \pi^2}{4\mu} (3+2\kappa) & 0 \\ 0 & 0 & 0 & 0 & \frac{rR_c \pi^2}{4\mu} (2\kappa-1) \end{bmatrix} \quad (7.77)$$

The appropriate coefficients of [K]* may be obtained from equation (6.23) with

$$TI = NG - 2N1 \quad (7.78)$$

as follows:

$$K^*[TI+\ell, TI+m] = K_c[\ell, m] + \sum_{i=1,3,5}^{2N1-1} \sum_{j=1,3,5}^{2N1-1} K[i, j] M[\ell] M[m] \quad (7.79)$$

$$K^*[TI+\ell, TI+m] = K_c[\ell, m] + \sum_{i=2,4,6}^{2N_1} \sum_{j=2,4,6}^{2N_1} K[i, j] N[\ell] N[m] \quad (7.80)$$

And due to symmetry:

$$K^*[TI+\ell, T[+m]] = K^*[TI+m, T[+\ell]] \quad (7.81)$$

And

$$K^*[TI+\ell, j] = \sum_{i=1,3,5}^{2N_1-1} K[j+2N_1, i] M[\ell] \quad j=1,2,3\dots 4N_1 \quad (7.82)$$

$$K^*[TI+\ell, j] = \sum_{i=2,4,6}^{2N_1} K[j+2N_1, i] N[\ell] \quad j=1,2,3\dots 4N_1 \quad (7.83)$$

where

$$\ell = m = 1, 2, 3, 4, 5$$

$$[M]^t = [H_2 \quad \cos\alpha \quad \sin\alpha \quad G_2 \quad F_2] \quad (7.84)$$

$$[N]^t = [H_1 \quad \sin\alpha \quad \cos\alpha \quad G_1 \quad F_1] \quad (7.85)$$

As the nodes on the interface are free from external loading

$$\{Q_2\}^* = \{0\} \quad (7.86)$$

Similar to the mode I case the partitioning will imply:

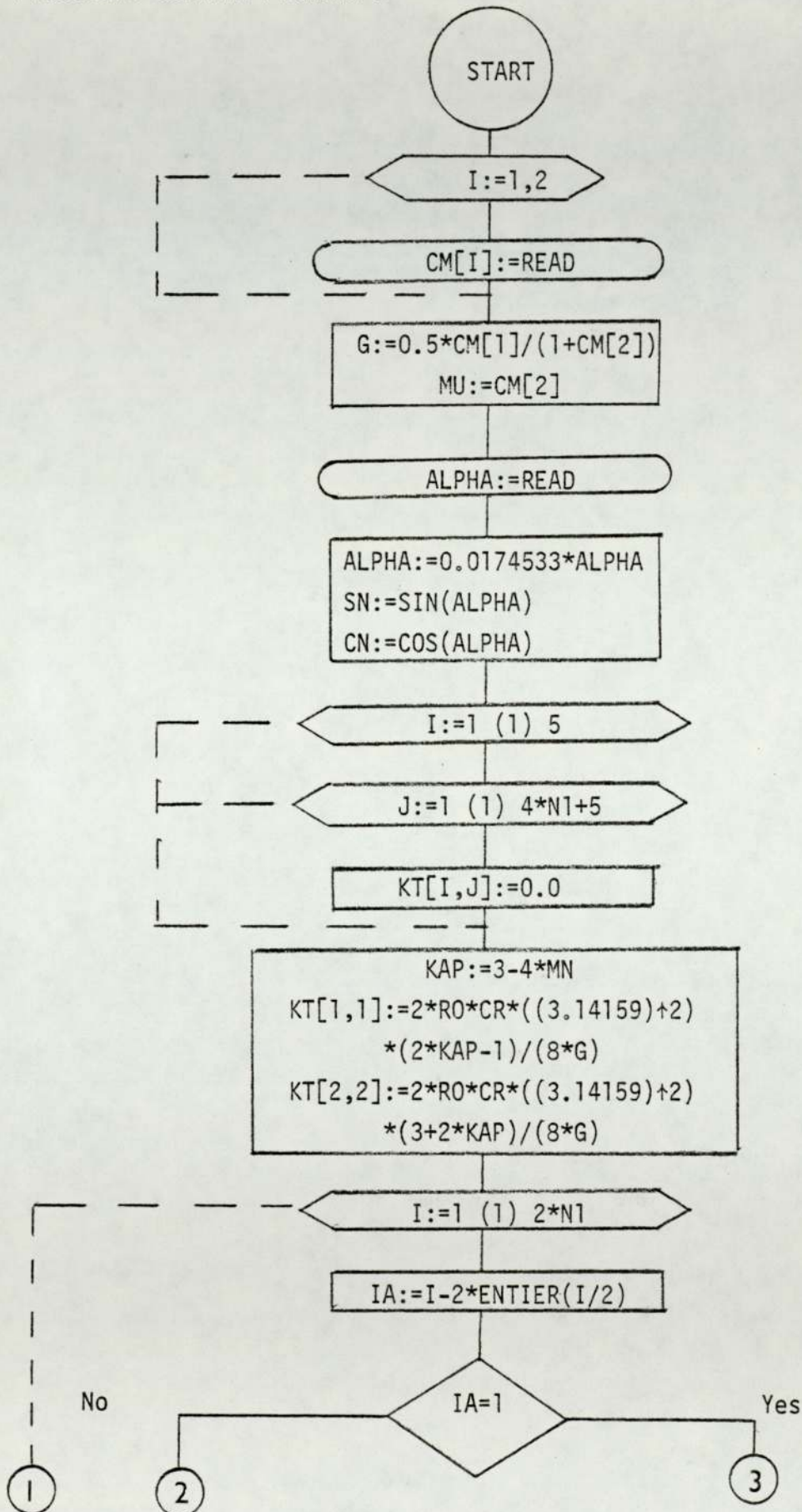
$$K[i, j]^* = K[i+2N_1, j+2N_1] \quad (7.87)$$

$$Q[i]^* = Q[i+2N_1] \quad (7.88)$$

where $i, j = 1, 2, \dots, NG-2N_1$

The steps of the procedure are similar to those of section (7.13) except for step (6) where there will be five rows instead of two at the bottom of $[K]^*$ corresponding to $\{\alpha\}_c$.

Procedure CCRMM12 Flowchart



1

2

3

$THETA := -3.14159 + 2 * (I / 2 - 1) * 3.14159 / (N1 - 1)$

$THETA := -3.14159 + 2 * ((I + 1) / 2 - 1) * 3.14159 / (N1 - 1)$

$FIO := (R0 / 2) + 0.5 * ((2 * KAP - 1) + \cos(THETA / 2) - \cos(1.5 * THETA)) / (4 * G)$
 $GIO := (R0 / 2) + 0.5 * ((2 * KAP + 3) * \sin(THETA / 2) + \sin(1.5 * THETA)) / (4 * G)$
 $FII := (R0 / 2) + 0.5 * ((2 * KAP + 1) * \sin(THETA / 2) - \sin(1.5 * THETA)) / (4 * G)$
 $GII := (R0 / 2) + 0.5 * (2 * KAP - 3) * \cos(THETA / 2) + \cos(1.5 * THETA) / (4 * G)$
 $HIO := -R0 * \sin(THETA)$
 $HI1 := R0 * \cos(THETA)$

No Yes
 IA = 1

$FIO := FIO * SN + FII * CN$
 $GIO := GIO * SN + GII * CN$
 $HIO := HIO * SN + HI1 * CN$
 $T2 := SN \quad T4 := CN$

$FIO := FIO * CN - FII * SN$
 $GIO := GIO * CN - GII * SN$
 $HIO := HIO * CN - HI1 * SN$
 $T2 := CN \quad T4 := -SN$

I = 1

$CI := I - \text{ADD}[I] + \text{ADD}[I - 1] + 1$

$CI := 1$

$J := CI (1) 2 * N1$

$J \leq I$

1

4

1

4

No Yes

J=1

CJ:=J-ADD[J]+ADD[J-1]+1

CJ:=1

DI:=I
DJ:=J

Yes

CJ>I

No

DI:=J

DJ:=I

JA:=J-2*ENTIER(J/2)

No

JA=1

Yes

THETA:=-3.14159+
2*(J/2-1)*3.14159
/(N1-1)

THETA:=-3.14159+2*
((J+1)/2-1)*
3.14149/(N1-1)

FJO:=(RO/2)+0.5*((2*KAP-1)*COS
(THETA/2)-COS(1.5*THETA))/(4*G)
GJO:=(RO/2)+0.5*((2*KAP+3)*SIN
(THETA/2)+SIN(1.5*THETA))/(4*G)
FJI:=(RO/2)+0.5*((2*KAP+1)*SIN
(THETA/2)-SIN(1.5*THETA))/(4*G)
GJI:=- (RO/2)+0.5*((2*KAP-3)*COS
(THETA/2)+COS(1.5*THETA))/(4*G)
HJO:= -RO*SIN(THETA)
HJ1:=RO*COS(THETA)

No

JA=1

Yes

FJO:=FJO*SN+FJ1*CN
GJO:=GJO*SN+GJ1*CN
HJO:=HJO*SN+HJ1*CN
T1:=CN T3:=SN

FJO:=FJO*CN-FJ1*SN
GJO:=GJO*CN-GJ1*SN
HJO:=HJO*CN-HJ1*SN
T1:=-SN T3:=CN

1

5

1 5

```
TEMP1:=ADD[DI]-DI+DJ
KT[1,1]:=KT[1,1]+FIO*FJO*K[TEMP1]
KT[1,2]:=KT[1,2]+GIO*FJO*K[TEMP1]
KT[2,2]:=KT[2,2]+GIO*GJO*K[TEMP1]
KT[1,4]:=KT[1,4]+FIO*T1*K[TEMP1]
KT[2,4]:=KT[2,4]+GIO*T1*K[TEMP1]
KT[3,4]:=KT[3,4]+T2*T1*K[TEMP1]
KT[1,3]:=KT[1,3]+T3*FIO*K[TEMP1]
KT[2,3]:=KT[2,3]+T3*GIO*K[TEMP1]
KT[3,3]:=KT[3,3]+T3*T2*K[TEMP1]
KT[4,4]:=KT[4,4]+T4*T1*K[TEMP1]
KT[1,5]:=KT[1,5]+FJO*HIO*K[TEMP1]
KT[2,5]:=KT[2,5]+GJO*HIO*K[TEMP1]
KT[3,5]:=KT[3,5]+T3*HIO*K[TEMP1]
KT[4,5]:=KT[4,5]*T1*HIO*K[TEMP1]
KT[5,5]:=KT[5,5]+HJO*HIO*K[TEMP1]
```

J:=6 (1) 4*N1+5

CJ:=J+2*N1-5-ADD[J+2*N1-5]
+ADD[J+2*N1-6]+1

I:=CJ (1) 2*N1

IA:=I-2*ENTIER(I/2)

No IA=1 Yes

THETA:=-3.14159+2*
(I/2-1)*3.14159/
(N1-1)

THETA:=3.14159+2*
((I+1)/2-1)*
3.14159/(N1-1)

6

```

FIO:=(RO/2)+0.5*((2*KAP-1)*COS(THETA/2)
-COS(1.5*THETA))/(4*G)
GIO:=(RO/2)+0.5*((2*KAP+3)*SIN(THETA/2)
+SIN(1.5*THETA))/(4*G)
FI1:=(RO/2)+0.5*((2*KAP+1)*SIN(THETA/2)
-SIN(1.5*THETA))/(4*G)
GI1:=-RO/2+0.5*((2*KAP-3)*COS(THETA/2)
+COS(1.5*THETA))/(4*G)
HIO:=-RO*SIN(THETA)
HI1:=RO*COS(THETA)

```

No

IA=1

Yes

```

FIO:=FIO*SN+FI1*CN
GIO:=GIO*SN+GI1*CN
HIO:=HIO*SN+HI1*CN
T1:=SN  T2:=CN

```

```

FIO:=FIO*CN-FI1*SN
GIO:=GIO*CN-GI1*SN
HIO:=HIO*CN-HI1*SN
T1:=CN  T2:=SN

```

```

TEMP1:=ADD[J+2*N1-5]-J-2*N1
+5+ I
KT[1,J]:=KT[1,J]+FIO*K[TEMP1]
KT[2,J]:=KT[2,J]+GIO*K[TEMP1]
KT[3,J]:=KT[3,J]+T1*K[TEMP1]
KT[4,J]:=KT[4,J]+T2*K[TEMP1]
KT[5,J]:=KT[5,J]+HIO*K[TEMP1]

```

TI:=NG-2*N1

I:=1 (1) TI

```

CI:=I+2*N1-ADD[I+2*N1]+ADD[I+
2*N1-1]+1

```

CI ≤ 2*N1

No

Yes

7

8

CI:=2*N1+1

ADD[I]:=I+2*N1-CI+ADD[I-1]+1

J:=CI (1) I+2*N1

K[ADD[I]-I+J-2*N1]:=K[ADD[I+2*N1]-I-2*N1+J]

I:=1 (1) 5

ADD[TI+I]:=ADD[TI+I-1]+TI+I

I:=1 (1) TI

J:=1 (1) NSETF

Q[I,J]:=Q[I+2*N1,J]

I:=1 (1) 5

J:=1 (1) NSETF

Q[I+TI,J]:=0.0

I:=1 (1) TI

J:=1 (1) 5

K[ADD[TI+J]-TI-J+I]:=0.0

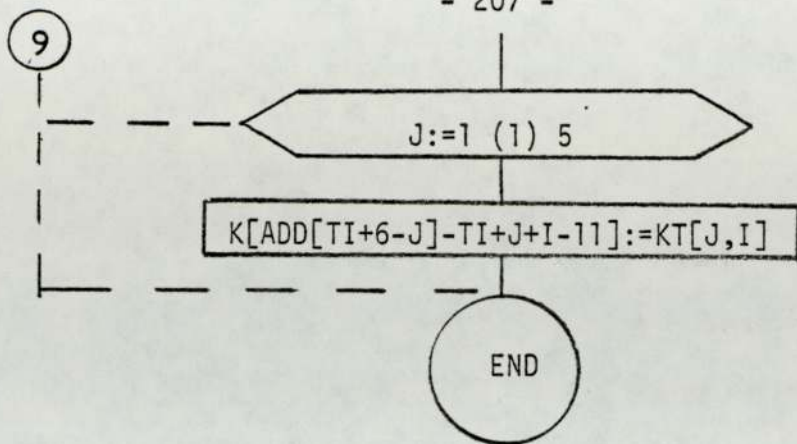
I:=1 (1) 5

J:=1 (1) I

K[ADD[TI+I]-I+J]:=KT[6-I,6-J]

I:=6 (1) 4*N1+5

9

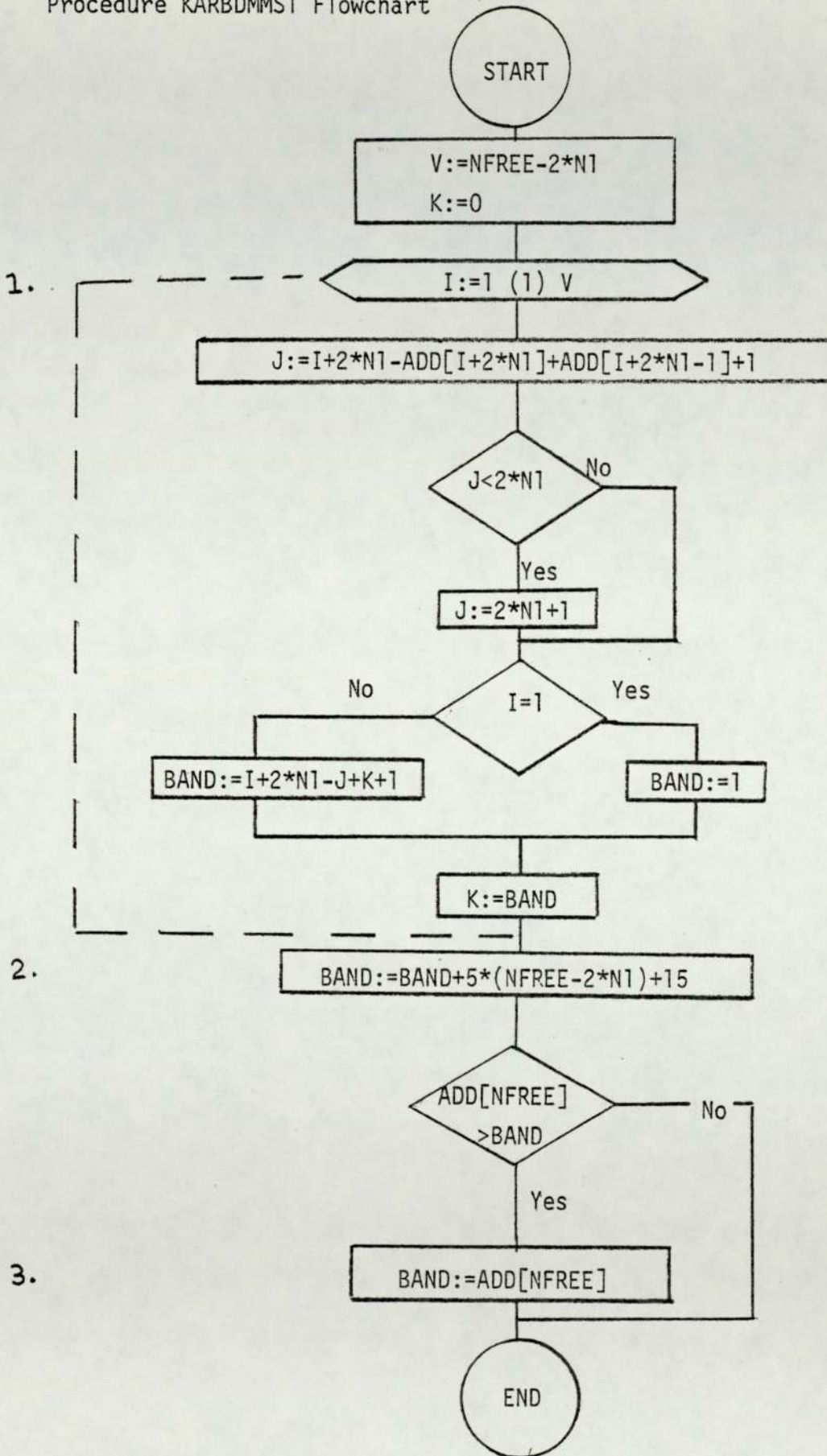


7.15 Procedure KARBDMMST

In order to declare the correct size array for storing the stiffness matrix coefficients, this procedure calculates the number of coefficients in $[K]^*$ and compares it with $[K]$ and fixes the size according to the larger one for the mixed mode I and II problem by the following steps:

1. After eliminating rows and columns (1) to $(2N_1)$ from $[K]$, the number of remaining coefficients is calculated.
2. Coefficients in rows $NG-2N_1+1$ to $NG - 2N_1+5$ are added to those steps (1) and the result is called BAND.
3. The new size is declared $K[1:BAND]$.

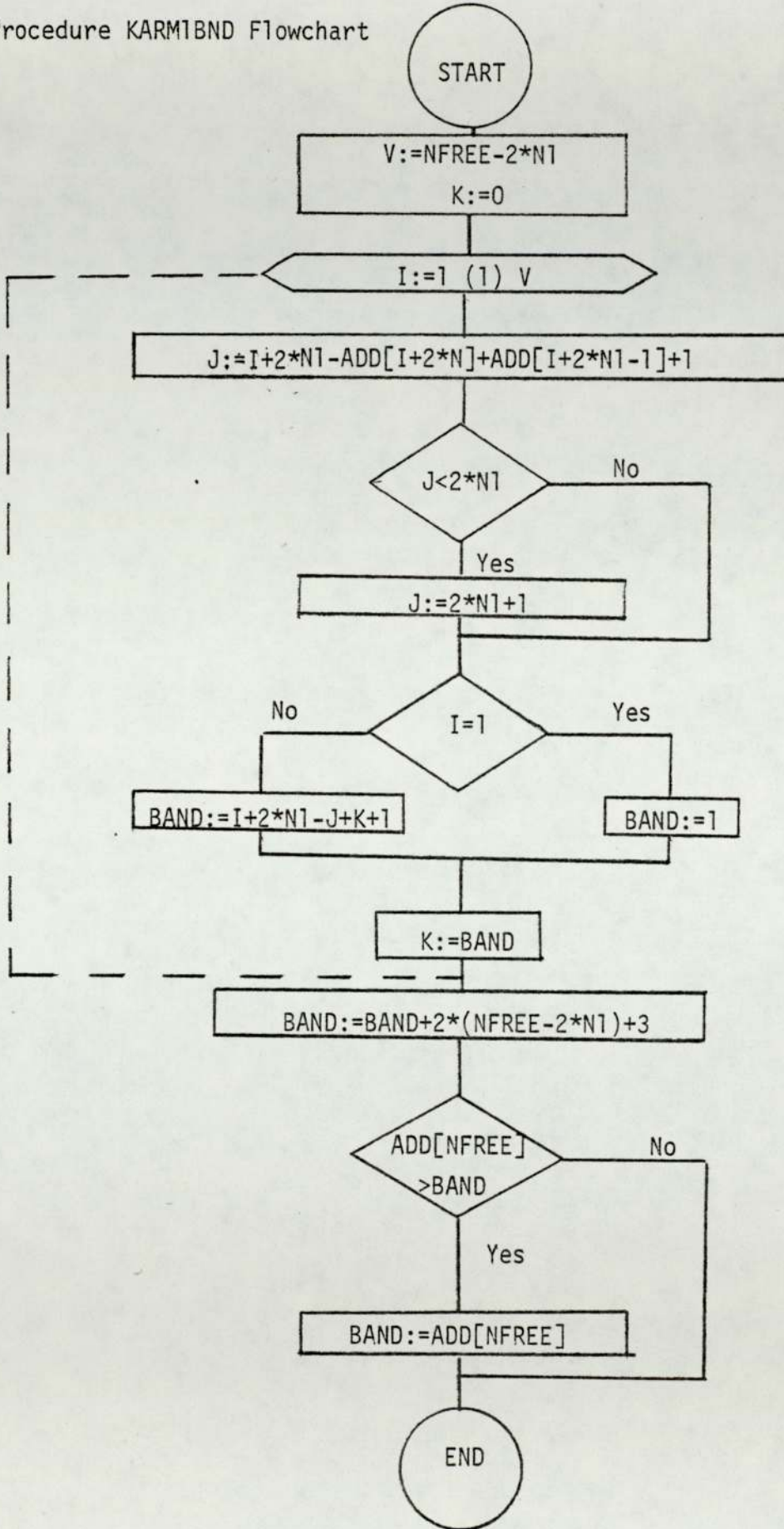
Procedure KARBDMMST Flowchart



7.16 Procedure KARM1BND

This procedure is similar to (KARBDMMST) of section (7.15) but is for mode I problems. The only difference between them is in step (2) where only coefficients of rows $NG-2N1+1$ and $NG-2N1+2$ are added.

Procedure KARM1BND Flowchart



7.17 Procedure RESIDUAL

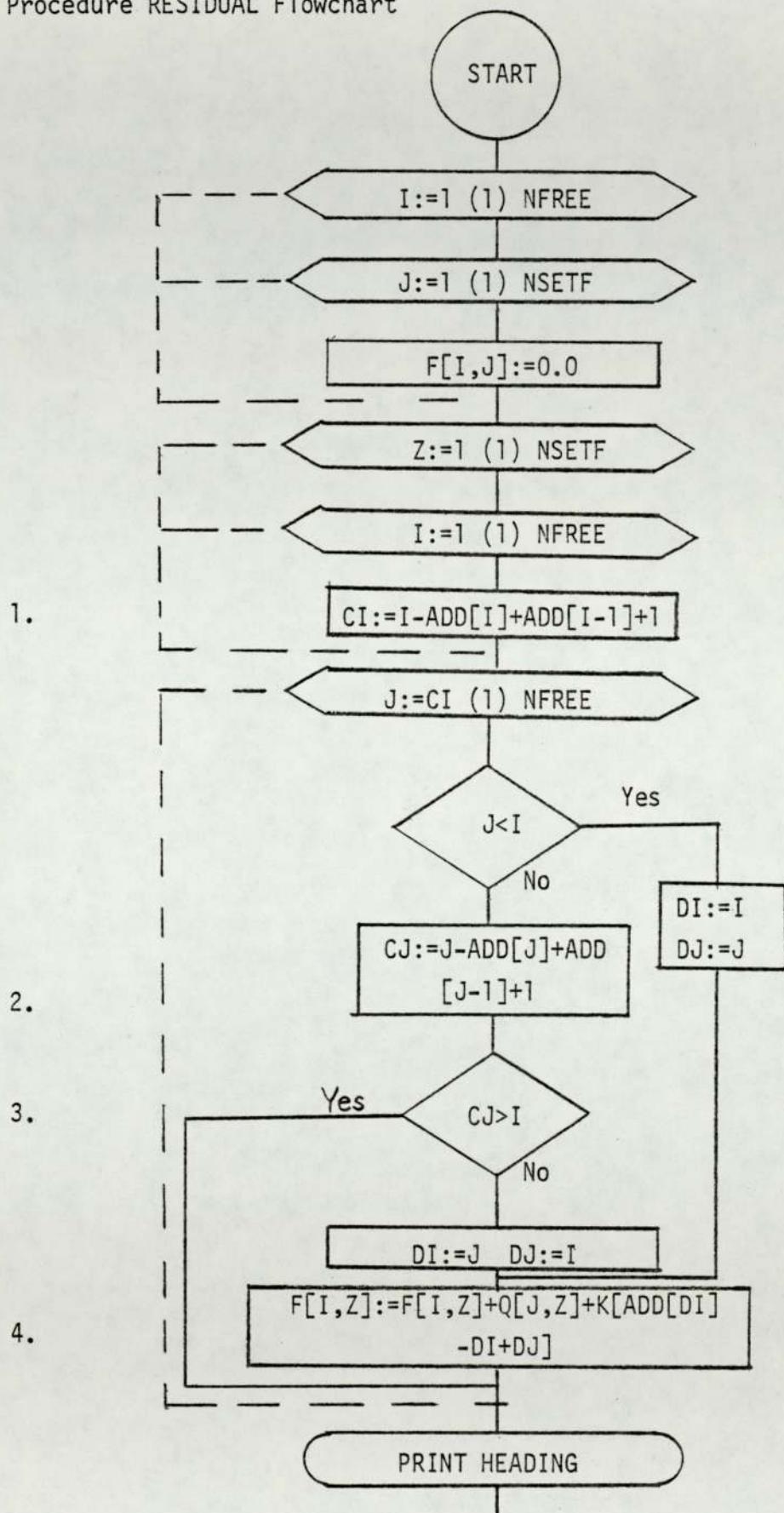
It will be shown later in section (8.4.1) that the conditioning of a set of linear simultaneous equilibrium equations may be checked by resubstituting the displacement vector $\{q\}$ obtained from the solution to evaluate a new load vector $\{Q\}'$ which is compared with the original $\{Q\}$. Also by examining $\{Q\}'$, any constraint applied by mistake to the structure can be spotted. It must be noted that for this purpose the original overall stiffness matrix and load vector before modification are required, therefore the back store of the computer is used to store them in their original form and they are recalled before calling this procedure. This was done to avoid having to store both versions of the stiffness matrix and load vector at the same time and hence not leaving enough room for the solution of the problem.

The procedure steps are as follows:

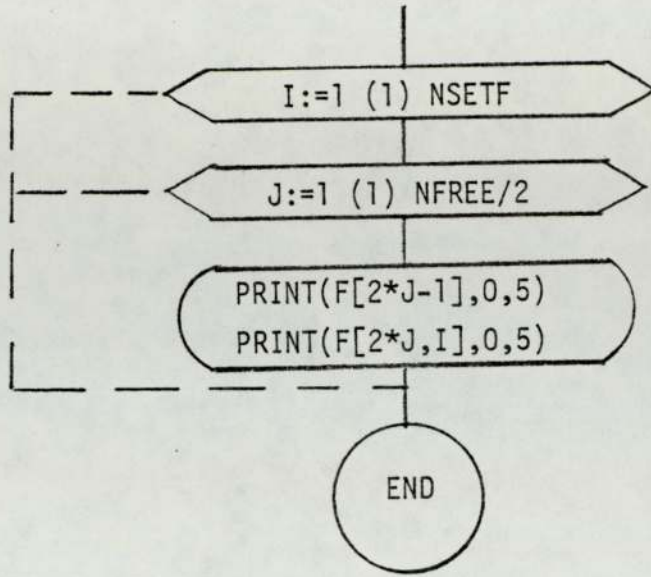
1. A loop I on the rows numbers from 1 to number of degrees of freedom is constructed.
2. The column number (CI) of the first non-zero coefficient of row I is calculated.
3. If $CI < I$ then the stiffness coefficient is stored and matrix multiplications are carried out to evaluate Q_I' .
4. If $CI > I$ then symmetry is used in the same way as shown in section (7.13) to obtain the appropriate stiffness matrix coefficient.

5. Coefficient of the new load vector $\{Q\}'$ are outputed for nodes $N1+1$ to number of nodes of the structure as the core nodes are free from external loading.

Procedure RESIDUAL Flowchart



5.



7.18 Procedure EO4AAA

This is a standard procedure from the Nottingham algorithm group library to minimize a function. It is used to minimize the strain energy density function given in equation (7.24) and finds the corresponding angle which according to Sih's criterion is the angle of crack initiation. With reference to Fig. (7.5) the method used basically works by fitting a quadratic through $(\theta_1, y(\theta_1))$, $(\theta_2, y(\theta_2))$, $(\theta_3, y(\theta_3))$ of the form:

$$Y = a + b\theta + c\theta^2 \quad (7.89)$$

It finds the point where $Y' = 0$ and calls it θ_4 and operates again using $\theta_2, \theta_3, \theta_4$ to find θ_5etc. The procedure requires specifying θ_1 , a range where the minimum value is expected, and the maximum number of iterations to be done. This procedure is stored in a precompiled form and no flowchart or listing is available for it.

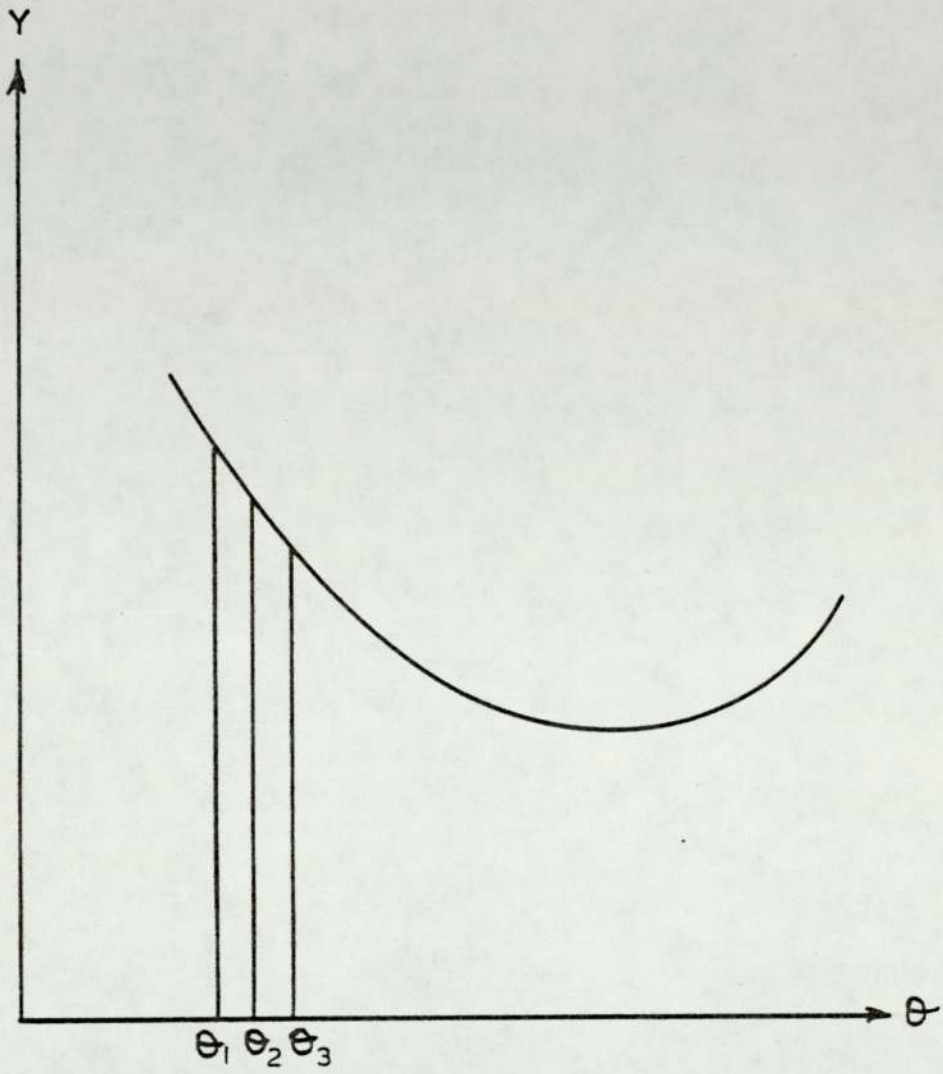


Fig. 7.5

CHAPTER 8

NUMERICAL EXAMPLES

8.1 Introduction

The various procedures needed to build up a finite element program to solve axisymmetric problems subject to axisymmetric loading, and those required to include a singular core element for the calculation of mode (I) or mixed mode (I) and (II) stress intensity factors have been presented in Chapter 7. The general finite element program was developed to form the basis for the fracture analysis programs, and its results were tested against known analytical solutions and others obtained by finite element methods using constant strain elements or linear strain ones formulating the elements stiffness matrices by explicit multiplication and term-by-term integration. The comparison between the results was used to draw conclusions about the superiority of the linear strain element and the accuracy of the numerical integration technique adopted. The fracture programs were tested by solving problems whose solution by other methods was available. The confidence gained in the program cleared the way to the solution of interesting fracture problems which have never appeared in the published literature. Among them were those investigating the effects of grooves, voids and inclusions of different material properties on mode (I) and mixed mode (I) and (II) stress intensity factors.

The complete programs, with input data instructions and results of sample problems, are presented in (Appendix 10.4).

This chapter is divided into four sections. The first is devoted to non fracture problems by the general finite element program while the second is for mode (I) fracture problems, and the third for mixed mode (I) and (II) fracture problems. In the fourth section, the influence of inclusions of different material properties on the values of the stress intensity factors in single and mixed mode fracture problems are studied.

8.2 Applications of the general axisymmetric program.

8.2.1 Thick cylinder under internal pressure

This example was solved to demonstrate the accuracy of the numerical integration technique used, and the superiority of the linear strain elements. The Lamé solution for the calculation of stresses in the thick cylinder are, [6]:

$$\sigma_r = \frac{a^2 p}{b^2 - a^2} \left(1 - \frac{b^2}{r^2}\right) \quad (8.1)$$

$$\sigma_\theta = \frac{a^2 p}{b^2 - a^2} \left(1 + \frac{b^2}{r^2}\right) \quad (8.2)$$

where: a = internal radius
 b = external radius
 P = internal pressure

Meek and Carey, [6], solved this problem using constant strain elements and linear ones with explicit multiplication and term-by-term integration to formulate the elements stiffness matrices.

In order to be able to draw conclusions regarding the accuracy of the numerical integration technique, the discretization used in solving this problem was the same as that used by Meek and Carey for their linear strain element, and is shown in Fig. (8.1).

Table (8.1) shows that the results obtained using linear strain elements and numerical integration are very close to those using term-by-term integration, and much closer to the theoretical results than those obtained from a constant strain element.

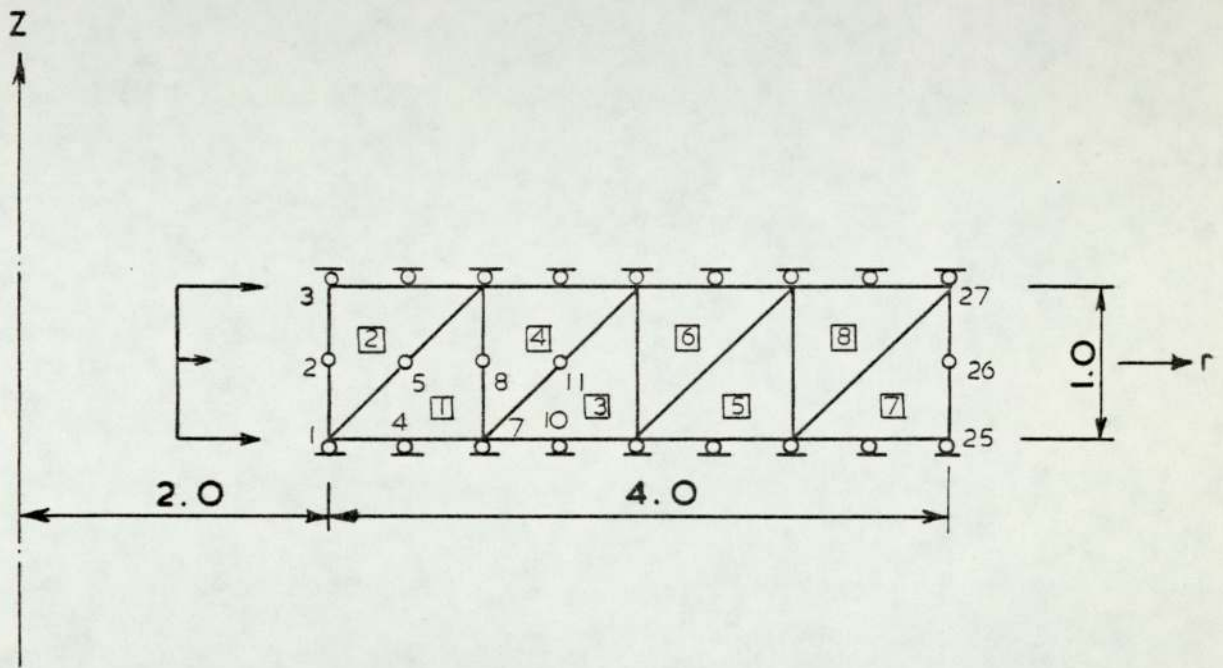


Fig. 8.1

Node No.	Axisym.F.E. prog.		M&C C.S. element		M&C L.S. element		Theoretical (Lame)	
	σ_{θ}	σ_r	σ_{θ}	σ_r	σ_{θ}	σ_r	σ_{θ}	σ_r
2	25.569	-18.606	26.21	-14.98	25.34	-18.7	25	-20
5	16.855	-11.877	16.10	-14.38	16.79	-12.32	16.9	-11.9
8	12.69	-7.009	12.03	-8.68	12.64	-6.89	12.5	-7.5
11	9.8642	-4.7782	9.53	-5.49	9.82	-4.92	9.846	-4.846
14	8.1793	-2.9762	7.89	-3.52	8.17	-2.94	8.125	-3.125
17	6.9468	-1.9207	6.77	-2.21	6.93	-1.97	6.94	-1.94
20	6.1184	-1.0436	5.96	-1.29	6.12	-1.03	6.1	-1.1
23	5.4734	-0.4678	5.35	-0.63	5.47	-0.49	5.475	-0.475
26	5.008	-0.029	4.84	-0.31	5.02	-0.08	5.0	0

TABLE (8.1)

8.2.2 Circular plate under uniform pressure

This problem was solved to study the effect of approximating the radius, for elements which have one side on the axis of revolution, to avoid computational problems arising from the calculation of the hoop strain (u/r) which was discussed in section (7.2).

In the physical situation, the applied pressure is uniformly distributed over the upper surface of the plate. When Meek and Carey, [6], solved a similar problem, they noted that distributing the load on the centroidal plane gives better agreement with theoretical solution obtained from plate bending (small deflections) theory than that on the upper surface. They concluded that this indicates that the simple mechanics solution is not a particularly accurate representation of the physical situation.

The problem of a simply supported circular plate under uniform pressure was solved with mesh (1) of Fig. (8.2) and the loads distributed on its upper and centroidal plane. The results are shown in Table (8.2), where it is seen that the values for the centroidal plane deflection and the hoop and radial stresses of the lower surface are very close for the two loading situations. The values however of the hoop and radial stress of the centroidal plane are different. When the centroidal plane is loaded, they are very small compared with the lower surface ones and therefore it can be assumed that the centroidal plane stresses are approximately equal to zero. But when the upper surface is loaded, the stresses of the centroidal plane are not very small and cannot be assumed to equal zero.

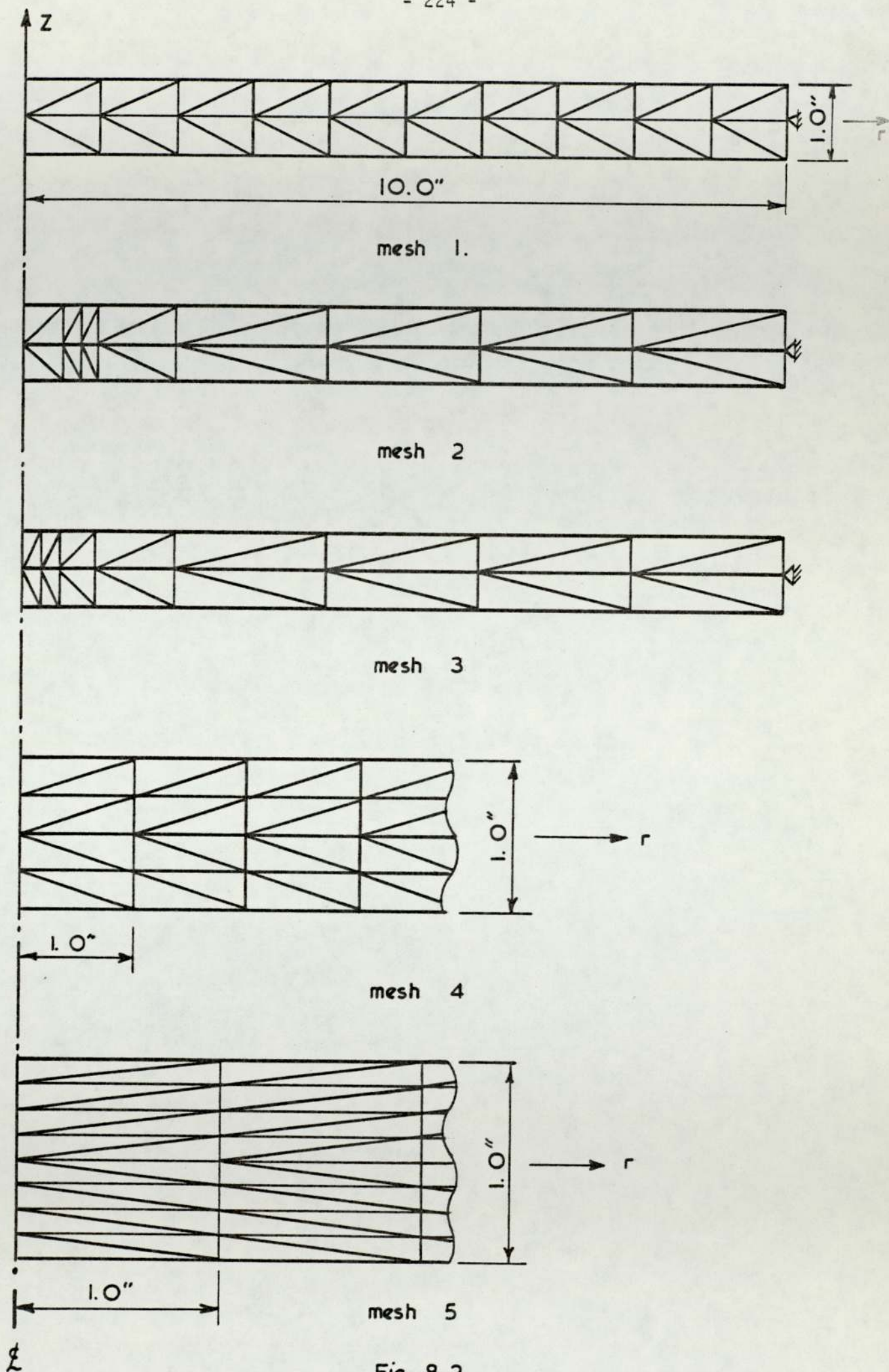


Fig. 8.2

Dist. from ζ (in)	$\delta_{\text{mid.pl.}}$	$\delta^*_{\text{mid.pl.}}$	σ_{θ} surface	σ^*_{θ} surface	σ_r surface	σ^*_r surface	σ_{θ} mid.pl.	σ^*_{θ} mid.pl.	σ_r mid.pl.	σ^*_r mid.pl.
0	2.69×10^{-4}	2.68×10^{-4}	-1.35	-0.96	4.39	4.58	3.76×10^{-6}	1.10×10^{-1}	2.89×10^{-6}	5.05×10^{-2}
1	2.66×10^{-4}	2.66×10^{-4}	236	236	191	191	6.90×10^{-6}	1.43×10^{-1}	6.30×10^{-6}	1.56×10^{-1}
2	2.56×10^{-4}	2.56×10^{-4}	122	122	120	120	4.51×10^{-6}	1.35×10^{-1}	5.49×10^{-6}	1.40×10^{-1}
3	2.40×10^{-4}	2.39×10^{-4}	117	117	112	112	3.97×10^{-6}	1.36×10^{-1}	2.16×10^{-6}	1.37×10^{-1}
4	2.18×10^{-4}	2.17×10^{-4}	112	112	103	103	3.14×10^{-6}	1.36×10^{-1}	1.29×10^{-6}	1.37×10^{-1}
5	1.90×10^{-4}	1.89×10^{-4}	106	106	92	92	2.52×10^{-6}	1.36×10^{-1}	-1.19×10^{-7}	1.37×10^{-1}
6	1.58×10^{-4}	1.57×10^{-4}	99	99	79	79	1.50×10^{-6}	1.37×10^{-1}	-8.10×10^{-7}	1.37×10^{-1}
7	1.21×10^{-4}	1.12×10^{-4}	91	91	63	63	9.56×10^{-7}	1.37×10^{-1}	-1.61×10^{-6}	1.37×10^{-1}
8	8.25×10^{-5}	8.23×10^{-5}	81	81	45	45	5.07×10^{-7}	1.37×10^{-1}	-2.17×10^{-6}	1.35×10^{-1}
9	4.15×10^{-5}	4.14×10^{-5}	70	70	24	24	5.80×10^{-8}	1.31×10^{-1}	-2.91×10^{-6}	1.52×10^{-1}
10	0	0	58	59	4	4	-2.08×10^{-6}	2.21×10^{-1}	-6.99×10^{-6}	3.73×10^{-1}

* denotes surface loading

TABLE (8.2)

Due to this the centroidal plane loading situation is closer to the elementary theory which assumes that plane sections remain plane.

The deflection of centroidal plane nodes obtained from the finite element solution are plotted with the theoretical values in Fig. (8.3). Two things are seen in this figure, the first is that the deflection curve of the finite element solution is lower than the theoretical one, and the second is that this difference is limited to the nodes near the axis of revolution after which very good agreement is obtained between them. It is thought that the first is possibly due to the fact that the centroidal plane nodes are under the direct influence of point loads simulating a uniform pressure and hence the deflections obtained are expected to be on the larger side of the theoretical values. The second is due to the effect of approximating the radius for elements on the axis of revolution. The influence of the mesh design on the accuracy near the axis of revolution due to approximating the radius was studied by solving the same problem with another four meshes 2 to 5 of Fig. (8.2). The results obtained are shown in Table (8.3) where it can be seen that they are very close with better results obtained from meshes with elements of the same size (1, 4 and 5).

The values of the radial and hoop stresses of the lower surface of the plate obtained from mesh (1) were plotted with the theoretical ones in Fig. (8.4). It is again seen that good agreement is obtained after a small distance from the axis of revolution.

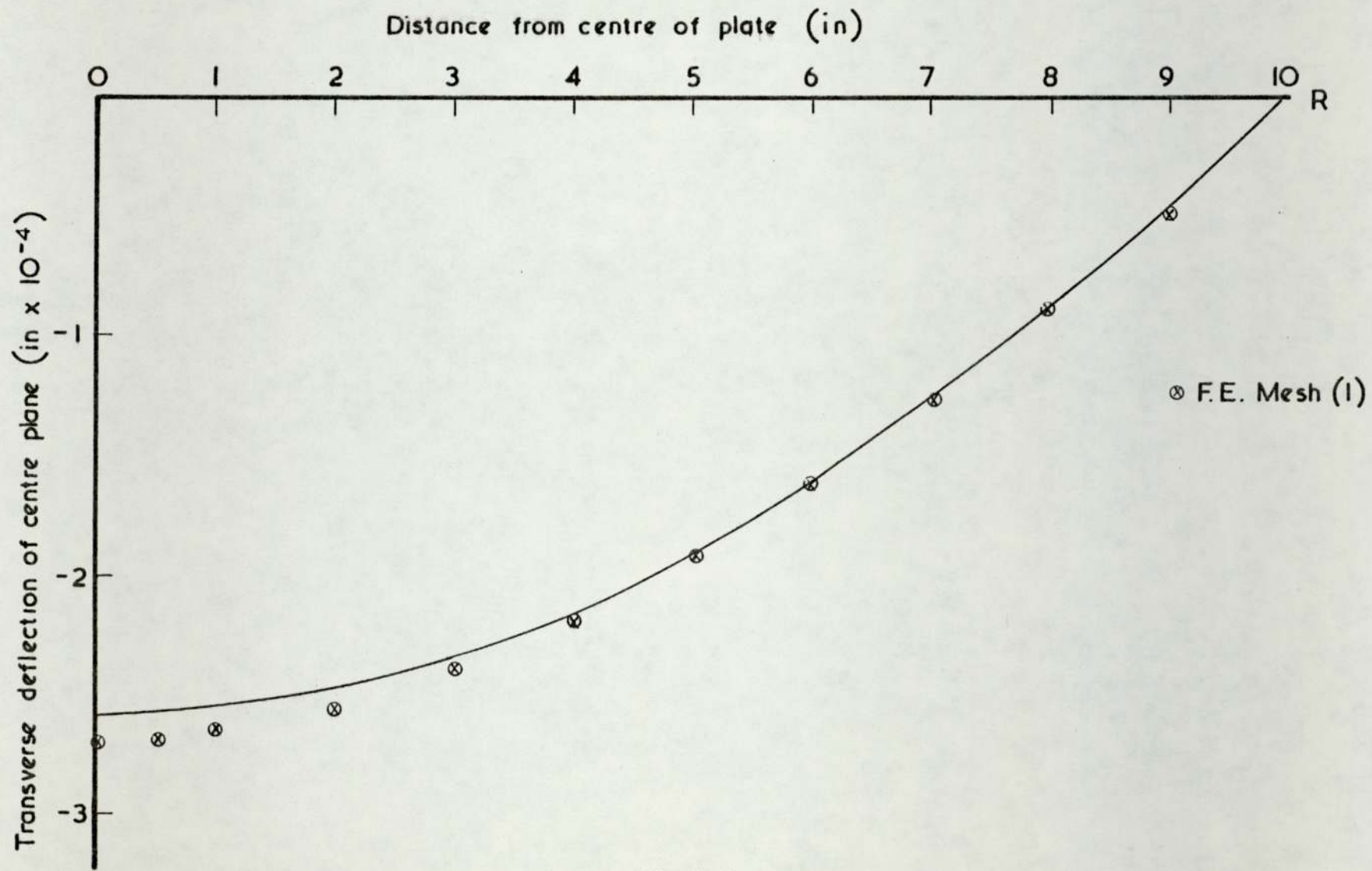


Fig. 8.3

Distance from ϕ (in)	Centroidal plane deflection (in $\times 10^{-4}$)				
	mesh(1)	mesh(2)	mesh(3)	mesh(4)	mesh(5)
0	2.690	2.716	2.716	2.681	2.637
1	2.667	2.684	2.682	2.655	2.614
2	2.566	2.582	2.581	2.557	2.526
3	2.403	2.418	2.417	2.396	2.369
4	2.180	2.194	2.193	2.175	2.151
5	1.904	1.914	1.914	1.899	1.879
6	1.580	1.588	1.588	1.577	1.560
7	1.217	1.223	1.222	1.215	1.202
8	0.825	0.829	0.828	0.824	0.815
9	0.415	0.416	0.416	0.415	0.410
10	0.0	0.0	0.0	0.0	0.0

TABLE (8.3)

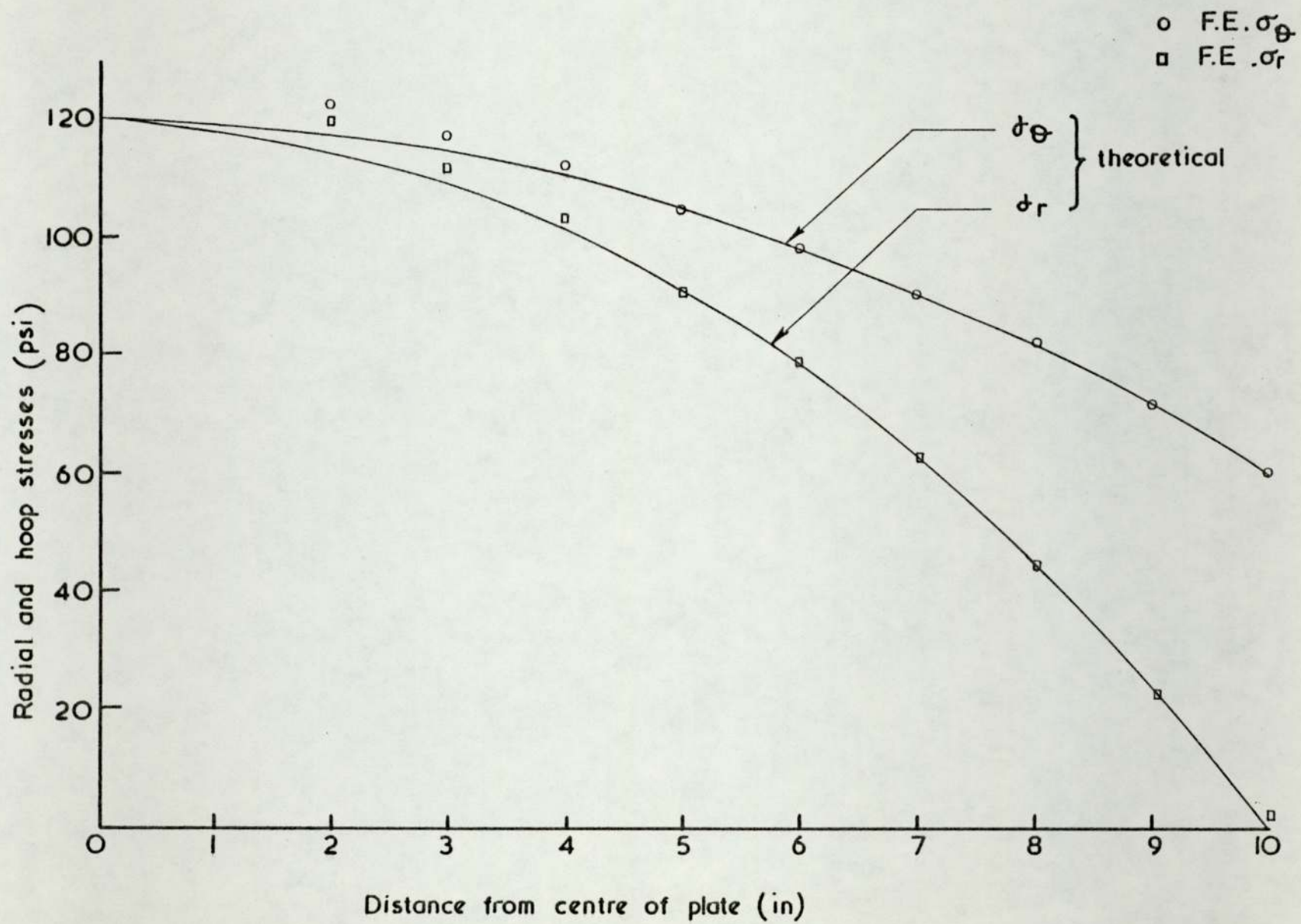


Fig. 8.4

The problem of a circular clamped plate under uniform pressure Fig. (8.5) was solved and the results of the centre plane deflection and the radial and hoop stresses of the lower surface were plotted with the theoretical ones obtained from, [80], in Figs. (8.6) and (8.7) respectively.

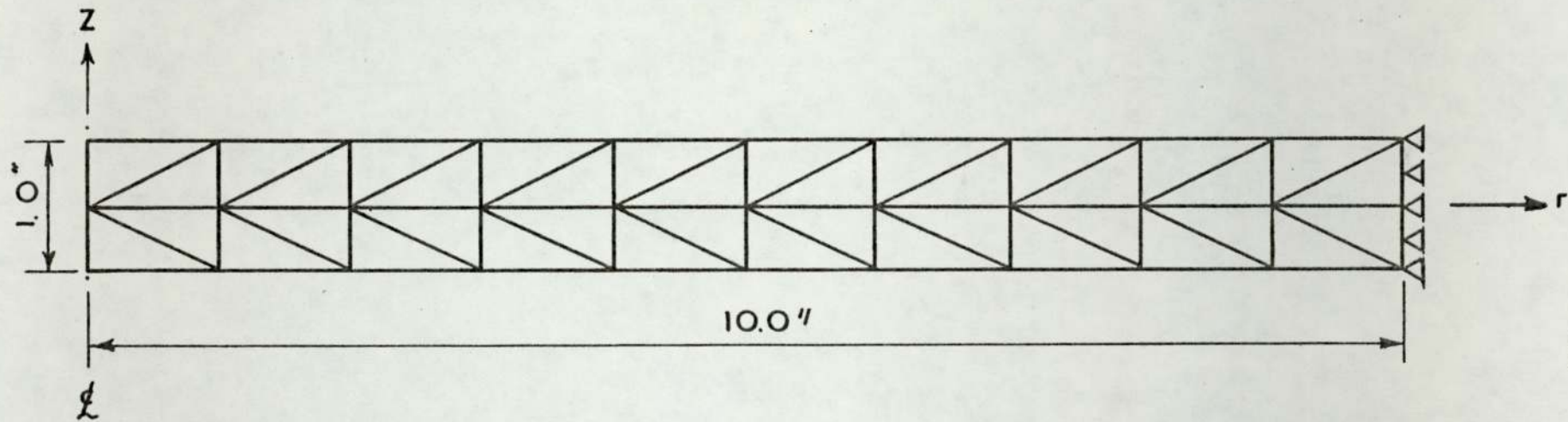


Fig . 8.5

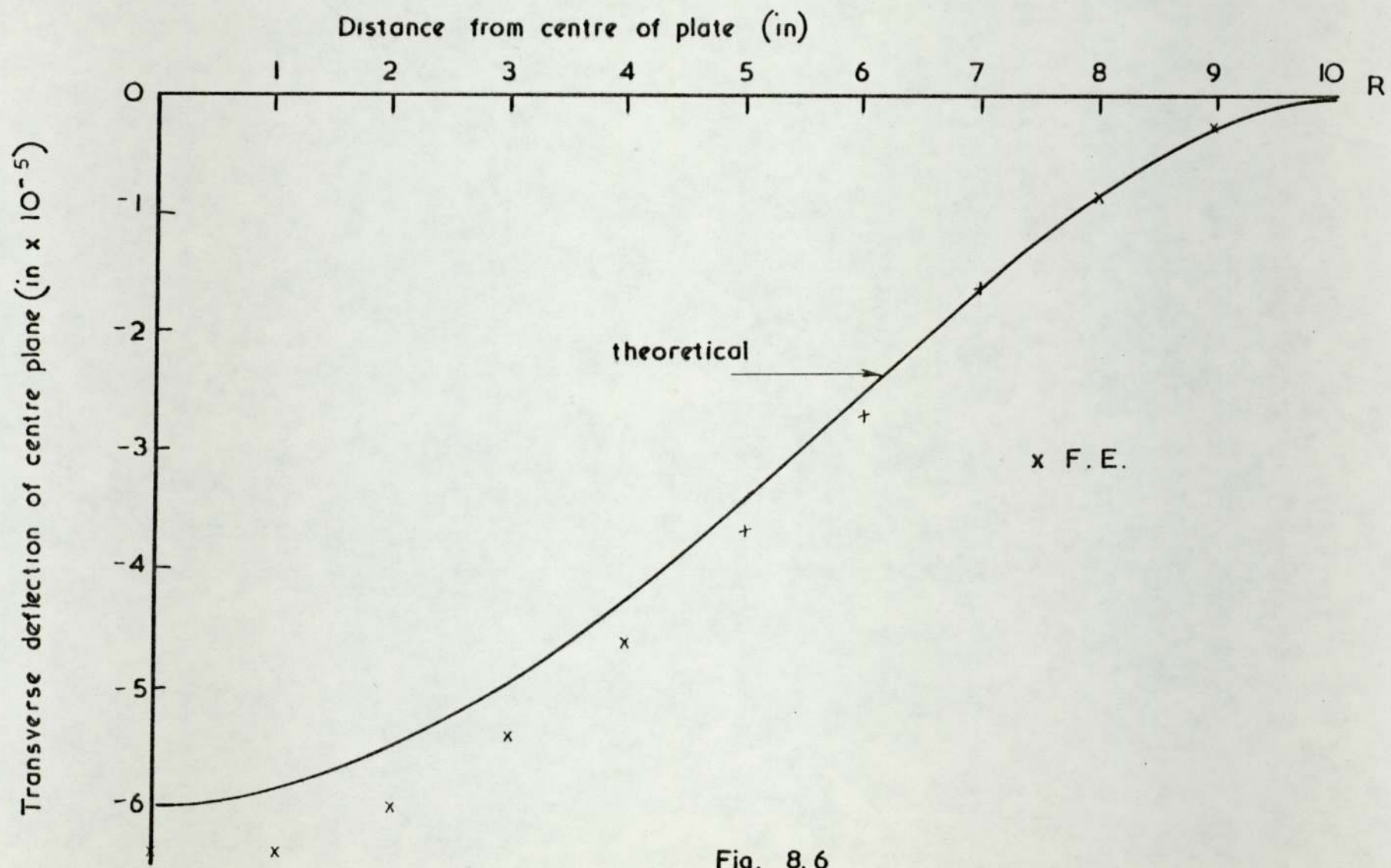


Fig. 8.6

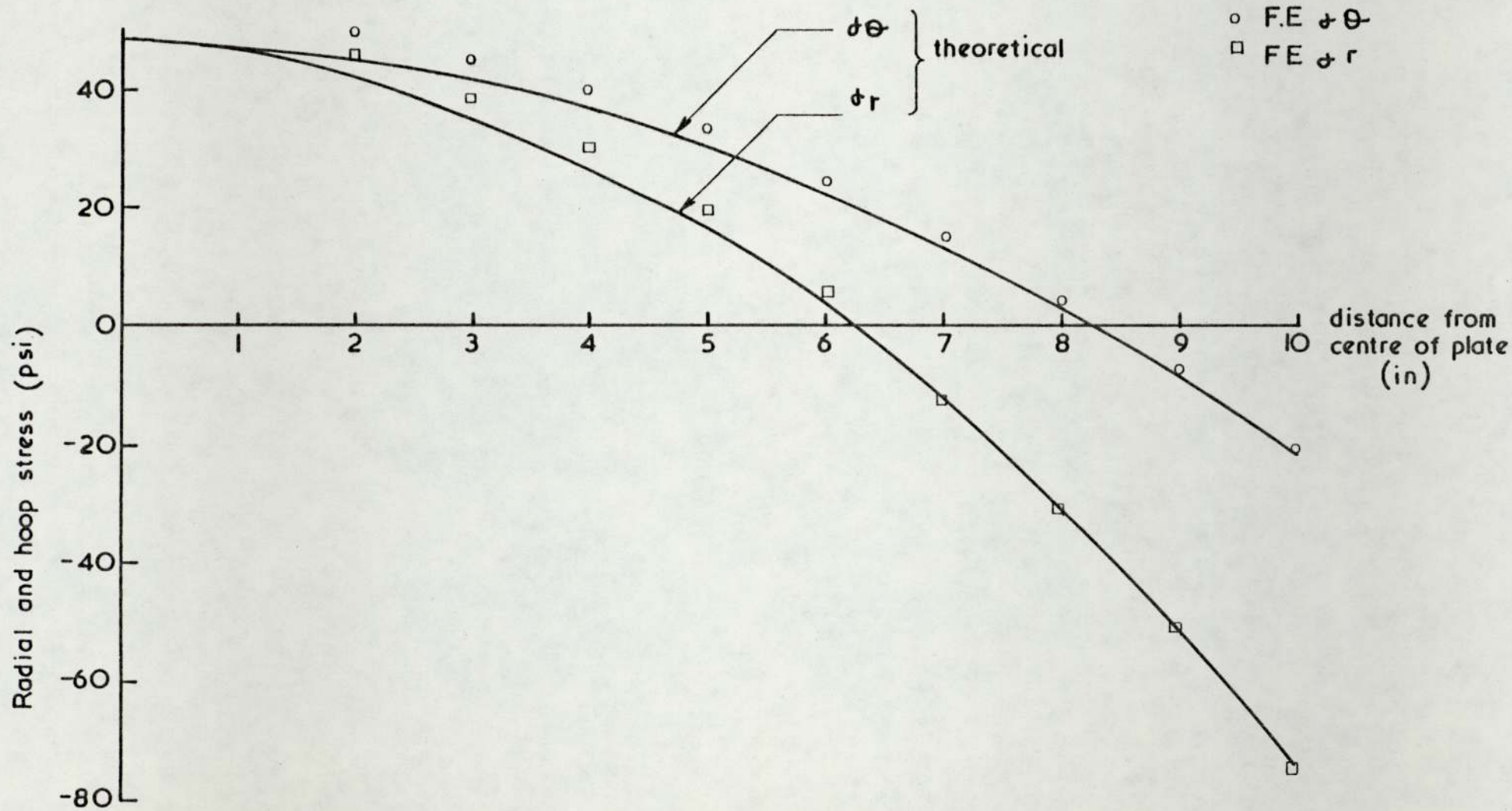


Fig. 8.7

8.3 Mode I fracture problems

8.3.1 A round bar with a circumferential normal edge crack

This problem was used to check the influence of core parameters, namely the core radius (R_c) and the number of nodes on the finite element/core interface (N_1), on the value of (K_I). The geometry of the round bar is shown in Fig. (8.8), with

$$h/R = 1.4$$

$$c/R = 0.5$$

The results were compared with a solution by finite elements to the same problem with ($N_1 = 21$) and the total number of nodes (297) by Hilton and Sih,[49], and another solution by Benthem and Koiter, [51].

Due to symmetry only half the bar need be considered and hence the core's shape is a semicircle. As ring elements are used, the mesh is generated for one quarter of the longitudinal section of the bar. The core radius was taken as ($R_c = 0.02 C$) and the number of nodes on the core ($N_1 = 19$). Two circular rings of elements surrounded the core with radii ($2R_c$) and ($3R_c$) after which the element distribution was changed to rectangular to match the boundary shape of the solid as shown in Fig. (8.9). The results for the dimensionless stress intensity factor (\bar{K}_I),

$$\bar{K}_I = \frac{K_I}{\sigma_0 (2R)^{\frac{1}{2}}} \quad (8.3)$$

where: σ_0 = the stress in the neck section

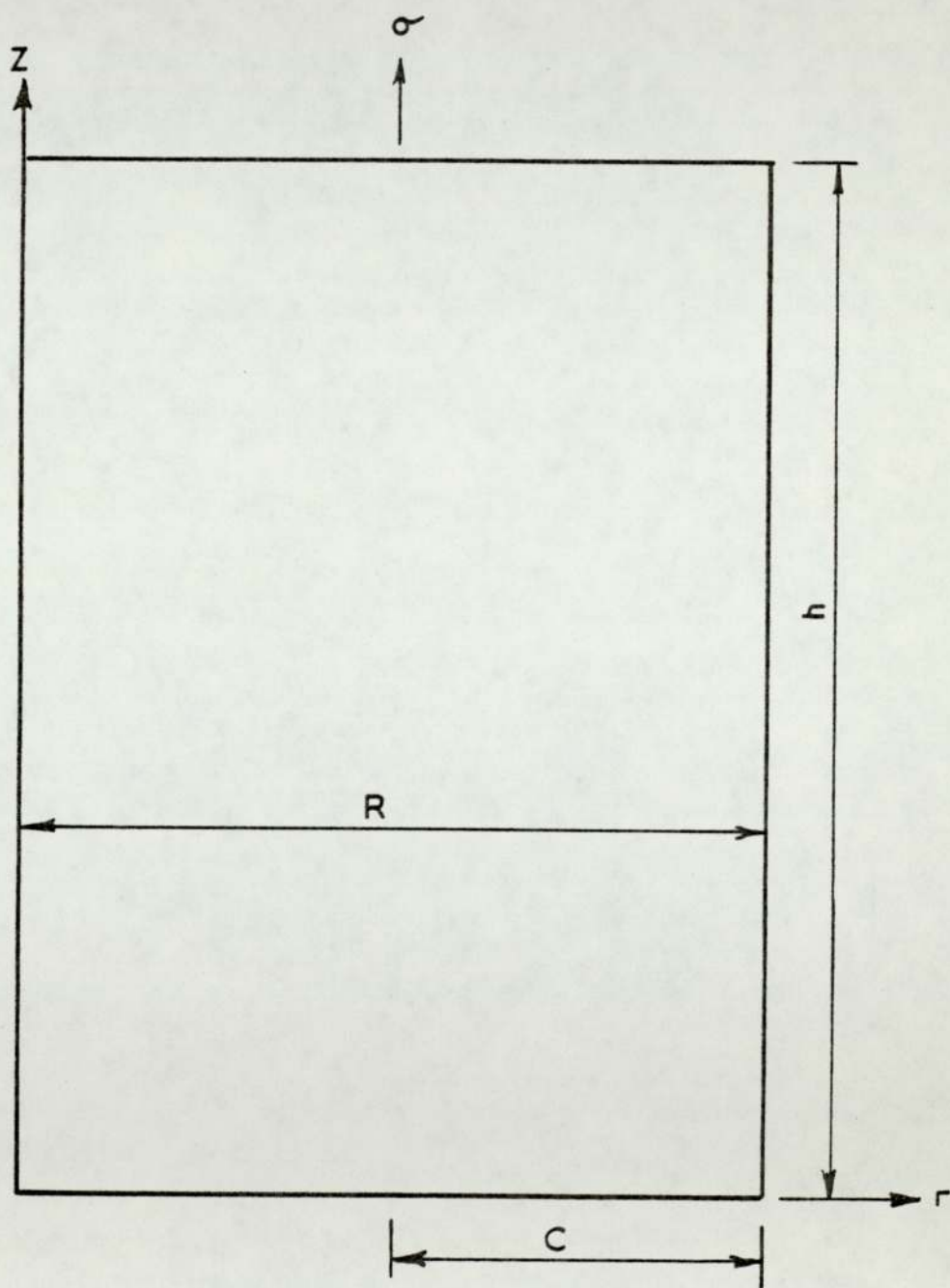
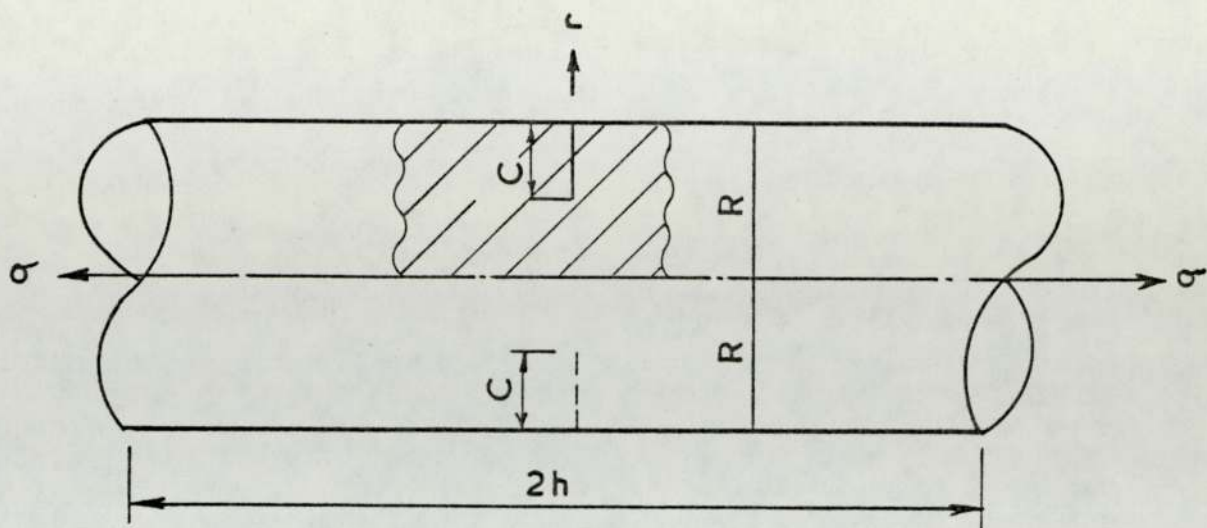


Fig . 8.8

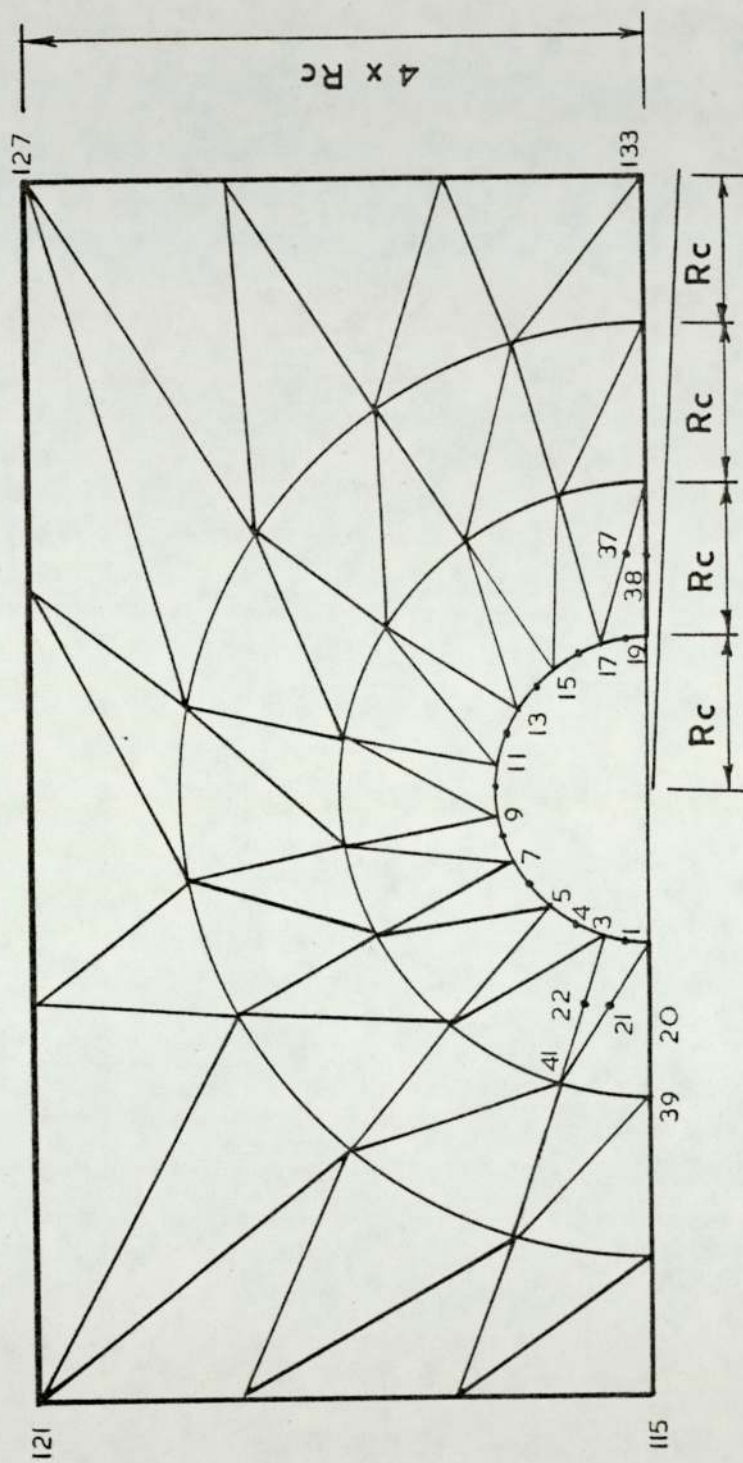


Fig. 8.9

are shown in Table (8.4); where it could be seen that although (N_1) was less than that used by Hilton and Sih and the total number of nodes was only (285), the results have good accuracy.

The problem was solved again with the same core radius but with ($N_1 = 15$) and ($N_1 = 9$), and total number of nodes (225) and (135) respectively. This had the effect of increasing the distance between nodes on the finite element/core interface, thus increasing the displacement incompatibility on the interface. To be able to assess the effect of the incompatibility the core radius in both cases was reduced in two stages to a value which made the distance between the nodes on the interface equal to that of ($N_1 = 19$) and ($R_c = 0.02C$). Therefore, the core radii for ($N_1 = 15$) were ($R_c = 0.0178C$) and ($R_c = 0.0156C$); and for ($N_1 = 9$) were ($R_c = 0.0145C$) and ($R_c = 0.009C$).

The value of (R_1) obtained in each case is compared to that of Benthem and Koiter in Table (8.5), and it is seen that the best agreement was obtained from the solution with ($N_1 = 19$) and ($R_c = 0.02C$). It could be concluded that although the displacement incompatibility at the interface affects the accuracy of the results, there are other factors involved which are the strain energy in the core and the back-up mesh expressed as the total number of nodes used.

It is noted that as the core radius decreases, the one term displacement expansions under estimates the strain energy in the core region resulting in large values for the stress intensity factor, [49]. The overall mesh should also be fine enough to filter through the effects of representing the distributed load by nodal forces and the effect of approximating the radius at elements on the axis of revolution.

	F.E. Program	F.E. Hilton & Sih	Benthem & Koiter
Dimensionless R_I	0.22774	0.235	0.237
% Accuracy	-	3.1	4.0

TABLE (8.4)

The total number of nodes which could be used depends on (N_1) and is limited by the computer store. Therefore, in order to stay within the capacity of the available computer, all the mode (I) problems solved had the parameters $(N_1 = 19)$ and a total number of nodes of (285).

N_1	R_C	K_I	Percentage discrepancy
9	0.009C	2477	+16%
	0.145C	2492	+17%
	0.02C	2504	+17.8%
15	0.0156C	2311	+8.7%
	0.0178C	2405	+13%
	0.02C	2441	+14.8%
19	0.02C	2050	-4%

TABLE (8.5)

8.3.2 A round bar with a normal penny shaped crack

This example consists of a bar similar in dimensions to the one in (8.3.1) with ($R_c = 0.02C$) but having a penny shaped crack, Fig. (8.10); and was solved also by Benthem and Koiter,[51]. The dimensionless stress intensity factor (\bar{K}_I),

$$\bar{K}_I = \frac{K_I}{\sigma(2R)^{\frac{1}{2}}} \quad (8.4)$$

where:

σ = the uniform stress remote
from the crack tip.

was found to be (0.34569) which is within (0.5%) of (0.34390) calculated by Benthem and Koiter.

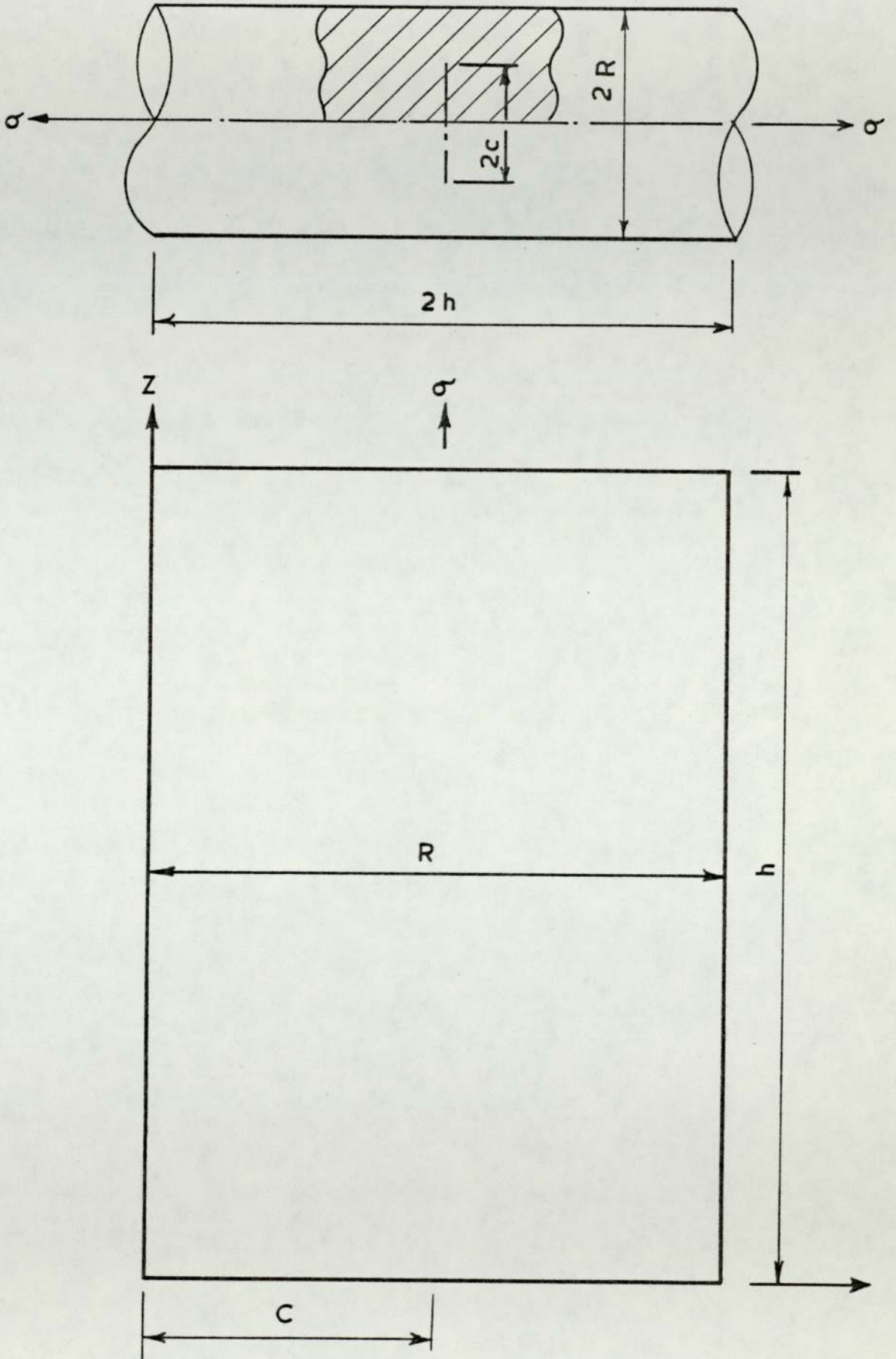


Fig. 8.10

8.3.3 A round bar with a crack extending from the base of an external groove

Cracks, when initiated, are often removed by cutting a groove in their place. A study was carried out to examine the results if a small part of the crack was left, or it initiates again from the base of the groove for various groove geometries. Another interesting example is the effect of a groove on a propagating crack. The different groove geometries studied, with reference to Fig. (8.11), were:

$$a/b = 0.0315, 0.063, 0.25, 0.5, 1, 2 \text{ and } 3.$$

$$\text{with } c/b = 0.28$$

The bar was subjected to a uniform tensile load remote from the crack plane of (500 psi) and the (K_I) results obtained are shown in Fig. (8.12). The two asymptotes in this figure represent the limits of the variation of (K_I). The upper one is the (K_I) value of a crack of length ($c + b$) in a bar of radius (R), and the lower one is that for a crack of length (c) in a bar of radius ($R - b$). The results show that (K_I) values are insensitive to small grooves and a reduction in them is not significant unless the groove opens up to an (a/b) ratio greater than unity.

For the example of a propagating crack from the base of a groove, a semicircular groove was chosen. The values of crack to groove ratio considered, with reference to Fig. (8.13), were:

$$c/b = 0.08, 0.09, 0.11, 0.13, 0.14, 0.2, 0.28, 0.34, 0.4, \\ \text{and } 0.44.$$

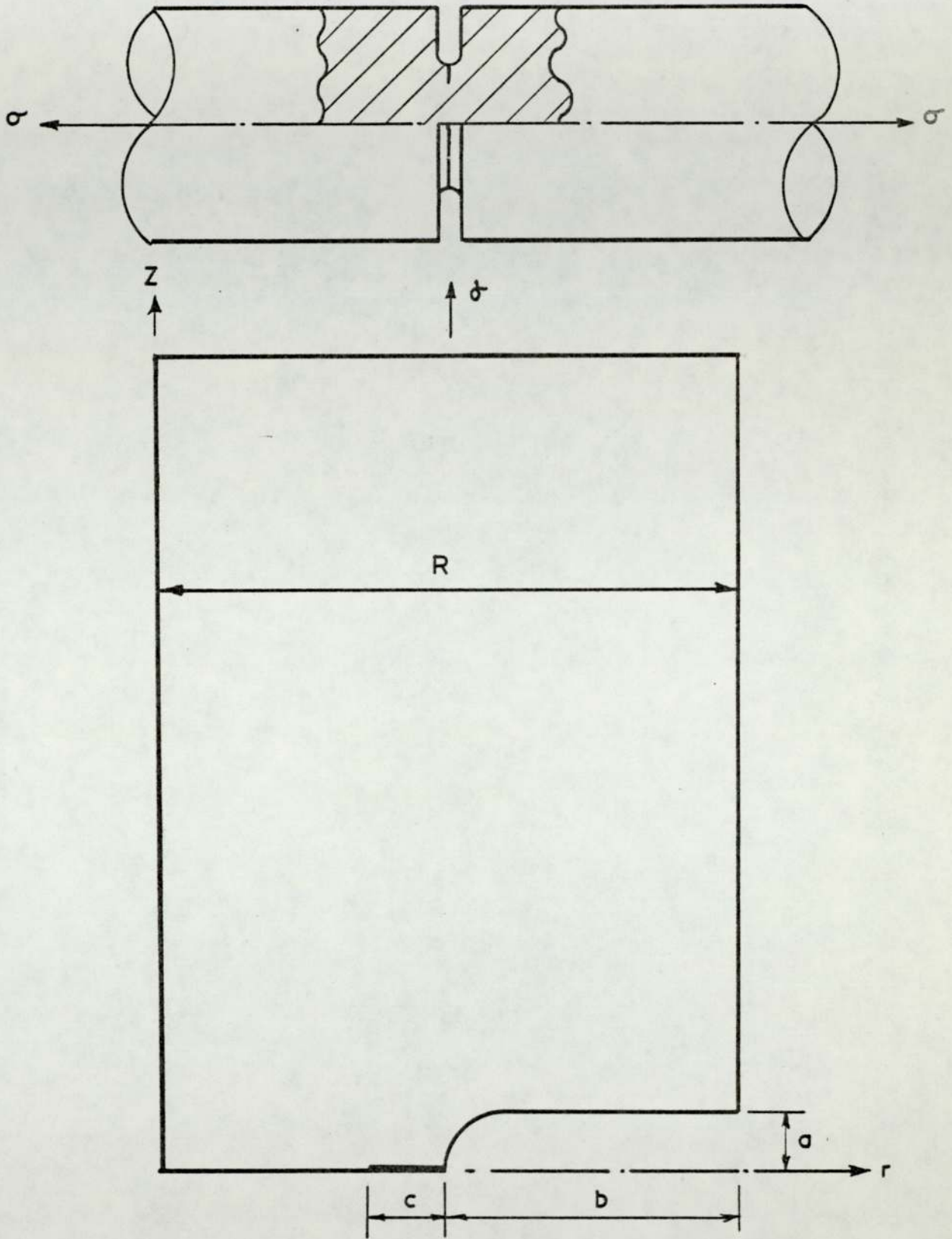


Fig. 8.11

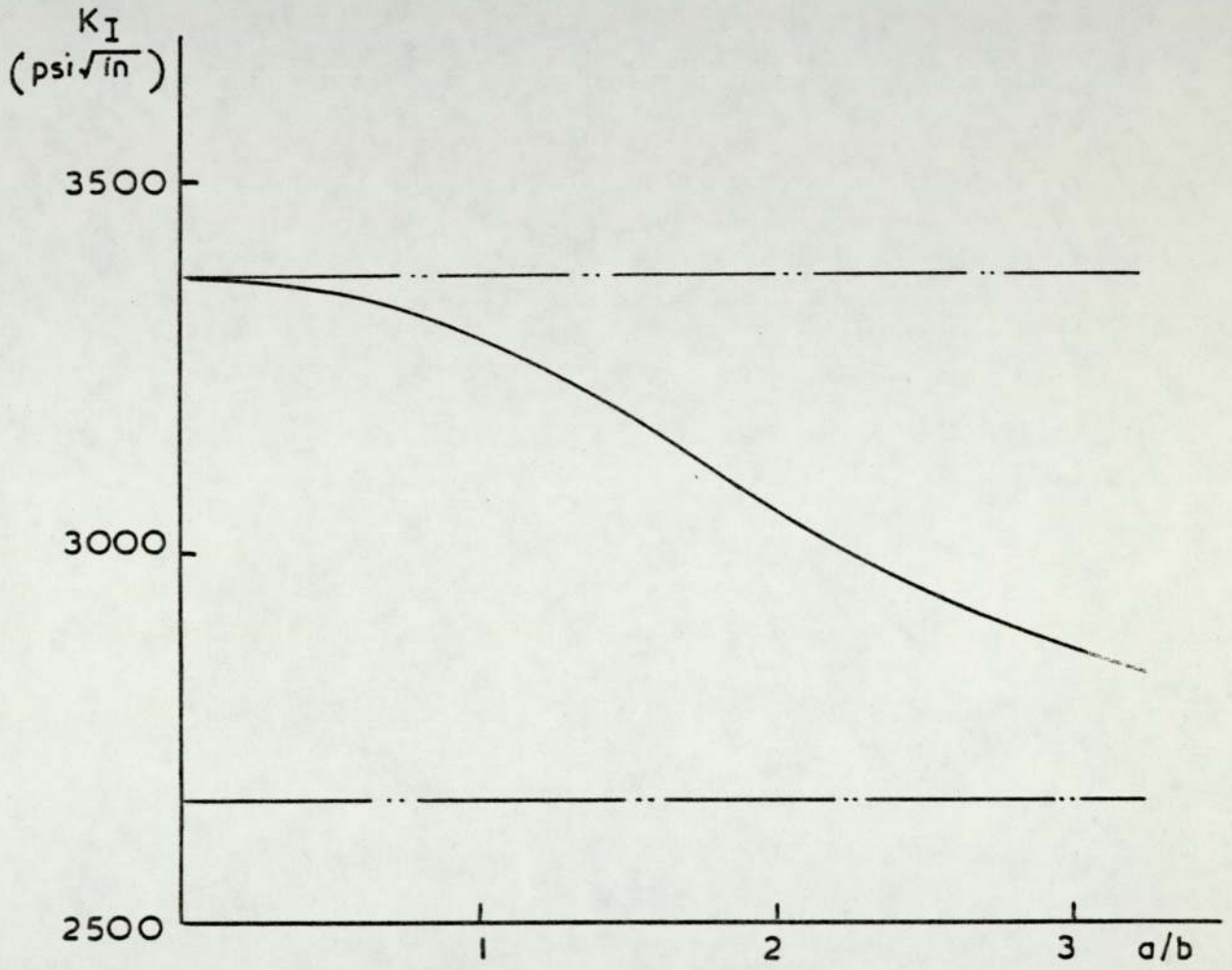


Fig. 8.12

Fig. (8.14) shows the percentage reduction in the value of (K_I) of a crack of length (c) due to the existence of a semicircular groove of radius (b) , from that of a circumferential normal crack of length $(b + c)$ in a solid bar.

The results show that when $(\frac{c}{b} > \frac{1}{3})$ this percentage reduction in (K_I) value is very small, while if $(\frac{c}{b} < \frac{1}{10})$ the reduction is large and is affected by very small changes in the (c/b) ratio.

An important general conclusion may be drawn from these results which is: if a crack is propagating from a groove whose (a/b) ratio is less than unity or it has propagated from any groove to a distance which is large compared to the groove depth, then a conservative estimate of (K_I) would be that of a crack of a depth equal to the sum of the depths of the groove and the crack in a solid bar. Such (K_I) values are available in charts and formulae and the need to resort to complicated numerical methods is eliminated.

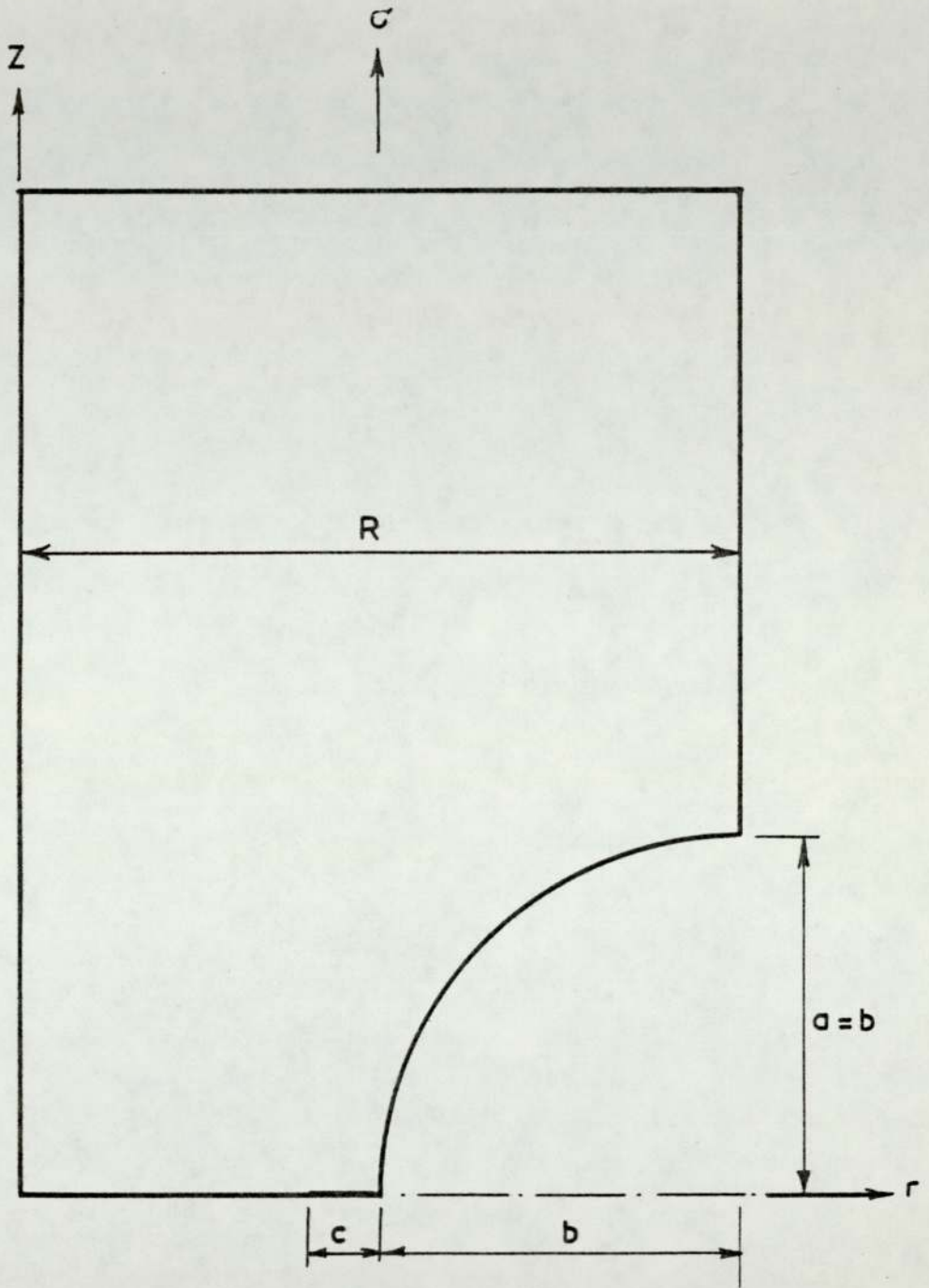


Fig . 8.13

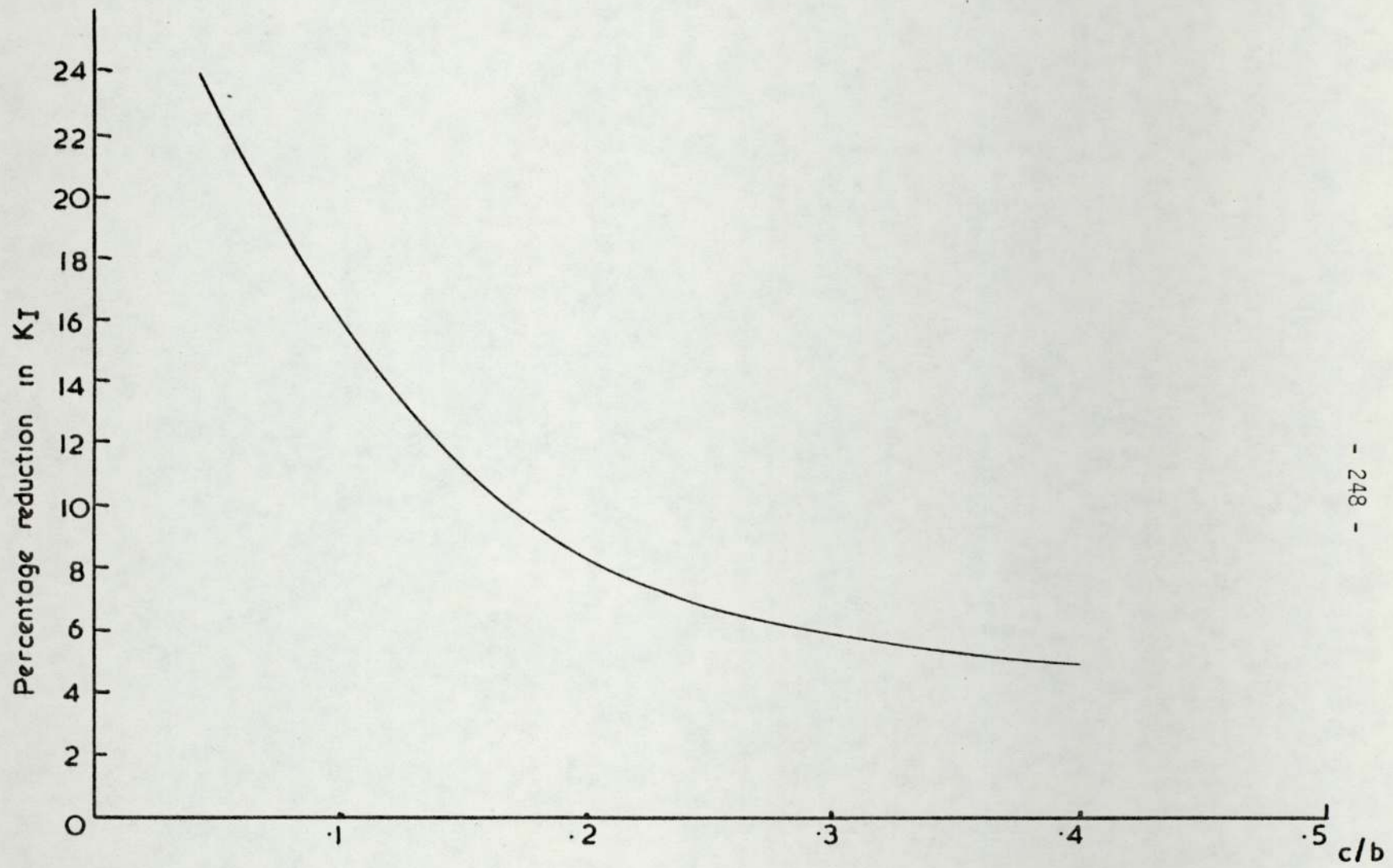


Fig. 8.14

8.3.4 A round bar with an internal void and a circumferential normal crack

A problem complementary to that in section (8.3.3) is of a bar containing a circumferential normal crack, but the geometric discontinuity is an axisymmetric void, Fig. (8.15). Here, the existence of the void reduces the net section further and thus elevating the local stress field near the crack tip. For the case of:

$\sigma = 500$ psi, where $\sigma =$ uniform stress remote
from the crack tip

$$c/R = 0.5$$

$$a/b = d/b = 0.09,$$

the (K_I) value was found to increase by (52%) due to the existence of the void.

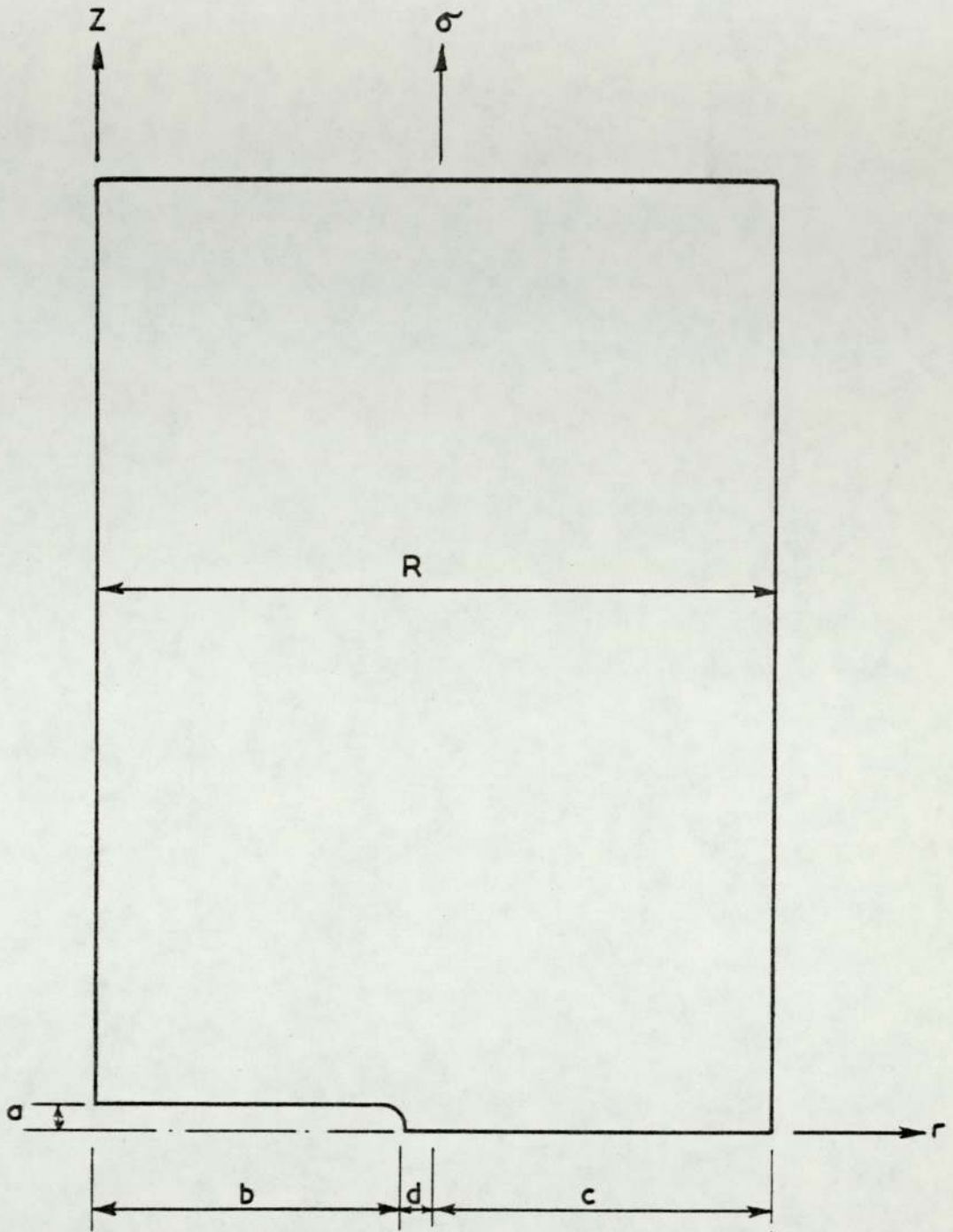


Fig 8.15 .

8.4 Mixed mode I and II fracture problems

8.4.1 A test problem

To the knowledge of the author, a mixed mode (I) and (II) axisymmetric fracture problem has not been solved previously and an example was not found in the published literature to be used as reference. In order to check the mixed mode (I) and (II) program, the mode (I) problem of section (8.3.1) was solved using it. In mixed mode problems there is no symmetry with respect to the crack plane to exploit, and hence the whole specimen is considered and the core's shape is a circle. However, as ring elements are used, half the longitudinal section need be discretized only. The new discretization of the problem required a redistribution of the elements both remote from and around the core, which implied either increasing the overall number of elements or reducing the number of nodes on the core/finite element interface. As the program was nearly occupying the full store of the available computer, the first alternative was not possible. Therefore, the nodes on the core/finite element interface were reduced to ($N_1 = 17$) and the total number of nodes became (289). It will be seen that this alteration did not affect the accuracy of the results obtained.

The problem was first solved using the discretization shown in Fig. (8.16) and the results obtained were compared with those for the same problem solved by the mode (I) program.

$$\frac{K_I(\text{mixed mode})}{K_I(\text{mode I})} = 0.9497$$

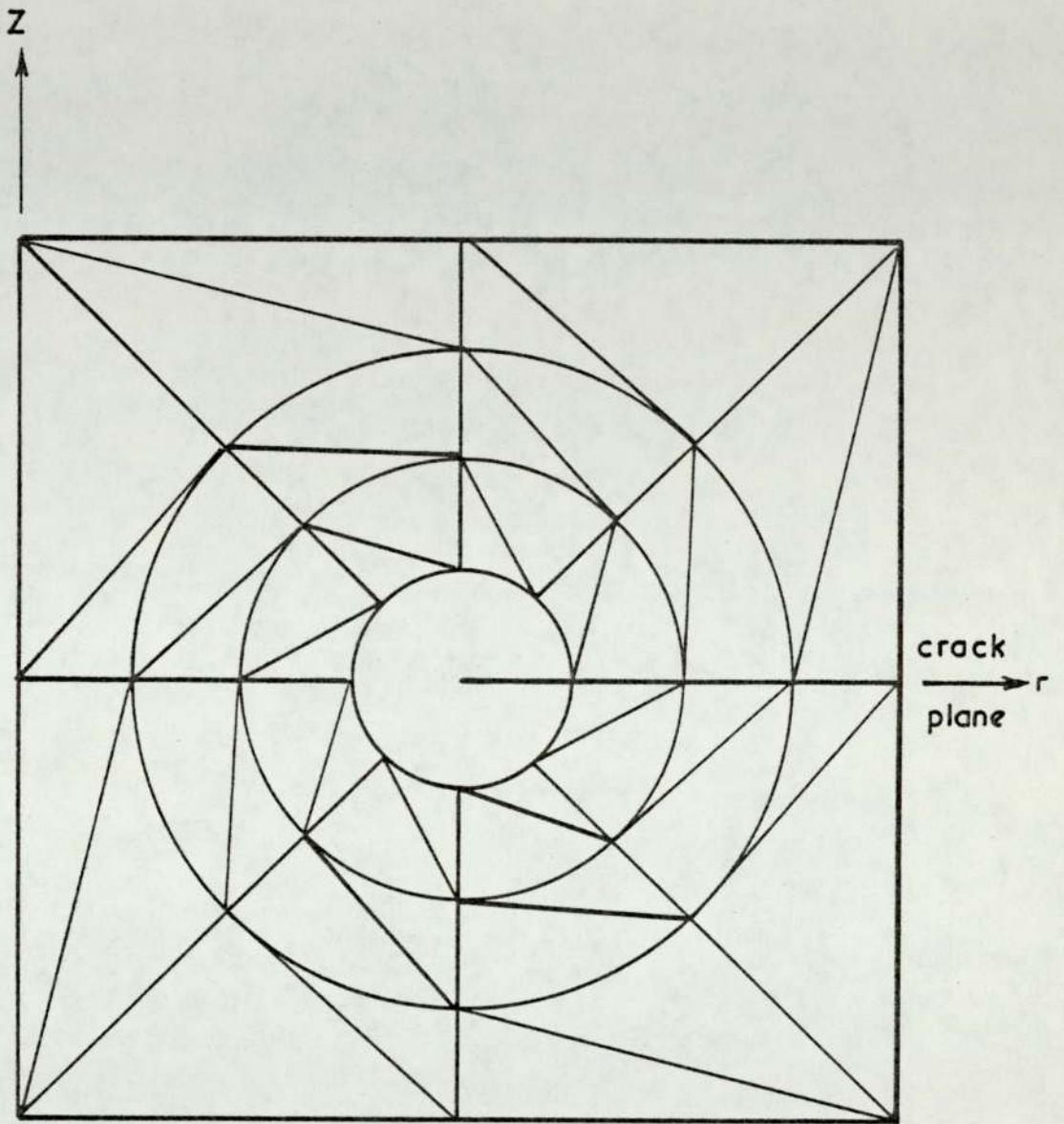


Fig. 8.16

and

$$\frac{K_{II}(\text{mixed mode})}{K_I(\text{mixed mode})} = 0.0441$$

They show that some influence in the mesh design exists leading to a small (K_{II}) value. It was noted that the mesh discretization was not symmetric with respect to the crack plane, therefore the problem was solved again with the discretization arranged symmetric in order to simulate isotropy as well as possible, Fig. (8.17), and the results obtained were:

$$\frac{K_I(\text{mixed mode})}{K_I(\text{mode I})} = 0.9956,$$

which shows very good accuracy, and the ratio of the values of (K_{II})/(K_I), which should be zero, was found to be:

$$\frac{K_{II}(\text{mixed mode})}{K_I(\text{mixed mode})} = 0.0067$$

This result was assumed to be acceptable. The small K_{II} value obtained, which is the consequence of generating some asymmetry in the discretization, is thought to be due to small errors resulting from the various numerical approximations in calculating the core displacement components and the nodal coordinates of the finite element mesh. The accuracy of the value of (π) used and the various angle functions which are calculated from series expansions are examples of these approximations. When inclined crack problems were solved, this symmetric discretization was limited to the neighbourhood of the crack tip (the first three element rings around the core).

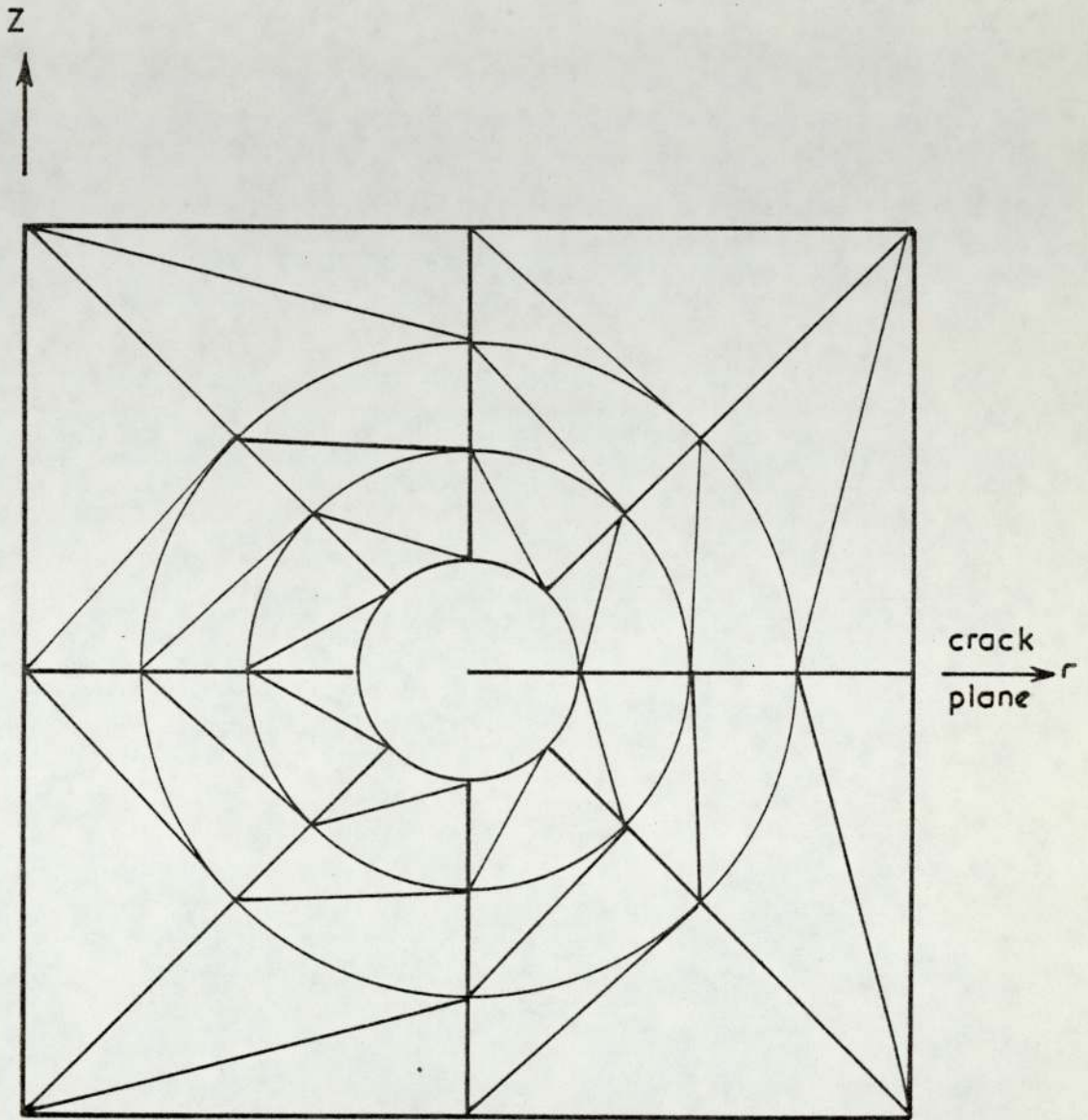


Fig . 8.17

The program was subjected to another test to check the conditioning of the equations.

The equilibrium equations are solved,

$$[K]\{q\} = \{Q\} \quad (8.5)$$

for the displacement $\{q\}$ given by:

$$\{q\} = [K]^{-1}\{Q\} \quad (8.6)$$

If the numerical values obtained for displacements from (8.6) are substituted in (8.5), they will yield a new force vector $\{Q\}^*$. The new force vector should theoretically be the same as the old one $\{Q\}$, however due to the approximations involved in the solution process some difference exists.

Substituting $\{Q\}^*$ in equation (8.5) will give a new displacement vector $\{q\}^*$. The difference between $\{q\}$ and $\{q\}^*$ may be used as a measure for the conditioning of the equations.

This iterative process was repeated three times giving three displacement vectors $\{q\}$, $\{q\}^*$, and $\{q\}^{**}$, and by comparing the three sets of stress intensity factors obtained as part of the displacement vector, it is seen from Table (8.6), that the difference between them is very small indeed indicating that the equations are well-conditioned.

	1st.iteration	2nd.iteration	3rd.iteration
K_I	2028.0818	2028.0820	2028.0819
K_{II}	13.70564	13.70566	13.70564

TABLE (8.6)

8.4.2 A round bar with a conical shaped circumferential crack

The problem is that of a round bar with a conical shaped edge crack inclined at an angle (θ) to the plane perpendicular to the axis of revolution, Fig. (8.18), with the ratios:

$$L/R = 2.4 \text{ and } c/R = 0.1$$

The cases studied were for angles:

$$\theta = 15^\circ, 30^\circ, 45^\circ, 60^\circ \text{ and } 75^\circ$$

The bar was subjected to a uniform tensile end load of (500) psi, and the values for (K_I) and (K_{II}) obtained are shown in Figs.(8.19) and (8.20). In the mixed mode situation, the direction of initial crack growth is not known beforehand, hence Sih's strain energy density criterion may be applied to find it, [25]. Fig. (8.21) shows the variation of the initial crack propagation angle (β) with the inclination of the crack.

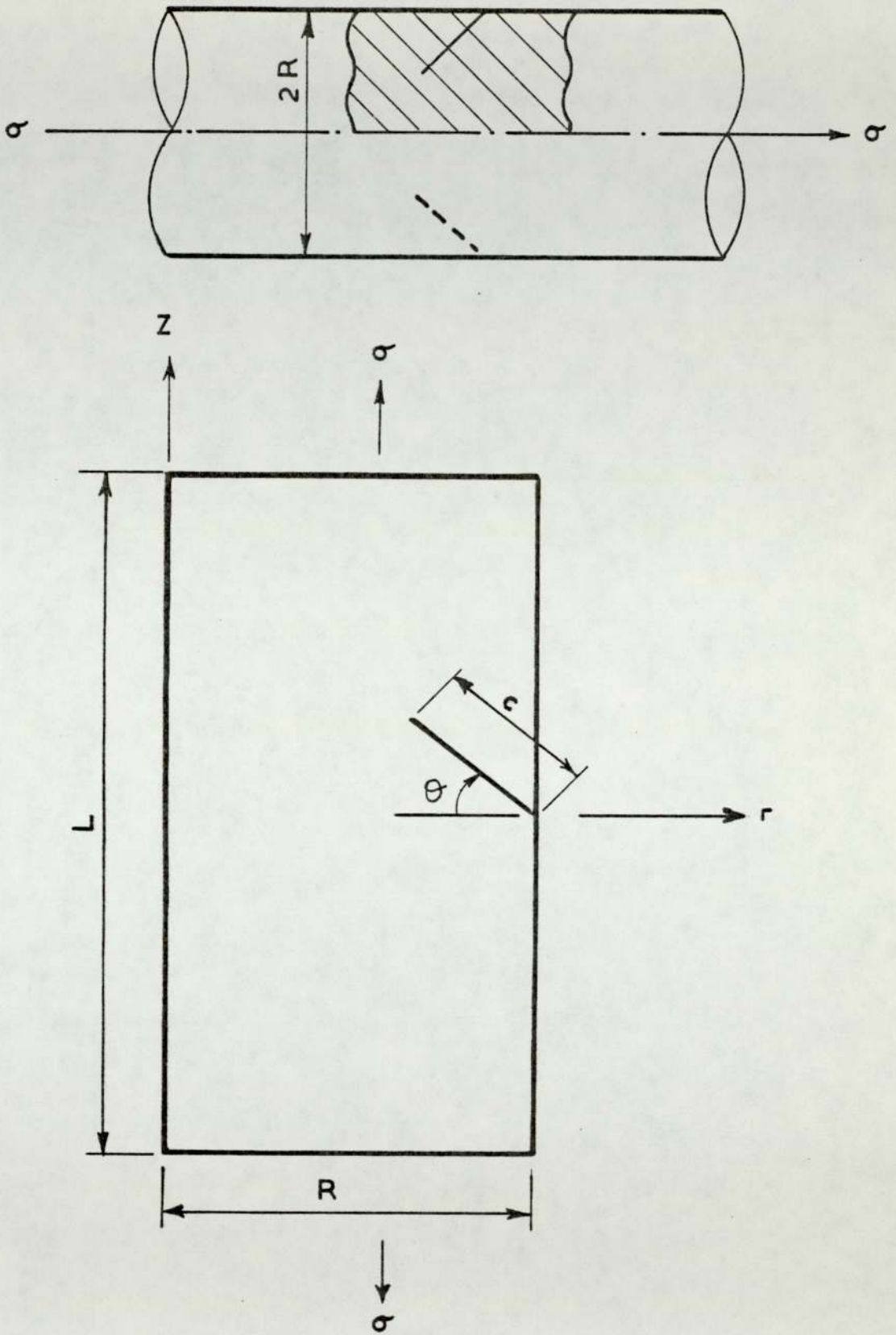


Fig. 8.18

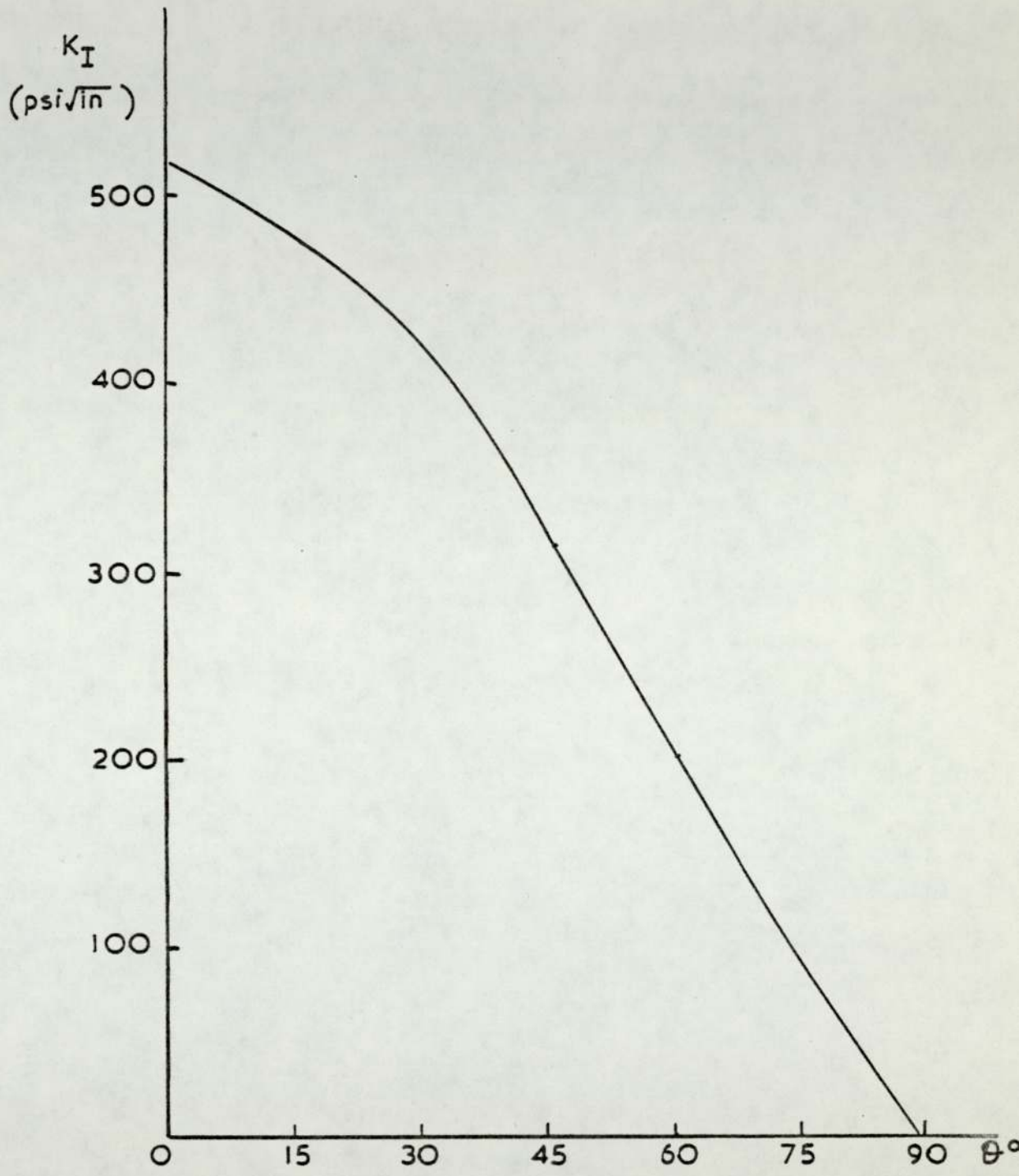


Fig. 8.19

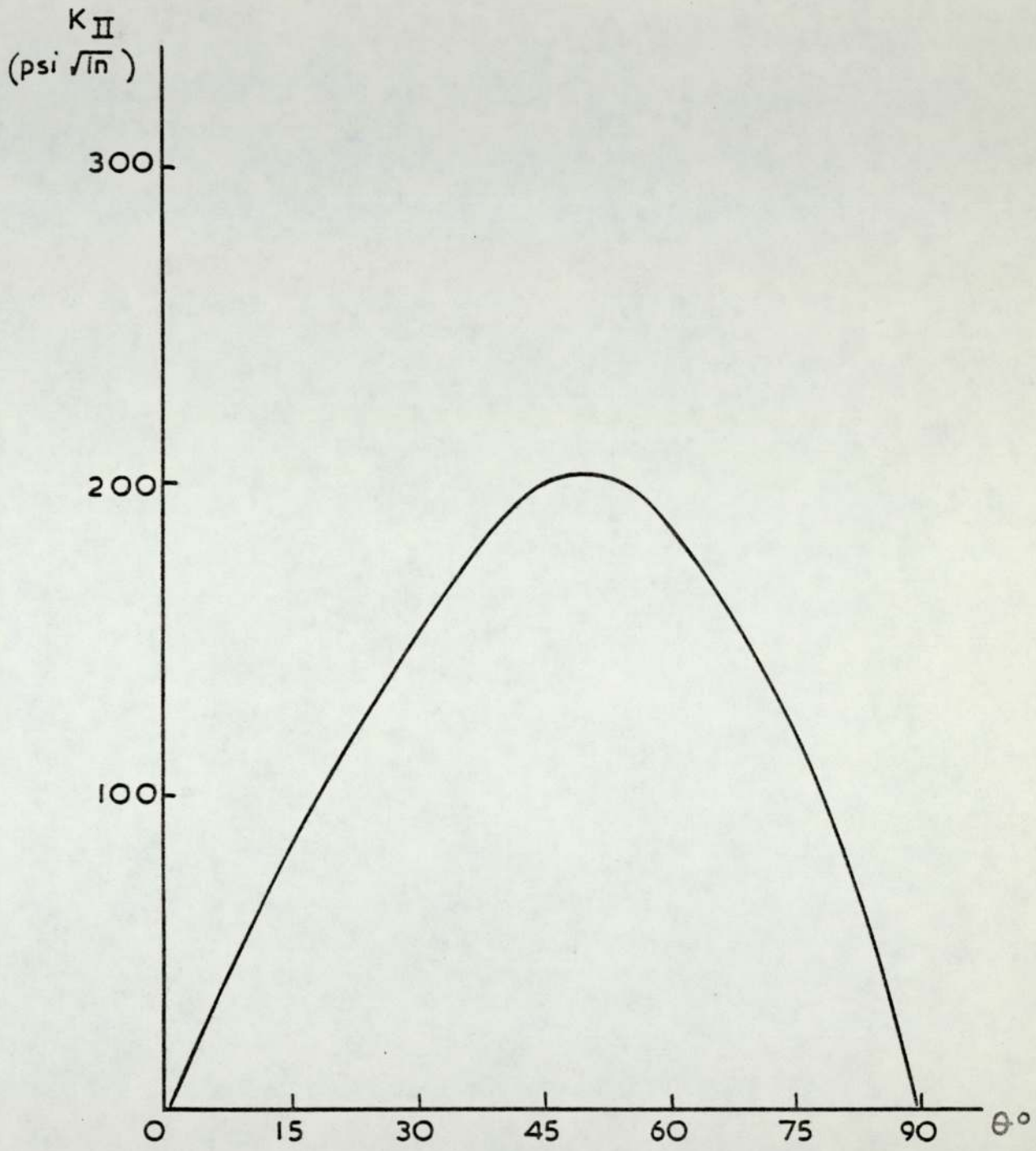


Fig. 8.20

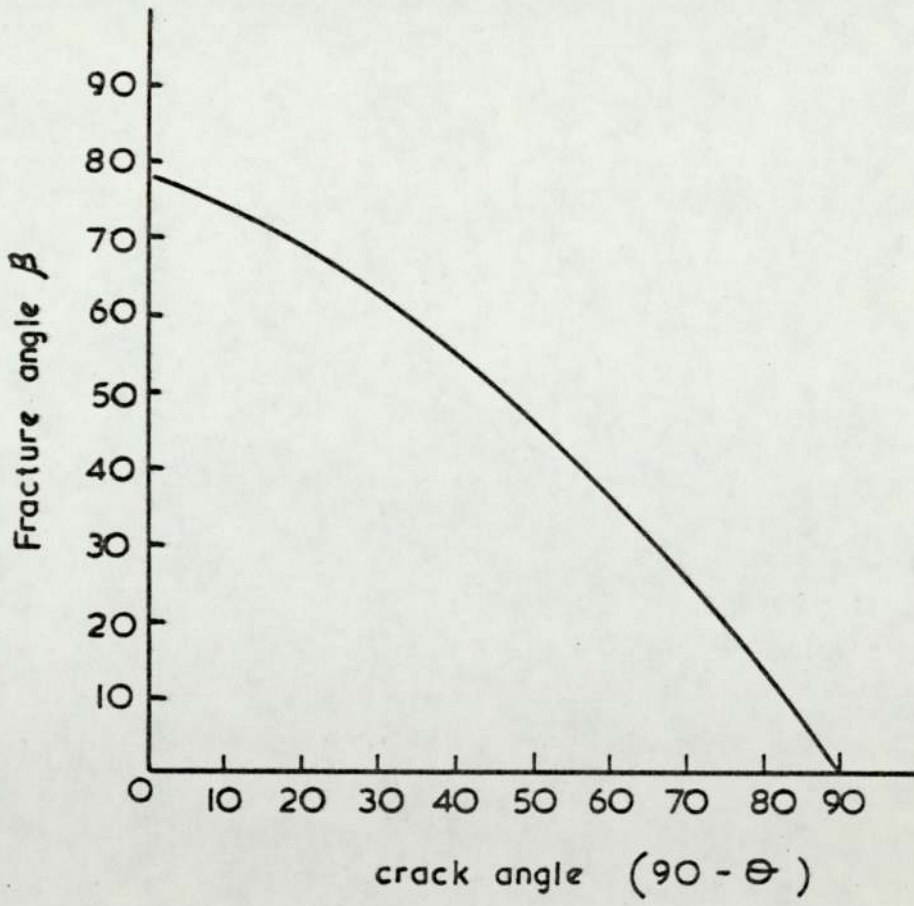
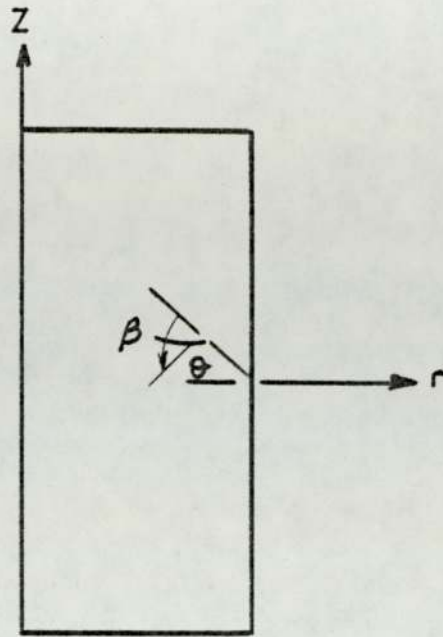


Fig. 8. 21

8.4.3 A shouldered bar with a circumferential conical crack emanating from its fillet

The influence of different shoulder fillet radii on the K_I and K_{II} values for various inclinations is studied in this problem. The range of possible geometries is infinite, and only a few representative situations were examined which are with reference to Fig. (8.22).

$$D/d = 2, c/D = 0.1, a/d = 0.125, L/D = 2.4$$

$$r/d = 0.05, 0.1, 0.15, 0.2, 0.25 \text{ and } 0.3$$

$$\theta = 15^\circ, 30^\circ, 45^\circ \text{ and } 60^\circ$$

The results are shown in Figs. (8.23), (8.24), (8.25), and (8.26) for a tensile load of (500) psi applied to the end with diameter (D).

In the cases of non-symmetric geometries or loadings the crack angle representing a pure mode (I) situation is not identified and curves such as those shown in Fig. (8.26) may be used to find it. It is seen from Fig. (8.24) that for the range of different fillet radii examined, their effect on (K_{II}) values is very little.

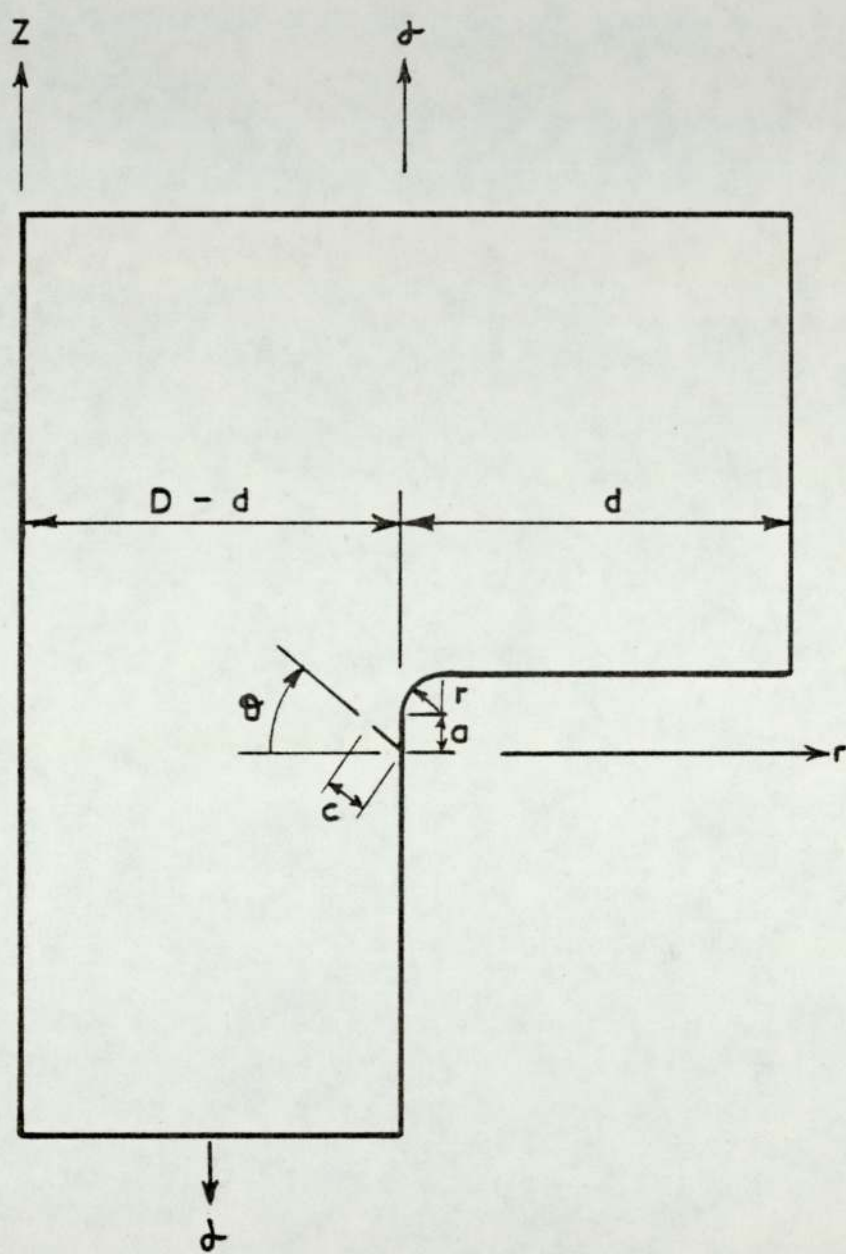


Fig. 8.22

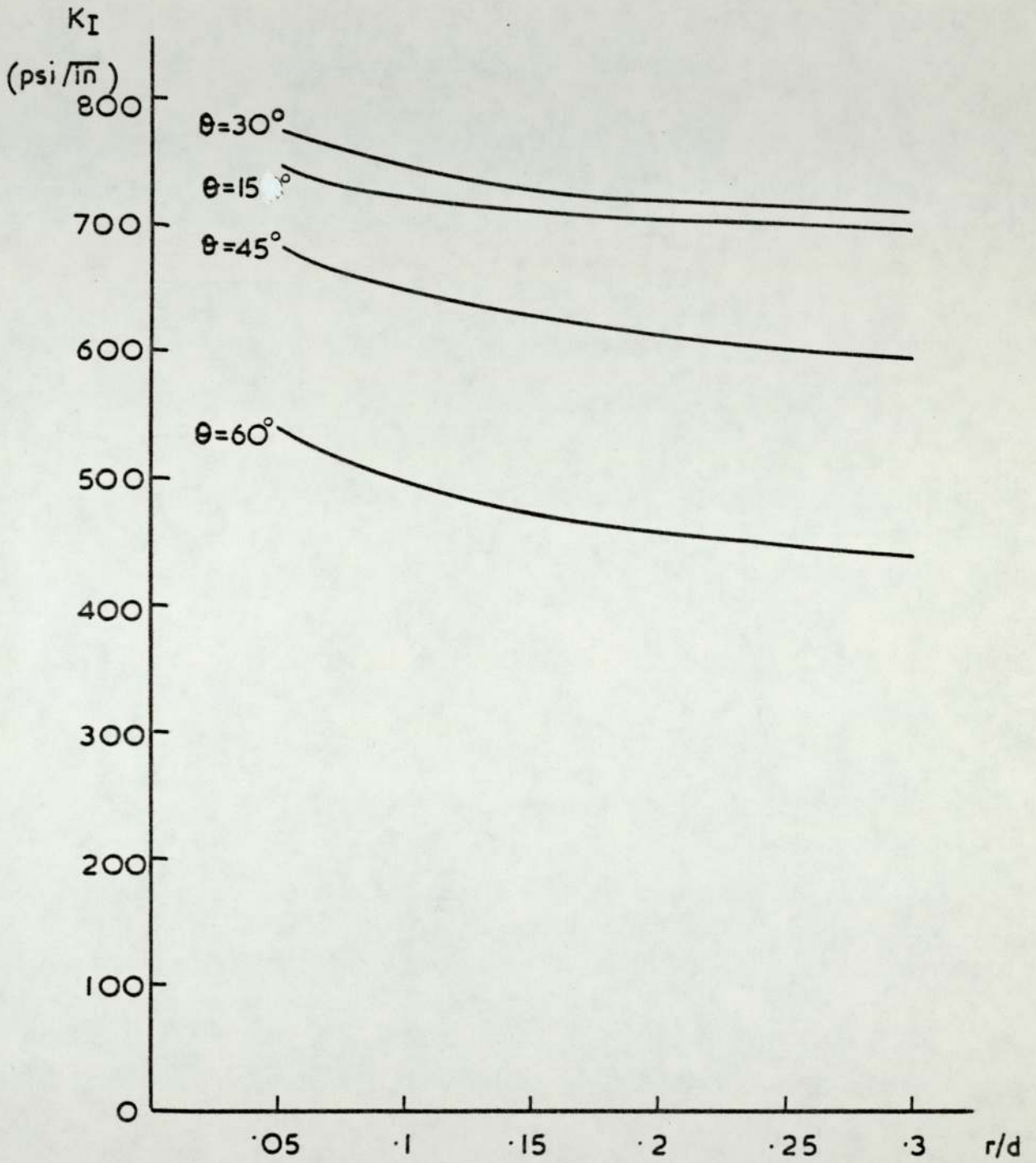


Fig . 8. 23

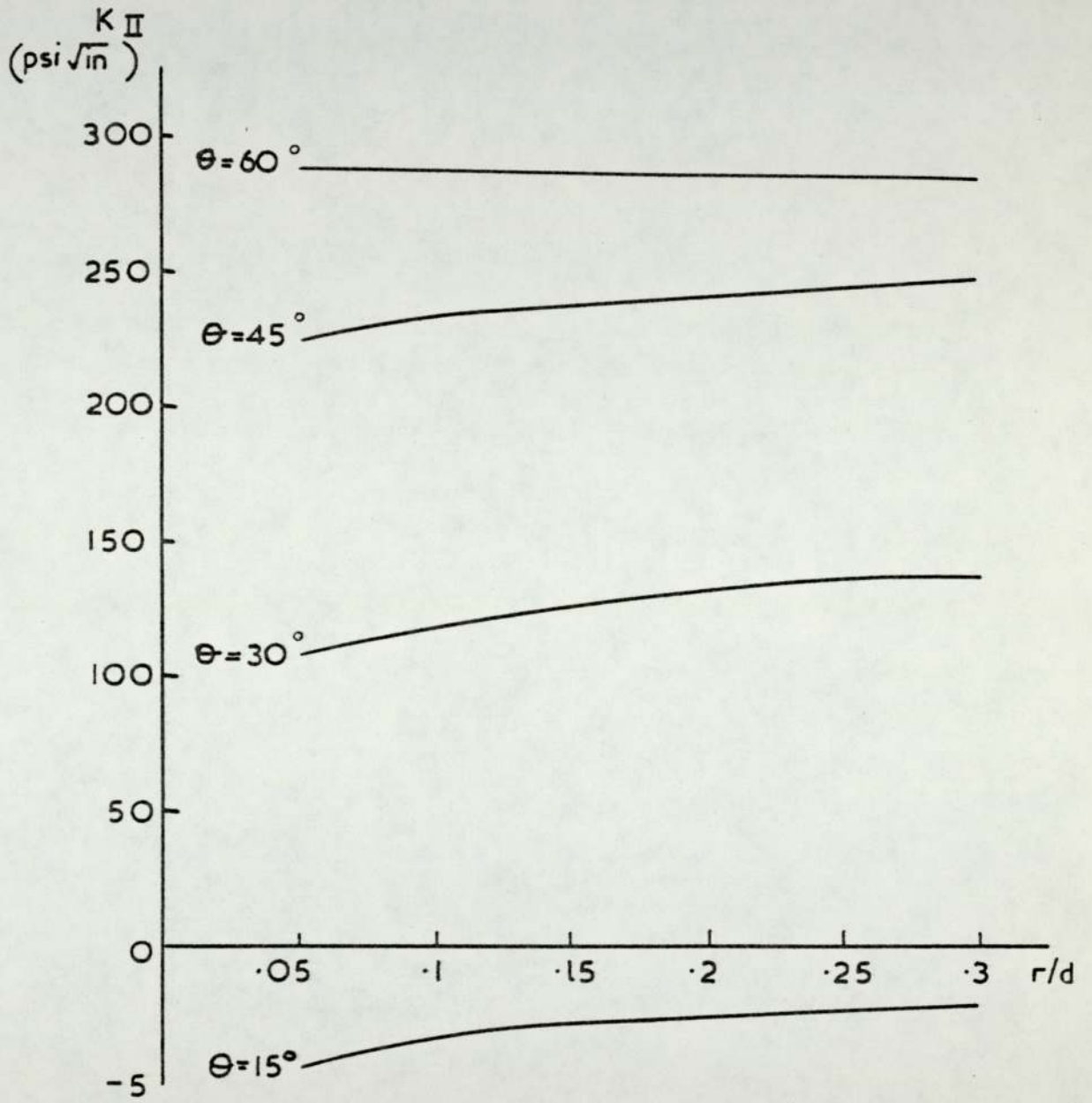


Fig 8.24

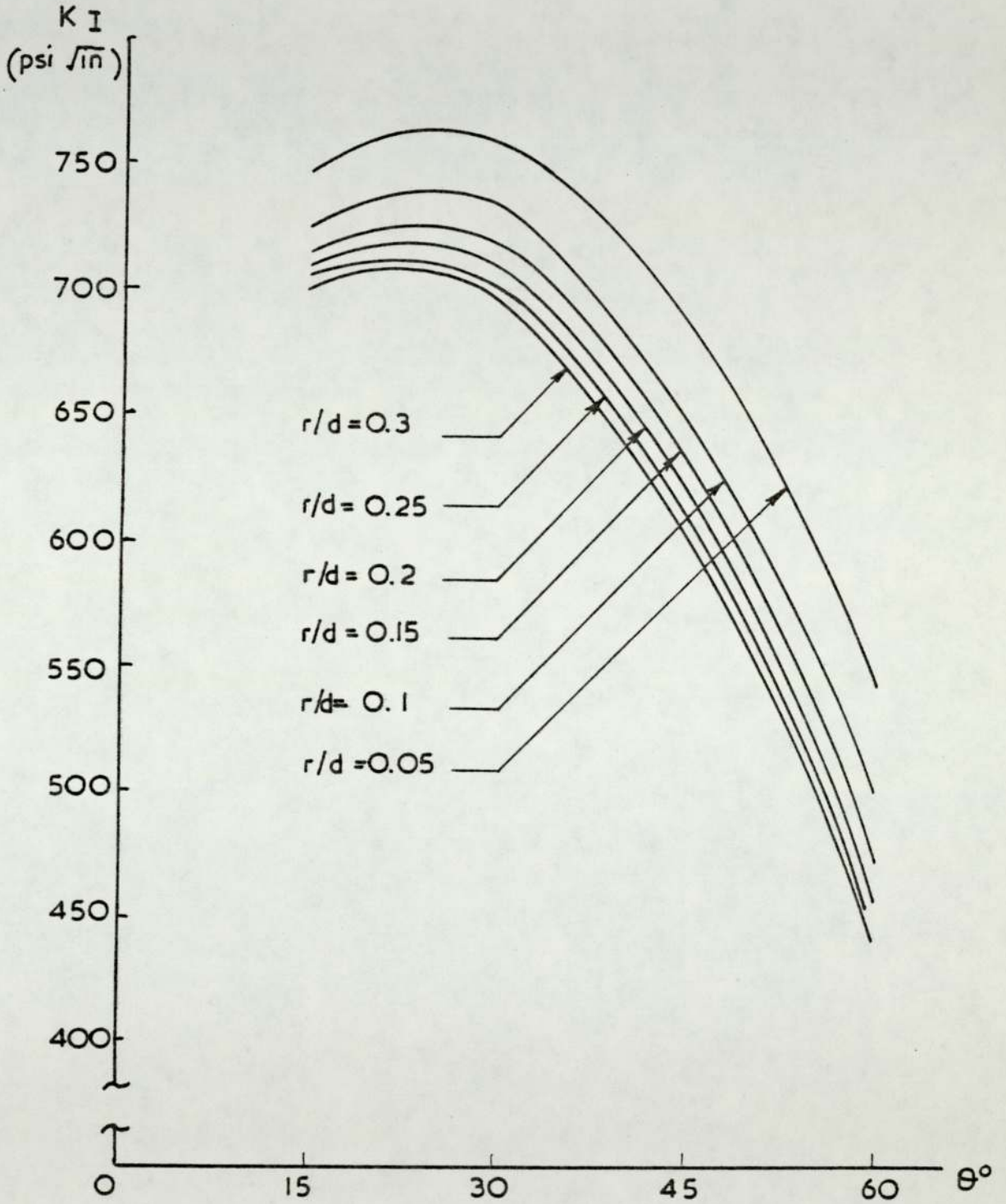


Fig. 8.25

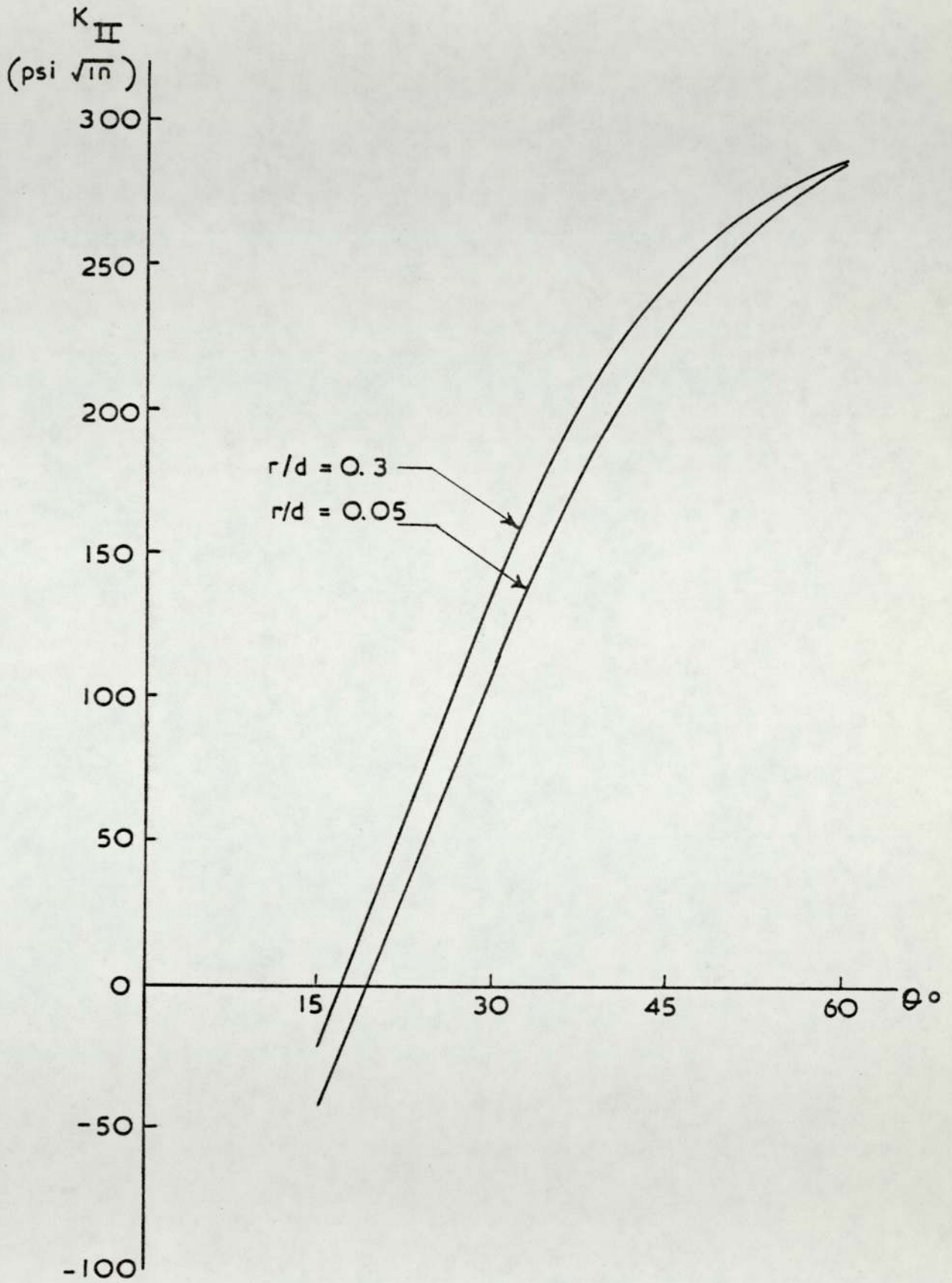


Fig . 8. 26

8.5 The effect of inclusions on the values of stress intensity factors

To illustrate the versatility of the present method for evaluating stress intensity factors, the mode (I) and mixed mode (I) and (II) fracture problems of axisymmetric composite bodies were examined. Two cases were chosen to represent mode (I) fracture problems. The first was that of a thin ring inclusion with its centre at the axis of revolution, Fig. (8.27), and the second was of a cylindrical inclusion around the axis of revolution, Fig. (8.28). For the mixed mode (I) and (II) fracture problems, the three cases of thin edge normal ring inclusions above, below, and both above and below a conical edge crack inclined with an angle of (30°) to the plane perpendicular to the axis of revolution were examined, Fig. (8.29).

The ratio (E_0/E) was taken as:

$$E_0/E = 10^3, 10, 10^{-1}, \text{ and } 10^{-6}$$

where

E_0 = Modulus of elasticity of inclusion

E = Modulus of elasticity of the remainder of the body.

Table (8.7) shows the ratio of the stress intensity factors obtained with the presence of the inclusions (K') to those obtained for the same problem with one material only (K).

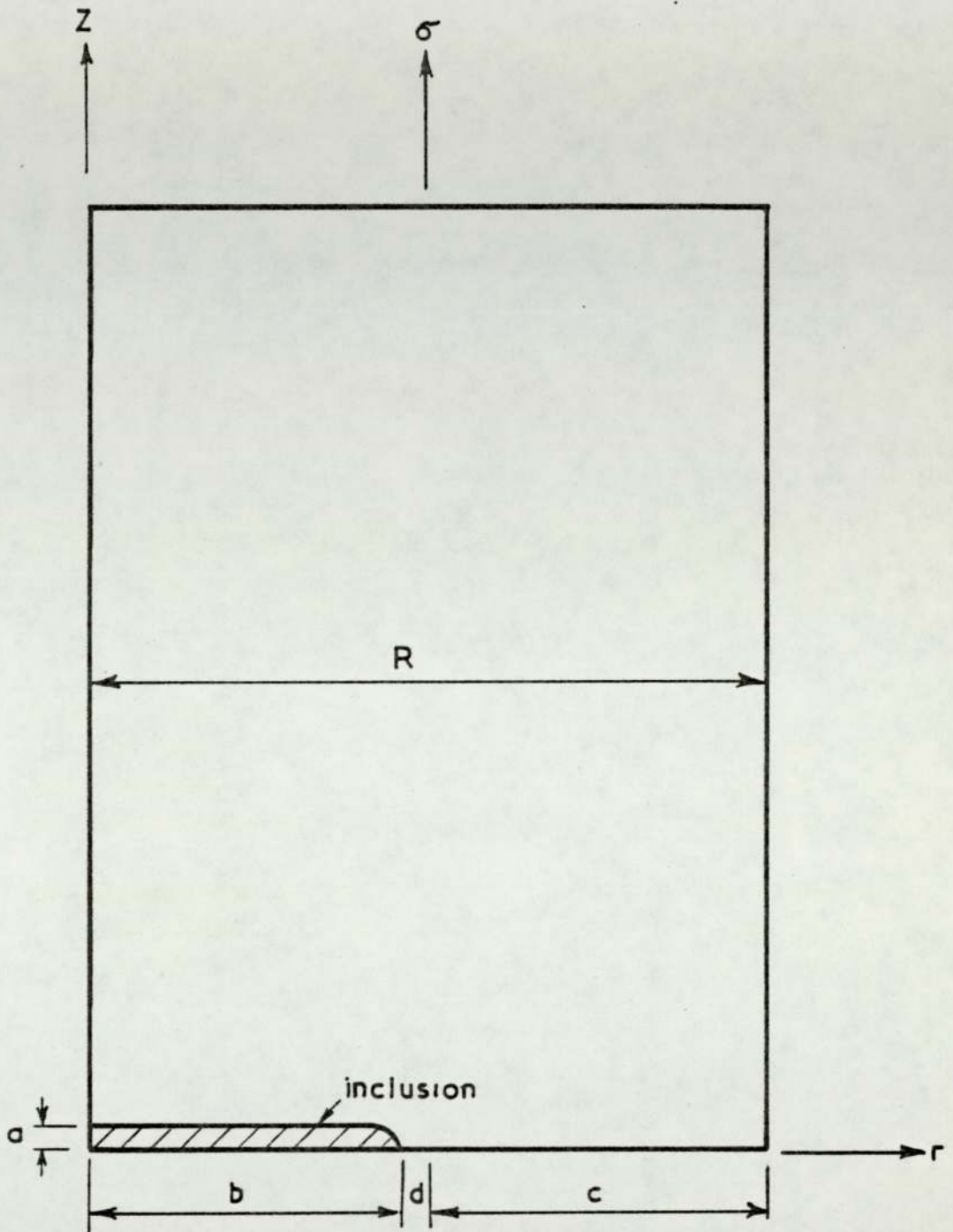
It can be concluded that the effect of the inclusions on the values of (K_I) and (K_{II}) can be characterized by the influence of the inclusions on the stress field in the neighbourhood of the crack. Hence, the

inclusion's geometry, its location with respect to the body, its distance from the crack tip, and its material properties are the factors which will ultimately effect the stress intensity values obtained.

E_0/E	Mode I ring inclusion	Mode II cylind. inclusion	Mixed Mode ring below crack		Mixed Mode ring above crack		Mixed Mode ring above & below crack	
	K'_I/K_I	K'_J/K_I	K'_I/K_I	K'_{II}/K_{II}	K'_I/K_I	K'_{II}/K_{II}	K'_I/K_I	K'_{II}/K_{II}
10^3	0.92	0.36	1.08	1.1	1.14	1.22	1.16	1.23
10	0.93	0.59	1.07	0.91	1.04	1.08	1.15	1.18
10^{-1}	1.03	1.09	0.56	0.77	0.79	0.41	.63	.46
10^{-6}	1.11	1.12	0.17	.03	0.10	0.36	$(-.04)^\dagger$	$(-.06)^\dagger$

† The negative sign implies a crack closure situation for which no special provision is made.

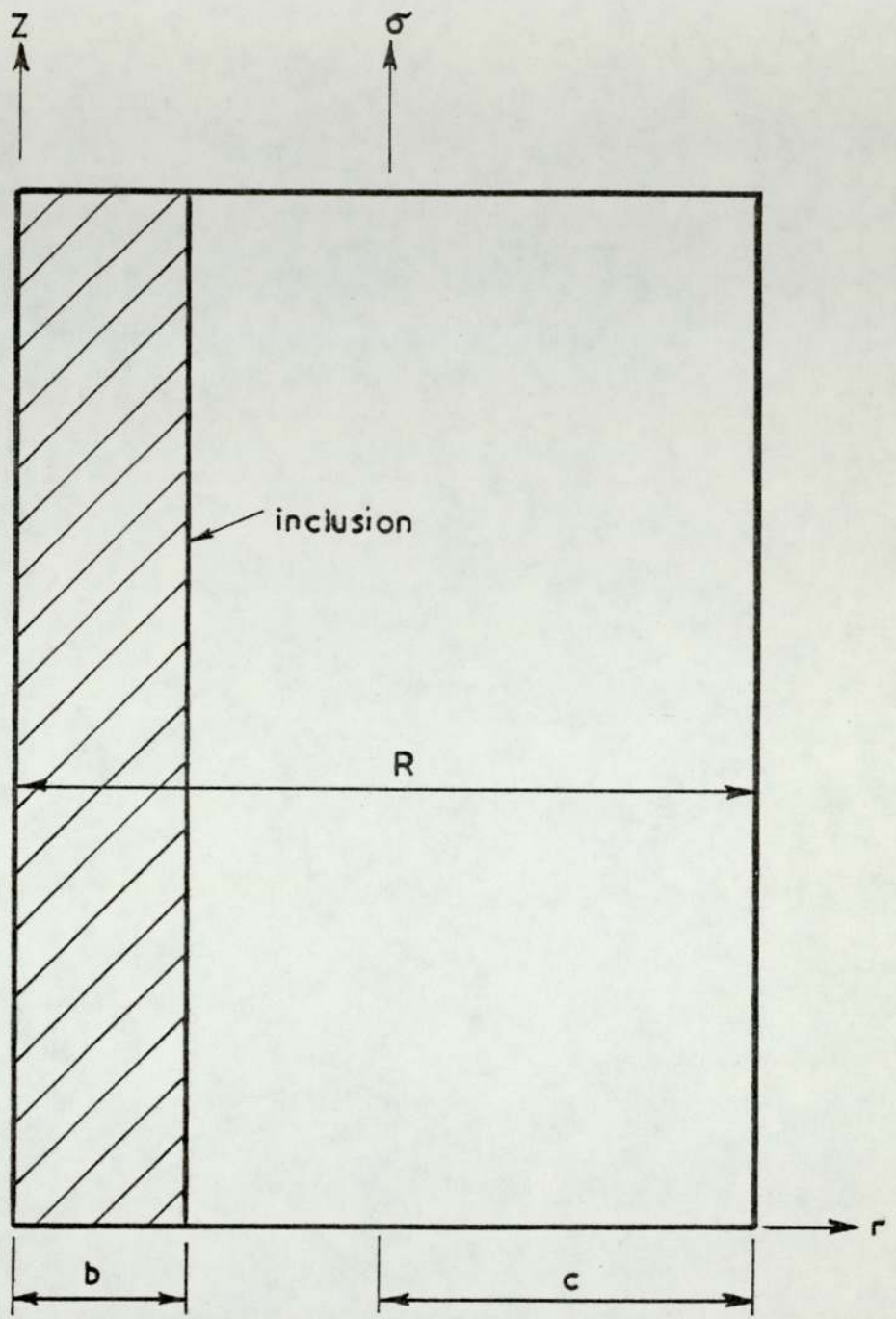
TABLE (8.7)



$$c/R = 0.5$$

$$a/c = d/c = 0.08$$

Fig. 8.27



$$c/R = 0.5$$

$$b/R = 0.23$$

Fig. 8. 28

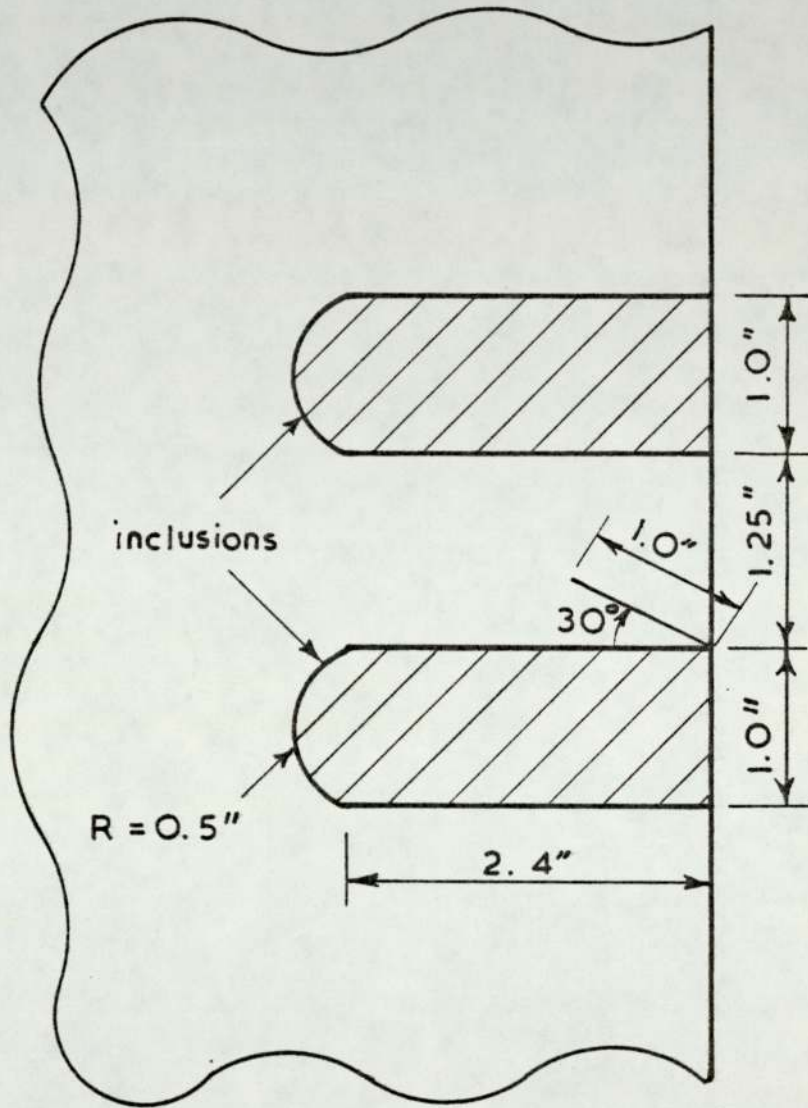


Fig 8.29

CHAPTER 9

DISCUSSIONS AND CONCLUSIONS

9.1 Discussions

A finite element computer program has been developed to solve the problem of axisymmetric solids subjected to axisymmetric loading and certain problems were solved to test it and assess its accuracy. This program was later modified to incorporate a special "core" element to provide a facility for determining data relevant to problems in linear fracture mechanics.

All the programs were divided into sub structures (procedures) each performing part of the solution. In addition to the simplification of the program structure, this method of programming enables corrections and alterations to be used to any procedure without affecting the main flow of the program.

The program's facilities are:

- a) Several loading sets for a given structure may be solved in one run.
- b) Several constraints sets per each loading set may be solved in one run.
- c) Each element can have different material properties which can either be isotropic or stratified orthotropic.
- d) Input facilities include:

i) for the general axisymmetric problem, the structures can have an arbitrary boundary shape (provided it is axisymmetric), it may or may not have symmetry with a diametral plane, and it can either be hollow or solid.

ii) for the mode (I) fracture case, the mesh is generated automatically by specifying the radius and half the length of the specimen, the crack length, the core radius, and the number of nodes on the core/finite element interface. Any grooves or voids are introduced by reading the coordinates of the nodes describing them together with those which have prescribed loads or displacements and hence overriding their previous values calculated by the mesh generation scheme.

iii) for the mixed mode (I) and (II) problem the angle of crack inclination with the r-axis measured from the direction of the crack tip is specified together with the parameters in (ii), and subsequently the mesh is generated automatically with symmetric discretization with respect to the crack plane in the neighbourhood of the crack tip.

e) Output facilities include:

i) for the general axisymmetric program, nodal displacements together with a choice of nodal and/or element centroidal values of stresses and strains

ii) for the mode (I) program, nodal displacements of unconstrained nodes with the value of K_I and the tip displacement in the r-direction.

iii) for the mixed mode (I) and (II) programs, nodal displacements of unconstrained nodes together with values of K_I , K_{II} , tip

displacement in r & z direction, rigid body rotation of the core element, angle of crack initiation, and the new force vector $\{Q\}^*$ resulting from resubstituting the obtained displacements into the equilibrium equations.

The choice of the isoparametric six node triangular ring element was shown to be a good one. The isoparametric concept enabled the mapping of curved boundaries which is particularly useful when matching the core circular shape. The economy achieved by using a linear strain element can best be illustrated by the example of a mode (I) situation solved by the mixed mode program. Seventeen nodes scurting a circular core gave results within (4%) of those obtained by Sih using twenty one nodes scurting a semicircular core of the same radius with a constant strain element, [49].

The solution was simplified by the use of numerical integration in evaluating the elements stiffness matrices instead of explicit multiplication and term by term integration which requires the solution of twenty one separate integrals. Although the order of integration chosen was quadratic, the results compare very well with those obtained by the term-by-term integration.

In solving the axisymmetric problem by finite element, some method of approximating the radii used in calculating the hoop strain (u/r) for elements on the axis of revolution must be used as a zero r value will cause computational problems. One possible method is to average the radii of nodal points of all elements on the axis of revolution,

which can give a high average value if the elements are large. Another method is to insert a hypothetical axial core by giving the nodes on the axis of revolution a small r-coordinate value. Both these methods affect all the nodes on the axis of revolution, however the method chosen distinguishes between elements with a side coinciding with the axis, which requires approximation, and others with only one node on it which do not require special consideration and thus limits the approximation to where it is needed only.

A considerable saving in computer storage is achieved by assembling the overall stiffness matrix as a one dimensional array and making use of symmetry by storing only the coefficients between the first non-zero one and the leading diagonal of each row, compared with a normal banded storage scheme. In the method used, the assembly is not penalised by bad nodal connections of some elements resulting in increasing the band width of the whole overall stiffness matrix.

The automatic mesh generation scheme developed relieved the user from the formidable task of preparing the large volume of input data required. It provided error free data for several crack configurations without the need for checking the mesh for every run.

It was seen that the solution of idealized situations in fracture mechanics problems may be obtained by powerful analytic techniques, but the complex geometries and loadings of real engineering problems forced the analyst to resort to numerical methods. The finite element method is an approximate numerical method which can easily model these shapes

and is widely used in structural analysis. However it was found that its direct application to fracture problems required a very fine mesh in the neighbourhood of the crack tip and its convergence to the right results is not guaranteed. A modified finite element formulation was suggested where the crack tip is surrounded by elements with the singularity condition built into their displacements functions. From the number of approaches possible, the idea of the singular core embedded in a normal finite element mesh was developed. It was chosen because it requires a little modification to the overall stiffness matrix and load vector only and enables the direct evaluation of stress intensity factors.

The computer programs developed using this method were tested by solving simple configurations to which analytical or other numerical solutions are available. Although the work was carried out on a modest size computer and the most complicated shape considered was represented by (578) degrees of freedom only, the results compare very well with the published alternatives. The range of axisymmetric fracture problems is very wide and hence only a number of problems of special interest were chosen to demonstrate the power of the technique. Among them are the problems of cracks extending from grooves, mixed mode (I) and (II) problems where the angle of crack initiation was calculated by Sih's strain energy density criterion, and shouldered bar problems with varying fillet radii and crack angles. The effects of inclusions of different material properties on the values of stress intensity factors was also examined for several illustrative examples.

The technique of Hilton and Hutchinson has been successfully further developed to solve a wide variety of axisymmetric crack configurations and results for problems previously untackled were obtained. However there still remains scope for further development of the technique to deal with areas other than those considered in this work which will be discussed in the following section.

9.2 Possible topics for further research

The topics which require further research and will be suggested may be divided into two groups. The first group consists of developments to the computer programs to increase their efficiency and improve their accuracy, while the second suggests different areas of problems which can be tackled by the technique. Following are the topics of each group.

9.2.1 Further developments to the computer programs

a) Improving the mesh generation scheme:

The scheme as it exists considers the structure as a whole and is purposely built for the various fracture problems. More flexibility in representing very complicated geometries, without risking over distortion of the elements due to bending element columns, may be achieved by dividing the structure into sub sections and making use of the isoparametric concept. The structure is represented by a rectangular shape divided into rectangular subsections whose number depends on the fineness of mesh required, in the auxiliary ζ - η plane. These rectangular shapes are later mapped to curved shape required in the r - z plane by using shape functions whose order can be chosen to give the best matching possible.

b) Improving the overall stiffness matrix storage scheme:

The variable bandwidth storage scheme can be improved further by using a more complex addressing system storing only the non zero coefficients between the first one and the leading diagonal of each row, together with a corresponding equation solving routine. However, the advantages gained from this reduction must be weighed against the disadvantages of complicating the storage procedure which will be reflected also on the re-assembling to incorporate the core element.

c) Developments to the core element:

There are two possible developments to the core element which can be incorporated:

i) To develop an elastic/plastic core element which takes into consideration the small plastic zone at the crack tip by including the plastic flow in the displacements field near the crack tip, [21].

ii) To develop a core element capable of solving problems of material anisotropy. Stratified orthotropic materials to which crack tip displacement fields are already available, [21], and cracks along the interface between two dissimilar materials are examples.

9.2.2 New areas for future research

a) To study the effects of core parameters and the back-up mesh:

The core's size, shape, number of nodes around it, and the back-up mesh of the remaining structure were determined by comparing the results obtained from each combination with known results, or by guidance from

the experience of other authors. In solving problems which were not tackled previously much judgement and testing is required to determine these parameters. It is desirable that any user and not just the expert is able to operate the computer program, therefore a criterion for the choice of these parameters needs to be developed.

The core in its present form is circular and centered around the crack tip. This was chosen to ease the integration required to evaluate (U_c). However, if numerical integration is used, this limitation can be eliminated and the core can have any shape with the crack position in it arbitrary. Hence the crack can extend within the core and its propagation path can be studied without having to change the mesh everytime. To offset the reduction in accuracy of the near tip displacement functions when the core is enlarged, more terms of the series expansions can be retained.

b) To include mode III fracture:

Wilson, [74], developed a computer program using the same principles as those employed in this research to evaluate K_{III} for axisymmetric solids. As K_{III} is uncoupled to K_I or K_{II} , this program can be added as a package to provide the calculation of all three stress intensity factors by one program.

c) To extend the method to solve axisymmetric problems subjected to non-axisymmetric loading:

Separation of variables may be used to represent a three dimensional problem by a two dimensional discretization by expressing the solution in one direction as a Fourier series. The orthogonality of the

trigonometric functions makes the Fourier series particularly valuable because it reduces the coupling of the simultaneous algebraic equations to be solved. For asymmetric loading the displacement vector is given by:

$$\{u_n\} = \begin{Bmatrix} \{u_n\} \\ \{v_n\} \\ \{w_n\} \end{Bmatrix} = \begin{bmatrix} \{N_1\}^t & \{0\}^t & \{0\}^t \\ \{0\}^t & \{N_1\}^t & \{0\}^t \\ \{0\}^t & \{0\}^t & \{N_1\}^t \end{bmatrix} \begin{Bmatrix} \{q_{un}\} \\ \{q_{vn}\} \\ \{q_{wn}\} \end{Bmatrix}$$

$$= [N]\{q_n\} \quad (9.1)$$

where:

$\{N_1\}^t$ = vector of shape functions given
by equation (3.21)

$\{q_{un}\}^t \dots \text{etc.} = [u_{1n} \ u_{2n} \dots u_{6n}] =$
vectors of nodal points displacements.

The strain vector is:

$$\{\epsilon_n\}^t = [\epsilon_{rn} \ \epsilon_{\theta n} \ \epsilon_{zn} \ \gamma_{r\theta n} \ \gamma_{\theta zn} \ \gamma_{zrn}] \quad (9.2)$$

and is related to displacements by

$$\{\epsilon_n\} = [B_n]\{q_n\} \quad (9.3)$$

Similar to the case of axisymmetric loading and by the application of the principle of minimum potential energy, the equilibrium equations are:

$$[K_n]\{q_n\} = \{Q_n\} \quad (9.4)$$

In equations (9.1) to (9.4) $n = 0, 1, 2, \dots, N_f$, where N_f is the number of terms in the Fourier series expansion necessary to represent the loading and is the number of independent sets of algebraic equilibrium equations resulting from the uncoupling obtained by making use of orthogonality.

Unlike the plane two dimensional problem, where there is an independent core for each of the two tips of a crack, the axisymmetric fracture problem core element is a circular ring and hence the same one for both crack opening and closure in a non-axisymmetric loading situation. Therefore the near tip displacements functions must be expressed as Fourier series as well. There are other difficulties associated with crack closure. One of them is the possible overlapping of crack surface resulting from assuming the crack nodes free to displace as dictated by the displacements of the discretization system given by the finite element method. Another difficulty is that concerning the frictional force due to closure of mode II fracture for which no theoretical solution is available.

d) Including thermal stresses:

These stresses can be included by assuming that initial strains (ϵ_0) exist in each element of the solid due to thermal expansions. They are called thermal strains and are variable within the element. However, for convenience they are assumed to be constant:

$$\{\epsilon_0\} = \begin{Bmatrix} \epsilon_{r0} \\ \epsilon_{\theta 0} \\ \epsilon_{z0} \\ \gamma_{rzo} \end{Bmatrix} \quad (9.5)$$

For an isotropic material:

$$\{\epsilon_0\} = \begin{Bmatrix} \alpha\theta^e \\ \alpha\theta^e \\ \alpha\theta^e \\ 0 \end{Bmatrix} \quad (9.6)$$

where θ^e is the average temperature rise in the element and α is the coefficient of thermal expansion. θ^e may be obtained by assuming a temperature distribution through the structure or by finite elements.

The stresses in the element due to external loading giving strains $\{\epsilon\}$ and thermal expansion giving strains $\{\epsilon_0\}$ is given by:

$$\{\sigma\} = [C](\{\epsilon\} - \{\epsilon_0\}) \quad (9.7)$$

where $[C]$ is the elasticity matrix.

The element stiffness matrix is the same as that with no initial strains. The force vector $\{Q\}$ is constructed by the superposition of external nodal forces $\{Q\}_\epsilon$ and nodal forces due to thermal strains $\{Q\}_{\epsilon_0}$. The element nodal forces due to thermal strains $\{Q\}_{\epsilon_0}^e$ are given by:

$$\{Q\}_{\epsilon_0}^e = -2\pi \int [B]^t [C] \{\epsilon_0\} r dr dz \quad (9.8)$$

This equation is similar to that for evaluating the element stiffness matrix and can be evaluated by a similar numerical integration technique. Therefore including thermal stresses is reduced to modifying the force vector by superposing the vector $\{Q\}_{\epsilon_0}$.

In applying this technique to fracture problems, the crack will have some effects on the temperature field in its vicinity, the extreme case being that of a completely insulated crack. A more realistic boundary condition is that of radiation across the crack proportional to the local discontinuity in temperature, [83]. In the embedded singularity method, the part of the crack enclosed in the core and including the tip is not subjected to the thermal stresses which is not an accurate representation of the physical situation.

9.3 Conclusions

The embedded singularity technique has been adapted to solve mode I and mixed mode I and II fracture problems of axisymmetric solids subjected to axisymmetric loading.

Computer programs incorporating a singular core element in a linear strain finite element mesh were developed from which results for a variety of crack configurations were obtained.

The method requires relatively little modification to a standard finite element program and is capable of tackling very complicated problems using a modest size computer with a core storage limit available to the user of (100K).

The computing time for typical runs with various degrees of freedom using an ICL 1904S computer and their cost according to the University Computer Centre costing system are shown in Table (9.1).

Type of problem	D.O.F.	Computing time (sec)	Cost (pence)
General axisymmetric	578	982	2202
Mode I fracture	570	594	1344
Mixed mode fracture	446	470	1082
Shouldered bar mixed mode fracture	502	573	1306

TABLE (9.1)

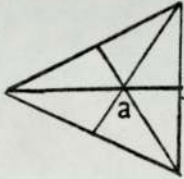
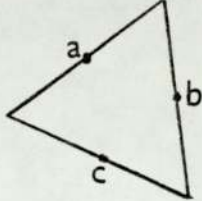
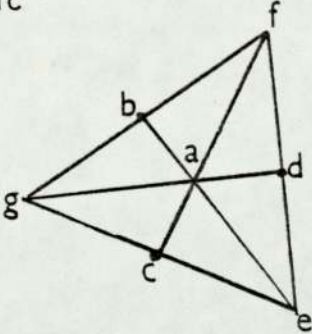
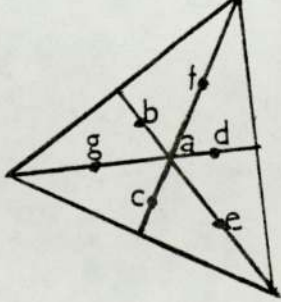
Due to the development of an automatic mesh generation scheme, a change of some of the parameters of a run examples of which are the core size, crack length, crack inclination, fillet radius, and different groove geometries, require changing one card of the input data only.

Approximately 40% of the job time is spent on generating the mesh automatically which shows a great saving compared with a manual system requiring one to two working days per job.

CHAPTER 10

APPENDICES

10.1 Numerical integration formulae for triangles

Order	Fig.	Error	points	Triangular coordinates	Weights W_K
linear		$R=O(h^2)$	a	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$	0.5
quadratic		$R=O(h^3)$	a b c	$\frac{1}{2}, \frac{1}{2}, 0$ $0, \frac{1}{2}, \frac{1}{2}$ $\frac{1}{2}, 0, \frac{1}{2}$	$\frac{1}{6}$ $\frac{1}{6}$ $\frac{1}{6}$
cubic		$R=O(h^4)$	a b c d e f g	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$ $\frac{1}{2}, \frac{1}{2}, 0$ $0, \frac{1}{2}, \frac{1}{2}$ $\frac{1}{2}, 0, \frac{1}{2}$ $1, 0, 0$ $0, 1, 0$ $0, 0, 1$	$\frac{27}{120}$ $\frac{8}{120}$ $\frac{3}{120}$
quentic		$R=O(h^6)$	a b c d e f g	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$ $\alpha_1, \beta_1, \beta_1$ $\beta_1, \alpha_1, \beta_1$ $\beta_1, \beta_1, \alpha_1$ $\alpha_2, \beta_2, \beta_2$ $\beta_2, \alpha_2, \beta_2$ $\beta_2, \beta_2, \alpha_2$	0.1125 0.066197075 0.06296959

where: $\alpha_1 = 0.05971587$
 $\beta_1 = 0.47014206$
 $\alpha_2 = 0.79742699$
 $\beta_2 = 0.10128651$

10.2 Near Crack tip equations for stresses, displacements and strains

According to Muskhelishvili, [42], an Airy stress function for the plane problems of isotropic elasticity may be written in terms of two complex functions, $\phi(z)$ and $\psi(z)$. The components of stresses and displacements associated with them for cartesian coordinates, Fig.(10.1) are given by:

$$\sigma_x + \sigma_y = 2[\phi'(z) + \overline{\phi'(z)}] \quad (10.1)$$

$$\sigma_y - \sigma_x + 2i\tau_{xy} = 2[\bar{z}\phi''(z) + \psi'(z)] \quad (10.2)$$

$$2\mu(u_x + iu_y) = \kappa\phi(z) - \overline{z\phi'(z)} - \overline{\psi(z)} \quad (10.3)$$

and for polar coordinates, Fig. (10.2) are given by:

$$\sigma_r + \sigma_\theta = 2[\phi'(z) + \overline{\phi'(z)}] \quad (10.4)$$

$$\sigma_\theta - \sigma_r + 2i\tau_{r\theta} = 2 e^{2i\theta} [\bar{z}\phi''(z) + \psi''(z)] \quad (10.5)$$

$$2\mu(u_r + iu_\theta) = e^{-i\theta} [\kappa\phi(z) - \overline{z\phi'(z)} - \overline{\psi'(z)}] \quad (10.6)$$

where:

μ = Modulus of rigidity

$\kappa = 3-4\nu$ for plane strain,

$= (3-\nu)/(1+\nu)$ for plane stress

The prime denotes the derivatives with respect to z and the bar indicates the complex conjugate number. To construct $\overline{\phi'(z)}$, the derivative is obtained by:

$$\phi'(z) = \frac{d}{dz} (\phi(z)) \quad (10.7)$$

and the result is converted to the conjugate.

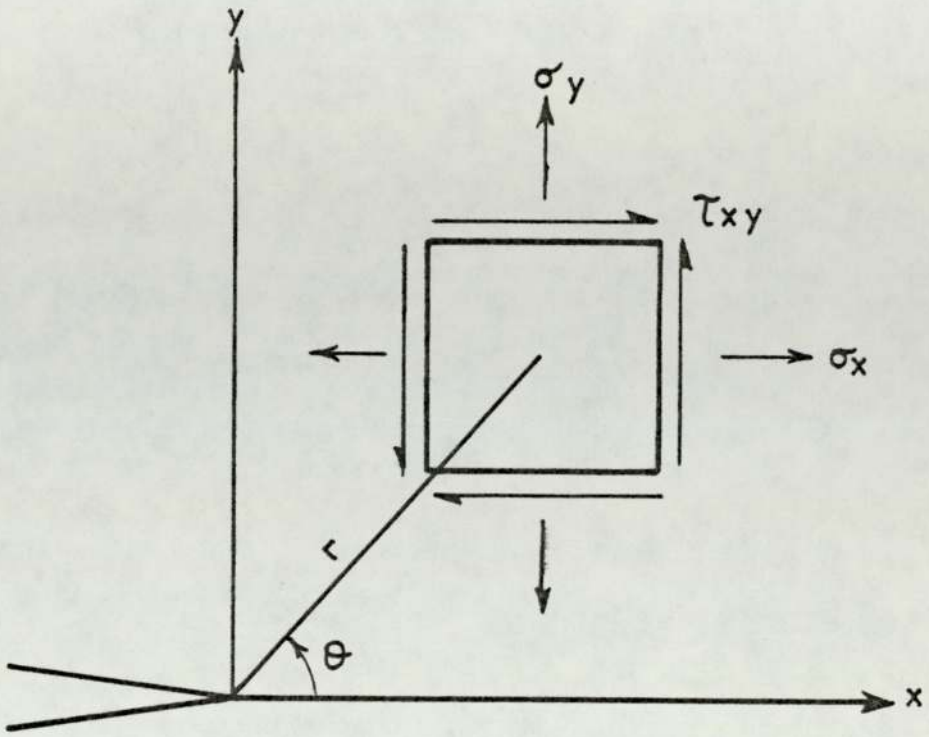


Fig. 10.1

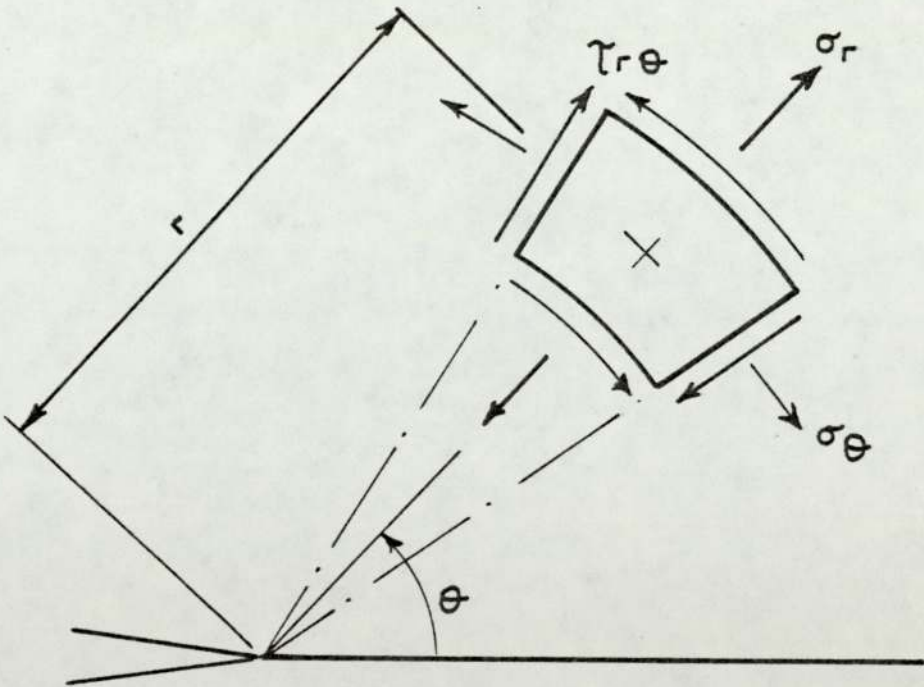


Fig. 10.2

The Goursat functions are taken as suitable forms for $\phi(z)$ and $\psi(z)$, [21], as:

$$\phi(z) = \sum_{n=0}^{\infty} A_n z^{\lambda_n} \quad (10.8)$$

$$\psi(z) = \sum_{n=0}^{\infty} B_n z^{\lambda_n+1} \quad (10.9)$$

where the A's and B's are complex constants to be determined from the boundary conditions and the λ 's are real eigenvalues.

By considering the boundary conditions of the crack configuration shown in Fig. (10.2), it can be shown that, [21]:

$$\begin{aligned} \sigma_{\theta} = \frac{1}{2\sqrt{r}} [a_1(1+\cos\theta)\cos\frac{\theta}{2} + a_2(3\sin\theta)\cos\frac{\theta}{2}] \\ + \alpha_1(1 - \cos 2\theta) \end{aligned} \quad (10.10)$$

$$\begin{aligned} \sigma_r = \frac{1}{2\sqrt{r}} [a_1(3-\cos\theta)\cos\frac{\theta}{2} - a_2(3\cos\theta-1)\sin\frac{\theta}{2}] \\ + 2\alpha_1(1 + \cos 2\theta) \end{aligned} \quad (10.11)$$

$$\begin{aligned} \tau_{r\theta} = \frac{1}{2\sqrt{r}} [a_1\sin\theta\cos\frac{\theta}{2} - a_2(3\cos\theta-1)\cos\frac{\theta}{2}] \\ - 2\alpha_1\sin 2\theta \end{aligned} \quad (10.12)$$

where a_1 and a_2 are related to the stress intensity factors K_I and K_{II} by:

$$a_1 + ia_2 = \frac{1}{\sqrt{2}} (K_I - iK_{II}) \quad (10.13)$$

This relation is sometimes expressed in the form:

$$a_1 + ia_2 = \frac{1}{\sqrt{2\pi}} (K_I - iK_{II}) \quad (10.14)$$

which leads to some confusion. However, most authors list the equations of the rear tip stresses and displacements used by them, and in this work, the form given in equation (10.13) is used. It yields the following stress and displacement components around the crack tip for the polar coordinates system:

$$\sigma_r = \frac{1}{2\sqrt{2r}} [K_I(3-\cos\theta)\cos\frac{\theta}{2} + K_{II}(3\cos\theta-1)\sin\frac{\theta}{2}] + 2\alpha_1(1+\cos 2\theta) \quad (10.15)$$

$$\sigma_\theta = \frac{1}{2\sqrt{2r}} [K_I(1+\cos\theta)\cos\frac{\theta}{2} - K_{II}(3\sin\theta)\cos\frac{\theta}{2}] + 4\alpha_1\sin^2\theta \quad (10.16)$$

$$\tau_{r\theta} = \frac{1}{2\sqrt{2r}} [K_I\sin\theta\cos\frac{\theta}{2} + K_{II}(3\cos\theta-1)\cos\frac{\theta}{2}] - 2\alpha_1\sin^2\theta \quad (10.17)$$

$$u_r = \frac{K_I}{4\mu} \left(\frac{r}{2}\right)^{\frac{1}{2}} [(2\kappa-1)\cos\frac{\theta}{2} - \cos\frac{3\theta}{2}] - \frac{K_{II}}{4\mu} \left(\frac{r}{2}\right)^{\frac{1}{2}} [(2\kappa-1)\sin\frac{\theta}{2} - 3\sin\frac{3\theta}{2}] + \alpha_1 \left(\frac{r}{2\mu}\right)(\kappa-1+2\cos 2\theta) \quad (10.18)$$

$$u_\theta = \frac{K_I}{4\mu} \left(\frac{r}{2}\right)^{\frac{1}{2}} [\sin\frac{3\theta}{2} - (1+2\kappa)\sin\frac{\theta}{2}] - \frac{K_{II}}{4\mu} \left(\frac{r}{2}\right)^{\frac{1}{2}} [(2\kappa+1)\cos\frac{\theta}{2} - 3\cos\frac{3\theta}{2}] - \alpha_1 \left(\frac{r}{\mu}\right) \sin 2\theta \quad (10.19)$$

The strains are obtained from the displacements and are given by:

$$\begin{aligned} \epsilon_r = & \frac{1}{8\mu\sqrt{2r}} \{ K_I [(2\kappa-1)\cos \frac{\theta}{2} - \cos \frac{3\theta}{2}] \\ & - K_{II} [(2\kappa-1)\sin \frac{\theta}{2} - 3\sin \frac{3\theta}{2}] \} \\ & + \alpha_1 \left(\frac{1}{2\mu}\right) (\kappa-1+2\cos 2\theta) \end{aligned} \quad (10.20)$$

$$\begin{aligned} \epsilon_\theta = & \frac{1}{4\mu\sqrt{2r}} \{ K_I \left[\frac{1}{2} \cos \frac{3\theta}{2} + (\kappa - \frac{3}{2}) \cos \frac{\theta}{2} \right] \\ & - K_{II} \left[(\kappa - \frac{3}{2}) \sin \frac{3\theta}{2} + \frac{3}{2} \sin \frac{\theta}{2} \right] \} \\ & + \alpha_1 \left(\frac{1}{2\mu}\right) (\kappa-1-2\cos 2\theta) \end{aligned} \quad (10.21)$$

$$\begin{aligned} \gamma_{r\theta} = & \frac{1}{4\mu\sqrt{2r}} [K_I (\sin \frac{\theta}{2} + \sin \frac{3\theta}{2}) + K_{II} (\cos \frac{\theta}{2} + 3\cos \frac{3\theta}{2})] \\ & - \alpha_1 \left(\frac{2}{\mu}\right) \sin 2\theta \end{aligned} \quad (10.22)$$

For a cartesian coordinate system, the stresses, displacements, and strains components are given by:

$$\begin{aligned} \sigma_x = & \frac{1}{\sqrt{2r}} [K_I \cos \frac{\theta}{2} (1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2}) \\ & - K_{II} \sin \frac{\theta}{2} (2 + \cos \frac{\theta}{2} \cos \frac{3\theta}{2})] \end{aligned} \quad (10.23)$$

$$\begin{aligned} \sigma_y = & \frac{1}{\sqrt{2r}} [K_I \cos \frac{\theta}{2} (1 + \sin \frac{\theta}{2} \sin \frac{3\theta}{2}) \\ & + K_{II} (\sin \frac{\theta}{2} \cos \frac{\theta}{2} \cos \frac{3\theta}{2})] \end{aligned} \quad (10.24)$$

$$\begin{aligned} \tau_{xy} = & \frac{1}{\sqrt{2r}} [K_I (\cos \frac{\theta}{2} \sin \frac{\theta}{2} \cos \frac{3\theta}{2}) \\ & + K_{II} \cos \frac{\theta}{2} (1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2})] \\ & + 2\alpha_1 (1 - \sin\theta \cos\theta) \end{aligned} \quad (10.25)$$

$$\begin{aligned} u_x = & \frac{1}{4\mu} \left(\frac{r}{2}\right)^{\frac{1}{2}} \{K_I [(2\kappa-1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2}] \\ & + K_{II} [(2\kappa+3) \sin \frac{\theta}{2} + \sin \frac{3\theta}{2}] \} \\ & + \alpha_1 \left(\frac{r}{2\mu}\right) (\kappa+1) \cos\theta \end{aligned} \quad (10.26)$$

$$\begin{aligned} u_y = & \frac{1}{4\mu} \left(\frac{r}{2}\right)^{\frac{1}{2}} \{K_I [(2\kappa+1) \sin \frac{\theta}{2} - \sin \frac{3\theta}{2}] \\ & - K_{II} [(2\kappa-3) \cos \frac{\theta}{2} + \cos \frac{3\theta}{2}] \} \\ & + \alpha_1 \left(\frac{r}{2\mu}\right) (\kappa-3) \sin\theta \end{aligned} \quad (10.27)$$

$$\begin{aligned} \epsilon_x = & \frac{1}{8\mu\sqrt{2r}} \{K_I [(2\kappa-3) \cos \frac{\theta}{2} + \cos \frac{5\theta}{2}] \\ & - K_{II} [(2\kappa+1) \sin \frac{\theta}{2} - \sin \frac{5\theta}{2}] \} \\ & + \alpha_1 \left(\frac{1}{2\mu}\right) (\kappa+1) \end{aligned} \quad (10.28)$$

$$\begin{aligned} \epsilon_y = & \frac{1}{8\mu\sqrt{2r}} \{K_I [(2\kappa-1) \cos \frac{\theta}{2} - \cos \frac{5\theta}{2}] \\ & - K_{II} [(2\kappa-5) \sin \frac{\theta}{2} - \sin \frac{5\theta}{2}] \} \\ & + \alpha_1 \left(\frac{1}{2\mu}\right) (\kappa-3) \end{aligned} \quad (10.29)$$

$$\gamma_{xy} = \frac{1}{4\mu\sqrt{2r}} [K_I(\sin \frac{5\theta}{2} - \sin \frac{\theta}{2}) + K_{II}(3\cos \frac{\theta}{2} - \cos \frac{5\theta}{2})] \quad (10.30)$$

For axisymmetric problems, the radial and axial displacements are the same as (10.26) and (10.27) for the plane strain case, [49]. The radial displacement component u_r is the same as u_x given by equation (10.26) for the case of a penny shaped crack. For a circumferential crack $u_r = -u_x$. The angle θ is always measured from the r direction ahead of the crack tip.

10.3 Strain energy of the core region

10.3.1 Mode I formulation

The core strain energy is derived from the strains, displacements, and stresses relationships derived from linear elastic fracture mechanics, and is given by:

$$U_c = \int_{V_c} SED \, dV_c \quad (10.31)$$

where: V_c = Core volume

$$SED/\text{unit volume} = \frac{1}{2} \sigma_{ij} \epsilon_{ij} \quad (10.32)$$

$$SED = \frac{1}{2} (\sigma_r \epsilon_r + \sigma_\theta \epsilon_\theta + \tau_{r\theta} \gamma_{r\theta}) \quad (10.33)$$

With reference to Fig. (10.3), equation (10.31) becomes:

$$U_c = \pi R \int_0^{\pi} \int_0^{r_0} (\sigma_r \epsilon_r + \tau_{r\theta} \epsilon_\theta + \tau_{r\theta} \gamma_{r\theta}) r \, dr \, d\theta \quad (10.34)$$

The stresses are obtained from equations (10.15, 10.16 and 10.17) by setting K_{II} and α_1 to zero:

$$\sigma_r = - \frac{K_I}{2\sqrt{2r}} (3\cos\theta) \cos \frac{\theta}{2} \quad (10.35)$$

$$\sigma_\theta = \frac{K_I}{2\sqrt{2r}} (1+\cos\theta) \cos \frac{\theta}{2} \quad (10.36)$$

$$\tau_{r\theta} = \frac{K_I}{2\sqrt{2r}} \sin\theta \cos \frac{\theta}{2} \quad (10.37)$$

Similarly for the strains:

$$\epsilon_r = \frac{K_I}{8\mu\sqrt{2r}} \left[(2\kappa-1)\cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \right] \quad (10.38)$$

$$\epsilon_\theta = \frac{K_I}{4\mu\sqrt{2r}} \left[\frac{1}{2} \cos \frac{3\theta}{2} + \left(\kappa - \frac{3\theta}{2} \right) \cos \frac{3\theta}{2} \right] \quad (10.39)$$

$$\gamma_{r\theta} = \frac{K_I}{4\mu\sqrt{2r}} \left[2\sin\theta \cos \frac{\theta}{2} \right] \quad (10.40)$$

By substitution and performing the necessary integrations in equation (10.34) the core strain energy is obtained:

$$U_c = 2\pi R \left[\frac{K_I^2 r_o \pi}{32\mu} (2\kappa-1) \right] \quad (10.41)$$

where: R = distance from the axis of revolution
to the core centroid

r_o = radius of the core

10.3.2 Mixed mode I and II formulation

The core strain energy for the mixed mode I and II situation is given by:

$$U'_c = \pi R \int_0^{2\pi} \int_0^{r_o} (\sigma_r \epsilon_r + \sigma_\theta \epsilon_\theta + \tau_{r\theta} \gamma_{r\theta}) r dr d\theta \quad (10.42)$$

The stresses and strains are obtained from Appendix (10.2) by setting α to zero only, and the rest of the evaluation is similar to that of section (10.3.1). The core strain energy obtained is:

$$U'_c = 2\pi R \left[\frac{K_I^2 r_o \pi}{16\mu} (2\kappa-1) + \frac{K_{II}^2 r_o \pi}{16\mu} (3+2\kappa) \right] \quad (10.43)$$

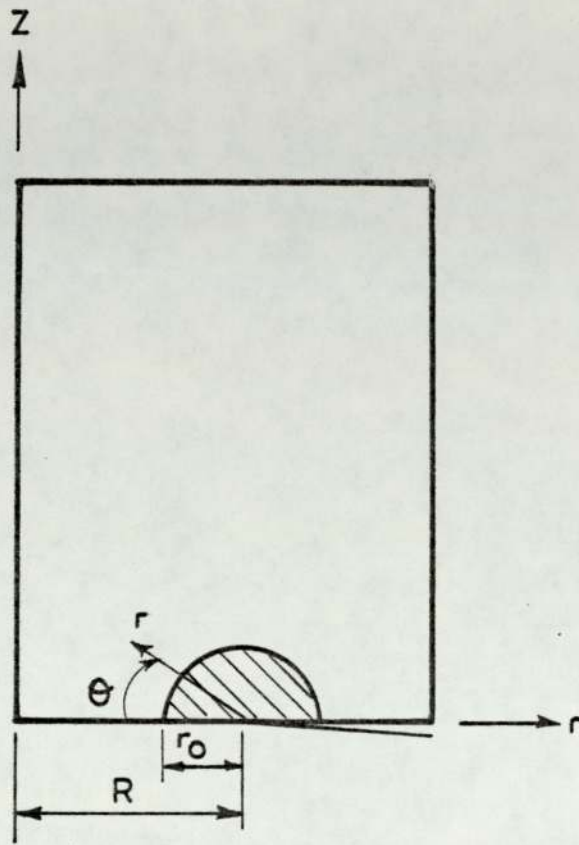


Fig. 10.3

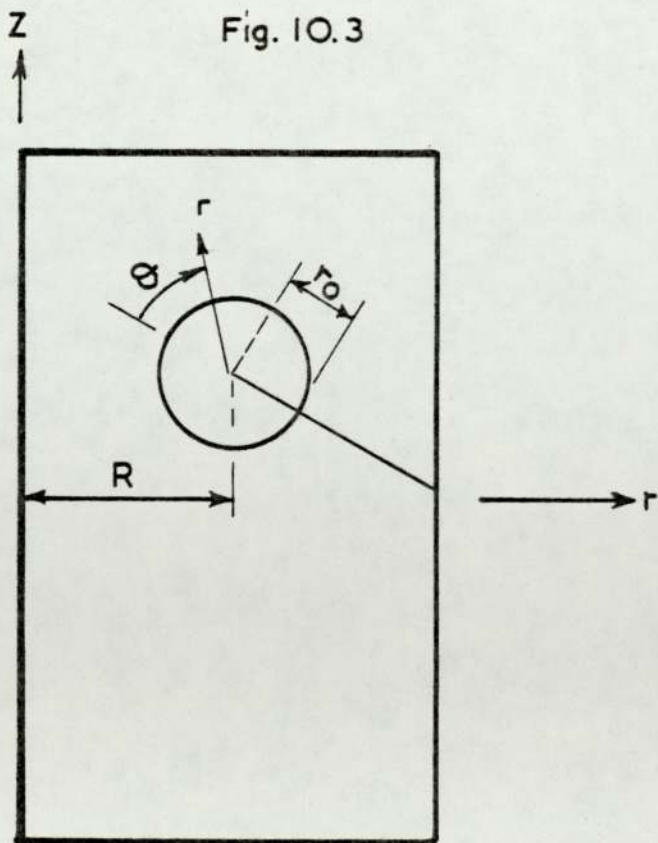


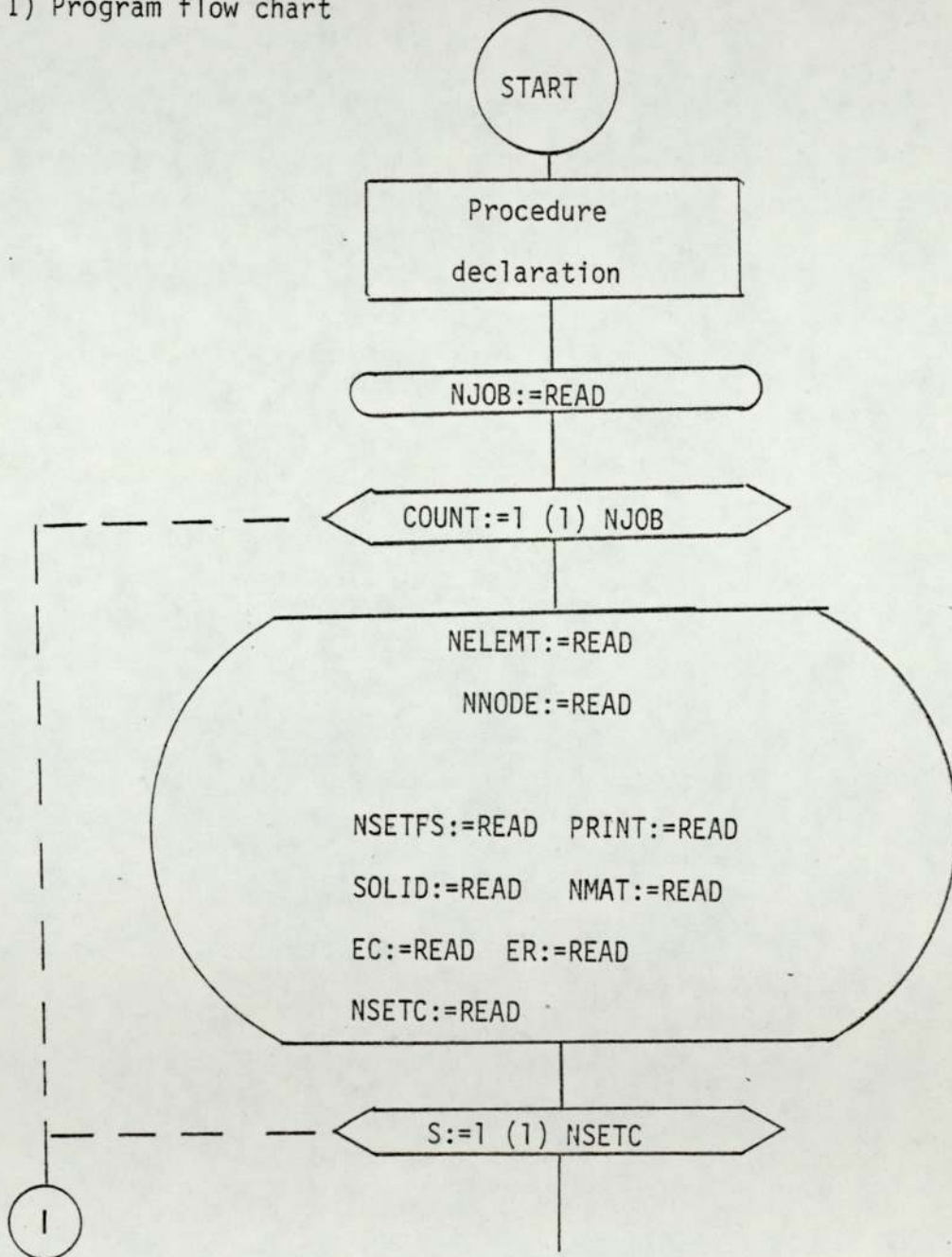
Fig. 10.4

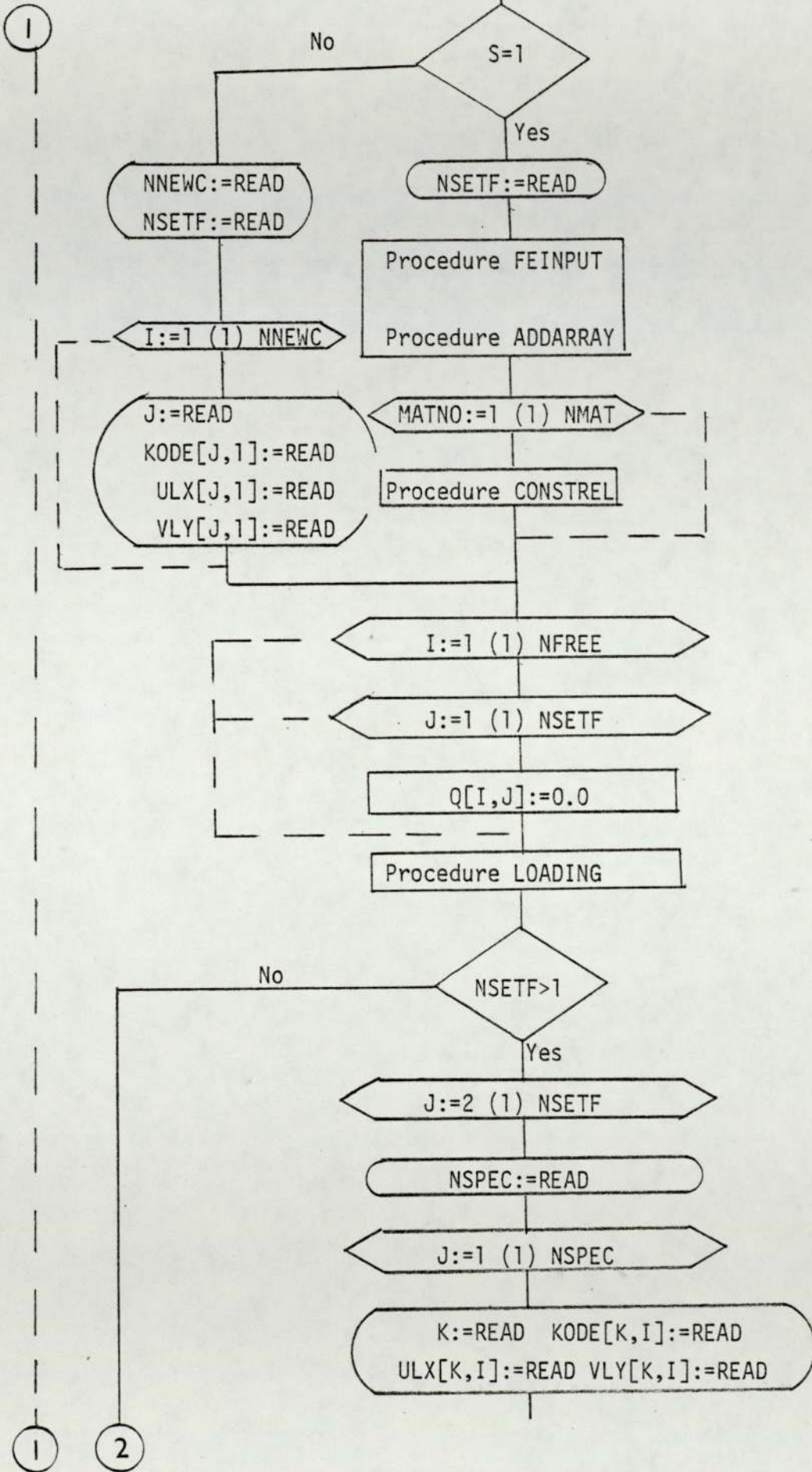
10.4 Programs Listings and sample problems

The complete program flow charts and listings will be presented in this section using the procedures described in Chapter 7 together with input data instructions and a sample problem for each type.

10.4.1 The general axisymmetric program

I) Program flow chart





II) Program listing

```
'PROGRAM' (AXXX)
'INPUT' O=CR0
'OUTPUT' O=LPO
'EXTENDED DATA'
'EXTENDED'
'TRACE' 2
'BEGIN' 'COMMENT' THIS PROG EMPLOYS THE ISOPARAMETRIC
          FORMULATION FOR AXISYMETRIC PROBLEMS
          THE ELEMENT USED IS A SIX-NODE TRIANGLE;
'COMMENT' CHANGES IN INPUT:-
          NODAL CONNECTIONS, MATERIAL NO.,
          AND ELEMENT PROPERTIES;
'INTEGER' NELEMT, NNODE, NSETS, I, J, K, V, U, NSETC, NNEIC, S,
NJOB, COUNT, PRNT, NMAT, NHTD, SYM, ER, NSEP, SOLID, CASE;
'REAL' DELTA, DETJ, RMEAN, RAVG, XS, XE, YS, YE;
'PROCEDURE' STRDIS(L1, L2, L3, B, X, Y, Z, ER, SO, IO);
'VALUE' L1, L2, L3, Z;
'INTEGER' Z, ER, SOLID;
'REAL' L1, L2, L3, U;
'INTEGER' 'ARRAY' N;
'ARRAY' X, Y, B;
'BEGIN'
'INTEGER' I, V;
'REAL' CHANGE, C;
'REAL' 'ARRAY' J[1:2, 1:2], NL[1:6], P[1:2, 1:6];
'COMMENT' THIS PROCEDURE EVALUATES THE JACOBIAN J ITS DETERMINANT U
          AND THE STRAIN-DISP ARRAY B;
J[1,1]:=X[N(Z,1)]*(4*L1-1)+X[N(Z,2)]*(4*L1+4*(L2-3))+4*L2*X[N(Z,4)]*
4*L2*X[N(Z,5)]+4*X[N(Z,6)]*(1-2*L1-L2);
J[1,2]:=Y[N(Z,1)]*(4*L1-1)+Y[N(Z,2)]*(4*L1+4*(L2-3))+4*L2*Y[N(Z,4)]*
4*L2*Y[N(Z,5)]+4*Y[N(Z,6)]*(1-2*L1-L2);
J[2,1]:=X[N(Z,2)]*(4*L2-1)+X[N(Z,3)]*(4*L1+4*(L2-3))+4*L1*X[N(Z,4)]*
4*X[N(Z,5)]*(1-L1-2*L2)-4*L1*X[N(Z,6)];
J[2,2]:=Y[N(Z,2)]*(4*L2-1)+Y[N(Z,3)]*(4*L1+4*(L2-3))+4*L1*Y[N(Z,4)]*
4*Y[N(Z,5)]*(1-L1-2*L2)-4*L1*Y[N(Z,6)];
'COMMENT' U REPLACES DETJ;
U:=J[1,1]*J[2,2]-J[1,2]*J[2,1];
'COMMENT' THE COEFFS OF [J] ARE REPLACED BY THOSE OF [U]=1;
CHANGE:=J[1,1];
J[1,1]:=J[2,2]/U;
J[1,2]:=-J[1,2]/U;
J[2,1]:=-J[2,1]/U;
J[2,2]:=CHANGE/U;
'COMMENT' NL IS THE DIS FUNCTION AND NOT ITS DERIVATIVES WRT X,Y;
NL[1]:=L1*(2*L1-1);
NL[2]:=L2*(2*L2-1);
NL[3]:=L3*(2*L3-1);
NL[4]:=4*L1*L2;
NL[5]:=4*L2*L3;
NL[6]:=4*L3*L1;
'IF' SOLID=0 'THEN'
```



```
C:=1
'ELSE'
C:=(2*ER)+1;
'IF' SOLID=0 'THEN' 'GOTO' SL1;
RMEAN:=(X[N(Z,1)]+X[N(Z,2)]+X[N(Z,3)]+
X[N(Z,4)]+X[N(Z,5)]+X[N(Z,6)])/6;
'FOR' Z:=1 'STEP' 1 'UNTIL' 6 'DO'
RAVG:=RMEAN;
SL1:'FOR' Z:=C 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN'
RAVG:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
RAVG:=RAVG+(X[N(Z,I)]*NL[I])/6;
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' V:=1,2 'DO' P[V,I]:=0.0;
'FOR' I:=1,2 'DO'
'BEGIN' P[I,1]:=J[I,1]*(4*L1-1);
P[I,2]:=J[I,2]*(4*L2-1);
P[I,3]:=J[I,1]*(1-4*L3)+J[I,2]*(1-4*L3);
P[I,4]:=4*(L2*J[I,1]-L1*J[I,2]);
P[I,5]:=4*(J[I,2]*L3-L2*(J[I,1]+J[I,2]));
P[I,6]:=4*(J[I,1]*L3-L1*(J[I,1]-J[I,2]));
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' V:=1,2,3,4 'DO' R[V,I]:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN'
B[1,(I*2-1)]:=B[4,(I*2)]:=P[1,I];
B[4,(I*2-1)]:=B[3,(I*2)]:=P[2,I];
B[2,(I*2-1)]:=NL[I]/RAVG;
'END';
'END' OF STRDIS;

'PROCEDURE' BOUNCONST(U,N,R,AK,NEQ,F);
'VALUE' U,N,NEQ,F;
'REAL' U;
'INTEGER' N,NEQ,F;
'ARRAY' R,AK;
'INTEGER' 'ARRAY' A;
'BEGIN' 'INTEGER' M,K,CJ;
'IF' N=1 'THEN' CJ:=1 'ELSE' CJ:=N-(A[N]-A[N-1])+1;
'FOR' K:=CJ 'STEP' 1 'UNTIL' N 'DO'
'BEGIN'
R[K,F]:=R[K,F]-AK[A[K]]*(R[K]-U);
AK[A[N]-N+K]:=0.0;
'END';
'IF' N+1 'GT' NEQ 'THEN' 'GOTO' L2;
'FOR' K:=N+1 'STEP' 1 'UNTIL' NEQ 'DO'
'BEGIN' CJ:=K-(A[K]-A[K-1])+1;
'IF' CJ 'LE' N 'THEN'
'BEGIN'
R[K,F]:=R[K,F]-AK[A[K]]*(R[K]-U);
AK[A[K]-K+1]:=0.0;
'END' 'ELSE'
'END';
```



```
LZ: AK(AEN)]=1.0;  
R(N,F):=U;  
'END' OF BOUNCONST ;
```

```
'PROCEDURE' LOADING(A,B,C,D,E,F);  
'VALUE' A,B,C,E,F;  
'INTEGER' A,E,F;  
'REAL' B,C;  
'ARRAY' D;  
'BEGIN'  
'INTEGER' K;  
'IF' A=3 'THEN' 'GOTO' KLAB1;  
K:=2*E;  
'IF' A=1 'THEN' 'GOTO' KLAB2;  
D[K-1,F]:=D[K-1,F]+B;  
'IF' A 'NE' 0 'THEN' 'GOTO' KLAB1;  
KLAB2: D[K,F]:=D[K,F]+C;  
KLAB1: 'END' OF LOADING;
```

```
'PROCEDURE' SYMBVSOL(A,L,S,B) DIMENSIONS: (N,K) FAILURE EXIT: (FAIL);  
'VALUE' N,R; 'ARRAY' A,L,B; 'INTEGER' 'ARRAY' S; 'INTEGER' N,K;  
'LABEL' FAIL;  
'BEGIN'  
'INTEGER' G,H,I,J,K,M,P,Q,T,U,V;  
'REAL' Y;  
H:=0;  
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'  
  'BEGIN'  
    T:=I+H-S[I]+1; G:=H+1;  
    P:=S[T]-1;  
    'FOR' J:=T 'STEP' 1 'UNTIL' I-1 'DO'  
      'BEGIN'  
        Q:=P+1; H:=H+1;  
        P:=S[J]; K:=J+Q-P;  
        V:=H-P; U:=G;  
        Y:=A[H];  
        'IF' K 'GT' T 'THEN' H:=U+K-T;  
        'FOR' U:=U 'STEP' 1 'UNTIL' H-1 'DO'  
          Y:=Y-L[U]*L[U-V];  
          Y:=Y/L[H-V]; L[H]:=Y;  
        'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'  
          B[I,M]:=B[I,M]-B[J,M]*Y;  
        'END' J;  
        Y:=A[H+1];  
        'FOR' U:=G 'STEP' 1 'UNTIL' H 'DO'  
          Y:=Y-L[U]*2;  
        'IF' Y 'LE' 0 'THEN' 'GOTO' FAIL;  
        H:=H+1; Y:=SQRT(Y);  
        L[H]:=Y;  
        'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'  
          B[I,M]:=B[I,M]/Y;  
        'END' I;  
'COMMENT' REDUCTION COMPLETED;  
'FOR' I:=N 'STEP' -1 'UNTIL' 1 'DO'
```



```
'BEGIN'  
V:=L[H];  
'FOR' M:=1 'STEP' 1 'UNTIL' ? 'DO'  
B[I,M]:=B[I,M]/V;  
'IF' I=1 'THEN' 'GOTO' COMPLETE;  
J:=I; P:=S[I-1];  
'FOR' H:=H-1 'STEP' -1 'UNTIL' -1 'DO'  
  'BEGIN'  
  J:=J-1; Y:=L[H];  
  'FOR' M:=1 'STEP' 1 'UNTIL' ? 'DO'  
  B[J,M]:=B[J,M]-B[I,H]*Y;  
  'END' H;  
H:=P;  
COMPLETE: 'END' I;  
'END' SYMVSOL;
```

```
'PROCEDURE' CONSTREL(Z,A,MATNO,CASE);  
'VALUE' MATNO;  
'INTEGER' MATNO,CASE;  
'ARRAY' A,Z;  
'BEGIN'  
'INTEGER' I;  
'REAL' C11,C12,C13,C14,  
C22,C23,C24,C33,C34,C44,P,C;  
'COMMENT' CALCULATION OF ELASTIC CONSTANTS FOR SEVERAL MATERIALS;  
CASE:=READ;  
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'  
A[I]:=READ;  
'IF' MATNO=1 'THEN'  
'BEGIN' WRTTEXT('('C'AC')'MATERIAL'PROPERTIES:=  
  ('2C')'MATERIAL'('21S')'ELASTIC'PROPERTIES  
  ('CS')'NUMBER'('30S')'E1'('11S')'E2'('3S')'  
  V1'('7S')'V2'('9S')'G12')';  
'END';  
NEWLINE(2); SPACE(1);  
PRINT(MATNO,2,0); SPACE(25); PRINT(A[1],0,3);SPACE(22);  
PRINT(A[4],0,3);SPACE(2);PRINT(A[2],1,2);SPACE(2);PRINT(A[5],1,2);  
SPACE(2);PRINT(A[3],0,3);  
C14:=C24:=C34:=0.0;  
'IF' CASE=0 'THEN'  
'BEGIN' C44:=A[3];  
A[3]:=A[1]/((1+A[5])+(1-A[5]-2*(A[4]/A[1])*A[2]*A[2]));  
C11:=A[3]*(1-A[5]+A[5]);  
C12:=C13:=C23:=A[3]*(A[4]/A[1])*A[2]*(1+A[5]);  
C22:=C33:=(A[4]/A[1])*(1-(A[4]/A[1])*A[2]-A[2])*A[3];  
'END' 'ELSE'  
'BEGIN'  
P:=(1+A[2])*(1-A[2]-2*(A[2]*A[5]));  
C:=A[1]/P;  
C11:=C22:=C*(1-(A[2]*A[5]));  
C12:=C*(A[2]+(A[2]*A[5]));  
C13:=C23:=(C*A[2])*(1+A[2]);  
C33:=C*(A[1]/A[4])*(1-(A[2]*A[2]));  
C44:=A[4]/(2*(1-A[5]));  
'END';
```



```
'BEGIN' Z[MATNO,1]:=C11; Z[MATNO,2]:=C12;
Z[MATNO,3]:=C13; Z[MATNO,4]:=C14;
Z[MATNO,5]:=C22; Z[MATNO,6]:=C23;
Z[MATNO,7]:=C24; Z[MATNO,8]:=C33;
Z[MATNO,9]:=C34; Z[MATNO,10]:=C44;
'END';
'END' OF PROCEDURE CONSTREL;
'PROCEDURE' ADDARRAY(NELEM,NNODE,ADD,NODE);
'INTEGER' NELEM,NNODE;
'INTEGER' 'ARRAY' NODE,ADD;
'BEGIN' 'INTEGER' W,CH,I,ADDTMP;
'FOR' W:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN' CH:=NODE[W,1];
'FOR' I:=2 'STEP' 1 'UNTIL' 6 'DO'
'IF' NODE[W,I] LT CH 'THEN' CH:=NODE[W,I];
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN' ADDTMP:=NODE[W,I]*2+1;
'IF' ADDTMP GT ADD[NODE[W,I]*2] 'THEN' ADD[NODE[W,I]*2]:=ADDTMP;
'END';
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'BEGIN' CH:=2*ADD[2*I]; I:=2*I;
ADD[W=1]:=ADD[W=2]*CH+1;
ADD[W]:=ADD[W-1]*CH;
'END';
'END' OF PROCEDURE ADDARRAY;
'PROCEDURE' PPINPUT(C,ADD,XX,YY,REP,NNODE,NSETF,KODE,NSPEC,NSETFS,
ULX,VLY,NELEM,NODE,NMAT,XS,XF,YS,YF,EC,ER,BSHP,VXS,VXF,VYS,VYF,SYM,
MTNO,NSTEL,STEL);
'VALUE' NFREE,NNODE,NSETF,NSPEC,NELEM,NSETFS,NNAT;
'INTEGER' NFREE,NNODE,NSETF,NSPEC,NELEM,NSETFS,NNAT,
EC,ER,BSHP,SYM;
'REAL' XS,XF,YS,YF;
'INTEGER' 'ARRAY' ADD,KODE,NODE,MMNO,NSTEL,STEL;
'REAL' 'ARRAY' XX,YY,ULX,VLY,C,VXS,VXF,VYS,VYF;
'BEGIN' 'INTEGER' I,J,U,A,B,L,D,N;
'REAL' 'ARRAY' DELTAX[1,(EC+2)+1],DELTAY[1,(EC+2)+1];
'FOR' I:=0 'STEP' 2 'UNTIL' NFREE 'DO' ADD[I]:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO' XX[I]:=0.00001;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' KODE[I,J]:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO'
ULX[I,J]:=VLY[I,J]:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' NNAT 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO' C[I,J]:=0.0;
BSHP:=READ;
'IF' BSHP=2 'THEN' 'GOTO' NL1;
'BEGIN'
XS:=READ; XF:=READ; YS:=READ; YF:=READ;
'FOR' N:=1 'STEP' 1 'UNTIL' (PP+2)+1 'DO'
'BEGIN'
VXS[N]:=XS;
VXF[N]:=XF; DELTAX[N]:=(XF-XS)/(EC+2);
'END';
```



```
'FOR' N:=1 'STEP' 1 'UNTIL' (ER*2)+1 'DO'  
'BEGIN'  
VYS[N]:=YS;  
VYF[N]:=YF; DELTAY[N]:=(YF-YS)/(ER*2);  
'END';  
'END';  
'IF' BSHp=1 'THEN' 'GOTO' HL2;  
HL1:'BEGIN'  
'FOR' N:=1 'STEP' 2 'UNTIL' (ER*2)+1 'DO'  
'BEGIN'  
VXS[N]:=READ;  
VXF[N]:=READ; DELTAX[N]:=(VXF[N]-VXS[N])/(ER*2);  
'END';  
'FOR' N:=1 'STEP' 2 'UNTIL' (ER*2)+1 'DO'  
'BEGIN'  
VYS[N]:=READ;  
VYF[N]:=READ; DELTAY[N]:=(VYF[N]-VYS[N])/(ER*2);  
'END';  
'END';  
HL2:A:=(ER*2)+1;  
'FOR' I:=1 'STEP' 2 'UNTIL' (ER*2)+1 'DO'  
'BEGIN'  
B:=I-1;  
'FOR' J:=1 'STEP' 2 'UNTIL' (ER*2)+1 'DO'  
'BEGIN'  
XX[(B*A)+J]:=VXS[J]+(I-1)*DELTAX[J];  
YY[(B*A)+J]:=VYS[J]+(J-1)*DELTAY[J];  
'END';  
'END';  
NSPEC:=READ;  
'FOR' I:=1 'STEP' 1 'UNTIL' NSPEC 'DO'  
'BEGIN'  
J:=READ;  
KODE[J,1]:=READ; ULX[J,1]:=READ; VLY[J,1]:=READ;  
'END';  
'COMMENT' THE NODE NUMBER MUST BE SPECIFIED ALONG WITH  
THE COORDS OF THE NODE;  
'FOR' W:=1 'STEP' 1 'UNTIL' NSPEC 'DO'  
MTNO[W]:=1;  
'BEGIN'  
'IF' NMAT=1 'THEN'  
'GOTO' RWA4  
'ELSE'  
'BEGIN'  
'FOR' I:=2 'STEP' 1 'UNTIL' NMAT 'DO'  
'BEGIN'  
NSTEL[I]:=READ;  
'FOR' J:=1 'STEP' 1 'UNTIL' NSTEL[I] 'DO'  
STEL[I,J]:=READ;  
'FOR' W:=1 'STEP' 1 'UNTIL' NSPEC 'DO'  
'BEGIN'  
J:=1;  
RWA1:'IF' W=STEL[I,J] 'THEN'  
'GOTO' RWA2  
'ELSE'
```



```
J:=J+1;
'IF' J 'LE' NSTELC[I] 'THEN'
'GOTO' RWA1
'ELSE'
'GOTO' RWA3;
RWA2:MTNO[W]:=I;
RWA3:'END';
'END';
'END';
RWA4:'END';
SYM:=READ;
'IF' SYM=1 'THEN' 'GOTO' NL1;
'FOR' I:=1 'STEP' 2 'UNTIL' NELENT-1 'DO'
'BEGIN'
L:=1;
D:=0;
ALI:'IF' I 'LE' (L*(A-1))-1 'THEN'
'BEGIN'
B:=I-((L*A)-A-L+2);
W:=I;
NODE[W,1]:=1+B*D;
NODE[W,2]:=(2*A)+1+B*D;
NODE[W,3]:=(2*A)+3+B*D;
NODE[W,4]:=A+1+B*D;
NODE[W,5]:=(2*A)+2+B*D;
NODE[W,6]:=A+2+B*D;
NODE[W,7]:=MTNO[W];
'END'
'ELSE'
'BEGIN'
L:=L+1;
D:=D+(2*A);
'GOTO' ALI;
'END';
'END';
'FOR' I:=2 'STEP' 2 'UNTIL' NELENT 'DO'
'BEGIN'
L:=1;
D:=0;
RANA:'IF' I 'LE' (L*(A-1)) 'THEN'
'BEGIN'
B:=I-((L*A)-A-L+3);
W:=I;
NODE[W,1]:=1+B*D;
NODE[W,2]:=(2*A)+3+B*D;
NODE[W,3]:=3+B*D;
NODE[W,4]:=A+2+B*D;
NODE[W,5]:=A+3+B*D;
NODE[W,6]:=2+B*D;
NODE[W,7]:=MTNO[W];
'END'
```



```
'ELSE'  
'BEGIN'  
L:=L+1;  
D:=D+(2*A);  
'GOTO' RANA;  
'END';  
'END';  
'IF' SYM=0 'THEN' 'GOTO' NL2;  
NL1: 'FOR' I:=1 'STEP' 2 'UNTIL' NELENT-1 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
LLI: 'IF' I 'LE' (L*(A-1))-1 'THEN'  
'BEGIN'  
B:=I-((L*A)-A-L+2);  
W:=I;  
'IF' I 'LE' (L*(A-1))-((A+1)/2) 'THEN'  
'BEGIN'  
NODE[W,1]:=1+B+D;  
NODE[W,2]:=(2*A)*1+B+D;  
NODE[W,3]:=(2*A)+3+B+D;  
NODE[W,4]:=A+1+B+D;  
NODE[W,5]:=(2*A)+2+B+D;  
NODE[W,6]:=A+2+B+D;  
NODE[W,7]:=MTNO[W];  
'END'  
'ELSE'  
'BEGIN'  
NODE[W,1]:=1+B+D;  
NODE[W,2]:=(2*A)*1+B+D;  
NODE[W,3]:=3+B+D;  
NODE[W,4]:=A+1+B+D;  
NODE[W,5]:=A+2+B+D;  
NODE[W,6]:=2+B+D;  
NODE[W,7]:=MTNO[W];  
'END';  
'END'  
'ELSE'  
'BEGIN'  
L:=L+1;  
D:=D+(2*A);  
'GOTO' LLI;  
'END';  
'END';  
'FOR' I:=2 'STEP' 2 'UNTIL' NELENT 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
LANA: 'IF' I 'LE' (L*(A-1)) 'THEN'  
'BEGIN'  
B:=I-((L*A)-A-L+3);  
W:=I;  
'IF' I 'LE' (L*(A-1))-((A-1)/2) 'THEN'  
'BEGIN'  
NODE[W,1]:=1+B+D;
```



```
NODE[U,2]:=(2-A)*3+B+D;
NODE[U,3]:=3+B+D;
NODE[U,4]:=A+2+B+D;
NODE[U,5]:=A+3+B+D;
NODE[U,6]:=2+B+D;
NODE[U,7]:=MTNO[U];
'END'
'ELSE'
'BEGIN'
NODE[W,1]:=3+B+D;
NODE[U,2]:=(2+A)+1+B+D;
NODE[W,3]:=(2*A)+3+B+D;
NODE[U,4]:=A+2+B+D;
NODE[U,5]:=(2*A)+2+B+D;
NODE[U,6]:=A+3+B+D;
NODE[U,7]:=MTNO[W];
'END';
'END'
'ELSE'
'BEGIN'
L:=L+1;
D:=D+(2*A);
'GOTO'LANA;
'END';
'END';
NL2:WRITETEXT('('('2C')'NODAL%CONNECTIONS%HAVE%BEEN%READ')');
'FOR' W:=1 'STEP' 1 'UNTIL' NLELEM 'DO'
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN' 'IF' XX[NODE[W,3+I]]=0.00001 'THEN'
'BEGIN'
'IF' I=1 'THEN' J:=1 'ELSE' 'IF' I=2 'THEN' J:=2 'ELSE' J:=0;
XX[NODE[W,3+I]]:=(XX[NODE[W,I]]+XX[NODE[W,1+J]])/2;
YY[NODE[W,3+I]]:=(YY[NODE[W,I]]+YY[NODE[W,1+J]])/2;
'END';
'END';
WRITETEXT('('('4C')'NODAL%POINT%DATA'('2C4S')'NODE'('5S')'R%COORD'
('5S')'Z%COORD'('5S')'TYPE'('7S')'RADIUS'('6S')'ZADIUS'('6S')'
OR%LOAD'('5S')'OR%LOAD')');
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'BEGIN' NEWLINE(1); SPACE(3);
PRINT(I,3,0); SPACE(3);
PRINT(XX[I,0,3]);
PRINT(YY[I,0,3]); SPACE(2);
PRINT(KODE[I,1],3,0); SPACE(2);
PRINT(CULX[I,1,0,3]);
PRINT(VLY[I,1,0,3]);
'END';
WRITETEXT('('('4C')'ELEMENT%DATA'('2C')'ELEMENT'('18S')'
NODAL%CONNECTIONS'('17S')'MATERIAL')');
'FOR' W:=1 'STEP' 1 'UNTIL' NLELEM 'DO'
'BEGIN' NEWLINE(1);
PRINT(W,3,0); SPACE(6);
'FOR' J:=1 'STEP' 1 'UNTIL' 7 'DO'
'BEGIN' PRINT(NODE[U,J],3,0);
SPACE(2);
```



```
'END';
'END';
'END' OF PROCEDURE FEINPUT;
'PROCEDURE' ASSEMBLY(NELEMT,NFREE,NSETF,
                    NSETF,STRDIS);
'VALUE' NELEMT,NFREE,NSETF;
'INTEGER' NELEMT,NFREE,NSETF; 'REAL' DETJ;
'REAL' 'ARRAY' K,XX,YY,C;'INTEGER' 'ARRAY' IODE,ADD;
'PROCEDURE' STRDIS;
'BEGIN' 'INTEGER' I,J,U,Z,V,SUB1,SUB2,SUB3;
'REAL' 'ARRAY' B[1:4,1:12],KE[1:12,1:12],U[1:4,1:4];
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' U[1,1]:=0,U[3,3]:=3;
W[1,2]:=W[1,3]:=W[2,3]:=W[2,4]:=W[3,2]:=W[3,4]:=0.5;
W[1,4]:=W[2,2]:=W[3,3]:=W[4,4]:=W[5,2]:=W[6,3]:=0.0;
W[4,2]:=W[4,3]:=W[5,3]:=W[5,4]:=W[6,2]:=W[6,4]:=1.0;
'FOR' I:=1 'STEP' 1 'UNTIL' ADD IODE[1:4] 'DO' KE[I]:=0.0;
'FOR' Z:=1 'STEP' 1 'UNTIL' NELEMT 'DO'
'BEGIN'
'FOR' I:=1 'STEP' 1 'UNTIL' 12 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 12 'DO' U[1,1]:=0.0;
'COMMENT' THE LOOP FOR THE NUMBER OF INT PTS IS CONSTRUCTED;
'FOR' U:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
STRDIS(W[U,2],W[U,3],U[U,4],B,XX,YY,DETJ,NODE,Z,ER,SOLID);
'FOR' J:=1 'STEP' 2 'UNTIL' 11 'DO'
'FOR' I:=J 'STEP' 2 'UNTIL' 11 'DO'
'BEGIN'
KE[J,I]:=KE[I,J]:=KE[I,J]+W[U,1]*C[1,J]*(C[NODE[Z,7],1]*B[1,I]+
C[NODE[Z,7],2]*B[2,I]+C[NODE[Z,7],4]*B[4,I])+
B[2,J]*(C[NODE[Z,7],2]*B[1,I]+C[NODE[Z,7],5]*B[2,I]
+C[NODE[Z,7],7]*B[4,I])+
B[4,J]*(C[NODE[Z,7],4]*B[1,I]+C[NODE[Z,7],7]*B[2,I]
+C[NODE[Z,7],10]*B[4,I]))*2*3.1415926*RAVG*0.5*DETJ;
KE[J,I+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]*(C[1,J]*(C[NODE[Z,7],5]*B[3,I+1]
+C[NODE[Z,7],4]*B[4,I+1])+
B[2,J]*(C[NODE[Z,7],6]*B[3,I+1]+C[NODE[Z,7],7]*B[4,I+1])+
B[4,J]*(C[NODE[Z,7],9]*B[3,I+1]+C[NODE[Z,7],10]*B[4,I+1]))
*2*3.1415926*RAVG*0.5*DETJ;
'END';
'FOR' J:=2 'STEP' 2 'UNTIL' 12 'DO'
'FOR' I:=J 'STEP' 2 'UNTIL' 12 'DO'
'BEGIN'
KE[J,I]:=KE[I,J]:=KE[I,J]+W[U,1]*C[3,I]*(C[NODE[Z,7],8]*B[3,I]+
C[NODE[Z,7],9]*B[4,I])+
B[4,J]*(C[NODE[Z,7],9]*B[3,I]+C[NODE[Z,7],10]*B[4,I])
*2*3.1415926*RAVG*0.5*DETJ;
'IF' I=12 'THEN' 'GOTO' L6 'ELSE'
KE[J,I+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]*(C[3,J]*(C[NODE[Z,7],5]*B[1,I+1]
+C[NODE[Z,7],6]*B[2,I+1]+C[NODE[Z,7],9]*B[4,I+1])+
B[4,J]*(C[NODE[Z,7],4]*B[1,I+1]+C[NODE[Z,7],7]*B[2,I+1]
+C[NODE[Z,7],10]*B[4,I+1]))*2*3.1415926*RAVG*0.5*DETJ;
L6: 'END';
'END';
'COMMENT' ASSEMBLY OF OVERALL STIFFNESS MATRIX AS A
ONE-DIMENSIONAL ARRAY;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
```



```

'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' V:=1,0 'DO'
'BEGIN' SUB1:=NODE[Z,1]*2-1;
        SUB2:=NODE[Z,J]*2-1;
        SUB3:=NODE[Z,1]*2;
'IF' SUB1 'LT' SUB2 'THEN' 'GOTO' LABA;
K[ADD[SUB1]-SUB1+SUB2]:=K[ADD[SUB1]-SUB1+SUB2]+KEI*2-1,J*2-V1;
LABA: 'IF' SUB3 'LT' SUB2 'THEN' 'GOTO' LABB;
K[ADD[SUB3]-SUB3+SUB2]:=K[ADD[SUB3]-SUB3+SUB2]+KEI*2,J*2-V1;
LABB: 'END';
'END';
'END' OF PROCEDURE ASSEMBLY;
'PROCEDURE' NODSTR(NSETF,NNODE,NELNBT,NODE,XX,YY,DETJ,D,C,STRDIS);
'VALUE' NSETF,NNODE,NELNBT; 'INTEGER' NSETF,NNODE,NELNBT;
'REAL' DETJ; 'INTEGER' 'ARRAY' NNODE;
'REAL' 'ARRAY' XX,YY,D,C; 'PROCEDURE' STRDIS;
'BEGIN' 'COMMENT' CALCULATION OF STRAINS AND STRESSES AT NODAL
        POINTS BY AVERAGING CONTRIBUTIONS FROM
        ADJACENT ELEMENTS;
'INTEGER' CNT1,CNT2,V,Z,U,J;
'REAL' SIGXX,SIGYY,SIGXY,EXX,EYY,EXY,ERTH,SIGRTH;
'REAL' 'ARRAY' STR[4],VE[6],E[6],E1[6],E2[6];
'FOR' V:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' Z:=2 'STEP' 1 'UNTIL' 4 'DO' NCV(Z):=0,C;
W[1,2]:=W[2,3]:=W[3,4]:=1,0;
W[4,2]:=W[4,3]:=W[5,3]:=W[5,4]:=W[6,2]:=W[6,4]:=0,5;
'FOR' V:=1 'STEP' 1 'UNTIL' NSETF 'DO'
'BEGIN'
WRITETEXT('('('5C20S')'STRAINS'('436')'STRESSES'('0')'NODE'('3S')'
ERR'('9S')'ETTH'('9S')'EZZ'('9S')'EZZ'('9S')'
SIGMA-R'('8S')'SIGMA-TTH'('8S')'SIGMA-Z'('3S')'SIGMA-PZ')');
'FOR' U:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'BEGIN' SIGXX:=SIGYY:=SIGXY:=EXX:=EYY:=EXY:=ERTH:=SIGRTH:=CNT2:=0,0;
'FOR' Z:=1 'STEP' 1 'UNTIL' NELNBT 'DO'
'BEGIN'
'FOR' CNT1 :=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN' 'IF' NODE[Z,CNT1]=0 'THEN'
'BEGIN'
STRDIS(W[CNT1,2],W[CNT1,3],W[CNT1,4],D,XX,YY,DETJ,NNODE,Z,CR,SOLID);
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN'
STR[J]:=(B[J,1]*Q[NODE[Z,1]*2-1,V]+B[J,2]*Q[NODE[Z,1]*2,V]
+B[J,3]*Q[NODE[Z,2]*2-1,V]+B[J,4]*Q[NODE[Z,2]*2,V]
+B[J,5]*Q[NODE[Z,3]*2-1,V]+B[J,6]*Q[NODE[Z,3]*2,V]
+B[J,7]*Q[NODE[Z,4]*2-1,V]+B[J,8]*Q[NODE[Z,4]*2,V]
+B[J,9]*Q[NODE[Z,5]*2-1,V]+B[J,10]*Q[NODE[Z,5]*2,V]
+B[J,11]*Q[NODE[Z,6]*2-1,V]+B[J,12]*Q[NODE[Z,6]*2,V]);
'END';
J:=NODE[Z,7];
SIGXX:=SIGXX+(C[J,1]*STR[1]+C[J,2]*STR[2]+C[J,3]*STR[3]+C[J,4]*
STR[4]);
SIGRTH:=SIGRTH+(C[J,2]*STR[1]+C[J,5]*STR[2]+
C[J,6]*STR[3]+C[J,7]*STR[4]);
SIGYY:=SIGYY+(C[J,3]*STR[1]+C[J,6]*STR[2]+
C[J,8]*STR[3]+C[J,9]*STR[4]);

```



```
SIGXY:=SIGXY+(C[J,4]*STR[1]+C[J,7]*STR[2]+  
C[J,9]*STR[3]+C[J,10]*STR[4]);  
EXX:=EXX+STR[1];  
ERTH:=ERTH+STR[2];  
EYY:=EYY+STR[3];  
EXY:=EXY+STR[4];  
CNT2:=CNT2+1;  
'END';  
'END';  
'END';  
SIGRTH:=SIGRTH/CNT2;  
SIGXX:=SIGXX/CNT2; SIGYY:=SIGYY/CNT2; SIGXY:=SIGXY/CNT2;  
ERTH:=ERTH/CNT2;  
EXX:=EXX/CNT2; EYY:=EYY/CNT2; EXY:=EXY/CNT2;  
NEWLINE(1);  
PRINT(U,2,0);  
PRINT(EXX,0,4);  
PRINT(ERTH,0,4);  
PRINT(EYY,0,4);  
PRINT(EXY,0,4); SPACE(10);  
PRINT(SIGXX,0,4);  
PRINT(SIGRTH,0,4);  
PRINT(SIGYY,0,4);  
PRINT(SIGXY,0,4);  
'END';  
'END';  
'END' OF PROCEDURE MODSTR;  
'PROCEDURE' ELESTR(NSETF,NELEMT,NODE,XX,YY,DETJ,C,STRDIS);  
'VALUE' NSETF,NELEMT; 'INTEGER' NSETF,NELEMT;  
'REAL' DETJ; 'INTEGER' 'ARRAY' NODE;  
'REAL' 'ARRAY' XX,YY,Q,C; 'PROCEDURE' STRDIS;  
'BEGIN' 'COMMENT' CALCULATION OF ELEMENT STRESSES AND STRAINS;  
'INTEGER' V,Z,J;  
'REAL' SIGXX,SIGYY,SIGXY,EXX,EYY,EXY,ERTH,SIGRTH;  
'REAL' 'ARRAY' STR[1:4],W[1,6,1:4],DET[4,1,12];  
'FOR' V:=1 'STEP' 1 'UNTIL' NSETF 'DO'  
'BEGIN'  
WRITETEXT('('('5020S')' STRAINS'('43S')' STRESSES'('0')' ELET'('3S')'  
ERR'('9S')' ETTT'('9S')' EZZ'('CS')' ETTT'('3S')'  
SIGMA-R'('8S')' SIGMA-TTH'('3S')' SIGMA-Z'('3S')' SIGMA-PZ'')));  
'FOR' Z:=1 'STEP' 1 'UNTIL' NELEMT 'DO'  
'BEGIN'  
SIGXX:=SIGYY:=SIGXY:=EXX:=EYY:=EXY:=ERTH:=SIGRTH:=0,V;  
STRDIS(0.3333,0.3333,0.3333,8,XX,YY,DETJ,NODE,Z,ER,SOLID);  
'FOR' J:=1 'STEP' 1 'UNTIL' 4 'DO'  
'BEGIN'  
STR[J]:= (B[J,1]*Q[NODE[Z,1]*2-1,V]+B[J,2]*Q[NODE[Z,1]*2,V]  
+B[J,3]*Q[NODE[Z,2]*2-1,V]+B[J,4]*Q[NODE[Z,2]*2,V]  
+B[J,5]*Q[NODE[Z,3]*2-1,V]+B[J,6]*Q[NODE[Z,3]*2,V]  
+B[J,7]*Q[NODE[Z,4]*2-1,V]+B[J,8]*Q[NODE[Z,4]*2,V]  
+B[J,9]*Q[NODE[Z,5]*2-1,V]+B[J,10]*Q[NODE[Z,5]*2,V]  
+B[J,11]*Q[NODE[Z,6]*2-1,V]+B[J,12]*Q[NODE[Z,6]*2,V]);  
'END';
```



```
J:=NUDELZ,Z];
SIGXX:=SIGXX+(C[J,1]*STR[1]+C[J,2]*STR[2]+C[J,3]*STR[3]+C[J,4]*
STR[4]);
SIGRTH:=SIGRTH+(C[J,2]*STR[1]+C[J,3]*STR[2]+
C[J,6]*STR[3]+C[J,7]*STR[4]);
SIGYY:=SIGYY+(C[J,3]*STR[1]+C[J,4]*STR[2]+
C[J,8]*STR[3]+C[J,9]*STR[4]);
SIGXY:=SIGXY+(C[J,4]*STR[1]+C[J,7]*STR[2]+
C[J,9]*STR[3]+C[J,10]*STR[4]);
EXX:=EXX+STR[1];
ERTH:=ERTH+STR[2];
EYY:=EYY+STR[3];
EXY:=EXY+STR[4];
NEWLINE(1);
PRINT(Z,2,0);
PRINT(EXX,0,4);
PRINT(ERTH,0,4);
PRINT(EYY,0,4);
PRINT(EXY,0,4); SPACE(10);
PRINT(SIGXX,0,4);
PRINT(SIGRTH,0,4);
PRINT(SIGYY,0,4);
PRINT(SIGXY,0,4);
'END';
'END';
'END' OF PROCEDURE ELESTR;
```

```
NJOB:=READ;
'FOR' COUNT:=1 'STEP' 1 'UNTIL' NJOB 'DO'
'BEGIN'
WRITETEXT('('('2C')'JOBNAME'-----)');
COPYTEXT('('END%OF%TITLE')');
NELEMT:=READ;
NNODE:=READ;
WRITETEXT('('('2C')'NO%OF%ELEMENTS'-----)');
PRINT(NELEMT,3,0);
WRITETEXT('('('2C')'NO%OF%NODES'-----)');
PRINT(NNODE,3,0);
NFREE:=NNODE*2;
NSETF:=READ;
PRNT:=READ; SOLID:=READ;
NMAT:=READ; EC:=READ; ER:=READ;
'BEGIN'
'INTEGER' BAND,NSPEC,Z,COMPA,NSETF;
'REAL' 'ARRAY' XX,YY[1:NNODE],ULX,VLY[1:NNODE],NSETF,
Q[1:NFREE,1:NSETF],C[1:NMAT,1:10],A[1:5],
VXS,VXF[1:(ER*2)+1],VYS,VYF[1:(ER*2)+1];
'INTEGER' 'ARRAY' NODE[1:NELEMT,1:1],KODE[1:NNODE,1:NSETF],ADD[0:NFREE]
MTNO[1:NELEMT],NSTEL[1:NMAT],STEL[1:NMAT,1:NELEMT];
NSETC:=READ;
'FOR' S:=1 'STEP' 1 'UNTIL' NSETC 'DO'
'BEGIN'
'IF' S=1 'THEN' 'BEGIN'
NSETF:=READ;
```



```
FEINPUT(C,ADD,XX,YY,HEREF,NODE,TYPE,KODE,NSPEC,NSETF,
ULX,VLY,NELEMT,NODE,NHAT,XS,XP,YS,YF,EC,ER,PSHP,VXS,VXF,VYS,VYF,SY,
MTNO,NSTEL,STEL);
ADDARRAY(NELEMT,NNODE,ADD,MODE);
'FOR' MATNO:=1 'STEP' 1 'UNTIL' NNODE 'DO'
CONSTREL (C,A,MATNO,CASE);
'END' 'ELSE'
'BEGIN' NNEWC:=READ;
      NSETF:=READ;
WRITETEXT('('('10C')'CONSTRAINTS--')');
PRINT(S,2,0);
'FOR' I:=1 'STEP' 1 'UNTIL' NNEWC 'DO'
'BEGIN' J:=READ;
      KODE[J,1]:=READ; ULX[J,1]:=READ; VLY[J,1]:=READ;
'END';
WRITETEXT('('('2C')'NODAL POINT DATA('204S')'NODE('15S')'XCOORD
('15S')'YCOORD('15S')'TYPE('15S')'X-DISP('15S')'Y-DISP('15S')'
ORXLOAD('4S')'ORLOAD')');
'FOR' I:=1 'STEP' 1 'UNTIL' NNEWC 'DO'
'BEGIN' NEWLINE(1); SPACE(3);
      PRINT(I,3,0); SPACE(5);
      PRINT(XX[I,1],0,3);
      PRINT(VLY[I,1],0,3); SPACE(5);
      PRINT(KODE[I,1],3,0); SPACE(2);
      PRINT(ULX[I,1],0,5);
      PRINT(VLY[I,1],0,3);
'END';
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' NNEWC 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' U[I,J]:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
      LOADING(KODE[I,1],ULX[I,1],VLY[I,1],0,1,1);
'IF' NSETF 'GT' 1 'THEN'
'BEGIN'
'FOR' I:=2 'STEP' 1 'UNTIL' NSETF 'DO'
'BEGIN'
NSPEC:=READ;
WRITETEXT('('('2C')'FORGENSET----')');
PRINT(I,3,0);
WRITETEXT('('('2C')'NODE('15S')'TYPE('8S')'X-DISP('7S')'Y-DISP
('1621S')'ORXLOAD('6S')'ORLOAD')');
'FOR' J:=1 'STEP' 1 'UNTIL' NSPEC 'DO'
'BEGIN'
      K:=READ; KODE[K,1]:=READ; ULX[K,1]:=READ; VLY[K,1]:=READ;
LOADING(KODE[K,1],ULX[K,1],VLY[K,1],0,1,1);
NEWLINE(2);
PRINT(K,3,0); SPACE(2);
PRINT(KODE[K,1],3,0); SPACE(2);
PRINT(ULX[K,1],0,4);
PRINT(VLY[K,1],0,4);
'END';
'END';
'END';
'BEGIN' 'REAL' 'ARRAY' K[1:ADDIFREE];
ASSEMBLY(NELEMT,K,XX,YY,DETJ,NODE,C,ADD,HEREF,NSETF,STDDIS);
'COMMENT' INTRODUCTION OF KINEMATIC CONSTRAINTS;
```



```
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'  
'BEGIN'  
'IF' KODE[I,1]=0 'THEN' 'GOTO' KC1;  
'IF' KODE[I,1]=2 'THEN' 'GOTO' KC2;  
BOUNCONST(CULX[I,1],2*I-1,Q,K,NFREE,1,ADD);  
'FOR' J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]=Q[I,1];  
'IF' KODE[I,1]=1 'THEN' 'GOTO' KC2;  
KC2: BOUNCONST(VLY[I,1],2*I,Q,K,NFREE,1,ADD);  
'FOR' J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]=Q[I,1];  
KC1: 'END';  
WRITETEXT('('('4C')'SYMBVSOL(NNODE,INS)');  
SYMBVSOL(K,K,ADD,Q,NFREE,NSETF,FAIL);  
'FOR' I:=1 'STEP' 1 'UNTIL' NSETF 'DO'  
'BEGIN' WRITETEXT('('('4C')'NODAL DISPLACEMENTS FOR % FORCE,SET)');  
PRINT(I,2,0);  
WRITETEXT('('('3CS')'NODE('5S')'R-DIRECTION('8S')'Z-DIRECTION  
'('8S')'NODE('5S')'R-DIRECTION('8S')'Z-DIRECTION)');  
W:=2*(NNODE/'2);  
'FOR' J:=2 'STEP' 2 'UNTIL' NNODE  
'BEGIN' NEWLINE(2); V:=2*(J-1);  
PRINT((J-1),3,0); SPACE(2); PRINT(Q[V-1,1],0,8); SPACE(4);  
PRINT(Q[V,1],0,8); SPACE(13); PRINT(J,3,0); SPACE(2);  
PRINT(Q[V+1,1],0,8); SPACE(2); PRINT(Q[V*2,1],0,8);  
'END';  
'IF' NNODE 'GT' W 'THEN'  
'BEGIN' NEWLINE(2); PRINT(NNODE,3,0); SPACE(2);  
V:=2*NNODE;  
PRINT(Q[V-1,1],0,8); SPACE(2); PRINT(Q[V,1],0,8);  
'END';  
'END';  
'IF' PRNT=1 'OR' PRNT=3 'THEN'  
NODSTR(NSETF,NNODE,NELENT,NODE,XX,YY,DETJ,Q,C,STRDIS);  
'IF' PRNT=2 'OR' PRNT=3 'THEN'  
ELESTR(NSETF,Q,NELENT,NODE,XX,YY,DETJ,C,STRDIS);  
'END';  
'END' OF CONSTRAINT LOOP;  
'END';  
FAIL: 'END' OF JOB LOOP;  
'END' OF PROGRAM;  
****
```

III) Input data instructions

A) Number of jobs to be solved.

B) For the first job:

1. The job name followed by end of title.
2. Number of elements.
3. Number of nodes.
4. Number of sets of forces⁽¹⁾.
5. Type of output⁽²⁾.
6. Type of the solid of revolution⁽³⁾.
7. Number of materials.
8. Number of element columns.
9. Number of element rows.
10. Number of sets of constraints.

C) For the first set of constraints.

1. Number of sets of forces
2. Type of boundary shape⁽⁴⁾.
3. r and z coordinates for the boundary shape⁽⁵⁾.
4. Number of specified nodes⁽⁶⁾.
5. For each specified node.
 - a) Node number
 - b) Kode⁽⁷⁾
 - c) Value of displacement or load in r direction
 - d) Value of displacement or load in z direction
6. IF number of materials >1 then:
 - a) The number of elements with different material properties for each differential material.
 - b) The element numbers of these elements.
7. Type of mesh⁽⁸⁾

8. For each material:
 - a) Type of material property⁽⁹⁾.
 - b) Elastic constants⁽¹⁰⁾.
 9. If the number of sets of forces for this set of constraints is >1 then:
 - a) Number of sets of forces.
 - b) For each force set:
 - Node number
 - Kode
 - value of prescribed load in r direction
 - value of prescribed load in z direction
- D) For subsequent sets of constraints:
1. Number of new constraints.
 2. Number of sets of forces in this set of constraints.
 3. For the first set of forces in the new set of constraints input C-9(b).
 4. For subsequent sets of forces input C-9.
- E) For the next job repeat from B.

Notes

- (1) The number of sets of forces is the maximum for any set of constraints.
- (2) Input 1 if nodal point stresses and strains are required.
2 if element stresses and strains are required.
3 if both 1 and 2 are required.
- (3) Input 0 if the structure is a hollow axisymmetric solid.
1 if the structure is a solid
- (4) Input 1 if the boundary shape is a square or a rectangle.
2 any other shape
- (5) If boundary shape is 1 input
 - a. r coordinate of the bottom left-hand corner
 - b. r coordinate of the bottom right-hand corner
 - c. z coordinate of the bottom left-hand corner
 - d. z coordinate of the upper left-hand cornerIf boundary shape is 2 input
 - a. left and right hand r coordinate of each main nodal row
 - b. lower and upper z coordinate of each main nodal column.
- (6) The specified nodes are those with prescribed loads and/or displacements.
- (7) Input 0 for prescribed loads in both r and z directions.
1 for prescribed displacement in r and load in z directions.
2 for prescribed load in r and z displacement in z directions.
3 for prescribed displacements in both r and z directions.
- (8) Input 0 if no symmetry with respect to a centroidal plane is required.
1 for a mesh symmetric with respect to the mid centroidal plane.

(9) Input 0 for isotropic material

1 for a stratified orthotropic material.

(10) Input $E \nu \mu E \nu$ for an isotropic material.

$E_1 \nu_1 \mu_2 E_2 \nu_2$ for a stratified orthotropic material.

IV Sample problem:

The input data required to solve the problem shown in Fig. (10.5)

is:

```
1
NBAR END OF TITLE
128,129,1,3,1,1,8,8,1,1,2
0,10,0,10,0,10,0,10,0,10,0,10,0,10,0,10,0,10
0,10,0,10,0,10,0,10,0,10,0,10,0,10,1.768,10,2.5,10
45
1,3,0,0
2,1,0,0
3,1,0,0
.
.
.
.
17,1,0,409
18,2,0,0
.
.
.
.
289,0,0,6136
0,0
30000000,0.3,12000000,30000000,0.3
****
```

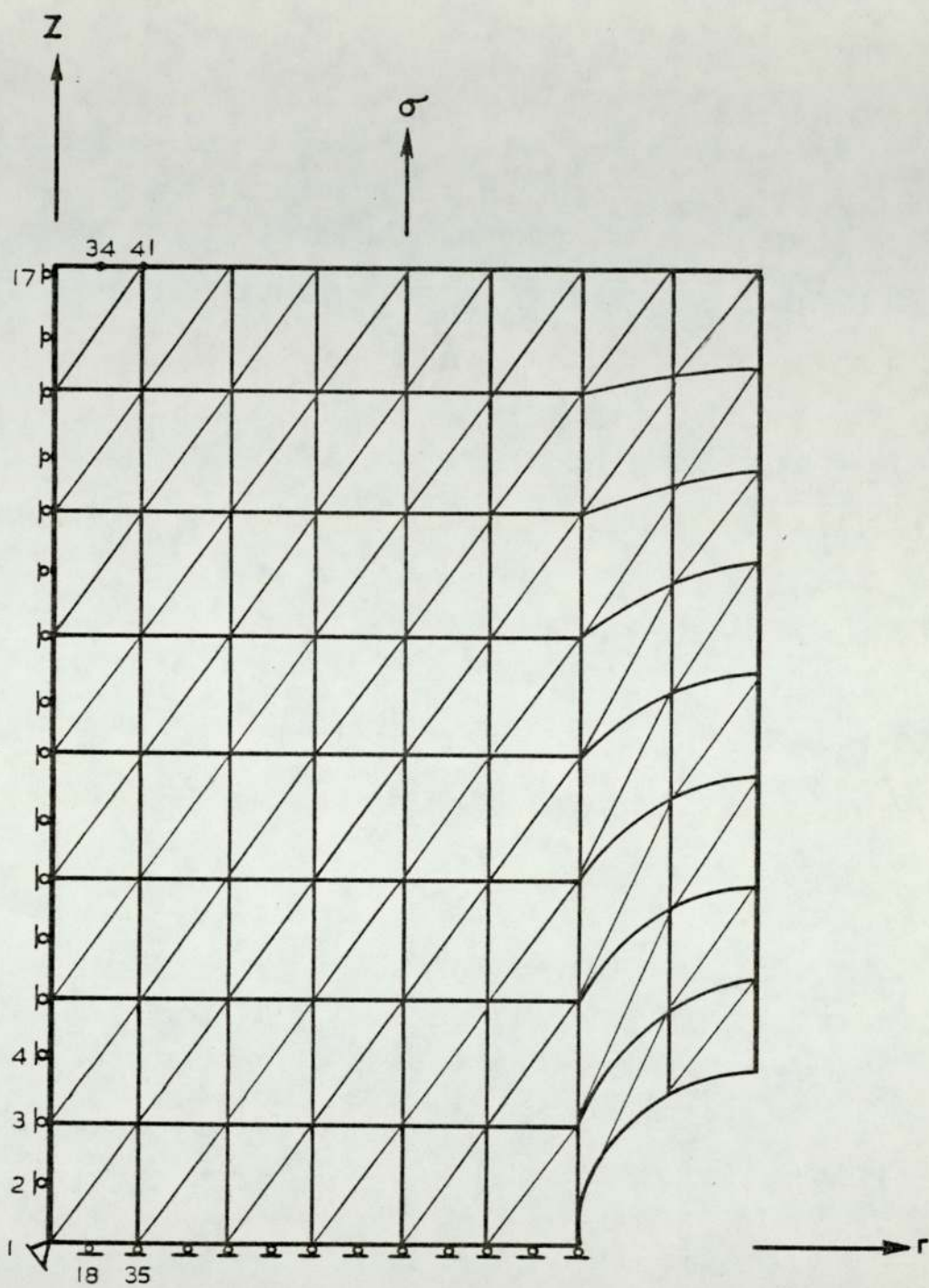


Fig. 10.5

8 NO OF ELEMENTS ---- 128

10 NO OF NODES ----- 289

12 NODAL CONNECTIONS HAVE BEEN READ

14

16 NODAL POINT DATA

18	NODE	R COORD	Z COORD	TYPE	R-DISP	Z-DISP
20					OR LOAD	OR LOAD
20	1	0.000& 0	0.000& 0	3	0.000& 0	0.000& 0
22	2	0.000& 0	6.250& -1	1	0.000& 0	0.000& 0
22	3	0.000& 0	1.250& 0	1	0.000& 0	0.000& 0
24	4	0.000& 0	1.875& 0	1	0.000& 0	0.000& 0
24	5	0.000& 0	2.500& 0	1	0.000& 0	0.000& 0
26	6	0.000& 0	3.125& 0	1	0.000& 0	0.000& 0
26	7	0.000& 0	3.750& 0	1	0.000& 0	0.000& 0
28	8	0.000& 0	4.375& 0	1	0.000& 0	0.000& 0
28	9	0.000& 0	5.000& 0	1	0.000& 0	0.000& 0
30	10	0.000& 0	5.625& 0	1	0.000& 0	0.000& 0
30	11	0.000& 0	6.250& 0	1	0.000& 0	0.000& 0
32	12	0.000& 0	6.875& 0	1	0.000& 0	0.000& 0
32	13	0.000& 0	7.500& 0	1	0.000& 0	0.000& 0
34	14	0.000& 0	8.125& 0	1	0.000& 0	0.000& 0
34	15	0.000& 0	8.750& 0	1	0.000& 0	0.000& 0
36	16	0.000& 0	9.375& 0	1	0.000& 0	0.000& 0
36	17	0.000& 0	1.000& 1	1	0.000& 0	4.000& 2
38	18	6.250& -1	0.000& 0	2	0.000& 0	0.000& 0
38	19	6.250& -1	6.250& -1	0	0.000& 0	0.000& 0
	20	6.250& -1	1.250& 0	0	0.000& 0	0.000& 0

ELEMENT DATA

ELEMENT		NODAL CONNECTIONS					MATERIAL
1	1	35	37	18	36	19	1
2	1	37	3	19	20	2	1
3	3	37	39	20	38	21	1
4	3	39	5	21	22	4	1
5	5	39	41	22	40	23	1
6	5	41	7	23	24	6	1
7	7	41	43	24	42	25	1
8	7	43	9	25	26	8	1
9	9	43	45	26	44	27	1
10	9	45	11	27	28	10	1
11	11	45	47	28	46	29	1
12	11	47	13	29	30	12	1
13	13	47	49	30	48	31	1
14	13	49	15	31	32	14	1
15	15	49	51	32	50	33	1
16	15	51	17	33	34	16	1
17	35	69	71	52	70	35	1
18	35	71	37	53	54	36	1
19	37	71	73	54	72	37	1
20	37	73	39	55	56	38	1
21	39	73	75	56	74	39	1
22	39	75	41	57	58	40	1
23	41	75	77	58	76	41	1
24	41	77	43	59	60	42	1
25	43	77	79	60	78	43	1
26	43	79	45	61	62	44	1
27	45	79	81	62	80	45	1
28	45	81	47	63	64	46	1
29	47	81	83	64	82	47	1
30	47	83	49	65	66	48	1

MATERIAL PROPERTIES: =

MATERIAL
NUMBER

ELASTIC PROPERTIES

E1

E2

V1

V2

G12

1

3.000E 7

3.000E 7

0.50

0.50

1.200E 7

SYNVBSOL BEGINS

NODAL DISPLACEMENTS FOR FORCE SET 1

NODE	R=DIRECTION	Z=DIRECTION	NODE	R=DIRECTION	Z=DIRECTION
1	0.00000000E 0	0.00000000E 0	2	0.00000000E 0	5.24858033E -6
3	0.00000000E 0	1.04756486E -5	4	0.00000000E 0	1.50902822E -5
5	0.00000000E 0	2.08472908E -5	6	0.00000000E 0	2.59435361E -5
7	0.00000000E 0	3.08855479E -5	8	0.00000000E 0	3.50570614E -5
9	0.00000000E 0	4.01412908E -5	10	0.00000000E 0	4.45092082E -5
11	0.00000000E 0	4.80573318E -5	12	0.00000000E 0	5.13413694E -5
13	0.00000000E 0	5.41089448E -5	14	0.00000000E 0	5.62889362E -5
15	0.00000000E 0	5.79525680E -5	16	0.00000000E 0	5.89345367E -5
17	0.00000000E 0	5.96346453E -5	18	-2.97292718E -7	0.00000000E 0
19	-6.04585170E -7	5.35143709E -6	20	-6.45254824E -7	1.05178300E -5

STRESSES

SIGMA=R		SIGMA=11d		SIGMA=Z		SIGMA=RZ	
1.0647%	2	1.2881%	2	3.2290%	2	1.0020%	-1
1.0593%	2	1.2765%	2	3.2085%	2	-4.7765%	-1
1.0230%	2	1.2089%	2	3.2067%	2	-1.2334%	0
9.8751%	1	1.2430%	2	3.1585%	2	-1.2455%	0
9.5066%	1	1.2243%	2	3.1302%	2	-1.5527%	0
9.1467%	1	1.1862%	2	3.0325%	2	-1.4421%	0
8.8103%	1	1.1519%	2	2.9585%	2	-1.6918%	0
8.4977%	1	1.1025%	2	2.8112%	2	-1.5914%	0
8.4912%	1	1.0226%	2	2.6895%	2	-1.6716%	0
6.2361%	1	9.2210%	1	2.4556%	2	-1.7249%	0
6.5375%	1	9.2245%	1	2.2210%	2	-1.5528%	0
6.0231%	1	8.4637%	1	1.9555%	2	-1.2462%	0
8.5071%	1	7.4899%	1	1.6459%	2	-1.3066%	0
8.1497%	1	6.5180%	1	1.3577%	2	-2.2461%	0
7.3568%	1	5.0142%	1	9.3572%	1	-2.4263%	-1
3.0253%	1	3.5005%	1	6.6428%	1	-3.6607%	0
1.5205%	1	1.5205%	1	3.5479%	1	-3.2845%	0
1.0704%	2	1.2705%	2	3.2284%	2	6.7065%	-2
1.0787%	2	1.2787%	2	3.2505%	2	1.5726%	0
1.0575%	2	1.2476%	2	3.1696%	2	2.3219%	0
1.0121%	2	1.2447%	2	3.2115%	2	4.5121%	0
9.9106%	1	1.2016%	2	3.1210%	2	3.5634%	0
9.2594%	1	1.1810%	2	3.0909%	2	9.5294%	0
6.8255%	1	1.1296%	2	2.9571%	2	1.0758%	1

STRAINS

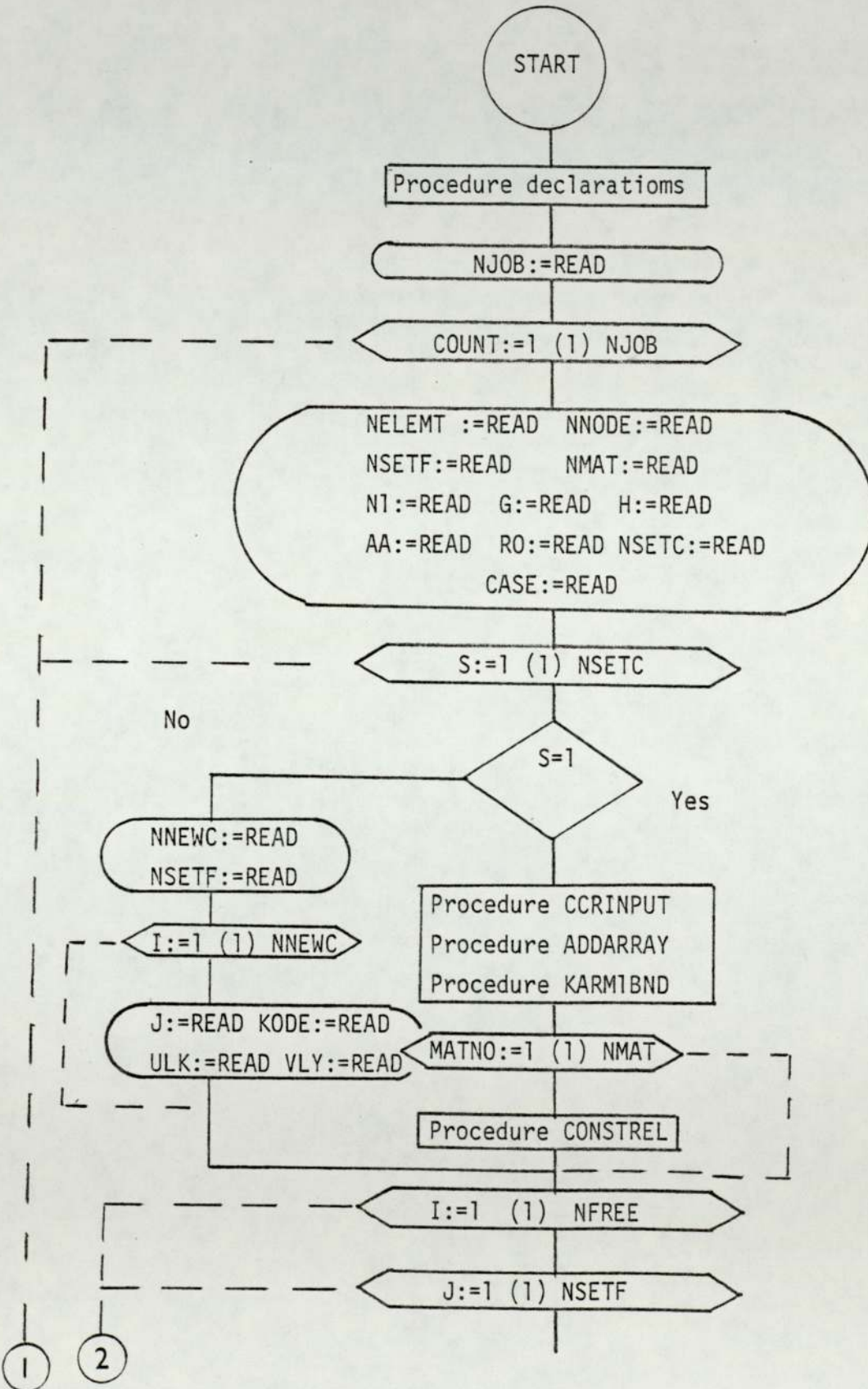
NODE	ERR	ETHH	EZZ	EZZ				
1	-9.6803%	-7	0.0000%	0	8.4104%	-6	8.3499%	-9
2	-1.0177%	-6	0.0000%	0	8.3805%	-6	-7.5156%	-6
3	-1.0654%	-6	0.0000%	0	8.3969%	-6	-1.0278%	-7
4	-1.1107%	-6	0.0000%	0	8.2973%	-6	-1.0363%	-7
5	-1.1856%	-6	0.0000%	0	8.2591%	-6	-1.2939%	-7
6	-1.1768%	-6	0.0000%	0	8.0366%	-6	-1.2018%	-7
7	-1.1735%	-6	0.0000%	0	7.8288%	-6	-1.4098%	-7
8	-1.0347%	-6	0.0000%	0	7.4046%	-6	-1.3262%	-7
9	-8.8164%	-7	0.0000%	0	6.9632%	-6	-1.3930%	-7
10	-6.0037%	-7	0.0000%	0	6.3528%	-6	-1.4374%	-7
11	-2.9755%	-7	0.0000%	0	5.6271%	-6	-1.2940%	-7
12	6.0395%	-8	0.0000%	0	4.8413%	-6	-1.6218%	-7
13	4.4078%	-7	0.0000%	0	3.8867%	-6	-1.0888%	-7
14	7.0707%	-7	0.0000%	0	3.0589%	-6	-1.2134%	-7
15	1.0151%	-6	0.0000%	0	1.8820%	-6	-2.0219%	-8
16	6.6085%	-7	0.0000%	0	1.3617%	-6	-3.0506%	-7
17	0.0000%	0	0.0000%	0	8.7853%	-7	-2.7371%	-7
18	-9.3109%	-7	-6.3711%	-8	8.4205%	-6	5.5869%	-9
19	-9.3348%	-7	-6.6792%	-8	8.4776%	-6	1.3105%	-7
20	-9.7886%	-7	-6.8612%	-8	8.5471%	-6	1.9349%	-7
21	-1.0824%	-6	-7.4444%	-8	8.4480%	-6	3.7600%	-7
22	-1.1190%	-6	-7.6773%	-8	8.2406%	-6	4.7195%	-7
23	-1.1854%	-6	-8.0219%	-8	8.1960%	-6	7.9412%	-7
24	-1.1351%	-6	-7.6676%	-8	7.8418%	-6	9.1514%	-7

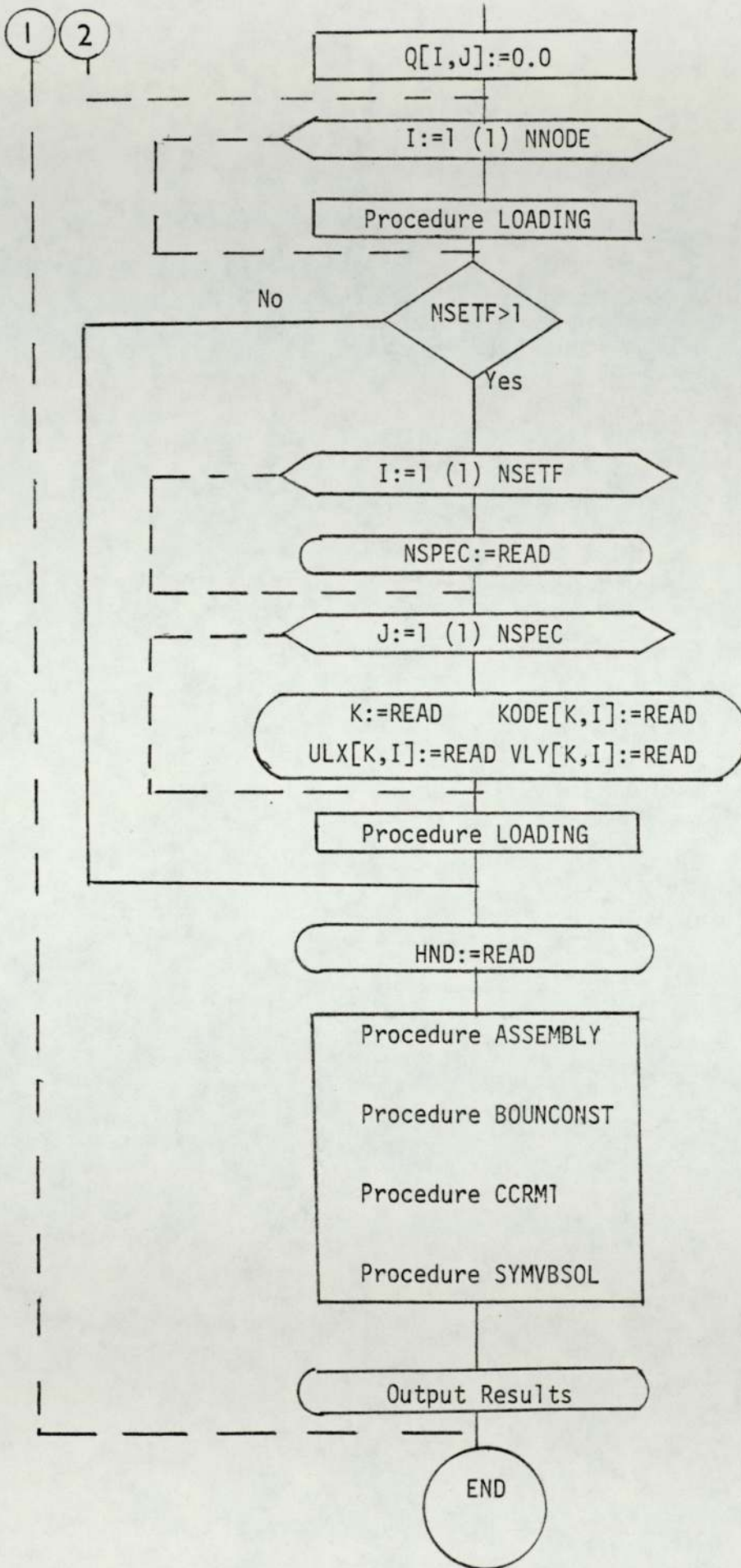
STRAINS								
ELET	ERR	ETTH	EZZ	ERZ				
1	-9.2708%	-7	-8.6688%	-3	3.5698%	-5	-1.4474%	-1
2	-9.7098%	-7	-4.5669%	-3	3.3240%	-6	1.5850%	-7
3	-1.0387%	-6	-9.7137%	-8	3.5412%	-5	3.0367%	-7
4	-1.0015%	-6	-5.0613%	-3	8.2448%	-6	3.0199%	-7
5	-1.1670%	-6	-1.0726%	-7	3.3476%	-6	7.7022%	-7
6	-1.1527%	-6	-5.2860%	-3	7.9442%	-6	5.7499%	-7
7	-1.1107%	-6	-1.0045%	-7	7.6036%	-6	1.4341%	-6
8	-9.8765%	-7	-4.4789%	-3	7.2614%	-6	9.1561%	-7
9	-7.7172%	-7	-6.7648%	-3	6.8079%	-6	2.1830%	-6
10	-5.4224%	-7	-2.3803%	-3	6.1317%	-6	1.2667%	-6
11	-2.0380%	-7	-1.2423%	-3	3.5871%	-5	2.3754%	-6
12	8.7098%	-8	5.8320%	-9	4.6053%	-6	1.5877%	-6
13	3.8020%	-7	4.6149%	-6	3.6642%	-6	3.4756%	-6
14	6.3707%	-7	3.1724%	-3	2.8384%	-6	1.8547%	-6
15	5.3101%	-7	6.6953%	-3	2.0050%	-6	3.0266%	-6
16	4.4052%	-7	2.0028%	-3	1.2213%	-6	1.8598%	-6
17	-7.7187%	-7	-1.8984%	-7	9.1100%	-6	3.2634%	-6
18	-7.4206%	-7	-1.6350%	-7	8.7441%	-6	3.6140%	-7
19	-1.0466%	-6	-2.2580%	-7	9.2137%	-6	7.1337%	-7
20	-9.7775%	-7	-1.9145%	-7	6.7490%	-6	1.0027%	-6
21	-1.3636%	-6	-2.6524%	-7	9.1665%	-6	1.7278%	-6
22	-1.1675%	-6	-2.0975%	-7	8.5441%	-6	2.1225%	-6
23	-1.4409%	-6	-2.5952%	-7	6.7147%	-6	3.5016%	-6

STRESSES							
SIGMA-R		SIGMA-TTH		SIGMA-Z		SIGMA-RZ	
1.0938%	2	1.2878%	2	0.3359%	2	-1.7285%	-1
1.0407%	2	1.2542%	2	6.3792%	2	1.6620%	0
1.0420%	2	1.2593%	2	6.3223%	2	3.5453%	0
9.7741%	1	1.2176%	2	6.2727%	2	3.5238%	0
9.5491%	1	1.1995%	2	6.3198%	2	9.4211%	0
9.0031%	1	1.1541%	2	6.0083%	2	6.8999%	0
6.0463%	1	1.1178%	2	2.9010%	2	1.7449%	1
8.5017%	1	1.0678%	2	5.5153%	2	1.0937%	1
8.3497%	1	1.0173%	2	5.2199%	2	2.6197%	1
8.3816%	1	9.5780%	1	4.7607%	2	1.5200%	1
8.4792%	1	8.9209%	1	4.2764%	2	3.4217%	1
8.3361%	1	8.1465%	1	3.7511%	2	1.9052%	1
7.9916%	1	7.2209%	1	5.1153%	2	4.1710%	1
7.5403%	1	6.1434%	1	2.5186%	2	2.2256%	1
5.7306%	1	4.6597%	1	1.6143%	2	4.5626%	1
3.9274%	1	2.9570%	1	1.1424%	2	2.2317%	1
1.2322%	2	1.5665%	2	7.0580%	2	6.5135%	-1
1.1854%	2	1.3189%	2	6.7774%	2	4.3368%	0
1.1529%	2	1.3224%	2	7.0405%	2	8.5869%	0
1.0863%	2	1.2677%	2	6.6949%	2	1.2033%	1
9.8911%	1	1.2430%	2	6.8531%	2	2.5133%	1
9.7099%	1	1.1920%	2	6.4605%	2	2.5470%	1
6.6148%	1	1.1541%	2	6.4931%	2	4.5214%	1

10.4.2 Mode I fracture program

1) Program flow chart





II) Program listing

```
'PROGRAM' (AXXX)
'INPUT' O=CR0
'OUTPUT' O=LPO
'EXTENDED DATA'
'EXTENDED'
'TRACE' 2
'BEGIN'
'INTEGER' NELEMT, NNODE, NS, TR, NRE, I, J, V, U, Z, CASE, NSETC, NINNO, S,
NJOB, COUNT, BAND, NIAT, NATI0;
'REAL' DELTA, DETJ, THICK, T, RAVG, RAVG;
'PROCEDURE' STRDIS(L1, L2, L3, B, X, Y, U, I, Z);
'VALUE' L1, L2, L3, Z;
'INTEGER' Z;
'REAL' L1, L2, L3, U;
'INTEGER' 'ARRAY' N;
'ARRAY' X, Y, B;
'BEGIN'
'INTEGER' I, V;
'REAL' CHANGE;
'REAL' 'ARRAY' P(1:2, 1:6), J(1:2, 1:2), NL(1:6);
'COMMENT' THIS PROCEDURE EVALUATES THE JACOBIAN J ITS DETERMINANT
AND THE STRAIN - ISP ARRAY B;
J[1, 1]=X[N[Z, 1]]*(4*L1-1)+X[N[Z, 3]]*(4*L1+4*L2-3)+4*L2*X[N[Z, 4]]-
4*L2*X[N[Z, 5]]+4*X[N[Z, 6]]*(1-2*L1*L2);
J[1, 2]=Y[N[Z, 1]]*(4*L1-1)+Y[N[Z, 3]]*(4*L1+4*L2-3)+4*L2*Y[N[Z, 4]]-
4*L2*Y[N[Z, 5]]+4*Y[N[Z, 6]]*(1-2*L1*L2);
J[2, 1]=X[N[Z, 2]]*(4*L2-1)+X[N[Z, 3]]*(4*L1+4*L2-3)+4*L1*X[N[Z, 4]]+
4*X[N[Z, 5]]*(1-L1-2*L2)+4*L1*X[N[Z, 6]];
J[2, 2]=Y[N[Z, 2]]*(4*L2-1)+Y[N[Z, 3]]*(4*L1+4*L2-3)+4*L1*Y[N[Z, 4]]+
4*Y[N[Z, 5]]*(1-L1-2*L2)+4*L1*Y[N[Z, 6]];
'COMMENT' U REPLACES DETJ;
U:=J[1, 1]*J[2, 2]-J[1, 2]*J[2, 1];
CHANGE:=J[1, 1];
J[1, 1]=J[2, 2]/U;
J[1, 2]=-J[1, 2]/U;
J[2, 1]=-J[2, 1]/U;
J[2, 2]=CHANGE/U;
NL[1]=L1*(2*L1-1);
NL[2]=L2*(2*L2-1);
NL[3]=L3*(2*L3-1);
NL[4]=4*L1*L2;
NL[5]=4*L2*L3;
NL[6]=4*L3*L1;
RMEAN:=(X[N[Z, 1]]+X[N[Z, 2]]+X[N[Z, 3]]+X[N[Z, 4]]+
X[N[Z, 5]]+X[N[Z, 6]])/6;
'IF' Z=109 'OR' Z=111 'OR' Z=113 'THEN'
RAVG:=RMEAN
'ELSE'
'BEGIN'
```



```

RAVG:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
RAVG:=(RAVG+(X(LN2, I)+X(L2, I)))/2;
'END';
'COMMENT' THE P ARRAY HAS THE BIP COEFFS OF RCII FOR L1 AND L2;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' V:=1,2 'DO' PIV, I:=0.0;
'FOR' I:=1,2 'DO'
'BEGIN' P(I,1):=J(I,1)*(4-L1-1);
P(I,2):=J(I,2)*(4-L2-1);
P(I,3):=J(I,1)*(1+4*L1)*J(I,2)*(1+4*L2);
P(I,4):=4*(L2+J(I,1)+L1+J(I,2));
P(I,5):=4*(J(I,2)+L1-L2*(J(I,2)+J(I,1)));
P(I,6):=4*(J(I,1)-L3-L1*(J(I,1)+J(I,2)));
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' 12 'DO'
'FOR' V:=1,2,3,4 'DO' PIV, I:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN'
B[1, (I*2-1)]:=B[4, (I*2)]:=P(I,1);
B[4, (I*2+1)]:=B[3, (I*2)]:=P(I,2);
B[2, (I*2-1)]:=NL[I]/RAVG;
'END';
'END' OF STRDIS;
'PROCEDURE' ADDARRAY(NFILE, NNODE, ADD, NNODE);
'INTEGER' NLEMT, NNODE;
'INTEGER' 'ARRAY' NNODE, ADD;
'BEGIN' 'INTEGER' W, CH, I, ADDTEMP;
'FOR' W:=1 'STEP' 1 'UNTIL' NLEMT 'DO'
'BEGIN' CH:=NNODE[W,1];
'FOR' I:=2 'STEP' 1 'UNTIL' 6 'DO'
'IF' NNODE[U,1] LT CH 'THEN' CH:=NNODE[W,1];
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN' ADDTEMP:=NNODE[U,1]-CH+1;
'IF' ADDTEMP GT ADD+NNODE[U,1]*2 'THEN' ADD+NNODE[W,1]*2:=ADDTEMP;
'END';
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'BEGIN' CH:=2+ADD[2*I]; U:=2+I;
ADD[W-1]:=ADD[U-2]+CH-1;
ADD[W]:=ADD[U-1]*2;
'END';
'END' OF PROCEDURE ADDARRAY;
'PROCEDURE' BOUNCONST(U, N, R, AK, NEQ, F, A);
'VALUE' U, N, NEQ, F;
'REAL' U;
'INTEGER' N, NEQ, F;
'ARRAY' R, AK;
'INTEGER' 'ARRAY' A;
'BEGIN' 'INTEGER' M, K, CJ;
'IF' N=1 'THEN' CJ:=1 'ELSE' CJ:=-(A[N]=A(N-1))+1;
'FOR' K:=CJ 'STEP' 1 'UNTIL' N 'DO'
'BEGIN'
R[K, F]:=R[K, F]-AK[A[N]-N+K]*1;
AK[A[N]-N+K]:=0.0;

```



```
'END';
'IF' N+1 'GT' NEQ 'THAT' 'BAND' 'LZ';
'FOR' K:=N+1 'STEP' 1 'UNTIL' NEQ '00';
'BEGIN' CJ:=K-(AK[K]-AK[K-1])*1;
      'IF' CJ 'LE' N 'THEN'
      'BEGIN'
          RK[F1]:=RK[F1]-AK[A[K]-K+1]*1;
          AK[A[K]-K-1]=0,0;
      'END' 'ELSE'
'END';
LZ: AK[A[N]]:=1.0;
R[N,F1]:=U;
'END' OF BOUNCONST;
'PROCEDURE' KARMBAND (N1, NFREE, ADD, BAND);
'VALUE' N1, NFREE;
'INTEGER' N1, NFREE, BAND;
'INTEGER' 'ARRAY' ADD;
'BEGIN'
'INTEGER' I, J, K, V;
V:=NFREE-2*N1; K:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' V 'DO';
'BEGIN'
J:=I+2*N1-ADD[I+2*N1]-ADD[I+2*N1-1]+1;
'IF' J 'LT' 2*N1 'THEN' J:=2*N1-1;
'IF' I=1 'THEN' BAND:=1 'ELSE'
BAND:=I+2*N1-J+K+1;
K:=BAND;
'END'
BAND:=BAND+2*(NFREE-2*N1)-3;
'IF' ADD[NFREE] 'GT' BAND 'THEN' BAND:=ADD[NFREE];
'END' OF PROCEDURE KARMBAND;
'PROCEDURE' LOADING(A, B, C, D, E, F);
'VALUE' A, B, C, E, F;
'INTEGER' A, E, F;
'REAL' B, C;
'ARRAY' D;
'BEGIN'
'INTEGER' K;
'IF' A=3 'THEN' 'GOTO' KLAB1;
K:=2*F;
'IF' A=1 'THEN' 'GOTO' KLAB2;
D[K-1, F1]:=D[K-1, F1]*B;
'IF' A 'NE' 0 'THEN' 'GOTO' KLAB1;
KLAB2: D[K, F1]:=D[K, F1]*C;
KLAB1: 'END' OF LOADING;
'PROCEDURE' SYMBSOL(A, L, S, DIMENSIONS: (L, R), ALLURE EXIT: (FAIL));
'VALUE' N, R; 'ARRAY' A, L, S; 'INTEGER' 'ARRAY' S; 'INTEGER' N, R;
'LABEL' FAIL;
'BEGIN'
'INTEGER' G, H, I, J, K, P, Q, T, U, V;
'REAL' Y;
H:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO';
'BEGIN'
T:=I+H-S[I]+1; G:=H+1;
P:=S[T]-1;
'FOR' J:=T 'STEP' 1 'UNTIL' I-1 'DO'
```



```
'BEGIN'  
Q:=P+1; H:=H+1;  
P:=S(J); K:=J+Q-2;  
V:=H-P; U:=G;  
Y:=A(H);  
'IF' K 'GT' T 'THEN' M:=U+K-T;  
'FOR' U:=U 'STEP' 1 'UNTIL' H 'DO'  
Y:=Y-L[U]*L[U-V];  
Y:=Y/L[H-V]; L(H)=Y;  
'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'  
B(I,M):=B(I,M)-B(J,M)*Y;  
'END' J;  
Y:=A(H+1);  
'FOR' U:=G 'STEP' 1 'UNTIL' H 'DO'  
Y:=Y-L[U]*2;  
'IF' Y 'LE' 0 'THEN' 'GOTO' FAIL;  
H:=H+1; Y:=SQRT(Y);  
L(H)=Y;  
'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'  
B(I,M):=B(I,M)/Y;  
'END' I;  
'COMMENT' REDUCTION COMPLETE;  
'FOR' I:=N 'STEP' -1 'UNTIL' 2 'DO'  
'BEGIN'  
Y:=L(H);  
'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'  
B(I,M):=B(I,M)/Y;  
'IF' I=1 'THEN' 'GOTO' COMPLETE;  
J:=I; P:=S(I-1);  
'FOR' H:=H-1 'STEP' -1 'UNTIL' P+1 'DO'  
'BEGIN'  
J:=J-1; Y:=L(H);  
'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'  
B(J,M):=B(J,M)-B(I,M)*Y;  
'END' H;  
H:=P;  
COMPLETE: 'END' I;  
'END' SYMBSQL;  
'PROCEDURE' CORR(N1,RO,NG,NSSET,CASE,CP,AA,CR,AA);  
'VALUE' N1,RO,NG,NSSET,CASE,CP,AA;  
'INTEGER' N1,NG,NSSET,CASE,IND;  
'REAL' RO,CR,AA;  
'INTEGER' 'ARRAY' ADD;  
'REAL' 'ARRAY' K,Q,CH;  
'BEGIN' 'INTEGER' IA,CI,DI,UJ,CV,IA;  
'REAL' UI,UJ,KT11,KT12,KT22,THETA,KAP,G,UI,UIT;  
'ARRAY' KT(1:2,3:4*H+2);  
'FOR' I:=1,2 'DO'  
CHEI:=READ;  
G:=0.5*CM(1)/(1+CM(2)); CU:=CM(2);  
WRITETEXT('('('20')' THIS PROCEDURE MODIFIES THE OVERALL  
STIFFNESS MATRIX FOR A SINGLE TIP UNDER A NODE 1 LOADING ('20')'  
NUMBER OF NODAL POINTS ON CORE = %)' );  
PRINT(N1,3,0);  
WRITETEXT('('('C')' CORE RADII = %)' ); PRINT(RO,0,5);  
WRITETEXT('('('C')' DIRECTION OF CRACK TIP = %)' ); PRINT(HND,3,0);
```



```
'FOR' I:=1,2 'DO'  
'FOR' J:=3 'STEP' 1 'UNTIL' 4*N1+2 'DO' KT(CI, J):=0.0;  
'IF' CASE=0 'THEN' KAP:=(C=0)/(C+0) 'ELSE' KAP:=3-4*U;  
'IF' HND=1 'THEN'  
KT11:=(2*KAP-1)*((3.14159/2)*RO)*2*CF/(16*G)  
'ELSE'  
KT11:=(2*KAP-1)*((3.14159/2)*RO)*2*AA/(16*G);  
KT12:=0.0;  
'FOR' I:=1 'STEP' 1 'UNTIL' 2*N1 'DO'  
'BEGIN' IA:=I-2*ENTIER(I/2);  
'IF' IA=0 'THEN'  
'BEGIN' THETA:=(I/2-1)*3.14159/(N1-1);  
UI:=(2*RO)*0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(8*G);  
'END' 'ELSE'  
'BEGIN' THETA:=((I+1)/2-1)*3.14159/(N1-1);  
UI:=(2*RO)*0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(8*G);  
'END';  
CI:=I-ADD[I]+ADD[I-1]+1;  
'IF' HND=2 'AND' IA=1 'THEN' UI:=UI 'ELSE' UI:=UI;  
'FOR' J:=CI 'STEP' 1 'UNTIL' 2*U1 'DO'  
'BEGIN' 'IF' J 'LE' I 'THEN'  
'BEGIN' DI:=I; DJ:=J;  
'GOTO' MSM1;  
'END' 'ELSE'  
'BEGIN' CJ:=J-ADD[J]+ADD[J-1]+1;  
'IF' CJ 'GT' I 'THEN' 'GOTO' MSM2;  
DI:=J; DJ:=I;  
'END';  
MSM1: JA:=J-2*ENTIER(J/2);  
'IF' JA=0 'THEN'  
'BEGIN' THETA:=(J/2-1)*3.14159/(N1-1);  
UJ:=(2*RO)*0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(8*G);  
'END' 'ELSE'  
'BEGIN' THETA:=((J+1)/2-1)*3.14159/(N1-1);  
UJ:=(2*RO)*0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(8*G);  
'END';  
'IF' HND=2 'AND' JA=1 'THEN' UJ:=UJ;  
KT11:=KT11+UI*UJ*K[ADD[DI]-DI+DJ];  
'IF' IA=1 'THEN' KT12:=KT12+UJ*K[ADD[DI]-DI+DJ];  
MSM2: 'END';  
'END';  
KT22:=0.0;  
'FOR' I:=1 'STEP' 2 'UNTIL' 2*N1-1 'DO'  
'BEGIN' CI:=I-ADD[I]+ADD[I-1]+1;  
IA:=CI-2*ENTIER(CI/2);  
'IF' IA=1 'THEN' 'GOTO' MSM5 'ELSE' CI:=CI+1;  
MSM5: 'FOR' J:=CI 'STEP' 1 'UNTIL' 2*N1-1 'DO'  
'BEGIN' 'IF' J 'LE' I 'THEN' 'BEGIN' DI:=I; DJ:=J; 'GOTO' MSM6; 'END'  
CJ:=J-ADD[J]+ADD[J-1]+1;  
'IF' CJ 'GT' I 'THEN' 'GOTO' MSM7;  
DI:=J; DJ:=I;  
MSM6: KT22:=KT22+K[ADD[DI]-DI+DJ];  
MSM7: 'END';  
'END';
```



```
'FOR' J:=1+2*N1 'STEP' 1 'UNTIL' 6*N1 'DO'  
'BEGIN' CJ:=J-ADD[J]-ADD[J-1]+1;'FOR' I:=CJ 'STEP' 1 'UNTIL' 2*N1 'DO'  
'BEGIN' 'IF' I 'LE' J 'THEN' 'BEGIN' DI:=I; DJ:=I; 'GOTO' MSM8; 'END'  
CI:=I-ADD[I]+ADD[I-1]+1;  
'IF' CI 'GT' J 'THEN' 'GOTO' MSM9;  
DI:=I; DJ:=J;  
MSM8: IA:=I-2*ENTIER(CI/2);  
'IF' IA=0 'THEN' 'BEGIN' THETA:=(I/2-1)+3.14159/(N1-1);  
UI:=(2*RO)↑0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(6*G);  
'END' 'ELSE' 'BEGIN'  
THETA:=(CI+1)/2-1)*3.14159/(N1-1);  
UI:=(2*RO)↑0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(3*G);  
'END';  
'IF' HND=2 'AND' IA=1 'THEN' UI:=-UI;  
KT[1,J-2*N1+2]:=KT[1,J-2*N1+2]+UI-K[ADD[D]-DI+DJ];  
MSM9: 'END';  
'END';  
'FOR' J:=1+2*N1 'STEP' 1 'UNTIL' 6*N1 'DO'  
'BEGIN'  
CJ:=J-ADD[J]+ADD[J-1]+1;  
JA:=CJ-2*ENTIER(CJ/2);  
'IF' JA=1 'THEN' 'GOTO' MSM10 'ELSE' CJ:=CJ+1;  
MSM10: 'FOR' I:=CJ 'STEP' 2 'UNTIL' 2*N1-1 'DO'  
'BEGIN' 'IF' I 'LE' J 'THEN' 'BEGIN' DI:=J; DJ:=I; 'GOTO' MSM11; 'END';  
CI:=I-ADD[I]+ADD[I-1]+1;  
'IF' CI 'GT' J 'THEN' 'GOTO' MSM12;  
DI:=I; DJ:=J;  
MSM11: KT[2,J-2*N1+2]:=KT[2,J-2*N1+2]+K[ADD[D]-DI+DJ];  
MSM12: 'END';  
'END';  
'FOR' I:=1 'STEP' 1 'UNTIL' NG-2*N1 'DO'  
'BEGIN' CI:=I+2*N1-(ADD[I+2*N1]-ADD[I+2*N1-1])+1;  
'IF' CI 'LE' 2*N1 'THEN' CI:=2*N1+1;  
'IF' I=1 'THEN' ADD[I]:=1 'ELSE' ADD[I]:=I+2*N1-CI+ADD[I-1]+1;  
'FOR' J:=CI 'STEP' 1 'UNTIL' I+2*N1 'DO'  
K[ADD[I]-I+J-2*N1]:=K[ADD[I+2*N1]-I-2*N1+J];  
'END';  
'FOR' I:=1 'STEP' 1 'UNTIL' NG-2*N1 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' NSRTE 'DO'  
Q[I,J]:=Q[I+2*N1,I];  
'FOR' I:=1 'STEP' 1 'UNTIL' NSRTE 'DO'  
Q[NG-2*N1+1,I]:=Q[NG-2*N1+2,I]:=0.0;  
ADD[NG-2*N1+1]:=ADD[NG-2*N1]+NG-2*N1+1;  
ADD[NG-2*N1+2]:=ADD[NG-2*N1+1]+NG-2*N1+4;  
'FOR' I:=1 'STEP' 1 'UNTIL' NG-2*N1 'DO'  
K[ADD[NG-2*N1+1]-NG+2*N1-1+I]:=K[ADD[NG-2*N1+2]-NG+2*N1-2+I]:=0.0;  
K[ADD[NG-2*N1+2]-1]:=KT12;  
K[ADD[NG-2*N1+1]]:=KT22;  
K[ADD[NG-2*N1+2]]:=KT11;  
'FOR' I:=3 'STEP' 1 'UNTIL' 4*N1+2 'DO'  
K[ADD[NG-2*N1+1]-NG+2*N1+1-3]:=KT[2,I];  
'FOR' I:=3 'STEP' 1 'UNTIL' 4*N1+2 'DO'  
K[ADD[NG-2*N1+2]-NG+2*N1+1-4]:=KT[1,I];  
'END' OF PROCEDURE CCRM1;
```



```
'PROCEDURE' CORINPOTCADD(X,X,YY,YY,NEEF,ENODE,NSETF,KODE,NSPEC,  
NMAT,HTNO,NSTEL,STEL,ULX,ULY,NELENT,NODE,I1,G,AA,R0):  
'VALUE' NFREE,NNODE,NSETF,NSPEC,NELENT;  
'INTEGER' NFREE,NNODE,NSETF,NSPEC,NELENT,I1,NMAT; 'REAL'G,AA,R  
'INTEGER' 'ARRAY' ADD,KODE,NSPEC,HTNO,NSTEL,STEL;  
'REAL' 'ARRAY' XX,YY,ULX,ULY;  
'BEGIN''INTEGER'I,J,N,B,L,0;  
'REAL'XS,XF,YS,YF,DELTA,DELTA,DELTA,DELTA;  
'FOR' I:=0 'STEP' 2 'UNTIL' NFREE 'DO' ADD(I):=0;  
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO' XX(I):=0.00001;  
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' KODE(I,J):=0;  
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO'  
ULX(I,J):=VLY(I,J):=0.0;  
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'  
'BEGIN'  
B:=I-1; F:=(I+1)/2;  
'FOR' J:=1 'STEP' 1 'UNTIL' NS 'DO'  
'BEGIN'  
TH:=(J-1)*(3.14159/18);  
XX[(B*N1)+J]:=G-AA-(F*R0*COS(TH));  
YY[(B*N1)+J]:=F*R0*SIN(TH);  
'END';  
'END';  
'FOR' I:=7 'STEP' 2 'UNTIL' 15 'DO'  
'BEGIN'  
B:=I-1;  
'FOR' J:=1,3,5 'DO'  
'BEGIN'  
YS:=0;  
YF:=(4*R0)+(((I-7)/2)*((H-(4*R0))/4));  
DELTA:=(YF-YS)/6;  
XX[(B*N1)+J]:=(G-AA-(4*R0))-(((I-7)/2)*((G-AA-(4*R0))/4));  
YY[(B*N1)+J]:=YS*(J-1)+DELTA;  
'END';  
'FOR' J:=7 'STEP' 2 'UNTIL' 15 'DO'  
'BEGIN'  
XS:=(G-AA-(4*R0))-(((I-7)/2)*((G-AA-(4*R0))/4));  
XF:=(G-AA+(4*R0))+(((I-7)/2)*((AA-(4*R0))/4));  
DELTA:=(XF-XS)/6;  
XX[(B*N1)+J]:=XS+(J-7)*DELTA;  
YY[(B*N1)+J]:=(4*R0)+(((I-7)/2)*((H-(4*R0))/4));  
'END';  
'FOR' J:=15,17,19 'DO'  
'BEGIN'  
YS:=(4*R0)+(((I-7)/2)*((H-(4*R0))/4));  
YF:=0;  
DELTA:=(YF-YS)/6;  
XX[(B*N1)+J]:=(G-AA+(4*R0))+(((I-7)/2)*((AA-(4*R0))/4));  
YY[(B*N1)+J]:=YS*(J-15)+DELTA;  
'END';  
'END';
```



```
NSPEC:=READ;
'FOR' I:=1 'STEP' 1 'UNTIL' NSPEC 'DO'
'BEGIN'
2 J:=READ; XX[I]:=READ; YVE[I]:=READ;
   KODE[J,1]:=READ; ULX[I,1]:=READ; VLY[I,1]:=READ;
4 'END';
   'FOR' W:=1 'STEP' 1 'UNTIL' NGLNT 'DO'
6 MTNO[W]:=1;
   'BEGIN'
8 'IF' NMAT=1 'THEN'
   'GOTO' RWA4
10 'ELSE'
   'BEGIN'
12 'FOR' I:=2 'STEP' 1 'UNTIL' NGLNT 'DO'
   'BEGIN'
14 NSTEL[I]:=READ;
   'FOR' J:=1 'STEP' 1 'UNTIL' NSTEL[I] 'DO'
16 STEL[I,J]:=READ;
   'FOR' W:=1 'STEP' 1 'UNTIL' NGLNT 'DO'
18 'BEGIN'
   J:=1;
20 RWA1:'IF' W=STEL[I,J] 'THEN'
   'GOTO' RWA2
22 'ELSE'
   J:=J+1;
24 'IF' J 'LE' NSTEL[I] 'THEN'
   'GOTO' RWA1
6 'ELSE'
   'GOTO' RWA3;
3 RWA2:MTNO[W]:=I;
   RWA3:'END';
1 'END';
   'END';
   RWA4:'END';
   'FOR' I:=1 'STEP' 2 'UNTIL' NGLNT-1 'DO'
   'BEGIN'
   L:=1;
   D:=0;
   ALI:'IF' I 'LE' (L*(N1-1))-1 'THEN'
   'BEGIN'
   B:=I-((L*N1)-N1-L+2);
   W:=I;
   NODE[W,1]:=1+B+D;
   NODE[W,2]:=(2*N1)+3+B+D;
   NODE[W,3]:=(2*N1)+1+B+D;
   NODE[W,4]:=N1+2+B+D;
   NODE[W,5]:=(2*N1)+2+B+D;
   NODE[W,6]:=N1+1+B+D;
   NODE[W,7]:=MTNO[W];
   'END'
   'ELSE'
   'BEGIN'
   L:=L+1;
   D:=D+(2*N1);
   'GOTO' ALI;
   'END';
'END';
```



```
'FOR' I:=2 'STEP' 2 'UNTIL' 'ELEMNT' 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
RANA:'IF' I 'LE' (L*(I-1)) 'THEN'  
'BEGIN'  
B:=I-((L*(I-1)-N1-L-3));  
W:=I;  
NODE[W,1]:=1+R+D;  
NODE[W,2]:=3+B+D;  
NODE[W,3]:=(2*N1)+3+D+D;  
NODE[W,4]:=2+B+D;  
NODE[W,5]:=N1+3+B+D;  
NODE[W,6]:=N1+2+B+D;  
NODE[W,7]:=MINV[N];  
'END'  
'ELSE'  
'BEGIN'  
L:=L+1;  
D:=D+(2*N1);  
'GOTO' RANA;  
'END';  
'END';  
WRITETEXT('('('('2C')'NODAL%CONNECTIONS%HAVE%BEEN%READ')');  
'FOR' W:=1 'STEP' 1 'UNTIL' 'ELEMNT' 'DO'  
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'  
'BEGIN' 'IF' XX[NODE[W,3+I]]=0.00001 'THEN'  
'BEGIN'  
'IF' I=1 'THEN' J:=1 'ELSE' 'IF' I=2 'THEN' J:=2 'ELSE' J:=0;  
XX[NODE[W,3+I]]:=(XX[NODE[W,I]]+XX[NODE[W,1+J]])/2;  
YY[NODE[W,3+I]]:=(YY[NODE[W,I]]+YY[NODE[W,1+J]])/2;  
'END';  
'END';  
WRITETEXT('('('('2C')'NODAL%POINT%DATA'('2C4S')'NODE'('5S')'R%COORD  
'('5S')'Z%COORD'('5S')'TYPE'('5S')'R-DISP'('5S')'Z-DISP'('C+6S')'  
OR%LOAD'('4S')'OR%LOAD')');  
'FOR' I:=1 'STEP' 1 'UNTIL' 'NODE' 'DO'  
'BEGIN' NEWLINE(1); SPACE(3);  
PRINT(I,3,0); SPACE(3);  
PRINT(XX[I],0,3);  
PRINT(YY[I],0,3); SPACE(2);  
PRINT(KODE[I,1],3,0); SPACE(2);  
PRINT(CULX[I,1],0,3);  
PRINT(VLY[I,1],0,3);  
'END';  
WRITETEXT('('('('4C')'ELEMENT%DATA'('2C')'ELEMENT'('18S')'  
NODAL%CONNECTIONS')');  
'FOR' W:=1 'STEP' 1 'UNTIL' 'ELEMNT' 'DO'  
'BEGIN' NEWLINE(1);  
PRINT(W,3,0); SPACE(6);  
'FOR' J:=1 'STEP' 1 'UNTIL' 7 'DO'  
'BEGIN' PRINT(NODE[W,J],3,0);  
SPACE(2);  
'END';  
'END';  
'END' OF PROCEDURE GCRINPUT;
```



```

'PROCEDURE' CONSTREL(Z,MATNO,A);
'INTEGER' MATNO; 'ARRAY' A,Z;
'BEGIN' 'INTEGER' I,J;
'REAL' C11,C12,C13,C14,C22,C23,C24,C33,C34,C44;
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO' A[I]:=READ;
WRITETEXT('('('('2045')'E1371000-STARTS'('2035')'E1X=Y')');
PRINT(A[1],0,6); WRITETEXT('('('135')'V1X=X')');
PRINT(A[2],0,6); WRITETEXT('('('2035')'E24=Y')');
PRINT(A[4],0,6); WRITETEXT('('('135')'V2X=X')');
PRINT(A[5],0,6); WRITETEXT('('('2035')'E12X=X')');
PRINT(A[3],0,6);
C14:=C24:=C34:=0.0;
C44:=A[3];
A[3]:=A[1]/((1+A[5])*(1-A[5])-2*(A[4]/A[1])*A[2]*A[2]);
C11:=A[3]+(1-A[5]*A[5]);
C12:=C13:=C23:=A[3]*(A[4]/A[1])*A[2]*(1+A[5]);
C22:=C33:=(A[4]/A[1])*(1-(A[4]/A[1])*A[4]*A[2])*A[3];
Z[MATNO,1]:=C11; Z[MATNO,2]:=C12;
Z[MATNO,3]:=C13; Z[MATNO,4]:=C14;
Z[MATNO,5]:=C22; Z[MATNO,6]:=C23;
Z[MATNO,7]:=C24; Z[MATNO,8]:=C33;
Z[MATNO,9]:=C34; Z[MATNO,10]:=C44;
'END' OF PROCEDURE CONSTREL;

'PROCEDURE' ASSEMBLY(NELEMT,K,XX,YY,DETJ,MODE,C,THICK,ADD);
'INTEGER' NELEMT; 'REAL' DETJ,THICK;
'REAL' 'ARRAY' K,XX,YY,C; 'INTEGER' 'ARRAY' MODE,ADD;
'BEGIN' 'INTEGER' I,J,U,Z,V,SUB2,SUB1,SUB3;
'REAL' 'ARRAY' B[1:4,1:12],KE[1:12,1:12],I[1:6,1:4];
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' W[1,1]:=0.33333333;
W[1,2]:=W[1,3]:=W[2,3]:=W[2,4]:=W[3,2]:=W[3,4]:=0.5;
W[1,4]:=W[2,2]:=W[3,3]:=W[4,4]:=W[5,2]:=W[6,3]:=0.0;
W[4,2]:=W[4,3]:=W[5,3]:=W[5,4]:=W[6,2]:=W[6,4]:=1.0;
'FOR' I:=1 'STEP' 1 'UNTIL' ADD[HEREE] 'DO' K[I]:=0.0;
'FOR' Z:=1 'STEP' 1 'UNTIL' NELEMT 'DO'
'BEGIN'
'FOR' I:=1 'STEP' 1 'UNTIL' 12 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 12 'DO' KE[I,J]:=0.0;
'COMMENT' FOR LATER ADAPTATION THIS WILL BE STORED AS
A ONE DIMENSIONAL ARRAY
THE LOOP FOR THE NUMBER OF INPUTS IS CONSTRUCTED;
'FOR' U:=1 'STEP' 1 'UNTIL' 5 'DO'
'BEGIN'
STRDIS(W[U,2],W[U,3],W[U,4],B,XX,YY,DETJ,MODE,Z);
'FOR' J:=1 'STEP' 2 'UNTIL' 14 'DO'
'FOR' I:=J 'STEP' 2 'UNTIL' 14 'DO'
'BEGIN'
KE[J,I]:=KE[I,J]:=KE[I,J]+W[U,1]*B[I,J]*(C[NODE[Z,7],1]*B[1,I]+
C[NODE[Z,7],2]*B[2,I]+C[NODE[Z,7],4]*B[4,I]+
B[2,J]*(C[NODE[Z,7],2]*B[1,I]+C[NODE[Z,7],5]*B[2,I]+
C[NODE[Z,7],7]*B[4,I]))+
B[4,J]*(C[NODE[Z,7],4]*B[1,I]+C[NODE[Z,7],7]*B[2,I]+
C[NODE[Z,7],10]*B[4,I]))*2*3.1415926*RAVG*U.5*DETJ;
KE[J,I+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]*(B[1,J]*(C[NODE[Z,7],3]*B[3,I+1]+
C[NODE[Z,7],4]*B[4,I+1]))+
B[2,J]*(C[NODE[Z,7],6]*B[3,I+1]+C[NODE[Z,7],7]*B[4,I+1])+
B[4,J]*(C[NODE[Z,7],9]*B[3,I+1]+C[NODE[Z,7],10]*B[4,I+1]))*
2*3.1415926*RAVG*0.5*DETJ;
'END';
'END';

```



```

'FOR' J:=2 'STEP' 2 'UNTIL' 12 'DO'
'FOR' I:=J 'STEP' 2 'UNTIL' 12 'DO'
'BEGIN'
KE[J,I]:=KE[I,J]:=KE[I,J]+B[4,I]*B[4,J]*(C[NODE[Z,7],9]*B[4,I]+
C[NODE[Z,7],9]*B[4,I])+
B[4,J]*(C[NODE[Z,7],9]*B[4,I]+C[NODE[Z,7],10]*B[4,I])
*2*3.1415926*RAVG*0.5*DETJ;
'IF' I=12 'THEN' 'GOTO' L6 'ELSE'
KE[J,I+1]:=KE[I+1,J]:=KE[I+1,J]+B[4,I]*B[4,J]*(C[NODE[Z,7],3]*B[1,I+1]+
C[NODE[Z,7],6]*B[2,I+1]+C[NODE[Z,7],9]*B[4,I+1])*
B[4,J]*(C[NODE[Z,7],4]*B[1,I+1]+C[NODE[Z,7],7]*B[2,I+1]+
C[NODE[Z,7],10]*B[4,I+1])*2*3.1415926*RAVG*0.5*DETJ;
L6: 'END';
'END';
'COMMENT' ASSEMBLY OF OVERALL STIFFNESS MATRIX AS A
ONE-DIMENSIONAL ARRAY;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' V:=1,0 'DO'
'BEGIN' SUB1:=NODE[Z,I]*2-1;
SUB2:=NODE[Z,J]*2-V;
SUB3:=NODE[Z,I]*2;
'IF' SUB1 'LT' SUB2 'THEN' 'GOTO' LABA;
K[ADD(SUB1)-SUB1+SUB2]:=K[ADD(SUB1)-SUB1+SUB2]+KE[I*2-1,J*2-V];
LABA: 'IF' SUB3 'LT' SUB2 'THEN' 'GOTO' LABB;
K[ADD(SUB3)-SUB3+SUB2]:=K[ADD(SUB3)-SUB3+SUB2]+KE[I*2,J*2-V];
LABB: 'END';
'END';
'END' OF PROCEDURE ASSEMBLY;

NJOB:=READ;
'FOR' COUNT:=1 'STEP' 1 'UNTIL' NJOB 'DO'
'BEGIN'
WRITETEXT('('('2C')'JOB#,'ABE'-'-'-'')');
COPYTEXT('('END%OF%TITLE')');
NELEMT:=READ; NNODE:=READ; NSETF:=READ;
WRITETEXT('('('2C')'NOKOF%ELEMENTS%'-'-'-'')');
PRINT(NELEMT,3,0);
WRITETEXT('('('2C')'NOKOF%NODES%'-'-'-'')');
PRINT(NNODE,3,0);
NFREE:=NNODE*2; NHAT:=READ;
'BEGIN' 'INTEGER' NSPEC,Z,CN,AA,II; 'REAL' G,H,AA,RO,CR;
'REAL' 'ARRAY' XX,YY[1:NNODE],ULX,VLY[1:NNODE,1:NSETF],
C[1:NHAT,1:10],AC[1:5],OC[1:NFREE,1:NSETF],CH[1:2];
'INTEGER' 'ARRAY' NODE[1:NELEMT,1:2],KODE[1:NNODE,1:NSETF],
ADD[0:NFREE],INTNO[1:NELEMT],NSTEL[1:NHAT],STEL[1:NHAT,1:NELEMT];
N1:=READ; G:=READ; H:=READ; AA:=READ; RO:=READ;
NSETC:=READ; CR:=G*AA; CAGE:=READ;
'FOR' S:=1 'STEP' 1 'UNTIL' NSETC 'DO'
'BEGIN' 'IF' S=1 'THEN'
'BEGIN'
CCRINPUT(ADD,XX,YY,NFREE,NNODE,NSETF,KODE,NSPEC,NHAT,INTNO,
NSTEL,STEL,ULX,VLY,NELEMT,NODE,N1,G,I,AA,RO);
ADDARRAY(NELEMT,NNODE,ADD,NODE);
KARM1BND(N1,NFREE,ADD,BA1D);

```



```
'FOR' MATNO:=1 'STEP' 1 'UNTIL' NMAT 'DO'  
CONSTREL(C,MATNO,1);  
'END' 'ELSE'  
'BEGIN' NNEWC:=READ; NSETF:=READ;  
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'  
'BEGIN' J:=READ;  
KODE(I,1):=READ; ULX(I,1):=READ; VLY(I,1):=READ;  
'END';  
'END';  
WRITETEXT('('('2C')'NORORNDGREEEORORFREEDOM---')');  
PRINT(NFREE,3,0);  
NEWLINE(2);  
'FOR' I:=1 'STEP' 1 'UNTIL' NFACE 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q(I,J):=0,0;  
'FOR' II:=1 'STEP' 1 'UNTIL' NNODE 'DO'  
LOADING(KODE(I,1),ULX(I,1),VLY(I,1),0,I,1);  
'IF' NSETF 'GT' 1 'THEN'  
'BEGIN'  
'FOR' II:=1 'STEP' 1 'UNTIL' NSETF 'DO'  
'BEGIN'  
NSPEC:=READ;  
WRITETEXT('('('2C')'FORCESSET---')');  
PRINT(I,3,0);  
WRITETEXT('('('2C')'NODE('5S')'TYPE('5S')'X-DISP('5S')'Y-DISP  
'('C10S')'ORXLOAD('4S')'ORLOAD('5')');  
'FOR' J:=1 'STEP' 1 'UNTIL' NSPEC 'DO'  
'BEGIN'  
K:=READ; KODE[K,II]:=READ; ULX[K,II]:=READ; VLY[K,II]:=READ;  
LOADING(KODE[K,II],ULX[K,II],VLY[K,II],C,K,II);  
NEWLINE(2);  
PRINT(K,3,0); SPACE(2);  
PRINT(KODE[K,II],0,4); SPACE(2);  
PRINT(ULX[K,II],0,4);  
PRINT(VLY[K,II],0,4);  
'END';  
'END';  
'END';  
NEWLINE(6);  
'BEGIN'  
'INTEGER' NG,HND;  
'REAL' 'ARRAY' K(1:ND);  
NG:=NFREE;  
HND:=READ;  
ASSEMBLY(NELEM,K,XX,YY,DETJ,NODE,C,THICK,ADD);  
'COMMENT' INTRODUCTION OF KINEMATIC CONSTRAINTS;  
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'  
'BEGIN'  
'IF' KODE(I,1)=0 'THEN' 'GOTO' KC1;  
'IF' KODE(I,1)=2 'THEN' 'GOTO' KC2;  
BOUNCONST(ULX(I,1),2*I-1,0,K,NFREE,1,ADD);  
'FOR' J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q(I,J):=Q(I,1);  
'IF' KODE(I,1)=1 'THEN' 'GOTO' KC1;  
KC2: BOUNCONST(VLY(I,1),2*I,0,K,NFREE,1,ADD);  
'FOR' J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q(I,J):=Q(I,1);  
KC1: 'END';  
WRITETEXT('('('C2S')'CCRM1$BEGIN$')');  
CCRM1(N1,20,NG,NSETF,CN,ADD,K,0,CASE,HND,CR,AN);  
NEWLINE(2);  
WRITETEXT('('('2C')'SYNTHSOL$BEGIN$')');
```



```
SYMBVSOL(K,K,ADD,Q,NG-2*N1+2,ISSET,FALL);
2 NEWLINE(6);
'FOR' I:=1 'STEP' 1 'UNTIL' 'ISSET' 'DO'
1 'BEGIN'
    WRITETEXT('('('40')'NODE#DISPLACEMENTS%FOR%FORCE%SET)');
    PRINT(I,2,0);
    WRITETEXT('('('30S')'NODE'('5S')'R-DIRECTION'('8S')'
3 Z-DIRECTION'('18S')'NODE'('5S')'R-DIRECTION'('8S')'
1 Z-DIRECTION')');
    U:=2*(NNODE-N1)/I;
    'FOR' J:=2 'STEP' 2 'UNTIL' U 'DO'
    'BEGIN' NEWLINE(2); V:=2*(J-1); Z:=J+1;
        PRINT(Z-1,3,0); SPACE(2); PRINT(Q[V-1,I],0,8);
        SPACE(2); PRINT(Q[V,I],0,8); SPACE(15);
        PRINT(Z,3,0); SPACE(2); PRINT(Q[V+1,I],0,8);
        SPACE(2); PRINT(Q[V+2,I],0,8);
    'END';
    'IF' NNODE-N1 'GT' 1 'THEN'
    'BEGIN' NEWLINE(2); PRINT(INODE-N1,3,0); SPACE(2);
        V:=2*(NNODE-N1);
        PRINT(Q[V-1,I],0,8);SPACE(2);PRINT(Q[V,I],0,8);
    'END';
    V:=NG-2*N1;
    WRITETEXT('('('20')'NODE#STRESS#INTENSITY#FACTOR,KI=')');
    IN%R-DIRECTION=');
    PRINT(Q[V+1,I],0,8);
    WRITETEXT('('('20')'NODE#STRESS#INTENSITY#FACTOR,KI=')');
    PRINT(Q[V+2,I],0,8);
'END';
'END';
'END';
'END';
FAIL: 'END';
'END';
****
```


III) Input data instructions

A) Number of jobs to be solved.

B) For the first job.

1. The job name followed by end of title.
2. Number of elements.
3. Number of nodes.
4. Number of sets of forces.
5. Number of materials.
6. Number of nodes on the singular core
7. The radius of the specimen⁽¹⁾
8. Half the length of the specimen⁽²⁾.
9. Crack length.
10. Singular core radius
11. Number of sets of constraints.
12. Case⁽³⁾.

C) For the first set of constraints

1. Number of specified nodes
2. For the specified nodes:
 - a) The node number
 - b) The node r coordinate
 - c) The node z coordinate
 - d) Kode
 - e) Value of prescribed load or displacement in r direction.
 - f) Value of prescribed load or displacement in z direction.
3. If number of materials >1 then:
 - a) The number of elements with different material properties for each different material.
 - b) The element numbers of these elements.

5. IF the number of sets of forces for this set of constraints is >1 then:
 - a) Number of specified forces.
 - b) For the number of specified forces
 - Node number
 - Kode
 - Value of prescribed load in r direction
 - Value of prescribed load in z direction.
6. Direction of crack tip⁽⁴⁾
7. Elastic constants E and ν of near tip material.
- D) For the subsequent sets of constraints:
 1. Number of new constraints
 2. Number of sets of forces for this set of constraints
 3. For the number of new constraints.
 - a) Node number
 - b) Kode
 - c) value of prescribed load or displacement in r direction.
 - d) value of prescribed load or displacement in z direction.
- E) For the next job strat from B.

Notes

- (1) Dimension G shown in Fig.(10.6)
- (2) Dimension H shown in Fig.(10.6)
- (3) Input 1 for all cases.
- (4) Input 1 for a circumferential crack
2 for a penny shaped crack.

IV) Sample problem

For the problem shown in Fig. (10.6), the required data is as follows:

```
1
CCRS1  END OF TITLE
126,285,1,19,10,14,5.0,0.1,1,1
29
20,r20,z20,2,0,0
39,r39,z39,2,0,0
.
.
.
.
.
267,0.0,0.0 3,0,0
.
.
.
.
.
279,10,14,0,0,14540
30000000,0.3,12000000,30000000,0,3,1
30000000,0.3
```

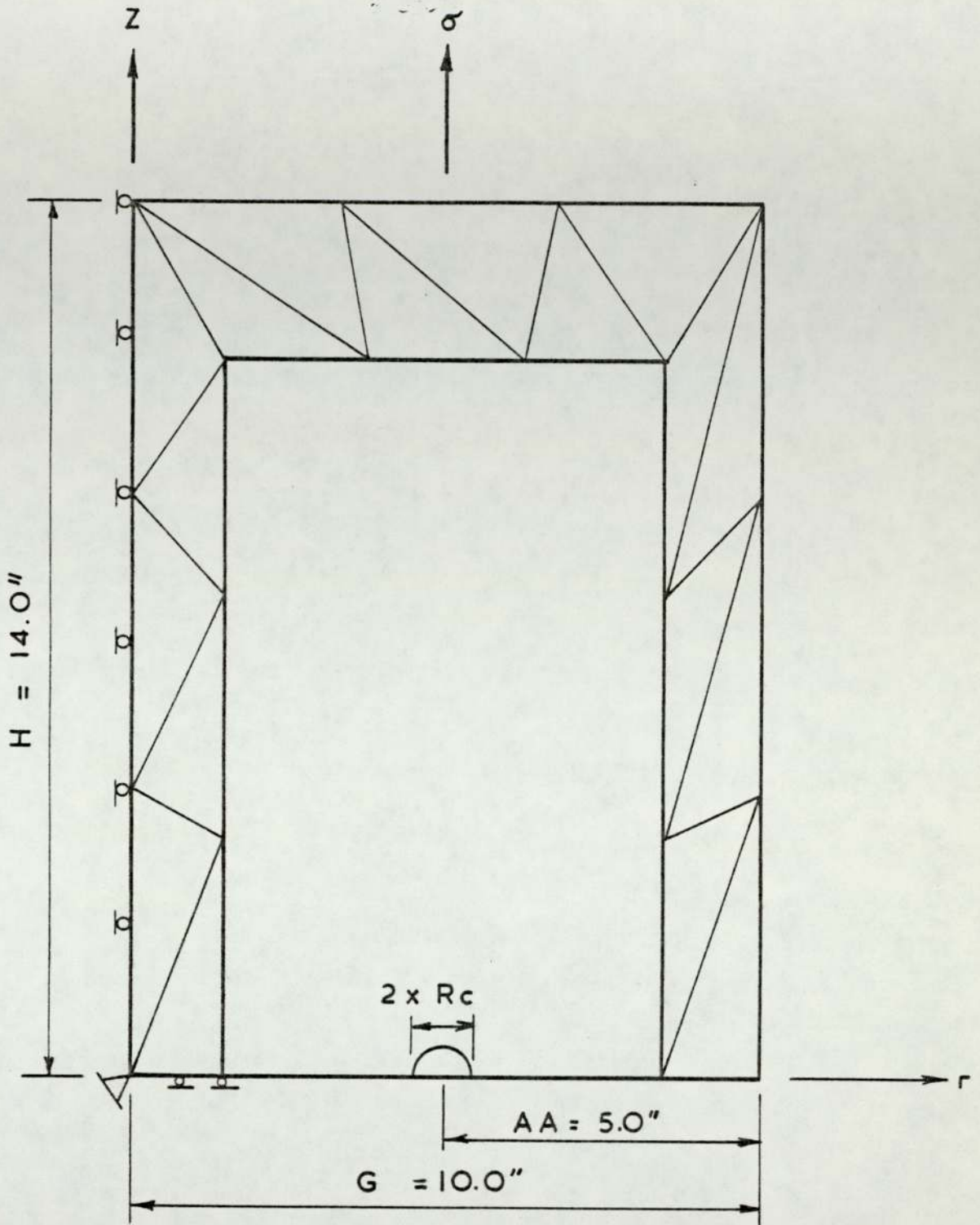
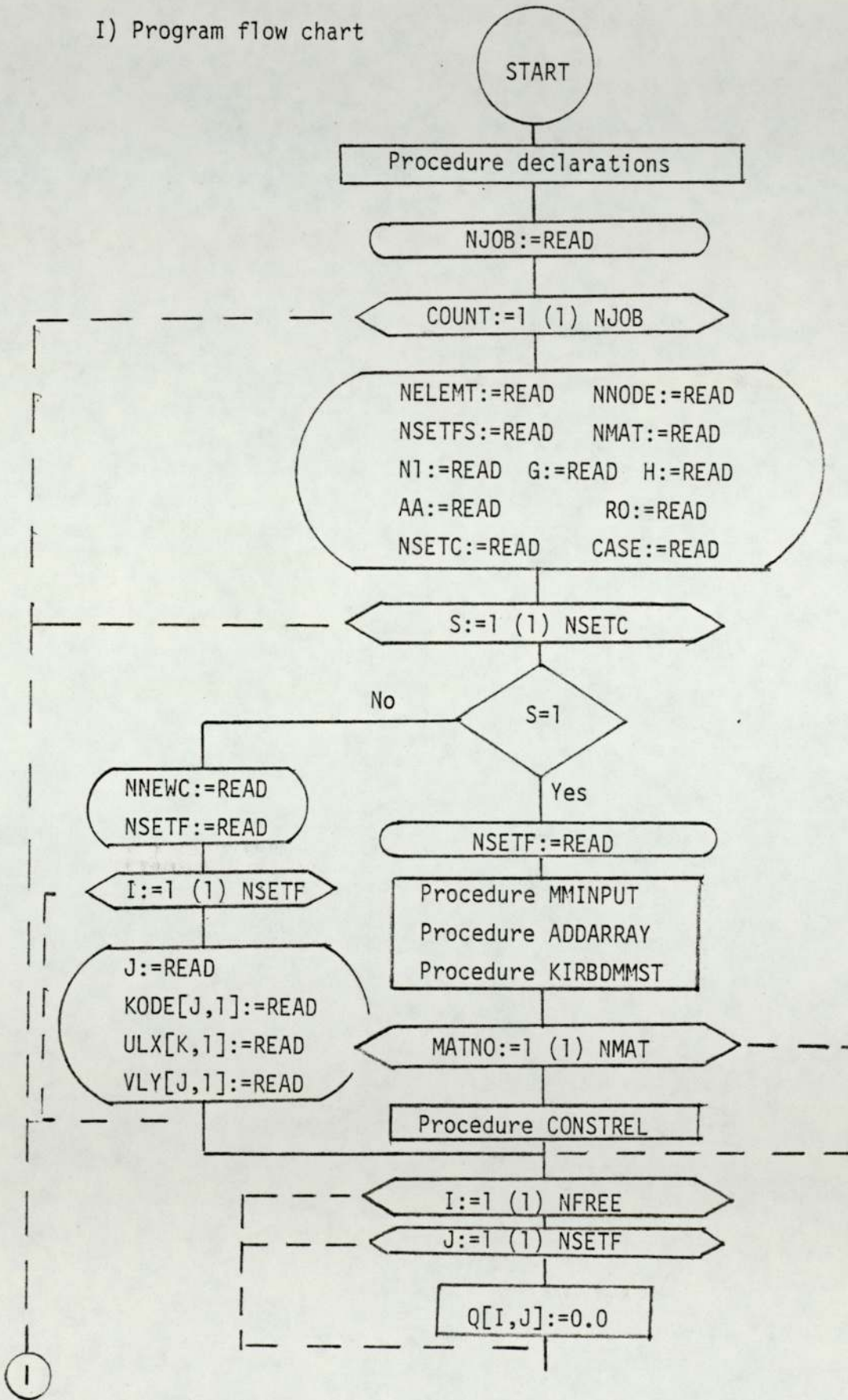
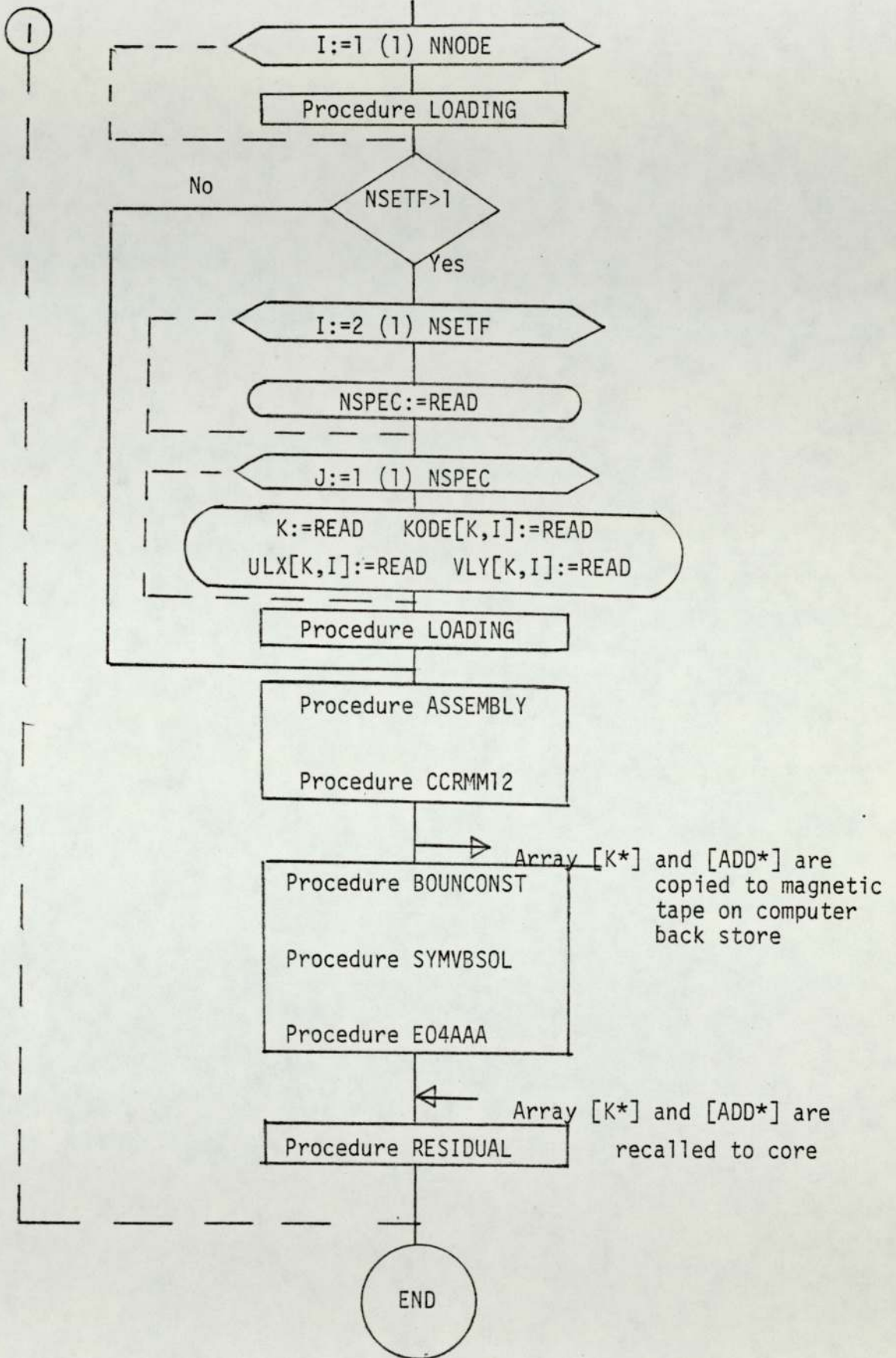


Fig. 10.6

10.4.3 Mixed mode I and II program

I) Program flow chart





II) Program listing

```
'PROGRAM' (AXXX)
'INPUT' O=CRO
'OUTPUT' O=LPO
'EXTENDED DATA'
'EXTENDED'
'TRACE' 2
'BEGIN'
'INTEGER' N,NELEMT,NNODE,NSSETS,PRPBE,I,J,K,V,U,C,SS,SETC,ANEHG,
NJOB,COUNT,PRNT,NVAB1,NUAT,MATNO,FAIL,MAXFUN;
'REAL' DELTA,DETJ,THICK,RAYG,REDA,SS,TH,ASSACC,
RELACC,XSTEP,CON,K1,K2,NU;
'PROCEDURE' CREATESTORE(N,S,T,G,L);
'VALUE' N,L,G; 'INTEGER' N,L,G; 'STRING' S,T; 'EXTERNAL';
'PROCEDURE' INSTORE(N,S,T,G);
'VALUE' N,G; 'INTEGER' N,G; 'STRING' S,T; 'EXTERNAL';
'PROCEDURE' PUTARRAY(N,K,A);
'VALUE' N; 'INTEGER' N,K; 'ARRAY' A; 'EXTERNAL';
'PROCEDURE' GETARRAY(N,K,A);
'VALUE' N; 'INTEGER' N,K; 'ARRAY' A; 'EXTERNAL';
'PROCEDURE' STRDIS(L1,L2,L3,B,X,Y,Z);
'VALUE' L1,L2,L3,Z;
'INTEGER' Z;
'REAL' L1,L2,L3,U;
'INTEGER' 'ARRAY' N;
'ARRAY' X,Y,B;
'BEGIN'
'INTEGER' I,V;
'REAL' CHANGE;
'REAL' 'ARRAY' P[1:2,1:6],J[1:2,1:2],NL[1:6];
'COMMENT' THIS PROCEDURE EVALUATES THE JACOBIAN J ITS DETERMINANT U
AND THE STRAIN-DISP ARRAY P;
J[1,1]:=X[N(Z,1)]*(4*L1-1)+X[N(Z,2)]*(4*L1+6*(L2-3))+2*X[N(Z,4)]*
4*L2*X[N(Z,5)]+4*X[N(Z,6)]*(1-2*L1-L2);
J[1,2]:=Y[N(Z,1)]*(4*L1-1)+Y[N(Z,2)]*(4*L1+6*(L2-3))+2*Y[N(Z,4)]*
4*L2*Y[N(Z,5)]+4*Y[N(Z,6)]*(1-2*L1-L2);
J[2,1]:=X[N(Z,2)]*(4*L2-1)+X[N(Z,1)]*(4*L1+6*(L2-3))+L1*X[N(Z,4)]*
4*X[N(Z,5)]*(1-L1-2*L2)-4*L1*X[N(Z,6)];
J[2,2]:=Y[N(Z,2)]*(4*L2-1)+Y[N(Z,1)]*(4*L1+6*(L2-3))+L1*Y[N(Z,4)]*
4*Y[N(Z,5)]*(1-L1-2*L2)-4*L1*Y[N(Z,6)];
'COMMENT' U REPLACES DETJ;
U:=J[1,1]*J[2,2]-J[1,2]*J[2,1];
'COMMENT' THE COEFFS OF [J] ARE REPLACED BY THOSE OF [J]=1;
CHANGE:=J[1,1];
J[1,1]:=J[2,2]/U;
J[1,2]:=-J[1,2]/U;
J[2,1]:=-J[2,1]/U;
J[2,2]:=CHANGE/U;
NL[1]:=L1*(2*L1-1);
NL[2]:=L2*(2*L2-1);
NL[3]:=L3*(2*L3-1);
NL[4]:=4*L1*L2;
NL[5]:=4*L2*L3;
NL[6]:=4*L3*L1;
RMEAN:=(X[N(Z,1)]+X[N(Z,2)]+X[N(Z,3)]+X[N(Z,4)]+
X[N(Z,5)]+X[N(Z,6)])/6;
```



```
'IF' Z=89 'OR' Z=01 'THEN'  
RAVG:=RMEAN  
'ELSE'  
'BEGIN'  
RAVG:=0.0;  
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'  
RAVG:=RAVG*(XEN(Z,I))*NL(I);  
'END';  
'COMMENT' THE P ARRAY HELPS THE USER CHECK THE CORRECTNESS OF THE WAVE LE AND LZ;  
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'  
'FOR' V:=1,2 'DO' P[V,I]:=0.0;  
'FOR' I:=1,2 'DO'  
'BEGIN' P[I,1]:=J[I,1]*(4*L1-1);  
P[I,2]:=J[I,2]*(4*L2-1);  
P[I,3]:=J[I,1]*(1-4*L3)+J[I,2]*(1+4*L3);  
P[I,4]:=4*(L2*J[I,1]+L1*J[I,2]);  
P[I,5]:=4*(J[I,2]+L3-L2*(J[I,1]+J[I,2]));  
P[I,6]:=4*(J[I,1]-L3-L1*(J[I,1]+J[I,2]));  
'END';  
'FOR' I:=1 'STEP' 1 'UNTIL' 12 'DO'  
'FOR' V:=1,2,3,4 'DO' B[V,I]:=0.0;  
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'  
'BEGIN'  
B[1,(I+2-1)]:=B[4,(I+2)]:=B[1,I];  
B[4,(I+2-1)]:=B[3,(I+2)]:=B[2,I];  
B[2,(I+2-1)]:=NL(I)/RAVG;  
'END';  
'END' OF STRDIS;  
  
'PROCEDURE' BOUNCONST(U,N,R,AK,NEQ,F,A);  
'VALUE' U,N,NEQ,F;  
'REAL' U;  
'INTEGER' N,NEQ,F;  
'ARRAY' R,AK;  
'INTEGER' 'ARRAY' A;  
'BEGIN' 'INTEGER' M,K,CJ;  
'IF' N=1 'THEN' CJ:=1 'ELSE' CJ:=N*(A[N]-A[N-1])+1;  
'FOR' K:=CJ 'STEP' 1 'UNTIL' N 'DO'  
'BEGIN'  
R[K,F]:=R[K,F]-AK[A[N]-N+K]*U;  
AK[A[N]-N+K]:=0.0;  
'END';  
'IF' N+1 'GT' NEQ 'THEN' 'GOTO' LZ;  
'FOR' K:=N+1 'STEP' 1 'UNTIL' NEQ 'DO'  
'BEGIN' CJ:=K-(A[K]-A[K-1])+1;  
'IF' CJ 'LE' N 'THEN'  
'BEGIN'  
R[K,F]:=R[K,F]-AK[A[K]-K+1]*U;  
AK[A[K]-K+1]:=0.0;  
'END' 'ELSE'  
'END';  
LZ: AK[A[N]]:=1.0;  
R[N,F]:=U;  
'END' OF BOUNCONST;
```



```
'PROCEDURE' LOADING(A,B,C,D,E,F);
'VALUE' A,B,C,E,F;
'INTEGER' A,E,F;
'REAL' B,C;
'ARRAY' D;
'BEGIN'
'INTEGER' K;
'IF' A=3 'THEN' 'GOTO' KLAB1;
K:=2*E;
'IF' A=1 'THEN' 'GOTO' KLAB2;
D[K-1,F]:=D[K-1,F]+B;
'IF' A 'NE' 0 'THEN' 'GOTO' KLAB1;
KLAB2: D[K,F]:=D[K,F]+C;
KLAB1: 'END' OF LOADING;

'PROCEDURE' SYMBVSOL(A,L,S,N)='SINGULAR' FAILURE EXIT;(FAIL)
'VALUE' N,R; 'ARRAY' A,L,B; 'INTEGER' 'ARRAY' S; 'INTEGER' N,R;
'LABEL' FAIL;
'BEGIN'
'INTEGER' G,H,I,J,K,H,P,Q,T,U,V;
'REAL' Y;
H:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
'BEGIN'
T:=I+H-S[I]+1; G:=H+1;
P:=S[T]-1;
'FOR' J:=T 'STEP' 1 'UNTIL' I-1 'DO'
'BEGIN'
Q:=P+1; H:=H+1;
P:=S[J]; K:=J+Q-1;
V:=H-P; U:=G;
Y:=A[H];
'IF' K 'GT' T 'THEN' U:=U+K-T;
'FOR' U:=U 'STEP' 1 'UNTIL' H-1 'DO'
Y:=Y-L[U]*L[U-V];
Y:=Y/L[H-V]; L[H]:=Y;
'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'
B[I,M]:=B[I,M]-B[J,H]*Y;
'END' J;
Y:=A[H+1];
'FOR' U:=G 'STEP' 1 'UNTIL' H 'DO'
Y:=Y-L[U]*2;
'IF' Y 'LE' 0 'THEN' 'GOTO' FAIL;
H:=H+1; Y:=SQRT(Y);
L[H]:=Y;
'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'
B[I,M]:=B[I,M]/Y;
'END' I;
'COMMENT' REDUCTION COMPLETE;
'FOR' I:=N 'STEP' -1 'UNTIL' 1 'DO'
'BEGIN'
Y:=L[H];
'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'
B[I,M]:=B[I,M]/Y;
```



```
'IF' I=1 'THEN' 'GOTO' COMPLETE;
J:=I; P:=S[I-1];
'FOR' H:=H-1 'STEP' -1 'UNTIL' -1 'DO'
  'BEGIN'
    J:=J-1; Y:=L[H];
    'FOR' M:=1 'STEP' 1 'UNTIL' P 'DO'
      B[J,M]:=B[J,M]-B[I,H]*Y;
    'END' H;
  H:=P;
COMPLETE; 'END' I;
'END' SYMVBSOL;
'PROCEDURE' ADDARRAY(NELEM,NNODE,ADD,NODE);
'INTEGER' NELEM,NNODE;
'INTEGER' 'ARRAY' NODE,ADD;
'BEGIN' 'INTEGER' W,CH,I,ADDTMP;
'FOR' W:=1 'STEP' 1 'UNTIL' NELEM 'DO'
  'BEGIN' CH:=NODE[W,1];
    'FOR' I:=2 'STEP' 1 'UNTIL' 6 'DO'
      'IF' NODE[W,I] LT CH 'THEN' CH:=NODE[W,I];
      'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
        'BEGIN' ADDTMP:=NODE[W,I]-CH+1;
          'IF' ADDTMP GT ADD[NODE[W,I]*2] 'THEN' ADD[NODE[W,I]*2]:=ADDTMP;
        'END';
      'END';
    'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
      'BEGIN' CH:=2*ADD[2*I]; W:=2*I;
        ADD[W-1]:=ADD[W-2]+CH-1;
        ADD[W]:=ADD[W-1]+CH;
      'END';
    'END' OF PROCEDURE ADDARRAY;

'PROCEDURE' GCRMM12(N1,RO,NG,NSETF,CASE,ADD,K,Q,CASE,THICK,CR);
'VALUE' N1,NG,RO,NSETF,CASE,THICK,CR; 'INTEGER' N1,NG,NSETF,CASE;
'REAL' RO,THICK,CR; 'INTEGER' 'ARRAY' ADD; 'REAL' 'ARRAY' K,Q,CH;
'BEGIN' 'COMMENT' MODIFICATION OF OVERALL STIFFNESS MATRIX FOR THE
      NODE I AND II STRESS INTENSITY FACTORS, SINGLE TIP;
  'INTEGER' IA,II,CI,DI,DJ,CJ,JA;
  'REAL' FIO,GIO,FJO,GJO,THETA,KAP,ALPHA,P1,Q1,T1,T2,T3,T4,T5,
    SN,CN,EJ1,GJ1,HI1,HJ1,HI0,HJ0,G,MU;
  'ARRAY' KT(1:5,1:4*N1+5);
  'FOR' I:=1,2 'DO'
    CM[I]:=READ;
    G:=0.5*CM[1]/(1+CH[2]);
    MU:=CH[2];
    ALPHA:=READ;
    WRITETEXT('('('('2C')'PROCEDURE'GCRMM12'BEGIN'US'('C')'('THIS'PROCEDURE'
      MODIFIES'THE'OVERALL'STIFNESS'MATRIX'FORMIXED'MODE'FRACTURE,SINGLE'
      CRACK'TIP'PRESENT)'('2C')'ANGLE'OF'CRACK'TIP'FO'THE'POSITIVE'R-AXIS,DE
      G='X')'); PRINT(ALPHA,0,4);
    WRITETEXT('('('('2C')'NUMBER'OF'NODES'ON'AC'RES'LEMENT'X='');
    PRINT(N1,3,0);
    WRITETEXT('('('('2C')'RADIUS'OF'COEFFICIENTS'X=''); PRINT(RO,0,4);
    WRITETEXT('('('('2C')'MODULUS'OF'RIGIDITY,GX='X')'); PRINT(G,0,4);
    WRITETEXT('('('('2C')'POISSON'S'RATIO,MU='X')'); PRINT(MU,0,4);
    ALPHA:=0.0174533*ALPHA; SN:=SIN(ALPHA); CN:=COS(ALPHA);
```



```

'FOR' I:=1 'STEP' 1 'UNTIL' 2*N1 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 4*N1 'DO' KTE[I,1]:=0.4;
'IF' CASE=0 'THEN' KAP:=(3-N1)/(3+N1) 'ELSE' KAP:=3-4*N1;
KT[1,1]:=2*RO*CR*((3.14159)↑2)*(2-KAP-1)/(3*G)*THICK;
KT[2,2]:=2*RO*CR*((3.14159)↑2)*(3-2*KAP)/(3*G)*THICK;
'FOR' I:=1 'STEP' 1 'UNTIL' 2*N1 'DO'
'BEGIN' IA:=I-2*ENTIER(I/2);
'IF' IA=1 'THEN' THETA:=-3.14159-((2*((I+1)/2-1)*3.14159)/(N1-1))
'ELSE' THETA:=-3.14159*((2*((I/2)-1)+3.14159)/(N1-1));
FIO:=(RO/2)↑0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(4*G);
GIO:=(RO/2)↑0.5*((2*KAP-1)*SIN(THETA/2)+SIN(1.5*THETA))/(4*G);
FI1:=(RO/2)↑0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(4*G);
GI1:=-((RO/2)↑0.5*((2*KAP+1)*COS(THETA/2)+COS(1.5*THETA))/(4*G));
HIO:=-RO*SIN(THETA); HI1:=RO*COS(THETA);
'IF' IA=1 'THEN' 'BEGIN' FI:=FIO*CN-FI1*SN;
GI:=GIO*CN-GI1*SN;
HI:=HI0*CN-HI1*SN;
T2:=CN; T3:=-SN;
'END' 'ELSE'
'BEGIN' FI0:=FIO*SN+FI1*CN;
GI0:=GIO*SN+GI1*CN;
HI0:=HI0*SN+HI1*CN;
T2:=GN; T3:=CN;
'END';
'IF' I=1 'THEN' CI:=1 'ELSE' CI:=I-ADD(I)+ADD(I-1)+1;
'FOR' J:=CI 'STEP' 1 'UNTIL' 2*N1 'DO'
'BEGIN' 'IF' J 'LE' I 'THEN'
'BEGIN' DI:=I; DJ:=J;
'GOTO' MSM1;
'END' 'ELSE'
'BEGIN' 'IF' J=1 'THEN' CJ:=1 'ELSE' CJ:=J-ADD(J)+ADD(J-1)+1;
'IF' CJ 'GT' I 'THEN' 'GOTO' MSM2;
DI:=J; DJ:=I;
'END';
MSM1: JA:=J-2*ENTIER(J/2);
'IF' JA=1 'THEN' THETA:=-3.14159-((2*((J+1)/2-1)*3.14159)/(N1-1))
'ELSE' THETA:=-3.14159*((2*((J/2)-1)+3.14159)/(N1-1));
FJO:=(RO/2)↑0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(4*G);
GJO:=(RO/2)↑0.5*((2*KAP-1)*SIN(THETA/2)+SIN(1.5*THETA))/(4*G);
FJ1:=(RO/2)↑0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(4*G);
GJ1:=-((RO/2)↑0.5*((2*KAP+1)*COS(THETA/2)+COS(1.5*THETA))/(4*G));
HJO:=-RO*SIN(THETA); HJ1:=RO*COS(THETA);
'IF' JA=1 'THEN' 'BEGIN' FJ0:=FJO*CN-FJ1*SN;
GJ0:=GJO*CN-GJ1*SN;
HJ0:=HJO*CN-HJ1*SN;
T1:=-SN; T3:=CN;
'END' 'ELSE'
'BEGIN' FJ0:=FJO*SN+FJ1*CN;
GJ0:=GJO*SN+GJ1*CN;
HJ0:=HJO*SN+HJ1*CN;
T1:=CN; T3:=SN;

```



```

'END';
TEMP1:=ADD[D1]-D1+D2;
KT[1,1]:=KT[1,1]*F10*FJ0*K[TEMP1];
KT[1,2]:=KT[1,2]+G10*FJ0*K[TEMP1];
KT[2,2]:=KT[2,2]+G10*GJ0*K[TEMP1];
KT[1,4]:=KT[1,4]+F10*T1*K[TEMP1];
KT[2,4]:=KT[2,4]+G10*T1*K[TEMP1];
KT[3,4]:=KT[3,4]+T2*T1*K[TEMP1];
KT[1,3]:=KT[1,3]+T3*F10*K[TEMP1];
KT[2,3]:=KT[2,3]+T3*G10*K[TEMP1];
KT[3,3]:=KT[3,3]+T3*T2*K[TEMP1];
KT[4,4]:=KT[4,4]+T4*T1*K[TEMP1];
KT[1,5]:=KT[1,5]+FJ0*H10*K[TEMP1];
KT[2,5]:=KT[2,5]+GJ0*H10*K[TEMP1];
KT[3,5]:=KT[3,5]+T3*H10*K[TEMP1];
KT[4,5]:=KT[4,5]+T1*H10*K[TEMP1];
KT[5,5]:=KT[5,5]+HJ0*H10*K[TEMP1];
MSM2:'END';
'END';
'FOR' J:=6 'STEP' 1 'UNTIL' 4*N1+5 'DO'
'BEGIN' CJ:=J+2*N1-5-ADD[J+2*N1-5]+ADD[J+2*N1-6]+1;
'FOR' I:=CJ 'STEP' 1 'UNTIL' 2*N1 'DO'
'BEGIN' IA:=I-2*ENTIER(I/2);
'IF' IA=1 'THEN' THETA:=-3.14159*((2*((I+1)/2-1)*3.14159)/(N1-1))
'ELSE' THETA:=-3.14159+((2*((I/2)-1)*3.14159)/(N1-1));
F10:=(R0/2)+0.5*((2*KAP-1)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
G10:=(R0/2)+0.5*((2*KAP+1)*SIN(THETA/2)+SIN(1.5*THETA))/(4*G);
F11:=(R0/2)+0.5*((2*KAP-1)*SIN(THETA/2)+SIN(1.5*THETA))/(4*G);
G11:=-R0/2+0.5*((2*KAP-3)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
H10:=-R0*SIN(THETA); H11:=R0*COS(THETA);
'IF' IA=1 'THEN' 'BEGIN' F10:=F10*CN-F11*SN;
G10:=G10*CN+G11*SN;
H10:=H10*CN-H11*SN;
T1:=CN; T2:=-SN;
'END' 'ELSE'
'BEGIN' F10:=F10*SN+F11*CN;
G10:=G10*SN+G11*CN;
H10:=H10*SN-H11*CN;
T1:=SN; T2:=CN;
'END';
TEMP1:=ADD[J+2*N1-5]-J-2*N1+5+1;
KT[1,J]:=KT[1,J]*K[TEMP1]*F10;
KT[2,J]:=KT[2,J]+K[TEMP1]*G10;
KT[3,J]:=KT[3,J]+T1*K[TEMP1];
KT[4,J]:=KT[4,J]+T2*K[TEMP1];
KT[5,J]:=KT[5,J]+H10*K[TEMP1];
'END';
'END';
TI:=NG-2*N1;
'FOR' I:=1 'STEP' 1 'UNTIL' TI 'DO'
'BEGIN' CI:=I+2*N1-ADD[I+2*N1]+ADD[I+2*N1-1]+1;
'IF' CI 'LE' 2*N1 'THEN' CI:=2*N1-1;
ADD[I]:=I+2*N1-CI+ADD[I-1]+1;
'FOR' J:=CI 'STEP' 1 'UNTIL' I+2*N1 'DO'
K[ADD[I]-I+J-2*N1]:=K[ADD[I+2*N1]-I-2*N1+1];

```



```
'END';  
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO' ADD[I+1]:=ADD[I+1]+TI+I;  
'FOR' I:=1 'STEP' 1 'UNTIL' TI 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]:=Q[I+2*N1,J];  
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q[I+TI,J]:=0.0;  
'FOR' I:=1 'STEP' 1 'UNTIL' TI 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' 5 'DO' K[ADD[I+J]-TI-J+1]:=0.0;  
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' I 'DO' K[ADD[I+I]-I+J]:=KT[6-I,6-J];  
'FOR' I:=6 'STEP' 1 'UNTIL' 4*N1*5 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' 5 'DO' K[ADD[I+6-J]-TI+J+I-11]:=KT[J,I];  
'END' OF PROCEDURE CCRMM12;
```

```
'PROCEDURE' KARBDHNST(NFREE,N1,DA,D,ADD);  
'VALUE' NFREE,N1;  
'INTEGER' BAND,NFREE,N1;  
'INTEGER' 'ARRAY' ADD;  
'BEGIN' 'INTEGER' I,J,K,V;  
V:=NFREE-2*N1; K:=0;  
'FOR' I:=1 'STEP' 1 'UNTIL' V 'DO'  
'BEGIN' J:=I+2*N1-(ADD[I+2*N1]-ADD[I+2*N1-1])+1;  
'IF' J 'LT' 2*N1 'THEN' J:=2*N1+1;  
'IF' I=1 'THEN' BAND:=1 'ELSE' BAND:=(I+2*N1-J+K+1);  
K:=BAND;  
'END';  
BAND:=BAND+5*(NFREE-2*N1)+15;  
'IF' ADD[NFREE] 'GT' BAND 'THEN' BAND:=ADD[NFREE];  
'END' OF PROCEDURE KARBDHNST;
```

```
'PROCEDURE' RESIDUAL(K,Q,ADD,NFREE,NSETF);  
'VALUE' NFREE,NSETF;  
'INTEGER' NFREE,NSETF;  
'INTEGER' 'ARRAY' ADD;  
'REAL' 'ARRAY' K,Q;  
'BEGIN' 'INTEGER' I,J,Z,DI,DJ,CI,CJ;  
'ARRAY' F(1:NFREE,1:NSETF);  
'FOR' I:=1 'STEP' 1 'UNTIL' NFREE 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' F[I,J]:=0.0;  
'FOR' Z:=1 'STEP' 1 'UNTIL' NSETF 'DO'  
'FOR' I:=1 'STEP' 1 'UNTIL' NFREE 'DO'  
'BEGIN' CI:=I-ADD[I]+ADD[I-1]+1;  
'FOR' J:=CI 'STEP' 1 'UNTIL' NFREE 'DO'  
'BEGIN' 'IF' J 'LE' I 'THEN'  
'BEGIN' DI:=I; DJ:=J;  
'GOTO' LAB1;  
'END' 'ELSE'  
'BEGIN' CJ:=J-ADD[J]+ADD[J-1]+1;  
'IF' CJ 'GT' I 'THEN' 'GOTO' LAB2;  
DI:=J; DJ:=I;  
'END';  
LAB1: F[I,Z]:=F[I,Z]+Q[J,Z]*K[ADD[DI]-DI+DJ];  
LAB2: 'END';  
'END';
```



```
WRITETEXT('('('4C')'RESIDUALNFORCES')');
'FOR' I:=1 'STEP' 1 'UNTIL' NSETE 'DO'
'BEGIN' NEWLINE(4);
'FOR' J:=1 'STEP' 1 'UNTIL' NFREE/2 'DO'
'BEGIN' PRINT(F[2+J-1,I],0,5);  SPACE(2);
      PRINT(F[2+J,I],0,5);
      NEWLINE(2);
'END';
'END';
'END' OF PROCEDURE RESIDUAL;

'PROCEDURE' CONSTREL(Z,MATNO,A);
'INTEGER' MATNO; 'ARRAY' A,Z;
'BEGIN' 'INTEGER' I,J;
'REAL' C11,C12,C13,C14,C22,C23,C24,C33,C34,C44;
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO' A[I]:=READ;
WRITETEXT('('('2C4S')'ELASTICITYCONSTANTS('2C3S')'E12=')');
PRINT(A[1],0,6); WRITETEXT('('('3S')'V11=')');
PRINT(A[2],0,6); WRITETEXT('('('2C3S')'E24=')');
PRINT(A[4],0,6); WRITETEXT('('('3S')'V21=')');
PRINT(A[5],0,6); WRITETEXT('('('2C3S')'E23=')');
PRINT(A[3],0,6);
C14:=C24:=C34:=0.0;
C44:=A[3];
A[3]:=A[1]/((1+A[5])*(1-A[5]-2*(A[4]/A[1])*(A[2]*A[2])));
C11:=A[3]*(1-A[5]*A[5]);
C12:=C13:=C23:=A[3]*(A[4]/A[1])*(A[2]*(1-A[5]));
C22:=C33:=(A[4]/A[1])*(1-(A[4]/A[1])*(A[2]*A[2]))*A[3];
Z[MATNO,1]:=C11; Z[MATNO,2]:=C12;
Z[MATNO,3]:=C13; Z[MATNO,4]:=C14;
Z[MATNO,5]:=C22; Z[MATNO,6]:=C23;
Z[MATNO,7]:=C24; Z[MATNO,8]:=C33;
Z[MATNO,9]:=C34; Z[MATNO,10]:=C44;
'END' OF PROCEDURE CONSTREL;

'PROCEDURE' MMINPUT(ADD,XX,YY,NFREE,NNODE,NSETE,NSPEC,NELEMT,ISETFS,
NMAT,MTNO,NSTEL,STEL,ULX,VLY,NELEMT1,NODE,N1,G,H,AA,RO);
'VALUE' NFREE,NNODE,NSETE,NSPEC,NELEMT,ISETFS;
'INTEGER' NFREE,NNODE,NSETE,NSPEC,NELEMT,ISETFS,N1,NMAT;
'REAL' G,H,AA,RO;
'INTEGER' 'ARRAY' ADD,KODE,NODE,MTNO,NSTEL,STEL;
'REAL' 'ARRAY' XX,YY,ULX,VLY;
'BEGIN' 'INTEGER' I,J,W,B,L,D,NM1,NELEMT1,NELEMT2;
'REAL' XS,XF,YS,YF,DELTA,DELTA,DELTA,PHI;
'FOR' I:=0 'STEP' 2 'UNTIL' NFREE 'DO' ADD[I]:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO' XXC[I]=0.0001;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSETE 'DO' X[DECI,J]:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSETE 'DO'
ULX[I,J]:=VLY[I,J]:=0.0;
PHI:=READ;
PHI:=0.0174533*PHI;
NM1:=N1-4;
NELEMT1:=3*(N1-1);
NELEMT2:=NELEMT-NELEMT1;
```



```
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'  
'BEGIN'  
B:=I-1; F:=(I+1)/2;  
'FOR' J:=1 'STEP' 1 'UNTIL' 11 'DO'  
'BEGIN'  
TH:=(2*(J-1)*3.14159)/(N1-1);  
XX[(B*N1)+J]:=(G-(AA*COS(PHI)))+( ((F*RO)-COS(TH))*COS(PHI))  
+(((F*RO)*SIN(TH))*SIN(PHI));  
YY[(B*N1)+J]:=((AA-(F*RO))*SIN(PHI))+((F*RO)*(1-COS(TH)))*SIN(PHI)  
+(((F*RO)*SIN(TH))*COS(PHI));  
'END';  
'END';  
'FOR' I:=7 'DO'  
'BEGIN'  
B:=I-1;  
'FOR' J:=1,3,5,17 'DO'  
XX[(B*N1)+J]:=G;  
'FOR' J:=7,15 'DO'  
XX[(B*N1)+J]:=G-((AA+(5*RO))/2);  
'FOR' J:=9,11,13 'DO'  
XX[(B*N1)+J]:=G-(AA+(5*RO));  
'FOR' J:=1,13,15,17 'DO'  
YY[(B*N1)+J]:=0,0;  
'FOR' J:=3,11 'DO'  
YY[(B*N1)+J]:=(AA+(5*RO))/2;  
'FOR' J:=5,7,9 'DO'  
YY[(B*N1)+J]:=AA+(5*RO);  
'END';  
'FOR' I:=1 'STEP' 1 'UNTIL' 7 'DO'  
'BEGIN'  
B:=I-1;  
'FOR' J:=1,13 'DO'  
XX[(B*NN1)+J+132]:=G;  
'FOR' J:=5,7,9 'DO'  
XX[(B*NN1)+J+132]:=(4-((I+1)/2))*((G-(AA+(5*RO)))/4);  
XX[(B*NN1)+3+132]:=XX[(B*NN1)+11+132]:=-XX[(B*NN1)+5+132]+  
((G-XX[(B*NN1)+5+132])/2);  
'FOR' J:=1,3,5 'DO'  
YY[(B*NN1)+J+132]:=AA+(5*RO)-((4-(AA+(5*RO)))/4)*((I+1)/2);  
'FOR' J:=9,11,13 'DO'  
YY[(B*NN1)+J+132]:=-((I+1)/2)*(4/4);  
YY[(B*NN1)+7+132]:=(YY[(B*NN1)+5+132]+YY[(B*NN1)+9+132])/2;  
'END';  
NSPEC:=READ;  
'FOR' I:=1 'STEP' 1 'UNTIL' NSPEC 'DO'  
'BEGIN'  
J:=READ;  
KODE[J,1]:=READ; ULXE[J,1]:=READ; VLYC[J,1]:=READ;  
'END';  
'FOR' W:=1 'STEP' 1 'UNTIL' 4 'DO'  
MTNO[W]:=1;  
'BEGIN'  
'IF' NMAT=1 'THEN'  
'GOTO' RWA4  
'ELSE'
```



```
'BEGIN'  
'FOR' I:=2 'STEP' 1 'UNTIL' NELENT 'DO'  
'BEGIN'  
NSTEL[I]:=READ;  
'FOR' J:=1 'STEP' 1 'UNTIL' NSTEL[I] 'DO'  
STEL[I,J]:=READ;  
'FOR' W:=1 'STEP' 1 'UNTIL' NELENT 'DO'  
'BEGIN'  
J:=1;  
RWA1:'IF' W=STEL[I,J] 'THEN'  
'GOTO' RWA2  
'ELSE'  
J:=J+1;  
'IF' J 'LE' NSTEL[I] 'THEN'  
'GOTO' RWA1  
'ELSE'  
'GOTO' RWA3;  
RWA2:MTNO[W]:=I;  
RWA3:'END';  
'END';  
'END';  
RWA4:'END';  
'FOR' I:=1 'STEP' 2 'UNTIL' NELENT-1 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
ALI:'IF' I 'LE' (L*(N1-1))-1 'THEN'  
'BEGIN'  
B:=I-((L*N1)-N1-L+2);  
W:=I;  
'IF' I 'LE' (L*(N1-1))-((N1+1)/2) 'THEN'  
'BEGIN'  
NODE[W,1]:=1+B+D;  
NODE[W,2]:=(2*N1)+1+B+D;  
NODE[W,3]:=(2*N1)+3+B+D;  
NODE[W,4]:=N1+1+B+D;  
NODE[W,5]:=(2*N1)+2+B+D;  
NODE[W,6]:=N1+2+B+D;  
NODE[W,7]:=MTNO[W];  
'END'  
'ELSE'  
'BEGIN'  
NODE[W,1]:=1+B+D;  
NODE[W,2]:=(2*N1)+1+B+D;  
NODE[W,3]:=3+B+D;  
NODE[W,4]:=N1+1+B+D;  
NODE[W,5]:=N1+2+B+D;  
NODE[W,6]:=2+B+D;  
NODE[W,7]:=MTNO[W];  
'END';  
'END'
```



```
'ELSE'  
'BEGIN'  
L:=L+1;  
D:=D+(2*N1);  
'GOTO' ALI;  
'END';  
'END';  
'FOR' I:=2 'STEP' 2 'UNTIL' NLEN+1 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
RANA: 'IF' I 'LE' (L*(N1-1)) 'THEN'  
'BEGIN'  
B:=I-((L*N1)-N1-L+3);  
W:=I;  
'IF' I 'LE' (L*(N1-1))-((N1-1)/2) 'THEN'  
'BEGIN'  
NODE[W,1]:=1+B*D;  
NODE[W,2]:=(2*N1)+3+B*D;  
NODE[W,3]:=3+B*D;  
NODE[W,4]:=N1+2+B*D;  
NODE[W,5]:=N1+3+B*D;  
NODE[W,6]:=2+B*D;  
NODE[W,7]:=MTNO[W];  
'END'  
'ELSE'  
'BEGIN'  
NODE[W,1]:=3+B*D;  
NODE[W,2]:=(2*N1)+1+B*D;  
NODE[W,3]:=(2*N1)+3+B*D;  
NODE[W,4]:=N1+2+B*D;  
NODE[W,5]:=(2*N1)+2+B*D;  
NODE[W,6]:=N1+3+B*D;  
NODE[W,7]:=MTNO[W];  
'END';  
'END'  
'ELSE'  
'BEGIN'  
L:=L+1;  
D:=D+(2*N1);  
'GOTO' RANA;  
'END';  
'END';  
'FOR' I:=1 'STEP' 2 'UNTIL' NLEN+1 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
ALI2: 'IF' I 'LE' (L*(N1-1))-1 'THEN'
```



```
'BEGIN'  
B:=I-((L*NN1)-NN1-L+2);  
W:=I+48;  
NODE[W,1]:=1+R+D+106;  
NODE[W,2]:=(2*NN1)+1+R+D+106;  
NODE[W,3]:=(2*NN1)+3+R+D+106;  
NODE[W,4]:=NN1+1+R+D+106;  
NODE[W,5]:=(2*NN1)+2+R+D+106;  
NODE[W,6]:=NN1+2+R+D+106;  
NODE[W,7]:=MTNO[W];  
'END'  
'ELSE'  
'BEGIN'  
L:=L+1;  
D:=D+(2*NN1);  
'GOTO' ALI2;  
'END';  
'END';  
'FOR' I:=2 'STEP' 2 'UNTIL' NELENT2 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
RANA2:'IF' I 'LE' (L*(NN1-1)) 'THEN'  
'BEGIN'  
B:=I-((L*NN1)-NN1-L+3);  
W:=I+48;  
NODE[W,1]:=1+R+D+106;  
NODE[W,2]:=(2*NN1)+3+R+D+106;  
NODE[W,3]:=3+R+D+106;  
NODE[W,4]:=NN1+2+R+D+106;  
NODE[W,5]:=NN1+3+R+D+106;  
NODE[W,6]:=2+R+D+106;  
NODE[W,7]:=MTNO[W];  
'END'  
'ELSE'  
'BEGIN'  
L:=L+1;  
D:=D+(2*NN1);  
'GOTO' RANA2;  
'END';  
'END';  
WRITETEXT('C'('2C')'NODALXCONNECTIONS%76/8808ENXREAD');  
'FOR' W:=1 'STEP' 1 'UNTIL' NELENT 'DO'  
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'  
'BEGIN' 'IF' XX[NODE[W,3+I]]=0.00001 'THEN'  
'BEGIN'  
'IF' I=1 'THEN' J:=1 'ELSE' 'IF' I=2 'THEN' J:=2 'ELSE' J:=0;  
XX[NODE[W,3+I]]:=(XX[NODE[W,I]]*XX[NODE[W,1+J]])/2;  
YY[NODE[W,3+I]]:=(YY[NODE[W,I]]+YY[NODE[W,1+J]])/2;  
'END';  
'END';  
WRITETEXT('C'('2C')'NODALXPOINT%DATA'('224S')'NODE'('5S')'XXCOORD'  
'('5S')'ZCOORD'('5S')'TYPE'('5S')'R-DISP'('5S')'Z-DISP'('046S')'  
'ORXLOAD'('4S')'OR%LOAD');  
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
```



```
'BEGIN' NEWLINE(1); SPACE(3);
PRINT(I,3,0); SPACE(3);
PRINT(XX(I),0,3);
PRINT(YY(I),0,3); SPACE(2);
PRINT(KODE(I,1),3,0); SPACE(2);
PRINT(ULX(I,1),0,3);
PRINT(VLY(I,1),0,3);
'END';
WRITETEXT('('('4C')'ELEMENTXDATA'('2C')'ELEMENT'('1BS')'
NODALXCONNECTIONS')));
'FOR' W:=1 'STEP' 1 'UNTIL' NELEMT 'DO'
```

```
'BEGIN' NEWLINE(1);
PRINT(W,3,0); SPACE(6);
'FOR' J:=1 'STEP' 1 'UNTIL' 7 'DO'
'BEGIN' PRINT(NODE(W,J),3,0);
SPACE(2);
'END';
'END';
```

```
'END' OF PROCEDURE MMINPUT;
'PROCEDURE' ASSEMBLY(NELEMT,K,XX,YY,DETJ,NODE,C,THICK,ADD);
'INTEGER' NELEMT; 'REAL' DETJ,THICK;
```

```
'REAL' 'ARRAY' K,XX,YY,C; 'INTEGER' 'ARRAY' NODE,ADD;
'BEGIN' 'INTEGER' I,J,U,Z,V,SUB2,SUB1,SUB3;
'REAL' 'ARRAY' B[1,4,1,12],KE[1,12,1,12],W[1,6,1,4];
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' W[I,1]=0.33333333;
W[1,2]=W[1,3]=W[2,3]=W[2,4]=W[3,2]=W[3,4]=0.5;
W[1,4]=W[2,2]=W[3,3]=W[4,4]=W[5,2]=W[6,3]=0.0;
W[4,2]=W[4,3]=W[5,3]=W[5,4]=W[6,2]=W[6,4]=1.0;
'FOR' I:=1 'STEP' 1 'UNTIL' ADD(NFREE) 'DO' K[I]=0.0;
```

```
'FOR' I:=1 'STEP' 1 'UNTIL' NELEMT 'DO'
'BEGIN' 'COMMENT' Z HSA REPLACED W TO OVERCOME A PROBLEM
OF AN INCORRECT IDENTIFIER PROBABLY CONFUSED WITH W ARRAY;
'COMMENT' THE ELEMENT STIFFNESS ARRAY IS INITIALISED;
'FOR' I:=1 'STEP' 1 'UNTIL' 12 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 12 'DO' KE[I,J]=0.0;
'COMMENT' FOR LATER ADAPTATION THIS WILL BE STORED AS
A ONE DIMENSIONAL ARRAY
THE LOOP FOR THE NUMBER OF INT PTS IS CONSTRUCTED;
```

```
'FOR' U:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
```

```
STRDIS(W[U,2],W[U,3],W[U,4],B,XX,YY,DETJ,NODE,Z);
```

```
'FOR' J:=1 'STEP' 2 'UNTIL' 11 'DO'
```

```
'FOR' I:=J 'STEP' 2 'UNTIL' 11 'DO' -
```

```
'BEGIN'
```

```
KE[J,I]=KE[I,J]=KE[I,J]+W[U,1]*(B[1,J]*(C[NODE[Z,7],1]*B[1,I]+
```

```
C[NODE[Z,7],2]*B[2,I]+C[NODE[Z,7],4]*B[4,I])+
```

```
B[2,J]*(C[NODE[Z,7],2]*B[1,I]+C[NODE[Z,7],5]*B[2,I]
```

```
+C[NODE[Z,7],7]*B[4,I])+
```

```
B[4,J]*(C[NODE[Z,7],4]*B[1,I]+C[NODE[Z,7],7]*B[2,I]
```

```
+C[NODE[Z,7],10]*B[4,I))*2*3.1415926*RAVG*0.5*DETJ;
```

```
KE[J,I+1]=KE[I+1,J]=KE[I+1,J]+W[U,1]*(B[1,J]*(C[NODE[Z,7],3]*B[3,I+1]
```

```
+C[NODE[Z,7],4]*B[4,I+1])+
```

```
B[2,J]*(C[NODE[Z,7],6]*B[3,I+1]+C[NODE[Z,7],7]*B[4,I+1])+
```

```
B[4,J]*(C[NODE[Z,7],9]*B[3,I+1]+C[NODE[Z,7],10]*B[4,I+1]))
```

```
*2*3.1415926*RAVG*0.5*DETJ;
```

```
'END';
```



```
'FOR' J:=2 'STEP' 2 'UNTIL' 12 'DO'  
'FOR' I:=J 'STEP' 2 'UNTIL' 12 'DO'  
'BEGIN'  
KE[J,I]:=KE[I,J]:=KE[I,J]+W[U,1]*(R[3,J]*(C[NODE[Z,7],8]*R[3,1]+  
C[NODE[Z,7],9]*B[4,1])+  
B[4,J]*(C[NODE[Z,7],9]*B[3,1]+C[NODE[Z,7],10]*B[4,1]))  
*2*3.1415926*RAVG*0.5*DETJ;  
'IF' I=12 'THEN' 'GOTO' L6 'ELSE'  
KE[J,I+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]*(B[3,J]*(C[NODE[Z,7],3]*R[1,I+1]+  
C[NODE[Z,7],6]*B[2,I+1]+C[NODE[Z,7],9]*B[4,I+1])+  
B[4,J]*(C[NODE[Z,7],4]*B[1,I+1]+C[NODE[Z,7],7]*B[2,I+1]+  
C[NODE[Z,7],10]*B[4,I+1]))*2*3.1415926*RAVG*0.5*DETJ;  
L6: 'END';  
'END';  
'COMMENT' ASSEMBLY OF OVERALL STIFFNESS MATRIX AS A  
ONE-DIMENSIONAL ARRAY;
```

```
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'  
'FOR' V:=1,0 'DO'  
'BEGIN' SUB1:=NODE[Z,I]*2-1;  
SUB2:=NODE[Z,J]*2-V;  
SUB3:=NODE[Z,I]*2;  
'IF' SUB1 'LT' SUB2 'THEN' 'GOTO' LABA;  
K[ADD(SUB1)-SUB1+SUB2]:=K[ADD(SUB1)-SUB1+SUB2]+KE[I*2-1,J*2-V];  
LABA: 'IF' SUB3 'LT' SUB2 'THEN' 'GOTO' LABB;  
K[ADD(SUB3)-SUB3+SUB2]:=K[ADD(SUB3)-SUB3+SUB2]+KE[I*2,J*2-V];  
LABB: 'END';  
'END';
```

```
'END' OF PROCEDURE ASSEMBLY;  
'PROCEDURE' E04AAA(X,F,ABSACC,RELACC,XSTEP,FUNCT,MAXFUN,IFAIL);  
'VALUE' ABSACC,RELACC,XSTEP,MAXFUN;  
'REAL' X,F,ABSACC,RELACC,XSTEP;  
'INTEGER' IFAIL,MAXFUN;  
'PROCEDURE' FUNCT; 'ALGOL';  
'PROCEDURE' FUNCT(TH,SS);  
'VALUE' TH;  
'REAL' TH,SS;  
SS:=(K1+2)*CON*((3-4*MU-COS(TH))*(1+COS(TH)))+  
((2*K1*K2)*CON*(2*SIN(TH))*(COS(TH)-1+2*MU))+  
((K2+2)*CON*((4*(1-MU)*(1-COS(TH)))+(1+COS(TH))*((3*COS(TH)-1)))));  
NJOB:=READ;  
'FOR' COUNT:=1 'STEP' 1 'UNTIL' NJOB 'DO'  
'BEGIN'  
WRITETEXT('('('2C')IJOBXNAMEX-----)');  
  
COPYTEXT('('ENDXOFX,I,LE)');  
NELEMT:=READ; NNODE:=READ;  
WRITETEXT('('('2C')NOXOFXELEMENTSX-----)');  
PRINT(NELEMT,3,0);  
WRITETEXT('('('2C')NOXOFXNODESX-----)');  
PRINT(NNODE,3,0);  
NFREE:=NNODE*2; NSETFS:=READ; NMAT:=READ;  
'BEGIN' 'INTEGER' BAND,NSPEC,Z,COMPA,NSETF,N1,NG;  
'REAL' RO,G,H,AA,CR,GG,ANGLE;  
'REAL' 'ARRAY' XX,YY[1;NNODE],ULX,VLY[1;NNODE,1;NSETFS],
```



```
C(1:NMAT,1:10),A(1:5),Q(1:NFREE,1:NSETFS),CM(1:2);
  'INTEGER' 'ARRAY' NODE(1:NELEMT,1:7),KODE(1:NNODE,1:NSETFS),
ADD(0:NFREE),MTNO(1:NELEMT),NSTEL(1:NMAT),STEL(1:NMAT,1:NELEMT);
N1:=READ; G:=READ; H:=READ; AA:=READ; RO:=READ;
NSETC:=READ; CR=G-AA; CASE:=READ;
'FOR' S,=1 'STEP' 1 'UNTIL' NSETC 'DO'
'BEGIN'
'IF' S=1 'THEN' 'BEGIN'
NSETF:=READ;
MMINPUT(ADD,XX,YY,NFREE,NNODE,NSETF,KODE,NSPEC,NSETFS,NMAT,MTNO,
NSTEL,STEL,ULX,VLY,NELEMT,NODE,N1,G,H,AA,RO);
ADDARRAY(NELEMT,NNODE,ADD,NODE);
NG:=NFREE;
KARBDMMST(NFREE,N1,BAND,ADD);
'FOR' MATNO,=1 'STEP' 1 'UNTIL' NMAT 'DO'
CONSTREL(C,MATNO,A);
WRITETEXT('('('('2C')'NOXOFXDEGREESXOFXFREEDOM----')');
PRINT(NFREE,3,0);
NEWLINE(2);
'END' 'ELSE'
'BEGIN' NNEWC:=READ;
      NSETF:=READ;
'FOR' I,=1 'STEP' 1 'UNTIL' NNEWC 'DO'

'BEGIN' J:=READ;
      KODE[J,1]:=READ; ULX[J,1]:=READ; VLY[J,1]:=READ;
'END';
'END';
'FOR' I,=1 'STEP' 1 'UNTIL' NFREE 'DO'
'FOR' J,=1 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]:=0.0;
'FOR' I,=1 'STEP' 1 'UNTIL' NNODE 'DO'
      LOADING(KODE(I,1),ULX(I,1),VLY(I,1),Q,I,1);
'IF' NSETF 'GT' 1 'THEN'
'BEGIN'
'FOR' I,=2 'STEP' 1 'UNTIL' NSETF 'DO'
'BEGIN'
NSPEC:=READ;
WRITETEXT('('('('2C')'FORCEXSET----')');
PRINT(I,3,0);
WRITETEXT('('('('2C')'NODE'('5S')'TYPE'('8S')'R-DISP'('7S')'Z-DISP
('C21S')'ORXLOAD'('6S')'ORXLOAD')');
'FOR' J,=1 'STEP' 1 'UNTIL' NSPEC 'DO'
'BEGIN'
      K:=READ; KODE[K,I]:=READ; ULX[K,I]:=READ; VLY[K,I]:=READ;
LOADING(KODE[K,I],ULX[K,I],VLY[K,I],Q,K,I);
NEWLINE(2);
PRINT(K,3,0); SPACE(2);
PRINT(KODE[K,I],3,0); SPACE(2);
PRINT(ULX[K,I],0,4);
PRINT(VLY[K,I],0,4);
'END';
'END';
'END';
NEWLINE(6);
'BEGIN' 'INTEGER' SUB1,SUB2,SUB3,SUBSP;
'REAL' 'ARRAY' K[1,BAND];
```



```
ASSEMBLY(NFELEM,K,XX,YY,DETJ,NODE,C,THICK,ADD);
NEWLINE(6);
CCRMM12(N1,RO,NG,NSETF,CH,ADD,K,Q,CASE,THICK,CR);
CREATESTORE(10,('ED'),'('KAFILEST)'),1,60000);
NVAB1:=1;
PUTARRAY(10,NVAB1,K);
PUTARRAY(10,NVAB1,ADD);
'COMMENT' INTRODUCTION OF KINEMATIC CONSTRAINTS;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE=N1 'DO'
  'BEGIN' J:=I+N1;
  'IF' KODE[Z,1]=0 'THEN' 'GOTO' KC1;
  'IF' KODE[Z,1]=2 'THEN' 'GOTO' KC2;
  BOUNCONST(ULX[Z,1],2*I-1,Q,K,NG-2*N1+5,1,ADD);
  'FOR' J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]:=Q[I,1];
  'IF' KODE[Z,1]=1 'THEN' 'GOTO' KC1;
  KC2: BOUNCONST(VLY[Z,1],2*I,Q,K,NG-2*N1+5,1,ADD);
  'FOR' J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]:=Q[I,1];
  KC1: 'END';
WRITETEXT('('('2C')'SYNVBSOLXXBEGIN')');
SYNVBSOL(K,K,ADD,Q,NG-2*N1+5,NSETF,FAIL);
NEWLINE(6);
'FOR' I:=1 'STEP' 1 'UNTIL' NSETF 'DO'
  'BEGIN' WRITETEXT('('('4C')'NODALXDISPLACEMENTS%FOR%FORCEXSET')');
  PRINT(1,2,0);
  WRITETEXT('('('3CS')'NODE('5S')'R=DIRECTION('3S')'Z=DIRECTION
('18S')'NODE('5S')'R=DIRECTION('8S')'Z=DIRECTION')');
  U:=2*((NNODE=N1)^(1/2));
  'FOR' J:=2 'STEP' 2 'UNTIL' U 'DO'
  'BEGIN' NEWLINE(2); V:=2*(J-1);

      Z:=J*N1;
      PRINT(Z=1,3,0);SPACE(2);PRINT(Q[V=1,I],0,8);SPACE(2);
      PRINT(Q[V,I],0,8);SPACE(13);PRINT(Z,3,0);SPACE(2);
      PRINT(Q[V+1,I],0,8);SPACE(2);PRINT(Q[V+2,I],0,8);

  'END';
  'IF' NNODE=N1 'GT' U 'THEN'
  'BEGIN' NEWLINE(2); PRINT(NNODE=N1,3,0);SPACE(2);
  V:=2*(NNODE=N1);
  PRINT(Q[V=1,I],0,8);SPACE(2);PRINT(Q[V,I],0,8);
  'END';
V:=NG-2*N1;
WRITETEXT('('('4C')'CRACKXTIPXDISPLACEMENT%INXZ=DIRECTION='));
PRINT(Q[V+3,I],0,10);
WRITETEXT('('('2C')'CRACKXTIPXDISPLACEMENT%INXZ=DIRECTION='));
PRINT(Q[V+2,I],0,10);
WRITETEXT('('('4C')'MODE1XSTRESS%INTENSITY%FACTOR%KI='));
PRINT(Q[V+5,I],0,10);
WRITETEXT('('('2C')'MODE2XSTRESS%INTENSITY%FACTOR%KI='));
PRINT((Q[V+4,I]*(-1)),0,10);
WRITETEXT('('('4C')'RIGIDXBODY%ROTATION%W='));
PRINT(Q[V+1,I],0,10);
```



```
NEWLINE(5);
MU:=CM[2];
GG:=0.5*CM[1]/(1+CM[2]);
CON:=(3.1415926*CR)/(8*GG);
K1:=Q[V*5,1];

K2:=Q[V*4,1];
TH:=-1.5707963;
RELACC:=0.0;
ABSACC:=0.0174;
XSTEP:=3.1415926;
MAXFUN:=150;
IFAIL:=0;
EO4AAA(TH,SS,ABSACC,RELACC,XSTEP,FUNCT,MAXFUN,IFAIL);
ANGLE:=(3.1415926*TH)*57.3;
WRITETEXT('MIN%AT%TH=');
PRINT(ANGLE,0,6);
NEWLINE(5);
'END';
NVAB1:=1;
GETARRAY(10,NVAB1,K);
GETARRAY(10,NVAB1,ADD);
RESIDUAL(K,Q,ADD,NG=2*N1*5,NSETF);
'END';
'END';
'END';
FAIL:'END';
'END';
```

III) Input data instructions

A) Number of jobs to be solved.

B) For the first job.

1. The job name followed by end of title.
2. The number of elements.
3. The number of nodes.
4. Number of sets of forces.
5. Number of materials.
6. Number of nodes on circular core
7. Radius of the specimen.
8. Half the length of the specimen.
9. Crack length.
10. Singular core radius
11. Number of sets of constraints.
12. Case

C) For the first set of constraints.

1. Angle of inclination of crack (θ)⁽¹⁾
2. Number of specified nodes.
3. For the specified nodes:
 - a) Node number
 - b) Kode
 - c) Value of prescribed load or displacement in r direction.
 - d) Value of prescribed load or displacement in z direction.
4. IF number of materials >1 then:
 - a) The number of elements with different material properties for each different material.
 - b) The element numbers of these elements.
5. Elastic constants.

6. If the number of sets of forces for this set of constraints >1 then:
 - a) Number of specified forces.
 - b) For the number of specified forces
 - Node number
 - Kode
 - Value of prescribed load in r direction.
 - Value of prescribed load in z direction.
 7. Elastic constants E and ν of near tip material.
 8. The angle $(180-\theta)$
- D) For the subsequent sets of constraints:
1. Number of new constraints.
 2. Number of sets of forces for this set of constraints.
 3. For the number of new constraints.
 - a) Node number
 - b) Kode
 - c) Value of prescribed load or displacement in r direction.
 - d) Value of prescribed load or displacement in z direction.
- E) For the next job start from B.

Notes

- (1) The angle θ is shown in Fig.(10.7)

IV) Sample problem

For the problem shown in Fig.(10.7), the required data is as follows:

```
1
MODE   END OF TITLE
96,223,1,1,17,10,12,1,0.02,1,1,1,30
13
211,0,0,19634
.
.
.
.
.
216,1,0,0
.
.
.
.
.
223,0,0,-19634
30000000,0.3,12000000,30000000,0.3
30000000,0.3
150
****
```

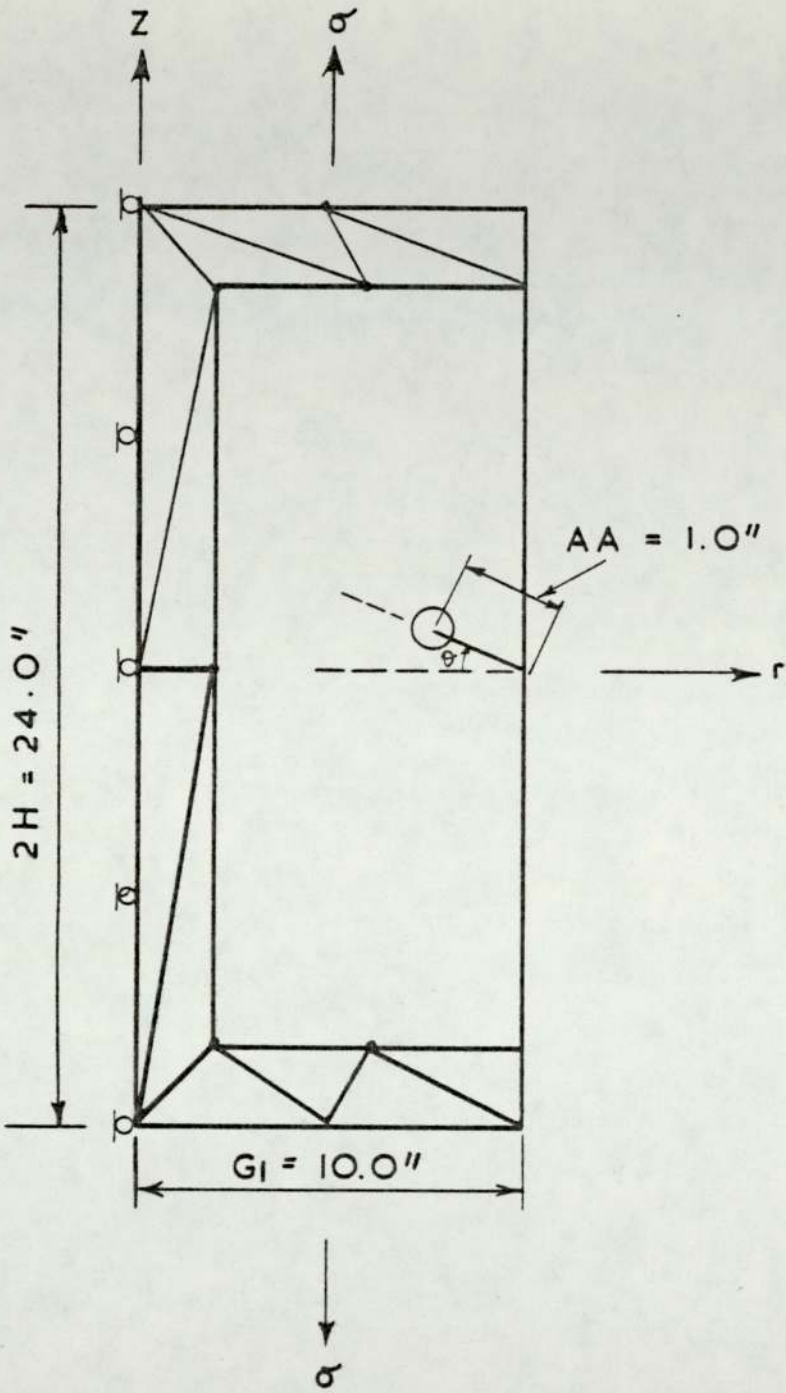



Fig. 10.7

10.4.4 Mixed mode shouldered bar program

This program is identical to the mixed mode program presented in section 10.4.3 except that procedure (SBINPUT) is used instead of (MMINPUT). The input data instructions are also the same except that the coordinates of the nodes specifying the fillet radius are inserted between C(1) and c(2) of the instructions for section (10.4.3) as will be seen in the sample problem.

I) Sample problem

For the problem shown in Fig.(10.8) the required data is as follows:

```
1
SBCRACK  END OF TITLE
108,251,1,1,17,10,12,1,0.02,1,1,1,45
20.0,4.25
13.0,4.25
11.5,3.848
9.0,3.848
3.848
17
235,0,0,11435
.
.
.
.
.
244,1,0,0
.
.
.
.
.
251,0,0,-19634
30000000,0.3,12000000,30000000,0.3
30000000,0.3
135
****
```

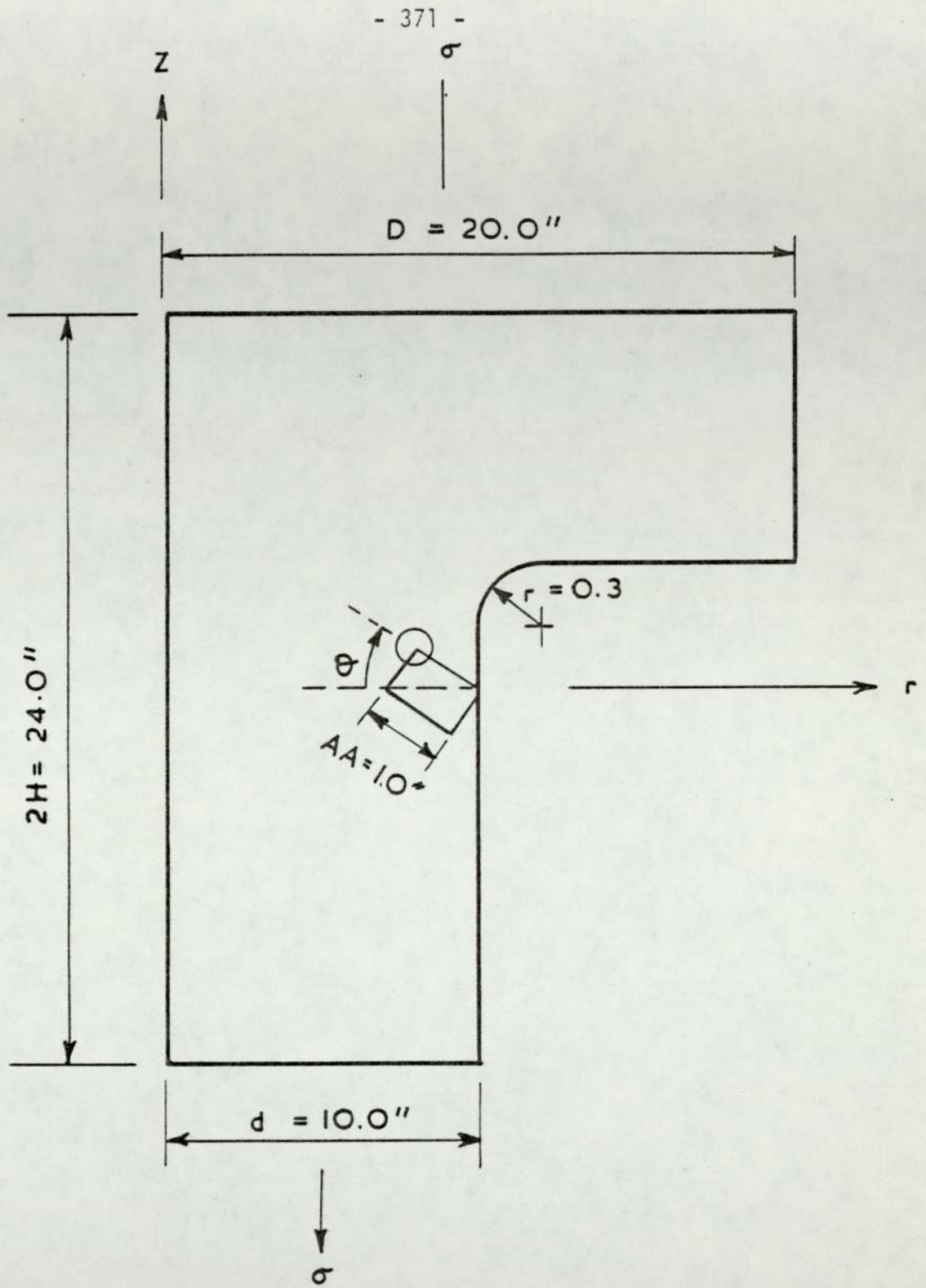



Fig . 10. 8

II) Program listing

```
'PROGRAM' (AXXX)
'INPUT' O=CRO
'OUTPUT' O=LP0
'EXTENDED DATA'
'EXTENDED'
'TRACE' 2
'BEGIN'
'INTEGER' N,LENT,MODE,NSETFS,NFREE,I,J,K,V,W,U,CASE,SEFC,ANRMC,
NIDP,COUNT,PRNT,NVAR1,NPAT,NATNO,IFAT1,IAKFUN;
'REAL' DELTA,DETJ,THICK,RAVG,RMEAN,SS,TH,ABSACC,
RELACC,XSTEP,CO1,T1,K2,DU;
'PROCEDURE' CREATESTORE(N,S,T,G,L);
'VALUE' N,L,G; 'INTEGER' N,L,G; 'STRING' S,T; 'EXTERNAL';
'PROCEDURE' INSTORE(N,S,T,G);
'VALUE' N,G; 'INTEGER' N,G; 'STRING' S,T; 'EXTERNAL';
'PROCEDURE' PUTARRAY(N,K,A);
'VALUE' N; 'INTEGER' I,K; 'ARRAY' A; 'EXTERNAL';
'PROCEDURE' GETARRAY(N,K,A);
'VALUE' N; 'INTEGER' I,K; 'ARRAY' A; 'EXTERNAL';
'PROCEDURE' STRDIS(L1,L2,L3,B,X,Y,U,N,Z);
'VALUE' L1,L2,L3,Z;
'INTEGER' Z;
'REAL' L1,L2,L3,U;
'INTEGER' 'ARRAY' N;
'ARRAY' X,Y,B;
'BEGIN'
'INTEGER' I,V;
'REAL' CHANGE;
'REAL' 'ARRAY' P[1:2,1:0],J[1:2,1:2],U[1:0];
'COMMENT' THIS PROCEDURE EVALUATES THE JACOBIAN J ITS DETERMINANT
AND THE STRAIN=DISP ARRAY B;
J[1,1]:=X[N[Z,1]]*(4*L1-1)+X[N[Z,3]]*(4*L1+4*L2-3)+4*L2*X[N[Z,4]]-
4*L2*X[N[Z,5]]+4*X[N[Z,0]]*(1-2*L1-L2);
J[1,2]:=Y[N[Z,1]]*(4*L1-1)+Y[N[Z,3]]*(4*L1+4*L2-3)+4*L2*Y[N[Z,4]]-
4*L2*Y[N[Z,5]]+4*Y[N[Z,0]]*(1-2*L1-L2);
J[2,1]:=X[N[Z,2]]*(4*L2-1)+X[N[Z,3]]*(4*L1+4*L2-3)+4*L1*X[N[Z,4]]+
4*X[N[Z,5]]*(1-L1-2*L2)-4*L1*X[N[Z,0]];
J[2,2]:=Y[N[Z,2]]*(4*L2-1)+Y[N[Z,3]]*(4*L1+4*L2-3)+4*L1*Y[N[Z,4]]+
4*Y[N[Z,5]]*(1-L1-2*L2)-4*L1*Y[N[Z,0]];
'COMMENT' U REPLACES DETJ;
U:=J[1,1]*J[2,2]-J[1,2]*J[2,1];
'COMMENT' THE COEFFS OF [J] ARE REPLACED BY THOSE OF [J]-1;
CHANGE:=J[1,1];
J[1,1]:=J[2,2]/U;
J[1,2]:=-J[1,2]/U;
J[2,1]:=-J[2,1]/U;
J[2,2]:=CHANGE/U;
```



```

NL[1]:=L1*(2*L1-1);
NL[2]:=L2*(2*L2-1);
NL[3]:=L3*(2*L3-1);
NL[4]:=4*L1*L2;
NL[5]:=4*L2*L3;
NL[6]:=4*L3*L1;
RMEAN:=(X[N[2,1]]+X[N[2,2]]+X[N[2,3]]+Y[N[2,4]]+
X[N[2,5]]+X[N[2,6]])/6;
'IF' Z=101 'OR' Z=103 'THEN'
RAVG:=RMEAN
'ELSE'
'BEGIN'
RAVG:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
RAVG:=RAVG+(X[N[2,I]]*NL[I]);
'END';
'COMMENT' THE P ARRAY REPS THE DIFF COEFFS OF PELL WRY L1 AND L2;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' V:=1,2 'DO' P[V,I]:=0.0;
'FOR' I:=1,2 'DO'
'BEGIN' P[1,I]:=J[I,1]*(4*L1-1);
P[1,2]:=J[I,2]*(4*L2-1);
P[1,3]:=J[I,1]*(1+4*L3)+J[I,2]*(1+4*L3);
P[1,4]:=4*(L2*J[I,1]+L1*J[I,2]);
P[1,5]:=4*(J[I,2]*L3-L2*(J[I,2]+J[I,1]));
P[1,6]:=4*(J[I,1]*L3-L1*(J[I,1]+J[I,2]));
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' 12 'DO'
'FOR' V:=1,2,3,4 'DO' P[V,I]:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN'
B[1,(I*2-1)]:=B[4,(I*2)]:=P[1,I];
B[4,(I*2-1)]:=B[3,(I*2)]:=P[2,I];
B[2,(I*2-1)]:=NL[I]/RAVG;
'END';
'END' OF STRDIS;

'PROCEDURE' BOUNCONST(U,N,R,AK,NEQ,P,A);
'VALUE' U,N,NEQ,P;
'REAL' U;
'INTEGER' N,NEQ,P;
'ARRAY' R,AK;
'INTEGER' I,ARRAY A;
'BEGIN' 'INTEGER' M,K,CJ;
'IF' N=1 'THEN' CJ:=1 'ELSE' CJ:=N-(A[N]-A[CJ-1])*1;
'FOR' K:=CJ 'STEP' 1 'UNTIL' N 'DO'
'BEGIN'
```



```

R(K,F):=R(K,F)-AK[A(K)]-N+K)*U;
AK[A(K)]=1+K:=0.0;
'END';
'IF' U+1 'GT' NEG 'THEN' 'GOTO' LZ;
'FOR' K:=N+1 'STEP' 1 'UNTIL' NEG 'DO';
'BEGIN' CJ:=K-(A(K)-AK(K-1))*1;
'IF' CJ 'LE' 0 'THEN'
'BEGIN'
R(K,F):=R(K,F)-AK[A(K)-K+1]*U;
AK[A(K)-K+1]=0.0;
'END' 'ELSE'
'END';
LZ: AK[A(N)]:=1.0;
R(N,F):=U;
'END' OF BOUNCONST;

'PROCEDURE' LOADING(A,B,C,D,E,F);
'VALUE' A,B,C,E,F;
'INTEGER' A,E,F;
'REAL' B,C;
'ARRAY' D;
'BEGIN'
'INTEGER' K;
'IF' A=3 'THEN' 'GOTO' KLAB1;
K:=2+E;
'IF' A=1 'THEN' 'GOTO' KLAB2;
D(K-1,F):=D(K-1,F)+B;
'IF' A 'NE' 0 'THEN' 'GOTO' KLAB1;
KLAB2: D(K,F):=D(K,F)+C;
KLAB1: 'END' OF LOADING;

'PROCEDURE' SYMBSOL(A,L,S,B) DIMENSIONS: (N,K) FAILURE EXIT: (FAIL)
'VALUE' N,R; 'ARRAY' A,L,S; 'INTEGER' 'ARRAY' S; 'INTEGER' N,F;
'LABEL' FAIL;
'BEGIN'
'INTEGER' G,H,I,J,K,U,P,Q,T,U,V;
'REAL' Y;
H:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
'BEGIN'
T:=I+H-S[I]*1; G:=H+1;
P:=S[T]-1;
'FOR' J:=T 'STEP' 1 'UNTIL' I-1 'DO'
'BEGIN'
Q:=P+1; H:=H+1;
P:=S[J]; K:=J+Q-P;
V:=H-Q; U:=K;
Y:=A[H];
'IF' K 'GT' T 'THEN' U:=U+K-T;

```



```
'FOR' U:=0 'STEP' 1 'UNTIL' H-1 'DO'  
Y:=Y-L[U]*L[U-V];  
Y:=Y/L[H-V]; L[H]:=Y;  
'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'  
B[I,M]:=B[I,M]-B[J,M]*Y;  
'END' J;  
Y:=A[H+1];  
'FOR' U:=0 'STEP' 1 'UNTIL' H 'DO'  
Y:=Y-L[U]^2;  
'IF' Y 'LE' 0 'THEN' 'GOTO' FAIL;  
H:=H+1; Y:=SORT(Y);  
L[H]:=Y;
```

```
'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'  
B[I,M]:=B[I,M]/Y;  
'END' I;  
'COMMENT' REDUCTION COMPLETE;  
'FOR' I:=N 'STEP' -1 'UNTIL' 1 'DO'  
'BEGIN'  
Y:=L[H];  
'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'  
B[I,M]:=B[I,M]/Y;  
'IF' I=1 'THEN' 'GOTO' COMPLETE;  
J:=I; P:=S[I-1];  
'FOR' M:=H-1 'STEP' -1 'UNTIL' P+1 'DO'  
'BEGIN'  
J:=J-1; Y:=L[P];  
'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'  
B[J,M]:=B[J,M]-B[I,M]*Y;  
'END' M;  
H:=P;  
COMPLETE: 'END' I;  
'END' SYMVSOL;  
'PROCEDURE' ADDARRAY(NELEMT, NNODE, ADD, NODE);  
'INTEGER' NELEMT, NNODE;
```

```
'INTEGER' I,ARRAY' NODE,ADD;  
'BEGIN' 'INTEGER' W,CH,I,ADDTMP;  
'FOR' W:=1 'STEP' 1 'UNTIL' NELEMT 'DO'  
'BEGIN' CH:=NODE[W,1];  
'FOR' I:=2 'STEP' 1 'UNTIL' NNODE 'DO'  
'IF' NODE[W,I]'LT'CH 'THEN' CH:=NODE[W,I];  
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'  
'BEGIN' ADDTMP:=NODE[W,I]-CH+1;  
'IF' ADDTMP'GT'ADD[NODE[W,I]+2] 'THEN' ADD[NODE[W,I]+2]:=ADDTMP;  
'END';  
'END';  
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
```



```
'BEGIN' CH:=2*ADD(2*1); I:=2+1;  
ADD(N-1):=ADD(N-2)+CH-1;  
ADD(N):=ADD(N-1)+,N;
```

```
'END';  
'END' OF PROCEDURE ADDARRAY;
```

```
'PROCEDURE' SCORPM12(N1,RO,NG,ISETF,CH,ADD,K,0,CASE,THICK,CR);  
'VALUE' N1,NG,RO,ISETF,CASE,THICK,CR; 'INTEGER' N1,NG,NSETF,CASE;  
'REAL' RO,THICK,CR; 'INTEGER' 'ARRAY' ADD; 'REAL' 'ARRAY' K,0,C;  
'BEGIN' 'COMMENT' MODIFICATION OF OVERALL STIFFNESS MATRIX FOR THE  
MODE I AND II STRESS INTENSITY FACTORS, SINGLE TI;  
'INTEGER' IA,TI,CI,DI,DJ,CJ,JA;  
'REAL' F10,G10,FJ0,GJ0,THETA,KAP,ALPHA,F11,G11,F201,T1,T2,T3,T4,T5,  
SN,CN,FJ1,CJ1,H11,HJ1,H10,HJ0,G,HU;  
'ARRAY' KT(1:5,1:4*N1*5);  
'FOR' I:=1,2 'DO'  
CH(I):=READ;  
G:=0.5*CH(1)/(1+CH(2));  
HU:=CH(2);  
ALPHA:=READ;  
WRITETEXT('('('20')'PROCEDURESCORPM12XBEI=S('G')'THIS PROCEDURE  
MODIFIES THE OVERALL STIFFNESS MATRIX FOR MIXED MODE FRACTURE, SINGLE  
CRACK TIP PRESENT)('20')'ANGLE OF CRACK TIP AT POSITIVE X AXIS, DEG  
%=''); PRINT(ALPHA,0,4);  
WRITETEXT('('('20')'NUMBER OF NODES X AND Y IN ELEMENTS='');  
PRINT(N1,3,0);  
WRITETEXT('('('20')'RADIUS OF CORE ELEMENTS=''); PRINT(RO,0,4);
```

```
WRITETEXT('('('20')'MODULUS OF RIGIDITY, G%=''); PRINT(G,0,4);  
WRITETEXT('('('20')'POISSON'S RATIO, NU%=''); PRINT(HU,0,4);  
ALPHA:=0.0174533*ALPHA; SN:=SIN(ALPHA); CN:=COS(ALPHA);  
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' 4*N1+5 'DO' KT(I,J):=0,0;  
'IF' CASE=0 'THEN' KAP:=(3*HU)/(1+HU) 'ELSE' KAP:=3-4*HU;  
KT(1,1):=2*RO*CR*((3.14159)*2)*(2*KAP-1)/(8*G)*THICK;  
KT(2,2):=2*RO*CR*((3.14159)*2)*(3+2*KAP)/(8*G)*THICK;  
'FOR' I:=1 'STEP' 1 'UNTIL' 2*N1 'DO'  
'BEGIN' IA:=I-2*BITTER(I/2);  
'IF' IA=1 'THEN' THETA:=-3.14159+((2*((I+1)/2-1)*3.14159)/(N1-1))  
'ELSE' THETA:=-3.14159*((2*((I/2)-1)*3.14159)/(N1-1));  
F10:=(RO/2)*0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(4*G);  
G10:=(RO/2)*0.5*((2*KAP+3)*SIN(THETA/2)*SIN(1.5*THETA))/(4*G);  
F11:=(RO/2)*0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(4*G);  
G11:=(RO/2)*0.5*((2*KAP-3)*COS(THETA/2)+COS(1.5*THETA))/(4*G);  
H10:=-RO*SIN(THETA); H11:=RO*COS(THETA);  
'IF' IA=1 'THEN' 'BEGIN' F10:=F10*CN-F11*SN;  
G10:=G10*CN-G11*SN;  
H10:=H10*CN-H11*SN;  
T2:=CN; T4:=SN;
```



```
'END' 'ELSE'
'BEGIN' F10:=F10*S1+F11*C1;
        G10:=G10*S1+G11*C1;
        H10:=H10*S1+H11*C1;
        T2:=S1; T4:=C1;
'END';
'IF' I=1 'THEN' C1:=1 'ELSE' C1:=I-RDN[I]+ADD[I-1]+1;
'FOR' J:=C1 'STEP' 1 'UNTIL' 2*I 'DO'
'BEGIN' 'IF' J 'LE' 1 'THEN'
        'BEGIN' D1:=I; D2:=J;
        'GOTO' HSH1;
        'END' 'ELSE'
        'BEGIN' 'IF' J=1 'THEN' C2:=1 'ELSE' C2:=J-ADD[J]+ADD[J-1]+1;
        'IF' C2 'GT' 1 'THEN' 'GOTO' HSH2;
        D1:=J; D2:=I;
        'END';
HSH1: JA:=J-2*ENTIER(J/2);
'IF' JA=1 'THEN' THETA:=3.14159*((2*(C1+1)/2-1)*3.14159)/(C1-1);
'ELSE' THETA:=3.14159*((2*(C1/2)-1)*3.14159)/(C1-1);
FJ0:=(R0/2)*0.5*((2*KAP-1)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
GJ0:=(R0/2)*0.5*((2*KAP+3)*SIN(THETA/2)*SIN(1.5*THETA))/(4*G);
FJ1:=(R0/2)*0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(4*G);
GJ1:=-R0/2*0.5*((2*KAP-3)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
HJ0:=-R0*SIN(THETA); HJ1:=R0*COS(THETA);
'IF' JA=1 'THEN' 'BEGIN' FJ0:=FJ0*CN-FJ1*SN;
        GJ0:=GJ0*CN-GJ1*SN;
        HJ0:=HJ0*CN-HJ1*SN;
        T1:=S1; T3:=C1;
        'END' 'ELSE'
        'BEGIN' FJ0:=FJ0*S1+FJ1*CN;
        GJ0:=GJ0*S1+GJ1*CN;
        HJ0:=HJ0*S1+HJ1*CN;
        T1:=CN; T3:=S1;
        'END';
TEMP1:=ADD[D1]-D1+D2;
KT[1,1]:=KT[1,1]*I0+FJ0*K[TEMP1];
KT[1,2]:=KT[1,2]*G10+FJ0*K[TEMP1];
```



```

KY[1,5]:=KY[1,5]+GJ0*HJ0*(CTEMP1);
KY[2,5]:=KY[2,5]+GJ0*HJ0*(CTEMP1);
KY[3,5]:=KY[3,5]+T3*HJ0*(CTEMP1);
KY[4,5]:=KY[4,5]+T1*HJ0*(CTEMP1);
KY[5,5]:=KY[5,5]+TJ0*HJ0*(CTEMP1);
MSH2: 'END';
'END';
'FOR' J:=6 'STEP' 1 'UNTIL' 4*N1+5 'DO'
'REGIN' CJ:=J+2*N1-5-ADD[J+2*N1-5]+ADD[CJ+2*N1-6]+1;
'FOR' I:=CJ 'STEP' 1 'UNTIL' 2*N1 'DO'
'REGIN' IA:=I-2*INT((I/2));
'IF' IA=1 'THEN' THETA:=3.14152+((2*((I+1)/2-1)*3.14159)/(I-1));
'ELSE' THETA:=3.14150+((2*((I/2)-1)*3.14152)/(I-1));
FIO:=(RO/2)+0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(4*G);
GIO:=(RO/2)+0.5*((2*KAP+1)*SIN(THETA/2)+SIN(1.5*THETA))/(4*G);
FI1:=(RO/2)+0.5*((2*KAP+1)*SIN(THETA/2)+SIN(1.5*THETA))/(4*G);
GI1:=(RO/2)+0.5*((2*KAP-1)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
HJ0:=RO*SIN(THETA); HJ1:=RO*COS(THETA);
'IF' IA=1 'THEN' 'BEGIN' FIO:=FIO*CN-FI1*SN;
GIO:=GIO*CN-GI1*SN;
HJ0:=HJ0*CN-HJ1*SN;
T1:=CN; T2:=SN;
'END' 'ELSE'
'BEGIN' FIO:=FIO*SN+FI1*CN;
GIO:=GIO*SN+GI1*CN;
HJ0:=HJ0*SN+HJ1*CN;
T1:=SN; T2:=CN;
'END';
TEMP1:=ADD[J+2*N1-5]-J-2*N1+5+1;
KT[1,J]:=KT[1,J]+K(TEMP1)*FIO;
KT[2,J]:=KT[2,J]+K(TEMP1)*GIO;
KT[3,J]:=KT[3,J]+T1*K(TEMP1);
KT[4,J]:=KT[4,J]+T2*K(TEMP1);
KT[5,J]:=KT[5,J]+HJ0*K(TEMP1);
'END';
'END';
TI:=HG-2*N1;
'FOR' I:=1 'STEP' 1 'UNTIL' TI 'DO'
'REGIN' CI:=I+2*N1-ADD[I+2*N1]+ADD[I+2*N1-1]+1;
'IF' CI 'LE' 2*N1 'THEN' CI:=2*N1+1;
ADD[CI]:=I+2*N1-CI+ADD[CI-1]+1;
'FOR' J:=CI 'STEP' 1 'UNTIL' I+2*N1 'DO'
K[ADD[CI]-I+J-2*N1]:=K[ADD[CI+2*N1]-I-2*N1+J];
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO' ADD[CI+I]:=ADD[CI+I-1]+1+1;
'FOR' I:=1 'STEP' 1 'UNTIL' TI 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q[1,J]:=Q[1+2*N1,J];
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q[I+1,J]:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' TI 'DO'

```



```
'FOR' J:=1 'STEP' 1 'UNTIL' 5 'DO' K(ADD(TI+J)-TI-J+1):=0,0;  
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' 1 'DO' K(ADD(TI+1)-I+J):=K(TI-1,0-J);  
'FOR' I:=6 'STEP' 1 'UNTIL' 6*1+5 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' 5 'DO' K(ADD(TI+6-J)-TI+J+1-1):=K(TI,1);  
'END' OF PROCEDURE CORR12;
```

```
'PROCEDURE' KARDDHST(NFREE,N1,BAND,ADD);  
'VALUE' NFREE,N1;  
'INTEGER' BAND,NFREE,N1;  
'INTEGER' 'ARRAY' ADD;  
'BEGIN' 'INTEGER' I,J,K,V;  
V:=NFREE-2*N1; K:=0;  
'FOR' I:=1 'STEP' 1 'UNTIL' V 'DO'  
'BEGIN' J:=I+2*N1-(ADD(I+2*N1)-ADD(I+2*N1-1))+1;  
'IF' J 'LT' 2*N1 'THEN' J:=2*N1+1;  
'IF' I=1 'THEN' BAND:=1 'ELSE' BAND:=I+2*N1-J+K+1;  
K:=BAND;  
'END';  
BAND:=BAND+2*(NFREE-2*N1)+1;  
'IF' ADD[NFREE] 'GT' BAND 'THEN' BAND:=ADD[NFREE];  
'END' OF PROCEDURE KARDDHST;
```

```
'PROCEDURE' RESIDUAL(K,N,ADD,NFREE,NSETF);  
'VALUE' NFREE,NSETF;  
'INTEGER' NFREE,NSETF;  
'INTEGER' 'ARRAY' ADD;  
'REAL' 'ARRAY' K,Q;  
'BEGIN' 'INTEGER' I,I1,Z,DI,DJ,CI,CJ;  
'ARRAY' F(1:NFREE,1:NSETF);  
'FOR' I:=1 'STEP' 1 'UNTIL' NFREE 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' F(I,J):=0,0;  
'FOR' Z:=1 'STEP' 1 'UNTIL' NSETF 'DO'  
'FOR' I:=1 'STEP' 1 'UNTIL' NFREE 'DO'  
'BEGIN' CI:=I+ADD(I)*ADD(I)+1;  
'FOR' J:=CI 'STEP' 1 'UNTIL' NFREE 'DO'  
'BEGIN' 'IF' J 'LT' I 'THEN'  
'BEGIN' DI:=I; DJ:=J;  
'GOTO' LAB1;  
'END' 'ELSE'  
'BEGIN' CJ:=J+ADD(J)*ADD(J)+1;  
'IF' CJ 'GT' I 'THEN' 'GOTO' LAB2;  
DI:=J; DJ:=I;  
'END';  
LAB1: F(I,Z):=F(I,Z)+Q(CJ,Z)*K(ADD(DI)-DI+DJ);  
LAB2: 'END';
```



```
'END';  
WRITETEXT('('('40')'RESIDUALS FORCES')');  
'FOR' I:=1 'STEP' 1 'UNTIL' NPTS 'DO'  
'BEGIN' NEWLINE(6);  
'FOR' J:=1 'STEP' 1 'UNTIL' NPTS/2 'DO'  
'BEGIN' PRINT(F(I2=J-1,I),0,5); SPACE(2);  
PRINT(F(I2=J,I),0,5);  
NEWLINE(2);  
'END';  
'END';  
'END' OF PROCEDURE RESIDUAL;
```

```
'PROCEDURE' CONST=EL(Z, MATNO, A);  
'INTEGER' MATNO; 'ARRAY' A, 4;  
'BEGIN' 'INTEGER' I, J;  
'REAL' C11, C12, C13, C14, C22, C23, C24, C33, C34, C44;  
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO' A[I]=READ;  
WRITETEXT('('('2048')'ELASTICITY CONSTANTS('2035')'E11=%')');  
PRINT(A[1],0,6); WRITETEXT('('('35')'V1=%')');  
PRINT(A[2],0,6); WRITETEXT('('('2035')'E23=%')');  
PRINT(A[4],0,6); WRITETEXT('('('35')'V2=%')');
```

```
PRINT(A[5],0,6); WRITETEXT('('('2035')'G12=%')');  
PRINT(A[3],0,6);  
C14:=C24:=C34:=0,0;  
C44:=A[3];  
A[3]:=A[1]/((1+A[5])*(1-A[5]-2*(A[4]/A[1])*A[2]*A[2]));  
C11:=A[3]*(1-A[5]*A[5]);  
C12:=C13:=C23:=A[3]*(A[4]/A[1])*A[2]*(1+A[5]);  
C22:=C33:=(A[4]/A[1])*(1-(A[4]/A[1])*A[2]*A[2]*A[3]);  
Z[MATNO,1]:=C11; Z[MATNO,2]:=C12;  
Z[MATNO,3]:=C13; Z[MATNO,4]:=C14;  
Z[MATNO,5]:=C22; Z[MATNO,6]:=C23;  
Z[MATNO,7]:=C24; Z[MATNO,8]:=C33;  
Z[MATNO,9]:=C34; Z[MATNO,10]:=C44;  
'END' OF PROCEDURE CONST=EL;
```

```
'PROCEDURE' SHINPUT(ADD, XX, YY, NFREE, NMODE, NSETF, KODE, NSPEC, NSETFS,  
NHAT, HTNO, NSTEL, STEL, DLX, VLY, NELEMT, NMODE, I, G, H, AA, R0);  
'VALUE' NFREE, NMODE, NSETF, NSPEC, NELEMT, NSETFS;  
'INTEGER' NFREE, NMODE, NSETF, NSPEC, NELEMT, NSETFS, N1, NHAT;
```

```
'REAL' G, H, AA, R0;  
'INTEGER' 'ARRAY' ADD, KODE, NMODE, MTNO, NSTEL, STEL;  
'REAL' 'ARRAY' XX, YY, DLX, VLY;  
'BEGIN' 'INTEGER' I, J, W, B, L, D, NM1, NELEMT1, NELEMT2;  
'REAL' XS, XF, YS, YF, DELTAX, DELTAY, F, TH, PHI;  
'FOR' I:=0 'STEP' 2 'UNTIL' NFREE 'DO' ADD[I]=0;  
'FOR' I:=1 'STEP' 1 'UNTIL' NMODE 'DO' X[I]=0, Y[I]=0;
```



```
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' N*NTESLOC K=PEL(I,J):=0;  
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' N*NTES 'DO'  
ULX[I,J]:=VLY[I,J]:=0.0;  
PHI:=READ;  
PHI:=0.0174533*PHI;  
NNT:=N1*4;  
NPLEMT1:=3*(N1-1);  
NPLEMT2:=NPLEMT-NPLEMT1-(N1-1);  
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'  
'BEGIN'  
B:=I-1; P:=(I+1)/2;  
'FOR' J:=1 'STEP' 1 'UNTIL' N1 'DO'  
'BEGIN'  
TH:=(2*(J-1)*3.14159)/(N1-1);  
XX[(B*N1)+J]:=(G-(AA*COS(PHI)))+(C*(F*RO)*COS(TH))*COS(PHI)  
+((C*RO)*SIN(TH))*SIN(PHI);  
YY[(B*N1)+J]:=(AA-(F*RO))*SIN(PHI)+(C*(F*RO)*(1-COS(TH))*SIN(PHI)  
+((C*RO)*SIN(TH))*COS(PHI);  
'END';  
'END';  
'FOR' I:=7 'DO'  
'BEGIN'  
B:=I-1;  
'FOR' J:=1,3,5,17 'DO'  
XX[(B*N1)+J]:=G;  
'FOR' J:=7,15 'DO'  
XX[(B*N1)+J]:=G-(AA+(5*RO))/2;  
'FOR' J:=9,11,13 'DO'  
XX[(B*N1)+J]:=G-(AA+(5*RO));  
'FOR' J:=1,13,15,17 'DO'  
YY[(B*N1)+J]:=0.0;  
'FOR' J:=3,11 'DO'  
YY[(B*N1)+J]:=(AA+(5*RO))/2;  
'FOR' J:=5,7,9 'DO'  
  
YY[(B*N1)+J]:=AA+(5*RO);  
'END';  
'FOR' I:=1 'STEP' 2 'UNTIL' 7 'DO'  
'BEGIN'  
B:=I-1;  
'IF' I=1 'THEN'  
'BEGIN'  
'FOR' J:=1,3,5,7 'DO'  
'BEGIN'  
XX[(B*N1)+J+132]:=READ;  
YY[(B*N1)+J+132]:=READ;
```



```
'END';
'FOR' J:=0 'DO'
  YV[(B*N1)+J+132]:=READ;
'END'
'ELSE'
  'BEGIN'

  'FOR' J:=1,3,5,7 'DO'
    'BEGIN'
    YY[(B*N1)+J+132]:=AA+(5*RO)+((H-(AA+(5*RO)))/4)*((I+1)/2);
    XX[(B*N1)+J+132]:=(2*G)-(((2*G)-(4-((I+1)/2))
      *((G-(AA+(5*RO)))/4))/4)*((J+1)/2));
    'END';
    'FOR' J:=0 'DO'
      YV[(B*N1)+J+132]:=AA+(5*RO)+((H-(AA+(5*RO)))/4)*((I+1)/2);
    'END';
    'FOR' J:=17 'DO'
      XX[(B*N1)+J+132]:=G;
    'FOR' J:=0,11,13 'DO'
      XX[(B*N1)+J+132]:=XX[(I*N1)+13+132]+
        ((G-(XX[(B*N1)+13+132]))/2);
    'FOR' J:=15,16,17 'DO'
      YY[(B*N1)+J+132]:=-((I+1)/2)*(H/4);
      YY[(B*N1)+11+132]:=(YV[(B*N1)+9+132]+YV[(I*N1)+13+132])/2;
    'END';
  NSPEC:=READ;
  'FOR' I:=1 'STEP' 1 'UNTIL' NSPEC 'DO'
    'BEGIN'
    J:=READ;
    KODE[J,1]:=READ;  PLX[J,1]:=READ;  VLY[J,1]:=READ;
    'END';
    'FOR' U:=1 'STEP' 1 'UNTIL' HEIGHT 'DO'
      HTNO[U]:=1;
    'BEGIN'
    'IF' NHAT=1 'THEN'
      'GOTO' RUA4
    'ELSE'
      'BEGIN'
      'FOR' I:=2 'STEP' 1 'UNTIL' NHAT 'DO'
        'BEGIN'
        NSTEL[I]:=READ;
        'FOR' J:=1 'STEP' 1 'UNTIL' NSTEL[I] 'DO'
          STEL[I,J]:=READ;
        'FOR' U:=1 'STEP' 1 'UNTIL' HEIGHT 'DO'
          'BEGIN'
          J:=1;
          RUA1:'IF' U=STEL[I,J] 'THEN'
            'GOTO' RUA2
          'ELSE'
            J:=J+1;
```



```
'IF' J 'LE' NSTE[L][I] 'THEN'  
'GOTO' RUA1  
'ELSE'  
'GOTO' RUA3;  
RUA2:HTND[U]:=I;  
RUA3:'END';  
'END';  
'END';  
RUA4:'END';  
'FOR' I:=1 'STEP' 2 'UNTIL' NELEMT1-1 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
ALL:'IF' I 'LE' (L*(N1-1))-1 'THEN'  
'BEGIN'  
B:=I-((L*N1)-1*(L-2));  
  
U:=I;  
'IF' I 'LE' (L*(N1-1))-((N1+1)/2) 'THEN'  
'BEGIN'  
NODE[W,1]:=1+L+D;  
NODE[U,2]:=(2*N1)+1+B+D;  
NODE[W,3]:=(2*N1)+3+B+D;  
NODE[U,4]:=N1+1+B+D;  
NODE[W,5]:=(2*N1)+2+B+D;  
NODE[U,6]:=1+2*B+D;  
NODE[U,7]:=HTND[U];  
'END'  
'ELSE'  
'BEGIN'  
NODE[W,1]:=1+B+D;  
NODE[U,2]:=(2*N1)+1+B+D;  
NODE[W,3]:=3+B+D;  
NODE[U,4]:=N1+1+B+D;  
NODE[W,5]:=N1+2+B+D;  
NODE[U,6]:=2+B+D;  
NODE[W,7]:=HTND[U];  
'END';  
'END'  
'ELSE'  
'BEGIN'  
L:=L*1;  
D:=D+(2*N1);  
'GOTO' ALL;  
'END';  
'END';  
'FOR' I:=2 'STEP' 2 'UNTIL' NELEMT1 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
RANA:'IF' I 'LE' (L*(N1-1)) 'THEN'
```

```
'BEGIN'  
B:=I-((L*NN1)-N1-L-3);  
W:=1;  
'IF' I 'LE' (L*(NN1-1))-((N1-1)/2) 'THEN'  
'BEGIN'  
NODE[U,1]:=1+B+D;  
NODE[U,2]:=(2*NN1)+1+B+D;  
NODE[U,3]:=3+B+D;  
NODE[U,4]:=N1+2+B+D;  
NODE[U,5]:=N1+3+B+D;  
NODE[U,6]:=2*B+D;  
  
NODE[U,7]:=HTNO[U];  
'END'  
'ELSE'  
'BEGIN'  
NODE[U,1]:=3+B+D;  
NODE[U,2]:=(2*NN1)+1+B+D;  
NODE[U,3]:=(2*NN1)+3+B+D;  
NODE[U,4]:=N1+2+B+D;  
NODE[U,5]:=(2*NN1)+2+B+D;  
NODE[U,6]:=N1+3+B+D;  
NODE[U,7]:=HTNO[U];  
'END'  
'ELSE'  
'BEGIN'  
  
L:=L+1;  
D:=D+(2*NN1);  
'GOTO' RANA;  
'END';  
'END';  
'FOR' I:=1 'STEP' 2 'UNTIL' NN1-2 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
B:=I-((L*NN1)-N1-L-2);  
W:=I+48;  
NODE[U,1]:=1+B+D+100;  
NODE[U,2]:=(2*NN1)+5+B+D+100;  
NODE[U,3]:=(2*NN1)+7+B+D+100;  
NODE[U,4]:=NN1+1+B+D+100;  
NODE[U,5]:=(2*NN1)+6+B+D+100;  
NODE[U,6]:=NN1+2+B+D+100;  
NODE[U,7]:=HTNO[U];  
'END';  
'FOR' I:=2 'STEP' 2 'UNTIL' NN1-1 'DO'
```



```
'BEGIN'  
L:=1;  
D:=0;  
B:=1-((L*N1)-N1-L*3);  
W:=I+48;  
NODE[W,1]:=1+L*D+106;  
NODE[W,2]:=(2*N1)+7+L+D+106;  
NODE[W,3]:=3+B*D+106;  
NODE[W,4]:=N1+2+B+D+106;  
NODE[W,5]:=N1*3+D+106;  
NODE[W,6]:=2+D+D+106;  
NODE[W,7]:=N1*100;  
'END';  
'FOR' I:=1 'STEP' 2 'UNTIL' N1*12-1 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
AL13:'IF' I 'LE' (L*(N1-1))-1 'THEN'  
'BEGIN'  
B:=1-((L*N1)-N1-L*2);  
W:=I+60;  
NODE[W,1]:=1+L*D+132;  
NODE[W,2]:=(2*N1)+1+B+D+132;  
NODE[W,3]:=(2*N1)-3+D+132;  
NODE[W,4]:=N1+1+B+D+132;  
NODE[W,5]:=(2*L1)*2+L*D+132;  
  
NODE[W,6]:=N1+2+B+D+132;  
NODE[W,7]:=N1*100;  
'END'  
'ELSE'  
'BEGIN'  
L:=L+1;  
D:=D+(2*N1);  
'GOTO' AL13;  
'END';  
'END';  
'FOR' I:=2 'STEP' 2 'UNTIL' N1*12 'DO'  
'BEGIN'  
L:=1;  
D:=0;  
  
RANAS:'IF' I 'LE' (L*(N1-1)) 'THEN'  
'BEGIN'  
B:=1-((L*N1)-N1-L*3);  
W:=I+60;  
NODE[W,1]:=1+D+D+132;  
NODE[W,2]:=(2*N1)+3+B+D+132;  
NODE[W,3]:=3+L+D+132;
```



```
NODE(I,4) := N1 + 2 * D + 132;  
NODE(I,5) := N1 + 3 * D + 132;  
NODE(U,6) := 2 * D + 132;  
NODE(U,7) := N1 + 132;  
'END'  
'ELSE'  
'BEGIN'  
L := L + 1;  
D := D + (2 * H1);  
'GOTO' RAHA3;  
'END';  
'END';  
WRITETEXT('('('('20')'NODAL%CONNECTIONS%HAVE%BEEN%READ')');  
'FOR' W:=1 'STEP' 1 'UNTIL' NELEMT 'DO'  
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'  
'BEGIN' IF XX[NODE(U,3+I)] = 0,0001 THEN  
'BEGIN'  
'IF' I=1 THEN J:=1 'ELSE' IF I=2 THEN J:=2 'ELSE' J:=0;  
XX[NODE(U,3+I)] := (XX[NODE(W,1)] + XX[NODE(W,1+J)]) / 2;  
YY[NODE(U,3+I)] := (YY[NODE(W,1)] + YY[NODE(W,1+J)]) / 2;  
'END';  
'END';  
WRITETEXT('('('('20')'NODAL%POINT%DATA'('2045')'NODE'('55')'RXCOORD  
'('55')'ZYCOORD'('55')'TY-A'('55')'E-DESP'('55')'Z-DISP'('6465')'  
OR%LOAD'('45')'OR%LOAD')');  
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'  
'BEGIN' NEWLINE(1); SPACE(3);  
PRINT(I,3,0); SPACE(3);  
PRINT(XX[I],0,3);  
PRINT(YY[I],0,3); SPACE(2);  
PRINT(NODE[I,1],3,0); SPACE(2);  
PRINT(ULX[I,1],0,3);  
PRINT(VLY[I,1],0,3);  
'END';  
WRITETEXT('('('('40')'ELEMENT%DATA'('20')'ELEMENT'('155')'  
NODAL%CONNECTIONS')');  
'FOR' W:=1 'STEP' 1 'UNTIL' NELEMT 'DO'  
'BEGIN' NEWLINE(1);  
PRINT(W,3,0); SPACE(6);  
'FOR' J:=1 'STEP' 1 'UNTIL' 7 'DO'
```



```
'BEGIN' PRINT(NODE(U,J),3,0);  
SPACE(2);  
'END';  
'END';  
'END' OF PROCEDURE SBINPUT;
```



```
'PROCEDURE' ASSEMBLYCLEFT(K,XX,YY,DETJ,ODE,C,THICK,ADD);
'INTEGER' NELEMT; 'REAL' DETJ,THICK;
'REAL' 'ARRAY' X,XY,YY,C; 'INTEGER' 'ARRAY' NNODE,LD;
'REGIM' 'INTEGER' I,J,U,Z,V,SUR2,SUR1,SUB3;
'REAL' 'ARRAY' B[4,1:12],B[1:12,1:12],C[1:6,1:4];
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' W[1,1]:=0.33333333;
W[1,2]:=W[1,3]:=W[2,3]:=W[2,4]:=W[3,2]:=W[3,4]:=0.57
W[1,4]:=W[2,2]:=W[3,5]:=W[4,4]:=W[5,2]:=W[6,5]:=0.07
W[4,2]:=W[4,3]:=W[5,5]:=W[5,4]:=W[6,2]:=W[6,4]:=1.07
'FOR' I:=1 'STEP' 1 'UNTIL' ADD(NPEREL 'DO' K[1]:=0.07
'FOR' Z:=1 'STEP' 1 'UNTIL' NELEMT 'DO'
'REGIM' 'COMMENT' Z HAS REPLACED W TO OVERCOME A PROBLEM
OF AN INCORRECT IDENTIFIER PROBABLY COMPOSED WITH W ARRAY;
'COMMENT' THE ELEMENT STIFFNESS ARRAY IS INITIALISED;
'FOR' I:=1 'STEP' 1 'UNTIL' 12 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 12 'DO' K[E[1,J]]:=0.07
'COMMENT' FOR LATER ADAPTATION THIS WILL BE STORED AS
A ONE DIMENSIONAL ARRAY
THE LOOP FOR THE NUMBER OF INT PTS IS CONSTRUCTED;
'FOR' U:=1 'STEP' 1 'UNTIL' 5 'DO'
'REGIM'
STRDIS(W[U,2],W[U,3],W[U,4],5,XX,YY,DETJ,ODE,Z);
'FOR' J:=1 'STEP' 2 'UNTIL' 11 'DO'
'FOR' I:=J 'STEP' 2 'UNTIL' 11 'DO'
'REGIM'
K[E[1,I]]:=K[E[1,J]]:=K[E[1,J]]+W[U,1]*(C[1,1]*C[1NODE[Z,7],1]*C[1,1]+
C[1NODE[Z,7],2]*B[2,1]+C[1NODE[Z,7],4]*B[4,1])+
B[2,J]*C[1NODE[Z,7],2]*B[1,1]+C[1NODE[Z,7],5]*B[2,1]
+C[1NODE[Z,7],7]*B[4,1])+
B[4,J]*(C[1NODE[Z,7],4]*B[1,1]+C[1NODE[Z,7],7]*B[2,1]
+C[1NODE[Z,7],10]*B[4,1]))*2*3.1415926*RAVG*0.5*DETJ;
K[E[1,I+1]]:=K[E[1+1,J]]:=K[E[1+1,J]]+W[U,1]*(B[1,J]*C[1NODE[Z,7],3]*B[3,1+1]
+C[1NODE[Z,7],4]*B[4,1+1])+
B[2,J]*C[1NODE[Z,7],6]*B[3,1+1]+C[1NODE[Z,7],7]*B[4,1+1])+
B[4,J]*(C[1NODE[Z,7],9]*B[3,1+1]+C[1NODE[Z,7],10]*B[4,1+1]))
*2*3.1415926*RAVG*0.5*DETJ;
'END';
'FOR' J:=2 'STEP' 2 'UNTIL' 12 'DO'
'FOR' I:=J 'STEP' 2 'UNTIL' 12 'DO'
'REGIM'
K[E[1,I]]:=K[E[1,J]]:=K[E[1,J]]+W[U,1]*(B[3,J]*C[1NODE[Z,7],8]*B[3,1]+
C[1NODE[Z,7],9]*B[4,1])+
B[4,J]*(C[1NODE[Z,7],9]*B[3,1]+C[1NODE[Z,7],10]*B[4,1]))
*2*3.1415926*RAVG*0.5*DETJ;
'IF' I=12 'THEN' 'GOTO' L6 'ELSE'
K[E[1,I+1]]:=K[E[1+1,J]]:=K[E[1+1,J]]+W[U,1]*(B[3,J]*C[1NODE[Z,7],8]*B[3,1+1]
+C[1NODE[Z,7],9]*B[4,1+1]+C[1NODE[Z,7],10]*B[4,1+1])+
B[4,J]*(C[1NODE[Z,7],4]*B[1,1+1]+C[1NODE[Z,7],7]*B[2,1+1]
+C[1NODE[Z,7],10]*B[4,1+1]))*2*3.1415926*RAVG*0.5*DETJ;
L6: 'END';
'END';
```



```
'COMMENT' ASSEMBLY OF OVER-ALL STIFFNESS MATRIX AS A  
ONE-DIMENSIONAL ARRAY
```

```
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'  
'FOR' J:=1 'STEP' 1 'UNTIL' 5 'DO'  
'FOR' V:=1,0 'DO'
```

```
'BEGIN' SUB1:=NODE[Z,I]*2-1;  
SUB2:=NODE[Z,J]*2-V;  
SUB3:=NODE[Z,I]*2;
```

```
'IF' SUB1 .LT. SUB2 'THEN' 'GOTO' LABA;  
K[ADD[SUB1]-SUB1+SUB2]:=K[ADD[SUB1]-SUB1+SUB2]+K[EI*2-1,J*2-V];  
LABA: 'IF' SUB3 .LT. SUB2 'THEN' 'GOTO' LABB;  
K[ADD[SUB3]-SUB3+SUB2]:=K[ADD[SUB3]-SUB3+SUB2]+K[EI*2,J*2-V];  
LABB: 'END';  
'END';
```

```
'END' OF PROCEDURE ASSEMBLY;  
'PROCEDURE' ED4AA(X,F,ABSACC,RELACC,XSTEP,FUNCT,MAXFUN,IFAIL);  
'VALUE' ABSACC,RELACC,XSTEP,MAXFUN;
```

```
'REAL' X,F,ABSACC,RELACC,XSTEP;
```

```
'INTEGER' IFAIL,MAXFUN;
```

```
'PROCEDURE' FUNCT; 'LOCAL';
```

```
'PROCEDURE' FUNCT(TH,SS);
```

```
'VALUE' TH;
```

```
'REAL' TH,SS;
```

```
SS:=((K1+2)*COS*(4*PI*NU-COS(TH))*(1+COS(TH)))+  
((2*K1+K2)*COS*(2*PI*NU*(TH))*(COS(TH)-1+2*NU))+  
((K2+2)*COS*(4*(1-NU)*(1-COS(TH))*(1+COS(TH))*(3*COS(TH)-1)))/2;
```

```
NJOB:=READ;
```

```
'FOR' COUNT:=1 'STEP' 1 'UNTIL' NJOB 'DO'
```

```
'BEGIN'
```

```
WRITETEXT('('('20')'JOBNAME-----')');
```

```
COPYTEXT('('END%OFXTITLE')');
```

```
NLENT:=READ; NNODE:=READ;
```

```
WRITETEXT('('('20')'NO%OF%ELEMENTS-----')');
```

```
PRINT(NLENT,3,0);
```

```
WRITETEXT('('('20')'NO%OF%NODES-----')');
```

```
PRINT(NNODE,3,0);
```

```
NFREE:=NNODE*2; NSETFS:=READ; NSETC:=READ; NMAT:=READ;
```

```
'BEGIN' 'INTEGER' NALD,NSPEC,2,COMPA,NSETF,NT,OB;
```

```
'REAL' RO,G,H,AA,CR,GG,ANGLE;
```

```
'REAL' 'ARRAY' XX,YY(1:NNODE),DLX,VLY(1:NNODE),NSETFS;
```

```
C(1:NNAT,1:10),A(1:5),Q(1:NFREE,1:NSETFS),C(1:2);
```

```
'INTEGER' 'ARRAY' NODE(1:NLENT,1:7),KDD(1:NNODE,1:NSETFS);
```

```
ADD(0:NFREE),ITNO(1:NLENT),NSTEL(1:NMAT),STEL(1:NMAT,1:NLENT);
```

```
NI:=READ; G:=READ; H:=READ; AA:=READ; RO:=READ;
```

```
NSETC:=READ; CR:=G-AA; CASE:=READ;
```

```
'FOR' S:=1 'STEP' 1 'UNTIL' NSETC 'DO'
```



```
'BEGIN'  
'IF' S=1 'THEN' 'BEGIN'  
  HSETF:=READ;  
  SFINPUT(ADD,XX,YY,PERFE,HNODE,SETE,KODE,SPED,NSETES,NMAT,NTIC,  
  NSTEL,STEL,ULX,VLY,NELENT,NODE,NI,G,H,AA,RO);  
  ADDARRAY(NELENT,HNODE,ADD,NODE);  
  NG:=NFREE;  
  KARDDHST(HEREE,NI,BA,0,ADD);  
  'FOR' NATNO:=1 'STEP' 1 'UNTIL' NMAT 'DO'  
  CONSTREL(C,NATNO,A);  
  WRITETEXT('('('20')'10%OFXDEGREESXOFFFREEDOM----')');  
  PRINT(HEREE,3,0);  
  NEWLINE(2);  
  'END' 'ELSE'  
  'BEGIN'  NNEWC:=READ;  
    HSETF:=READ;  
    'FOR' I:=1 'STEP' 1 'UNTIL' NNEWC 'DO'  
    'BEGIN' J:=READ;  
      KODE[J,1]:=READ;  ULX[J,1]:=READ;  VLY[J,1]:=READ;  
    'END';  
  
  'END';  
  'FOR' I:=1 'STEP' 1 'UNTIL' NFREE 'DO'  
  'FOR' J:=1 'STEP' 1 'UNTIL' HSETF 'DO'  Q[I,J]:=0,0;  
  'FOR' I:=1 'STEP' 1 'UNTIL' HNODE 'DO'  
    LOADING(KODE[I,1],ULX[I,1],VLY[I,1],Q,I,1);  
  'IF' HSETF 'GT' 1 'THEN'  
  'BEGIN'  
  'FOR' I:=2 'STEP' 1 'UNTIL' HSETF 'DO'  
  'BEGIN'  
  NSPEC:=READ;  
  WRITETEXT('('('20')'FORCEKSET-----')');  
  
  PRINT(I,3,0);  
  WRITETEXT('('('20')'KODE('55')'TYPE('65')'Z-DISP('75')'Z-DISP  
  ('82S')'ORLOAD('85')'ORLOAD')');  
  'FOR' J:=1 'STEP' 1 'UNTIL' NSPEC 'DO'  
  'BEGIN'  
    K:=READ;  KODE[K,I]:=READ;  ULX[K,I]:=READ;  VLY[K,I]:=READ;  
  LOADING(KODE[K,I],ULX[K,I],VLY[K,I],Q,K,I);  
  NEWLINE(2);  
  PRINT(K,3,0);  SPACE(2);  
  PRINT(KODE[K,I],3,0);  SPACE(2);  
  PRINT(ULX[K,I],0,4);  
  PRINT(VLY[K,I],0,6);  
  'END';  
  'END';  
  'END';  
  NEWLINE(6);
```



```
'BEGIN' 'INTEGER' SUB1, SUB2, SUB3, SUBSCP;  
'REAL' 'ARRAY' K(4,4,4);  
ASSEMBLY(DELETE, K, XX, YY, DETJ, NODE, C, THICK, ADD);  
NEWLINE(6);  
CCRUM12(N1, R0, NG, NSETF, C1, ADD, K, Q, CASE, THICK, CR);  
CREATESTOR(10, 'C'ED', 'C'FARILEST', 1, SQUO);  
NVAB1:=1;  
PJTARRAY(10, NVAB1, K);  
PJTARRAY(10, NVAB1, ADD);  
'COMMENT' 'INTRODUCTION OF KINEMATIC CONSTRAINTS';  
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE-N1 'DO'  
'BEGIN' Z:=I+N1;  
'IF' KODE(Z,1)=0 'THEN' 'GOTO' K01;  
'IF' KODE(Z,1)=2 'THEN' 'GOTO' K02;  
BOUNDCONST(ULX(Z,1), 2*I-1, 2, K, NG-2*I+5, 1, ADD);  
'FOR' J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q(1, J):=Q(1, 1);  
'IF' KODE(Z,1)=1 'THEN' 'GOTO' K01;  
K02: BOUNDCONST(ULY(Z,1), 2*(I+Q, K, NG-2*I+5, 1, ADD);  
'FOR' J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q(1, J):=Q(1, 1);  
K01: 'END';  
WRITETEXT('('('2C')'SYNVSOLX$BEGIN$')');  
SYNVSOL(K, K, ADD, 0, NG-2*I+5, 'SETF, FAL');  
NEWLINE(6);  
'FOR' I:=1 'STEP' 1 'UNTIL' NSETF 'DO'  
'BEGIN' WRITETEXT('('('4C')'NODAL$DISPLACEMENTS$POK$FORCE$SET$')');  
PRINT(1, 2, 0);  
WRITETEXT('('('3CS')'MODE('5S')'R-DIRECTION('6S')'Z-DIRECTION  
'('6S')'MODE('5S')'R-DIRECTION('3S')'Z-DIRECTION')');  
U:=2*((NNODE-N1)+'/2);  
'FOR' J:=2 'STEP' 2 'UNTIL' U 'DO'  
'BEGIN' NEWLINE(2); V:=2*(J-1);  
Z:=J*N1;  
PRINT(Z-1, 3, 0); SPACE(2); PRINT(Q[V-1, 1], 0, 0); SPACE(2);  
PRINT(Q[V, 1], 0, 0); SPACE(15); PRINT(Z, 3, 1); SPACE(2);  
  
PRINT(Q[V+1, 1], 0, 0); SPACE(2); PRINT(Q[V+2, 1], 0, 0);  
'END';  
'IF' NNODE-N1 'GT' U 'THEN'  
'BEGIN' NEWLINE(2); PRINT(NNODE-N1, 3, 0); SPACE(2);  
V:=2*(NNODE-N1);  
PRINT(Q[V-1, 1], 0, 0); SPACE(2); PRINT(Q[V, 1], 0, 0);  
'END';  
V:=NG-2*N1;  
WRITETEXT('('('4C')'CRACK$TIP$DISPLACEMENTS$IN$R-DIRECTION$='')');  
PRINT(Q[V+3, 1], 0, 10);  
  
WRITETEXT('('('2C')'CRACK$TIP$DISPLACEMENTS$IN$Z-DIRECTION$='')');  
PRINT(Q[V+2, 1], 0, 10);  
WRITETEXT('('('4C')'MODE$STRESS$INTENSITY$FACTOR$KI$='')');  
PRINT(Q[V+5, 1], 0, 10);
```



```
WRITE TEXT('('C'(12))' 'MODE2 ST PAS I TUBS IY SECT (K1)');
PRINT((Q[V+4, I]*(-1)), 0, 10);
WRITE TEXT('('C'(40))' 'IGL BODY SECT (14)');
PRINT(Q[V+1, I], 0, 10);
NEWLINE(5);
MU:=CH[2];
GG:=0.5*Q[11]/(1+CH[2]);
CON:=(3.1415926*CR)/(D+GG);
K1:=Q[V+3, I];
K2:=Q[V+4, I];
TH:=-1.5707963;
RELACC:=0.0;
ABSACC:=0.0174;
XSTEP:=3.1415926;
MAXFUN:=100;
IFAIL:=0;
EQGAACTH, SS, ABSACC, RELACC, XSTEP, PUNCT, MAXFUN, IFAIL);
ANGLE:=(3.1415926+TH)*37.3;
WRITE TEXT('('D'(12))' 'TH');
PRINT(ANGLE, 0, 0);
NEWLINE(5);
'END';
HVAB1:=1;
GETARRAY(10, HVAB1, K);
GETARRAY(10, HVAB1, ADD);
RESIDUAL(R, Q, ADD, GG-2*NI+5, NSETF);
'END';
'END';
'END';
FAIL: 'END';
'END';
```

CHAPTER 11

REFERENCES

1. Timoshenko, S., and Goodier, J.N., "Theory of Elasticity", McGraw-Hill, 1934.
2. Sokolnikoff, I.S., "Mathematical theory of elasticity", McGraw-Hill, 1956.
3. Desai, C.S., and Abel, J.F., "Introduction to the finite element method", Van Nostrand Reinhold Company, 1972.
4. Zienkiewicz, O.C., "The finite element method in engineering science", McGraw-Hill, 1971.
5. Martin, H.C., and Carey, G.F., "Introduction to finite element analysis", McGraw-Hill, 1973.
6. Meek, J.L., and Carey, G.F., "Axisymmetric solution of elastic plastic problems by finite element methods", Civil Eng.Dept., Univ. of Queensland, Australia, Bull. No.11, May, 1969.
7. Jaeger, J.C., "Elasticity, Fracture and flow", Methuen & Co.Ltd., 1969.
8. Murnaghan, F.D., "Finite deformation of an elastic solid", J.Wiley and Son, 1951.
9. Robertson, A.W., "On the stress analysis of cracked bodies by means of finite elements", Ph.D.Thesis, University of Aston in Birmingham, 1976.
10. Becker, E.B., and Brisband, J.J., "Application of the finite element method to stress analysis of solid propellant rocket grains", S-76, Rohm and Haas Co., Alabama, 1966.

11. Frederick, O.C., Wong, Y.C., and Edge, F.W., "Two-dimensional automatic mesh generation for structural analysis", *Int.J.Num.Methods in Eng.*, vol.2, No.1, p.133, 1970.
12. Akyuz, F.A., "Natural coordinate systems, an automatic input data scheme for a finite element method", *Num.Eng.Design*, 11, p.195, 1970.
13. Zienkiewicz, O.C., and Phillips, D.V., "An automatic mesh generation scheme for plane and curved surfaces by isoparametric coordinates", *Int.J.Num.Methods in Eng.*, 3, p.519, 1971.
14. Anderson, W.E., "Some design-oriented views on brittle fracture", *Gulf Coast Metals Conf.*, Houston, Texas, Feb., 1969.
15. Sih, G.C., and Macdonald, B., "Fracture mechanics applied to engineering problems - strain energy density factor criterion", *Int.J.Frac.Mech.*, 6, pp.361-386, 1974.
16. Griffith, A.A., "The phenomena of rupture and flow in solids", *Phil.Trans.Roy.Soc.*, A221, p.163, 1921.
17. Bueckner, H.F., "Propagation of cracks and the energy of elastic deformation", *Trans.ASME* 80E, p.1225 1958.
18. Griffith, A.A., "The theory of rupture", *Proc.1st.Int. Cong.Appl. Mech.*, Delft, 1924.
19. Irwin, G.R., "Analysis of stresses and strains near the end of a crack traversing a plate", *J.Appl.Mech.*, 24, p.361, 1957.
20. Lawn, B.R., and Wilshaw, T.R., "Fracture of brittle solids", *Cambridge Solid State Science Series*, 1975.

21. Sih, G.C., and Liebowitz, H. "Mathematical theories of brittle fracture", in "Fracture", vol.2, ed., Liebowitz, Academic Press, 1968.
22. Hayes, D.J., "Origins of the stress intensity factor approach to fracture", J.of Strain Analysis, 10, p.198, Oct., 1975.
23. Sih, G.C., Paris, P.C., and Erdogan, F., "Crack-tip stress intensity factors for plane extension and plate bending problems", Int.J.of Frac.Mech., 29, p.306, 1962.
24. Oglesby, J.J., and Lomacky, O., "An evaluation of finite element methods for the computation of elastic stress intensity factors", NSRDC, Report No. 3751, 1971.
25. Sih, G.C., "Strain energy density factor applied to mixed mode crack problems", Int.J.of Frac., 10, p.305, Sept., 1974.
26. Broek, D., "Elementary engineering fracture mechanics", Noordhoff Int., 1974.
27. Duffy, A.R., "Fracture design practice for pressure piping", Fracture, vol.I., Liebowitz ed., Academic Press, 1968
28. Rooke, D.P., "Elastic yield zone round a crack tip", Roy.Aircr.Est., Farnborough, Tech.note CPM 29, 1963.
29. Irwin, G.R., "Fracture dynamics", Fracture of metals, ASME, pub., pp.147-166, 1948.
30. Orwan, E., "Energy criteria of fracture", Welding J., 20, pp.157S - 160S, 1955.

31. Irwin, G.R., "Plastic zone near a crack and fracture toughness", Proc. 7th. Sagamore Conf., p.IV-63, 1960.
32. Jerram, K., and Hellen, T.K., "The use of finite element techniques in fracture mechanics", CEEGB, RD/B/N2478, Dec. 1972.
33. Emery, A.F., Barrett, C.F., and Kobayashi, A.S., "Temperature distributions and thermal stresses in a partially filled annulus", Expl.Mech., 6, p.602, 1966.
34. Smith, D.G., and Smith, C.W., "A photoelastic evaluation of the influence of closure and other effects upon the local bending stresses in cracked plates", Int.J.of Frac.Mech., 6, p.305, 1970.
35. Smith, D.G., and Smith, C.W., "Photoelastic determination of mixed mode stress intensity factors", Eng.Frac.Mech., 4, p.357,1972.
36. Irwin, G.R., and Kies, J.A., "Critical energy rate analysis of fracture strength", Weld.J., 33, p.193, 1954.
37. Wells, A.A., "Unstable crack propagation in metals cleavage and fast fracture", Proc. Crack Propagation Symposium, pp.210-230, Cranfield, 1961.
38. Burdekin, F.M., and Stone, D.E.W., "The crack opening displacement approach to fracture mechanics in yielding", J.Strain analysis, 1, pp.145-153, 1966.
39. Westergaard, H.M., "Bearing pressure and cracks", J.Appl.Mech., 61, p.49, 1939.

40. Cartwright, D.J., and Rook, D.P., "Methods of determining stress intensity factors", R.A.E., Tech. report. 73031, May, 1973.
41. Sih, G.C., "On the Westergaard method of crack analysis", Int.J.Frac.Mech., 2, 628, 1966.
42. Mushkelishvili, N.I., "Some basic problems of the mathematical theory of elasticity", P.Noordhoff Ltd., 1953.
43. Sih, G.C. "Application of Mushkelishvili's method to fracture mechanics", Trans.Chines Assoc. for Advanced Studies, 3, p.25, 1962.
44. Rice, J.R., and Tracey, D.M., "Computational fracture mechanics", in "Numerical and Computer methods in structural mechanics", by Fenves, S.J., Perron, N., Robinson, A.R., and Schnobrich, W.C., Academic Press, 1973.
45. Williams, M.S. "On the stress distribution at the base of a stationary crack:", J.App.Mech., 24, p.109, 1957.
46. Cartwright, D.J., and Rook, D.P., "Evaluation of stress intensity factors", J.Strain Analysis, vol.10, pp.217-224, Oct. 1975.
47. Gross, B., Srawley, J.E., and Brown, W.F., "Stress intensity factors for a single edge notch tension specimen by boundary collocation of a stress function", NASA TN,D-2395, 1964.
48. Gross, B., and Srawley, J.E., "Stress intensity factors for three point bend specimen by boundary collocation", NASA TN-D-3092,1965.
49. Sih, G.C., "Methods of analysis and solutions of crack problems", Noordhoff, 1973.

50. Neuber, H., "Theory of notch stresses"; Springer, 1958.
51. Sih, G.C., "Handbook of stress intensity factors", Inst. of fracture and solid mech., Lehigh Univ., 1973.
52. Erdogan, F., "On the stress distribution in plates with colinear cuts under arbitrary loads", Proc. 4th. U.S. Natn.Congr.Appl.Mech., Berkeley, 1962.
53. Rooke, D.P., and Sneddon, I.N., "The crack energy and the stress intensity factor for a cruciform crack deformed by internal pressure", Int. J. Eng.Sci., 7, pp.1079-1089, 1969.
54. Tweed, J., "The solution of certain triple integral equations involving inverse mellin transforms", Glasgow math. J., 14, pp.65-72, 1973.
55. Romonaldi, J.P., Fraiser, J.T., and Irwin, G.R., "Crack extension force near a riveted stringer", N.R.L.report 4956, Washington D.C., 1957.
56. Irwin, G.R., Fracture, Handbook der Physik, vol.VI, Springer, 1958.
57. Koiter, W.T., "Rectangular tensile sheet with symmetric edge crack", J.App.Mech.,32,p.237, 1965.
58. Kobayashi, A.S., et al. "Application of the method of finite element analysis to two-dimensional problems in fracture mechanics", ASME Meeting, Los Angeles, Calif., 69-WA/PVP, Nov., 1969.
59. Miamoto, H., Shiratori, M., and Miyoshi, T., "Applications of the finite element method to fracture mechanics", J. of Faculty of Eng. Univ. of Tokyo (B), vol.31, No.1, 1971.

60. Mowbary, D.F., and Andrews, W.R., G.E.C. report No. KAPL-P-3815, New York, 1969.
61. Dixon, J.R., and Pook, L.P., "Nature", 224, No.5215, pp.116-117, 1969.
62. Jerram, K., "Discussions to a paper by Smith and Alavi", Int.Conf. on Pressure Vessel Tech., part III Discussions, ASME, New York, 1970.
63. Rice, J.R., "A path independent integral and the approximate analysis of strain concentration by notches and cracks", J.App.Mech., 35, p.379, 1968.
64. Chan, S.K., Tuba, I.S., and Wilson, W.K., "On the finite element method in linear elastic fracture mechanics", Eng.Frac.Mech., 2, pp.1-17, 1970.
65. Tong, P., and Pian, T.H.H., "On the convergence of the finite element method for problems with singularity", Int.J.Solids. Structures, 9, pp.313-321, 1973.
66. Bysokov, E., "The calculation of stress intensity factors using the finite element method with cracked elements", Int.J.Frac.Mech., 6, p.159, 1972.
67. Tracey, D.M., "Finite elements for the determination of crack tip elastic stress intensity factors", Eng.Frc.Mech., 3, p.255, 1971.
68. Henshell, R.D., "Crack tip finite elements are unnecessary", Int.J.Num.Methods.Eng., 9, p.495, 1975.
69. Barsoum, R.S., "On the use of isoparametric finite elements in linear fracture mechanics", Int.J.Num.Methods.Eng., 10, p.25, 1976.

70. Tong, P., Pian, H.H., and Lasry, S.J., "A hybrid element approach to crack problems in plane elasticity", *Int.J.Num.Methods.Eng.*, 7, p.297, 1973.
71. Blackburn, W.S., "Calculation of stress intensity factors at crack tips using special finite elements", *Conf. on Maths. of finite elements and App.*, Brunell Univ., April, 1972.
72. Paris, P.C., and Sih, G.C., "Fracture toughness testing and its applications", *ASTM,STP 381*, Philadelphia, pp.30-81, 1965.
73. Hilton, P.D., and Hutchinson, J.W., "Plastic stress intensity factors for cracked plates", *Harvard Univ. report SM-34*, 1969.
74. Wilson, W.K., Clark, W.G., and Wessel, E.T., "Fracture mechanics technology for combined loading and low-to-intermediate strength metals", *Westinghouse Research Lab.*, report No. 69-1E7-FMECH-R1, Pittsburgh, Penn., 1969.
75. Levey, M., and Marcal, P.V., "Three dimensional elastic plastic stress and strain analysis for fracture mechanics", *Brown Univ. Tech.report No.12*, UCCND sub.cont. 3152, 1970.
76. Kassir, M.K. and Sih, G.C., "Three dimensional stress distribution around an elliptical crack under an arbitrary loading", *J.App.Mech.* p.601, 1966.
77. Sih, G.C., "Three dimensional stress distribution near a sharp crack in a plate of finite thickness", *Air force Mat.Lab. AFML-TR-66-242*, 1966.

78. Tracey, D.M., "Progress report on elastic plastic finite element analysis of cracked solids", Brown Univ., NASA grant NOL-40-002-080, 1971.
79. Richards, T.H., "On using the finite element method and sub region singular solutions in fracture mechanics", Dept. of Mech.Eng.report, Univ. of Aston in Birmingham, Nov., 1974.
80. Timoshenko, S., and Woinowsky-Krieger, S., "Theory of plates and shells", McGraw-Hill, 1959.
81. Knott, J.F., "Fundamentals of fracture mechanics", Butterworths, 1973.
82. Jennings, A., and Tuff, A.D., "A direct method of the solution of large sparse symmetric simultaneous equations", Int.Conf. on large sparse sets of linear equations, Oxford, 1970.
83. Barber, J.R., "The disturbance of a uniform steady state heat flux by a partially conducting plane crack", Dept. of Mech.Eng., Univ of Newcastle upon Tyne, 1976.