

Some pages of this thesis may have been removed for copyright restrictions.

If you have discovered material in Aston Research Explorer which is unlawful e.g. breaches copyright, (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please read our [Takedown policy](#) and contact the service immediately (openaccess@aston.ac.uk)

ON THE STRESS ANALYSIS OF CRACKED BODIES BY MEANS
OF FINITE ELEMENTS

BY

A. W. ROBERTSON, B.Sc., M.Sc.

UNIVERSITY OF ASTON

BIRMINGHAM

524.04 R: F

- 8 OCT 1976

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

February 1976

SUMMARY

The work described in this thesis deals with the development and application of a finite element program for the analysis of several cracked structures.

In order to simplify the organisation of the material presented herein, the thesis has been subdivided into two Sections :

In the first Section the development of a finite element program for the analysis of two-dimensional problems of plane stress or plane strain is described. The element used in this program is the six-mode isoparametric triangular element which permits the accurate modelling of curved boundary surfaces. Various cases of material anis^otropy are included in the derivation of the element stiffness properties. A digital computer program is described and examples of its application are presented.

In the second Section, on fracture problems, several cracked configurations are analysed by embedding into the finite element mesh a sub-region, containing the singularities and over which an analytic solution is used. The modifications necessary to augment a standard finite element program, such as that developed in Section I, are discussed and complete programs for each cracked configuration are presented. Several examples are included to demonstrate the accuracy and flexibility of the technique.

TABLE OF CONTENTS

	<u>Page No.</u>
Preface	1
Acknowledgements	2
Notation and List of Symbols	3
SECTION I - FORMULATION AND APPLICATION OF TWO-DIMENSIONAL PLANE STRESS/ PLANE STRAIN FINITE ELEMENT PROGRAM	
1. Summary of Section I	5
2. Introduction	6
3. Finite Element Formulation	15
3.1 Introduction	15
3.2 Natural Coordinate System for Plane Triangles	16
3.3 Interpolation Formula for Six-Node Triangle	19
3.3.1 Use of Shape Functions for Element Mapping	25
3.4 Constitutive Relations	26
3.4.1 Elasticity Matrix	27
3.4.2 Orthotropic Material	28
3.4.3 Isotropic Material	32
3.4.4 Transversely Isotropic Material	32
3.4.5 Transformation of the Elasticity Matrix Under a Transformation of the Coordinate System	35
3.5 Generation of Element and Overall Stiffness Matrices and Load Vectors	37
3.5.1 Derivation of Strain-Element Nodal Displacement Matrix	38
3.5.2 Formulation of Element Stiffness Matrix and Load Vector	42
3.5.3 Assembly of Overall Stiffness Matrix and Load Vector	46
4. Numerical Procedures	48
4.1 Introduction	48
4.2 Generation of Elasticity Matrix	50
4.3 Format for Input Data	54

4.4	Generation of Strain-Element Nodal Displacement Matrix	59
4.5	Storage of the Overall Stiffness Matrix	63
4.6	Assembly of Stiffness Matrices	67
4.6.1	Numerical Integration	67
4.6.2	Element Stiffness Matrix	69
4.6.3	Overall Stiffness Matrix	72
4.7	Application of Boundary Conditions	78
4.7.1	Applied Concentrated and Distributed Loading	78
4.7.2	Prescribed Nodal Displacements	85
4.7.2.1	Procedure SKEWEDCON	88
4.7.2.2	Procedure SKEWLOAD	93
4.7.2.3	Procedure GEOMBC	96
4.8	Solution of System Equations	100
4.9	Calculation of Element and Nodal Point Stresses and Strains	102
4.9.1	Procedure FEELSTR	103
4.9.2	Procedure FENOSTR	106
5.	Application of the Six-Node Isoparametric Triangular Finite Element	110
5.1	Introduction	110
5.2	Program Flowchart, Listing and Input Data	111
5.3	Numerical Examples	131
5.3.1	Circular Hole in a Tension Plate	131
5.3.2	Circular Hole in Rectangular Plate, Material Anisotropy	131
5.3.3	Thick Cylinder under Internal Pressure	138
6.	Discussion and Conclusions	145
7.	References	148
8.	Appendices	
8.A	Change of Variables for Area Integrals	151

8.B	Plane Stress and Plane Strain	154
8.C	Program for Plotting Defined Finite Element Mesh	157

SECTION II - FRACTURE PROBLEMS

1.	Summary of Section II	161
2.	Introduction and Literature Review	162
2.1	The Griffith Concept	163
2.2	The Stress Intensity Factor Concept	164
2.3	Equivalence of Griffith and Stress Intensity Factor Concepts	168
2.4	Modes of Crack Extension	169
2.5	Role of Plasticity	171
2.6	Determination of Stress Intensity Factors	173
2.6.1	Analytical Methods	174
2.6.2	Experimental Methods	177
2.6.3	Approximate Methods	178
2.6.3.1	Boundary Collocation	178
2.6.3.2	Finite Element Method	179
	Methods Based on a Standard Finite Element Program	180
	(a) Crack Tip Stress	180
	(b) Crack Tip Displacement	181
	(c) Energy Methods	183
c.i	Strain Energy(Compliance) Method	183
c.ii	Line Integral Method	185
c.iii	Crack Opening Energy Method	187

Methods Based on Singular Crack Tip Elements	189
(a) Byskov Element	189
(b) Tracey Element	190
(c) Hilton and Hutchinson Element	195
3. Crack Border Stress/Strain and Displacement Fields	200
3.1 Introduction	200
3.2 Derivation of Near Tip Field Equations	201
4. Application of the Hilton Element for Determining Stress Intensity Factors	208
4.1 Introduction	208
4.2 Modifying a Finite Element Program to Include the Hilton Element	208
5. Numerical Procedures for Various Crack Configurations	216
5.1 Introduction	216
5.2 Single Tip, Symmetric Mode I Case	222
5.3 Several Tips, Symmetric Mode I Case	241
5.4 Single Tip, Combined Modes I and II Case	264
5.5 Double Tip, Combined Modes I and II Case	284
5.5.1 Modification of the Overall System Equations	284
5.5.2 Crack Closure	304
6. Numerical Examples	312
6.1 Introduction	312
6.2 Single Tip, Symmetric Mode I	312
6.3 Several Tips, Symmetric Mode I	327
6.4 Single Tip, Combined Modes I and II	331
6.5 Double Tip, Combined Modes I and II	339
7. Discussion and Suggestions for Further Work	346
8. Conclusions	358
9. References	359

10. Appendices	365
10.A Field Equations for Strains/Stresses and Displacements Near a Crack Tip	365
10.B Strain Energy of Core Region	368
10.C Program for Plotting Defined Finite Element Mesh	371
10.D Fracture Program Listings	375
10.D.1 Single Tip, Symmetric Mode I	379
10.D.2 Several Tips, Symmetric Mode I	390
10.D.3 Single Tip, Combined Modes I and II	402
10.D.4 Double Tip, Combined Modes I and II	415

PREFACE

In order to simplify the organisation of the material presented herein, this thesis has been subdivided into two Sections. The first Section deals with the development and application of a two-dimensional plane stress/plane strain finite element program and the second Section deals with the adaptation and application of this program to the analysis of certain fracture problems.

The numbering of Chapters, Equations, Figures, Tables and References of both Sections is independant. The programs of Section II are based on the finite element program presented in the first Section, to which frequent reference is made. No reference is made to Section II in Section I.

ACKNOWLEDGEMENTS

The author wishes to express his deepest gratitude to his supervisor, Mr. T.H. Richards, for his encouragement and assistance throughout this work.

The research was carried out in the Department of Mechanical Engineering at the University of Aston in Birmingham. The author is grateful to Professor A.J. Ede for providing the facilities for this research.

Finally, the author is grateful to Mrs. J.M. Atherton for the typing of this thesis.

NOTATION AND LIST OF SYMBOLS

A list of the more commonly used symbols and their meaning follows. All symbols are defined when introduced in the text. Vector notation is used throughout the thesis, where $\{ \}$, $\{ \}^t$, $[]$ and $[]^t$ refer to a column vector, row vector, matrix and matrix transpose, respectively.

u, v	Components of displacement
P	Nodal loads
k_{ij}	Stiffness coefficients
δ_{ei}	Element nodal point displacements
V	Potential energy
U	Strain energy
Ω	Potential energy of applied loads.
ξ, η and L_1, L_2, L_3	Natural coordinates
h	Thickness
σ_{ij}	Components of stress
ϵ_{ij}	Components of strain
C_{ij}	Coefficients of the elasticity matrix
E, ν, G	Elastic constants
K	$= (3 - 4\nu)$ for plane strain $= (3 - \nu)/(1 + \nu)$ for generalised plane stress
$[S], [T]$	Transformation matrices
$[J]$	Jacobian matrix
w_i	Weight coefficients
L_{1i}, L_{2i}, L_{3i}	Gauss integrating points
$[I]$	Identity matrix
$[L], [L]^t$	Lower and upper triangular matrices respectively.
r, θ	Polar coordinates with origin at crack tip.

G, G_I	Strain energy release rate with crack extension per unit length of crack border. Subscript I refers to opening mode of crack extension; without subscript the mode is unspecified.
G_c	Critical values of strain energy release rate.
K, K_I, K_{II}, K_{III}	Stress Intensity Factors, where subscripts I, II and III refer to the modes of crack extension; without subscript the mode is unspecified.
$K_{Ic}, K_{IIc}, K_{IIIc}$	Fracture toughness values
$\delta_x, \delta_y, \omega$	Rigid body displacements of core element
U_c	Elastic strain energy of core element
σ_{ys}	Uniaxial yield stress
r_y	Radius of crack tip plastic zone.
$\psi(z), \phi(z)$	Complex functions
∇^2, ∇^4	Harmonic and bi-harmonic operators
λ	Compliance (inverse spring constant)
Γ_1	Interface between core element and finite element mesh
\bar{T}	Traction vector ($T_i = \sigma_{ij} n_j$)
\bar{n}	Vector outward normal
J	Rice's path independent integral
R_c	Core element radius
N_1	Number of nodes of core element/finite element mesh interface.

SECTION I

FORMULATION AND APPLICATION OF TWO-DIMENSIONAL
PLANE STRESS/PLANE STRAIN FINITE ELEMENT PROGRAM

1. SUMMARY OF SECTION I

The purpose of this Section is to describe the development of a finite element program for the analysis of two-dimensional problems of plane stress or plane strain.

A six-node isoparametric triangular element is used which permits the accurate modelling of curved boundary surfaces. Various cases of material anisotropy are included in the derivation of the element stiffness properties.

A digital computer program is described and examples of its application are presented.

2. INTRODUCTION

Of the general finite element methods, for the solution of continuum mechanics problems, the subclass known as the 'direct stiffness method' with displacements as primary unknowns is one of the most powerful and well developed. This method is employed exclusively in this thesis and before proceeding to a general description it is worthwhile to summarise the relevant steps involved :-

(A) Structure Idealisation

The behaviour of the actual continuum, Fig. 2.1, is assumed to be approximated by the assemblage of simply connected domains, called finite elements, Fig. 2.2. The elements are connected at a finite number of nodal points situated on their boundaries and the displacements of these nodal points are the basic unknowns of the problem. Thus the infinite number of degrees of freedom of the actual structure has been replaced by a finite number as determined by the number of nodes present.

(B) Displacement Field Discretisation

A set of functions are chosen to define uniquely the state of displacement within each element in terms of its nodal displacements. The amplitudes of these nodal displacements are known as the generalised coordinates or generalised displacements of the system.

(C) Derivation of Element Stiffness

The element stiffness matrices are derived, relating element nodal displacements $\{\delta\}_e$ to the associated nodal forces $\{F\}_e$, in the form :-

$$[k] \{\delta\}_e = \{F\}_e \quad (2.1)$$

Stiffness matrices are functions of the geometry and the elastic properties of the elements.

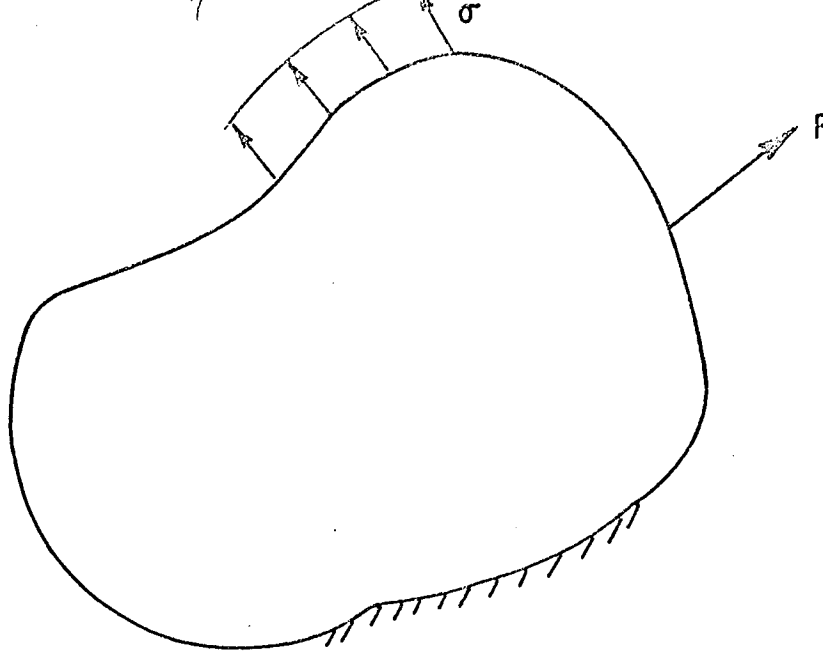


FIG 2.1 TWO DIMENSIONAL CONTINUUM.

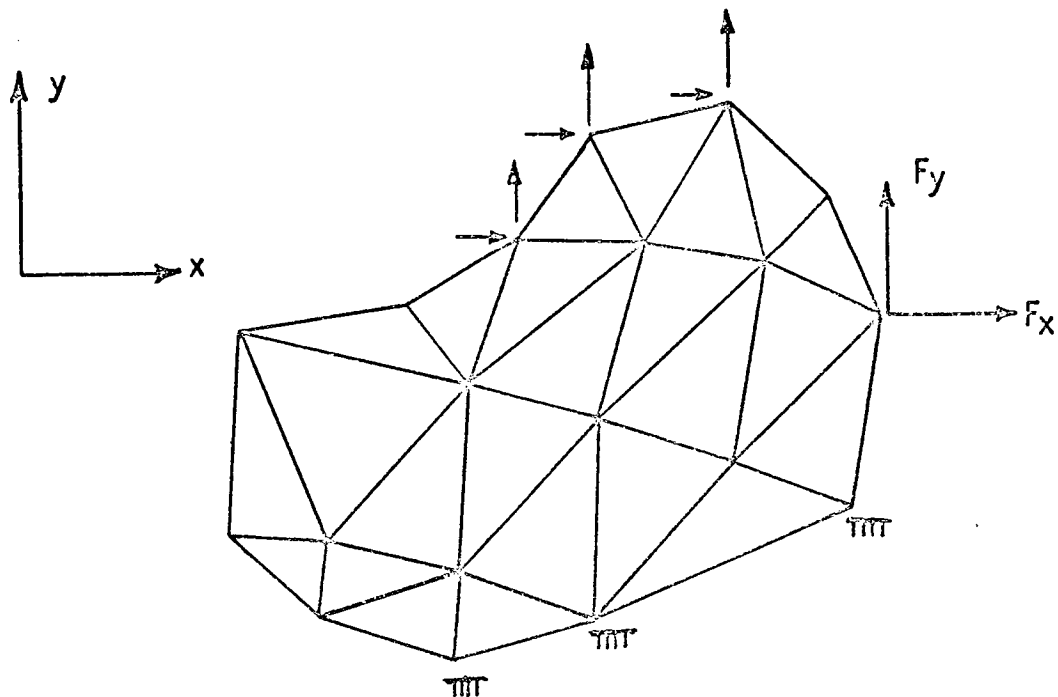


FIG 2.2 IDEALISATION OF CONTINUUM

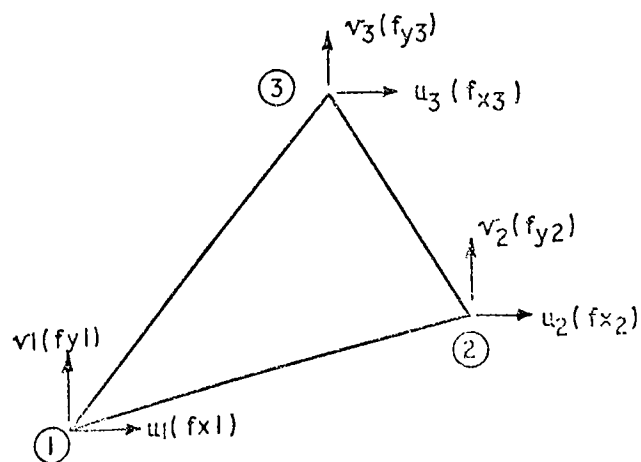


FIG 2.3 TRIANGULAR FINITE ELEMENT.

(D) Assembly of the Stiffness Equation for the Complete Structure

The individual element stiffness coefficients contributing to each nodal point are summed to obtain the total assemblage nodal stiffness matrix. Finally, kinematic constraints (prescribed nodal displacements) are introduced.

(E) Displacement Solution

The algebraic equations, assembled as in Step (D), are solved for the unknown nodal displacement components.

(F) Stress and Strain Computation

Element or nodal stresses and strains may be computed from the nodal displacement solution. They are generally computed as average values from surrounding elements or nodes.

The foundations of the finite element method, as applied to the solution of continuum problems, were laid by Argyris⁽¹⁾ and Turner et al⁽²⁾ in the 1950s. These and other researchers derived stiffness matrices for various elements, the derivation generally being performed by semi-intuitive methods. It was later recognised⁽³⁾ that the most important single factor which determines the eventual convergence or divergence of the final solution was the stiffness matrix for the element employed. In his paper Melosh⁽³⁾ bases the derivation of the element properties on the extremum (variational) theorems of elasticity, i.e. as the minimisation of some energy functional. In the case of the direct stiffness method the potential energy is minimised with respect to the element nodal displacements. This realisation and method of approach had a profound effect on both the respectability and the subsequent development of the general class of finite element methods. In the first place the method was given a sound theoretical foundation, and among the many benefits which accrued from such a step, it was possible⁽³⁾ to define finite element solution bounds to the true solution. Secondly,

it permitted greater flexibility in the formulation of different elements

- (a) by using higher derivatives of displacements as generalised displacements, e.g. curvatures and twists in the case of a bent plate, or
- (b) by using the variety of variational principles available; in addition to the well known principles of Minimum Potential Energy and Minimum Complementary Energy, there is the more general Reissner Principle, employed in (4), and also a number of modified principles presented in (5).

Finally, the scope of the method was broadened to include non-structural field problems which can be expressed in terms of a variational principle; the method has been applied to problems of fluid mechanics, heat transfer, electromagnetic phenomena and others⁽⁶⁾.

It is worth noting that there are several alternative approaches to finite element formulation which do not use the variational methods. These include the methods of Point Collocation and Residuals, both of which are described by Zienkiewicz⁽⁹⁾.

In the direct stiffness method, the variational principle used to derive the stiffness equations is the Principle of Minimum Potential Energy and can be stated : the potential energy is stationary with regard to all kinematically admissible variations of displacements from the equilibrium position. Displacements are kinematically admissible when they satisfy continuity within the continuum and also the prescribed kinematic boundary conditions. This may be written in symbolic form :-

$$\delta V = 0 \quad (2.2)$$

where $V = U + \Omega$ the total potential energy, U is the strain energy of the body and Ω is the potential energy of the applied loads.

Having established a basis for the derivation of element stiffness properties attention can now be centred on the selection of suitable displacement functions for the element chosen. In general, the displacements within an element are of the form :-

$$u(x,y) = [N(x,y)] \{\delta\}_e \quad (2.3)$$

where $\{\delta\}_e$ is a vector of element nodal point displacements (generalised displacements) and $[N(x,y)]$ is a matrix of suitable shape functions.

The problem was first approached by Melosh⁽³⁾ who stated essential as well as some optional characteristics for the shape functions. The paper was subsequently reviewed by Irons⁽⁷⁾ and the selection criteria for the stiffness method were listed in the now generally accepted form;

- (1) It must be possible to represent all the rigid body modes of an element.

For two-dimensional plane problems the rigid body modes of an element are translation in the two coordinate (x and y) directions and rotation in the coordinate (xOy) plane. Failure to satisfy this requirement causes strains to be induced by the rigid body modes, an increase in the strain energy results and the overall equilibrium conditions of the element are violated.

- (2) It must be possible to represent states of constant strain.

This condition may be appreciated physically if the mesh is considered to be finely subdivided to the stage where infinitesimal strain is constant over an element. Failure to allow for this requirement means that there is no guarantee of convergence of strains to continuous functions across element boundaries; in general⁽⁸⁾, they will not converge at all!

(3) Where neighbouring elements meet between nodes there must be no discontinuity of displacements between the adjacent elements.

This compatibility criterion ensures that no holes or overlaps will appear in the continuum after the solution and can be related to the kinematic admissibility of displacements as previously stated in the definition of the Principle of Minimum Potential Energy.

It can be appreciated at this early stage that a practical application of the finite element procedure will involve the generation, organisation and manipulation of large amounts of numbers. For this reason the method is generally formulated using the concise matrix notation. Indeed the method lends itself so well to this notation that it is often classified under the general heading of 'Matrix Methods of Structural Analysis'. Another essential adjunct to a finite element analysis is the use of electronic digital computers capable of performing the voluminous numerical operations involved. The status the method enjoys in the field of numerical stress analysis can be directly attributed to the power of such computers.

The earliest elements used for the analysis of two-dimensional problems were simple triangles with three nodes (the apices of the triangles) and two displacements at each node. The stresses in these elements are constant and consequently, large numbers of elements have to be used in order to predict stress gradients and stress concentrations with any reliable accuracy. With large numbers of elements, data preparation becomes extremely time-consuming and therefore considerable effort has been directed towards discovering better types of elements. One approach to the problem was to increase the number of nodes present per element, which results in families of elements, Fig. 2.4, with more boundary and internal nodes⁽⁹⁾. Using these complex straight sided elements, fewer elements need to be used in an analysis but a poor geometric representation of an arbitrary curved boundary will result.

This difficulty was overcome and a new phase in finite element theory was originated by Irons⁽¹⁰⁾ who presented a two-dimensional curved finite element which was later termed 'isoparametric'. (The term is derived from the fact that the same parametric representation is used to describe the element geometry and displacement pattern.) The family was later extended by Ergatoudis et al⁽¹¹⁾ and by Irons⁽¹²⁾. A more general formulation was given by Argyris⁽¹³⁾ but no numerical results were included, a comprehensive bibliography of problems solved using isoparametric elements may be found in Refs. 14 and 6.

To facilitate the modelling of curved element boundaries, the element is mapped from the physical x, y plane to a new ξ, η plane, Fig. 2.5. This mapping can be achieved by using the basic element shape functions, simultaneously used for representing the variation of the unknown displacements, and the transformation of coordinates. A standard procedure is followed to construct the required derivatives of the displacements and the integration necessary for forming the element stiffness matrices is performed numerically. These and other relevant steps will be discussed in subsequent chapters when the computer programs are described.

Apart from the obvious advantage of permitting curved element boundaries to be modelled accurately, the method has other important advantages. Irons⁽⁸⁾ refers to the idea of 'mathematical programming' where, for example, standard subroutines are used to calculate the shape functions and their derivatives and numerical integration relieves the programmer of the tedious and lengthy algebra required for exact integration. The programming effort for new elements and the probability of algebraic errors are thus greatly reduced. Irons⁽¹⁰⁾ also pointed out that elements need not have the same number of nodes on each side, Fig. 2.5. The theory for this type of degraded element is identical to that

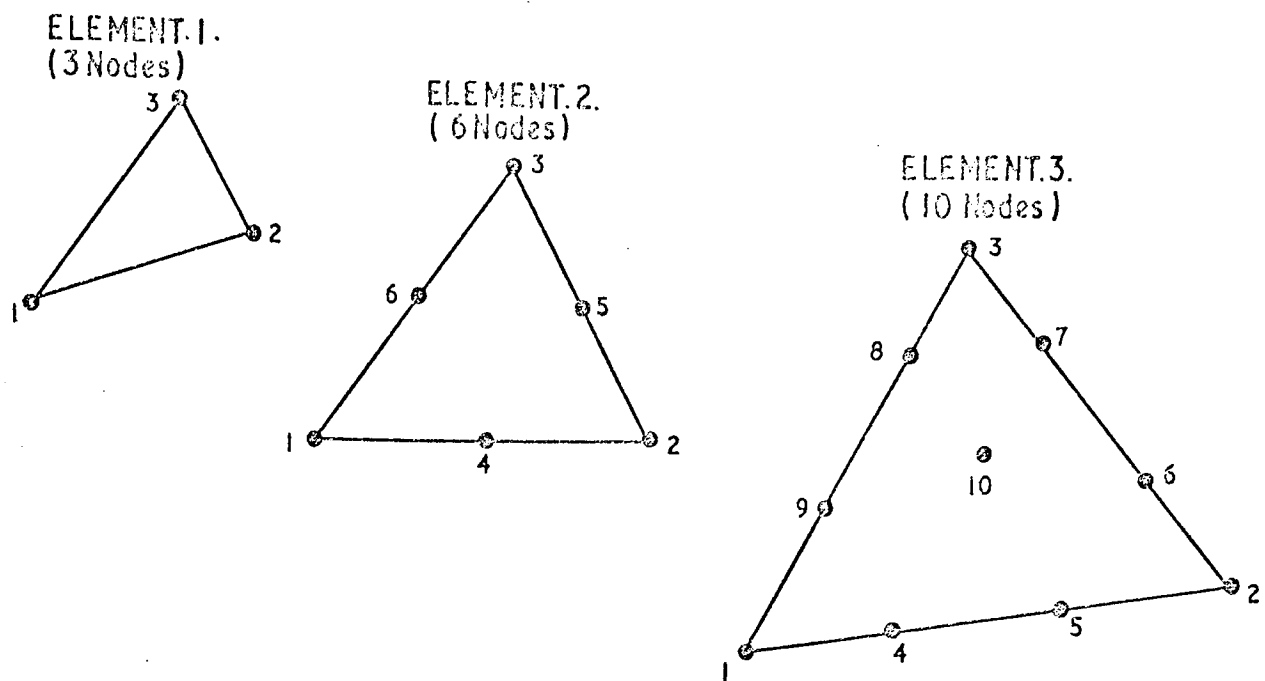


FIG 2.4. FAMILY OF STRAIGHT SIDED ELEMENTS.

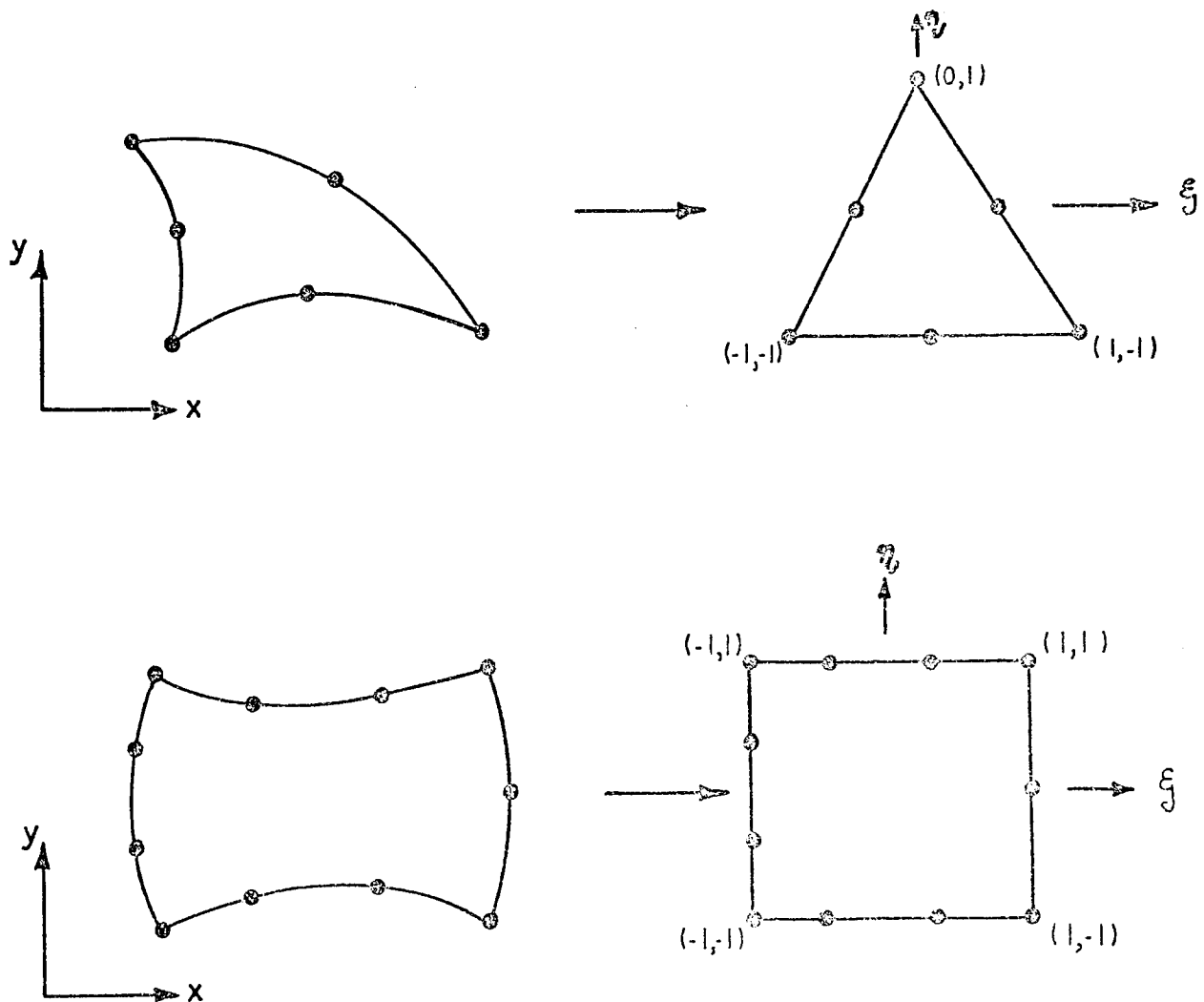


FIG 2.5 MAPPING ELEMENT SHAPES.

of the standard form. Bond et al⁽¹⁵⁾ present shape functions for an 18 node element which was used in the investigation of concentration of stress which occurs in the region of a fillet radius.

In the subsequent chapters of this section the finite element theory is applied to the analysis of two dimensional problems of plane stress and plane strain. The element employed is the six-node triangle, formed using the isoparametric approach. A finite element program is presented at the end of the Section with examples to demonstrate its application and accuracy.

3. FINITE ELEMENT FORMULATION

3.1 INTRODUCTION

The general principles of the method having been discussed in the previous Chapter, it is the purpose of this Chapter to apply these principles to the analysis of two-dimensional plane problems using a six-node triangular element. Relevant matrix equations are derived in a form suitable for their implementation in a computer program.

In Section 3.2, a natural coordinate system for plane triangles is introduced which proves convenient in the derivation of the element stiffness properties. The concept of interpolation is discussed in Section 3.3 where the element shape functions are derived.

The importance of interpolation functions, where a function has unit value at one node and equals zero at all other nodal points, cannot be overstressed. These functions can be used for representing distributed loading, thickness variation across an element and any other variable of significance in an element⁽²¹⁾.

In Section 3.4 elasticity matrices are derived for various cases of material anisotropy, the particular case of a transversely isotropic material is considered in some detail. Laminated structures are an important class of materials whose properties may be described as transversely isotropic.

Finally, in Section 3.5, the overall equilibrium equations for the assembled structure are derived. In the process, the individual element stiffness properties are determined and described.

3.2 NATURAL COORDINATE SYSTEM FOR PLANE TRIANGLES

A local coordinate system is one that is defined for a particular element, whereas the coordinate system used for the entire body is known as the global system, Fig. 3.1. A natural coordinate system is a local system which permits the specification of a point within an element by a set of dimensionless numbers whose magnitudes never exceed unity. Such a coordinate system considerably simplifies the element stiffness formulation, particularly with regard to the integration which must be carried out over the element area.

A convenient natural coordinate system, for plane triangles⁽⁹⁾, is one based on areas A_1 , A_2 and A_3 , Fig. 3.2. The position of a point P is defined by its natural coordinates L_1 , L_2 , L_3 as $P(L_1, L_2, L_3)$, where :

$$L_1 = \frac{A_1}{A} , \quad L_2 = \frac{A_2}{A} , \quad L_3 = \frac{A_3}{A} \quad (3.1)$$

and A is the total area of the triangle.

For the three node triangle, nodes 1, 2 and 3 of Fig. 3.1, the relation between Cartesian and local coordinates is given by :

$$\begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} \quad (3.2)$$

It is important to note that only two of the three natural coordinates are independent and that the sum

$$L_1 + L_2 + L_3 = 1 \quad (3.3)$$

The set of coordinates L_i ($i = 1, 2, 3$) are independent of the global reference system and of the shape of the element. Equations $L_i = \text{constant}$ represent lines parallel to the corresponding sides, Fig. 3.3. The three apices have coordinates $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$, the midpoints of the sides $(\frac{1}{2}, \frac{1}{2}, 0)$, $(0, \frac{1}{2}, \frac{1}{2})$, $(\frac{1}{2}, 0, \frac{1}{2})$ and the centroid $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

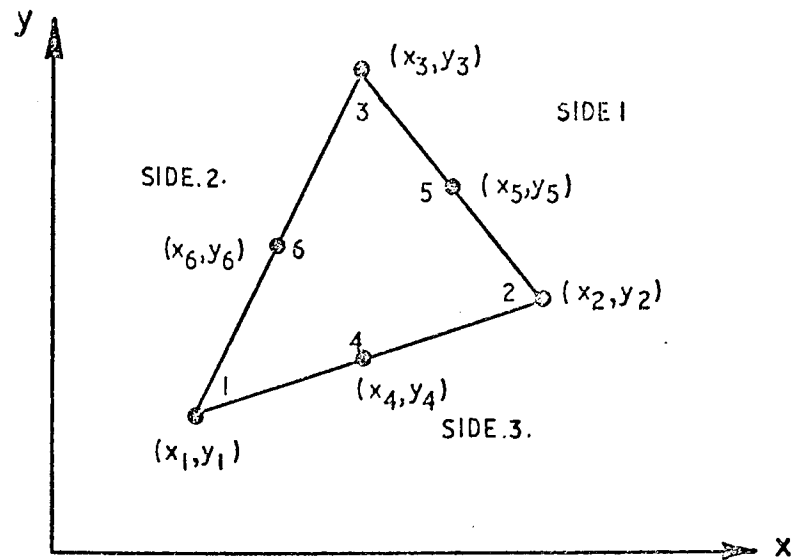


FIG. 3.1 GLOBAL COORDINATE SYSTEM

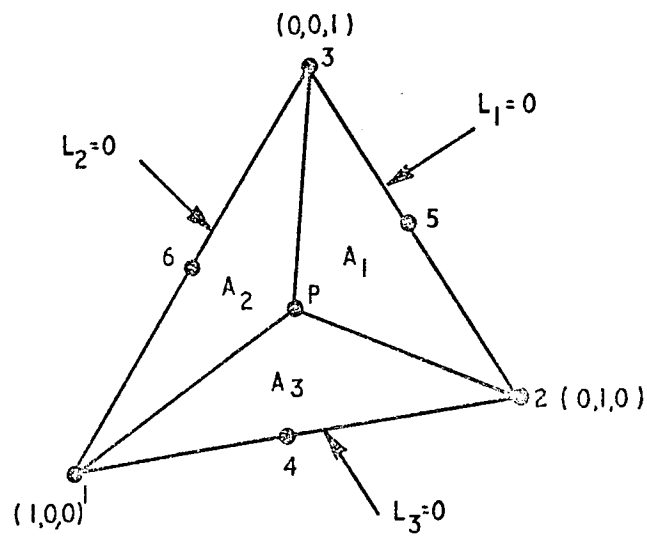


FIG. 3.2 NATURAL COORDINATE SYSTEM

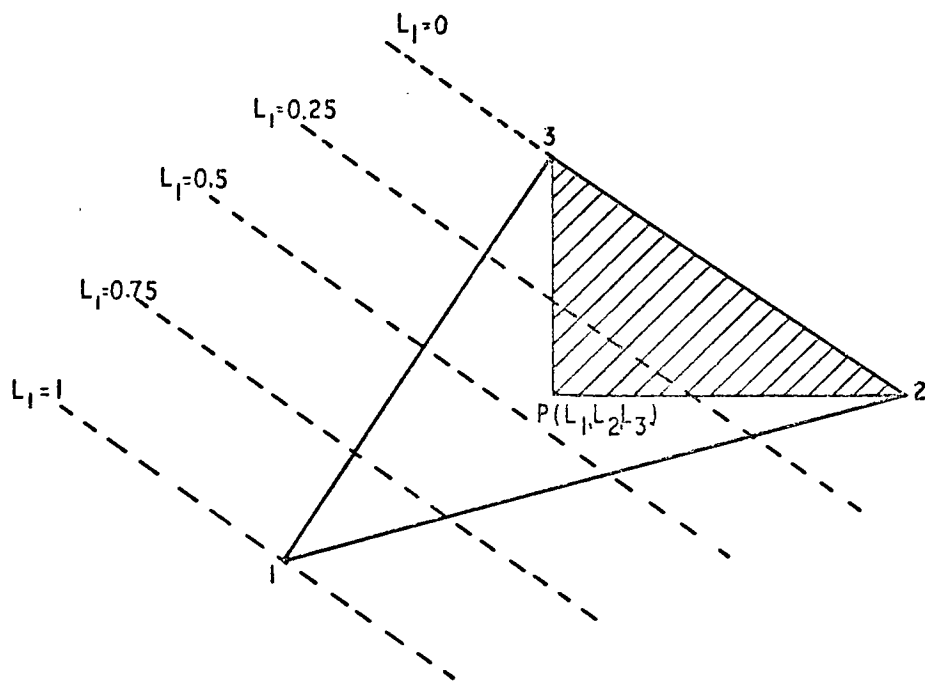


FIG. 3.3 AREA COORDINATES

3.3 INTERPOLATION FORMULA FOR SIX-NODE TRIANGLE

To introduce the idea of interpolation it is instructive to reconsider one of the basic concepts of the finite element method.

The finite element method is a piecewise approximation in which the solution to a complex problem is obtained by subdividing the region into an assemblage of elements and representing the solution within each element by a simple function. These functions are known as shape functions and relate the displacement components within an element to the displacements of the nodal points in the element.

To illustrate one possible method of obtaining suitable shape functions, consider the triangular element, Fig. 3.4 and assume polynomial expansions for the displacements in the form :

$$\begin{aligned} u &= \alpha_1 + \alpha_2 x + \alpha_3 y \\ v &= \alpha_4 + \alpha_5 x + \alpha_6 y \end{aligned} \quad (3.4)$$

where u and v are the displacement components and α_i are the generalised coordinates.

The choice of equations 3.4 is not arbitrary since the displacements must satisfy conditions of inter-element compatibility, constant strain and rigid body modes as discussed in Chapter 2.

Substituting coordinates of the various nodes, a set of simultaneous equations will be obtained.

$$\begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & y_1 \\ 1 & x_2 & y_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_2 & y_2 \\ 1 & x_3 & y_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_3 & y_3 \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{Bmatrix}$$

(3.5)

$$\text{or } \{\delta\}_e = [D] \{\alpha\} \quad (3.6)$$

Formally,

$$\{\alpha\} = [D]^{-1} \{\delta\}_e \quad (3.7)$$

Equation 3.4 may now be written as :

$$\{\phi\} = [P] \{\alpha\} = [P] [D]^{-1} \{\delta\}_e \quad (3.8)$$

in which

$$[P] = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix}$$

Thus the shape function for the element is defined by :

$$\{\phi\} = [N] \{\delta\}_e \quad (3.9)$$

and the shape function matrix $[N]$ can be found from :

$$[N] = [P][D]^{-1} \quad (3.10)$$

The process described by equations 3.4 to 3.10 is frequently employed for obtaining shape functions although considerable algebraic difficulty is always experienced in the inversion of $[D]$ in general terms suitable for all element geometries. It is of value, therefore, to consider alternative methods for obtaining shape functions $N_i(x,y)$ directly without recourse to the matrix operations described. Before continuing, however, one important property of these functions will be mentioned.

Using the defining relation, equation 3.9, and noting that this equation is valid for all $\{\phi\}_e$, then :

$$N_i = 1$$

at node i and zero for all other nodes. The typical form of the shape functions for a six node triangle is illustrated isometrically for a mid-side and corner node in Fig. 3.5

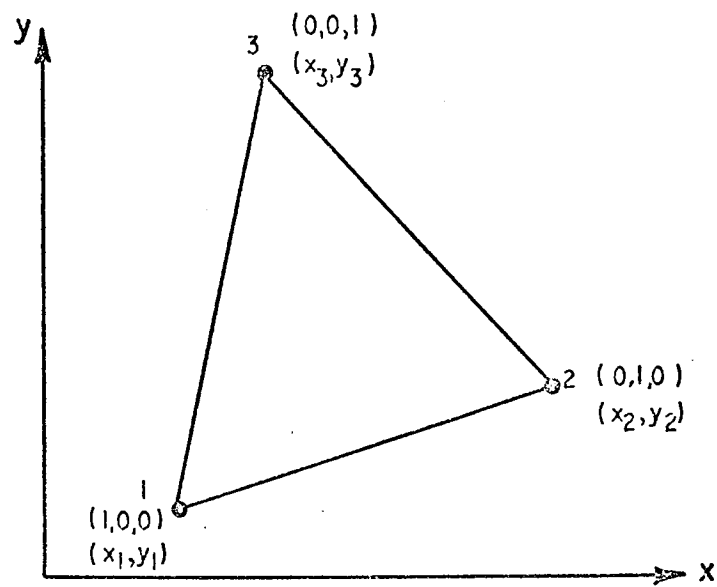


FIG.3.4. THREE NODE TRIANGULAR ELEMENT.

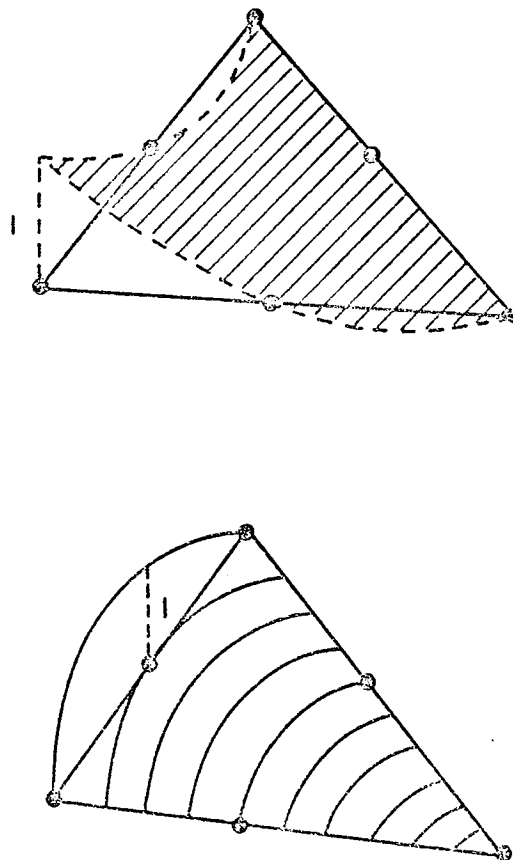


FIG.3.5 SHAPE FUNCTIONS FOR 6-NODE TRIANGLE.

By selecting an interpolation function to represent a shape function the inversion of $[D]$, equation 3.7, may be avoided. An interpolation function is, by definition, a function which has unit value at one nodal point and equals zero at all other nodal points. Interpolation functions are most conveniently expressed in terms of the natural coordinates introduced in the previous section, 3.2.

In many cases, particularly for simple element shapes, the interpolation formulae may be written directly. For the example of Fig. 3.4,

$$N_1 = L_1, \quad N_2 = L_2 \quad \text{and} \quad N_3 = L_3$$

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} L_1 & 0 & L_2 & 0 & L_3 & 0 \\ 0 & L_1 & 0 & L_2 & 0 & L_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} \quad (3.11)$$

$$\{\phi\} = [N]\{\delta\}_e$$

To derive shape functions for other elements in the family a simple recurrence relation can be derived^(9,16).

Consider a triangle of order n for which the shape functions are known, it is required to find the shape functions for a triangle of order $n + 1$, Fig. 3.6. For a typical node i of the n^{th} order triangle

$$N_i(L_1^n, L_2^n, L_3^n) \quad (3.12)$$

This shape function can also be expressed in terms of the area coordinates of the larger triangle $1\ 2^*\ 3^*$ if the area coordinates can be related.

It will then still retain a value of unity at node i and at all other nodes of the new triangle with the exception of those along the base $2^*\ 3^*$. As L_1^{n+1} is zero along the base of the larger triangle it can be seen that

$$N_i^{n+1} = C L_1^{n+1} N_i^n \quad (3.13)$$

will be the desired shape function providing C is so scaled as to achieve

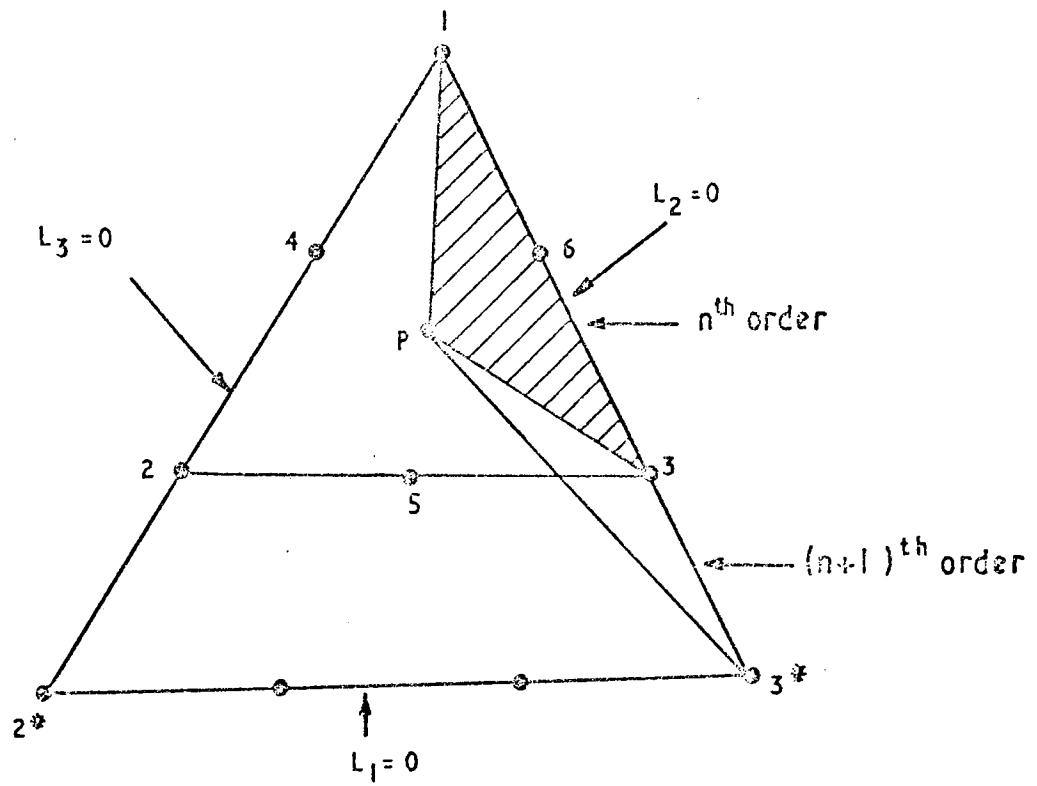


FIG. 3.6 RECURRENCE GENERATION OF SHAPE FUNCTIONS FOR TRIANGLES.

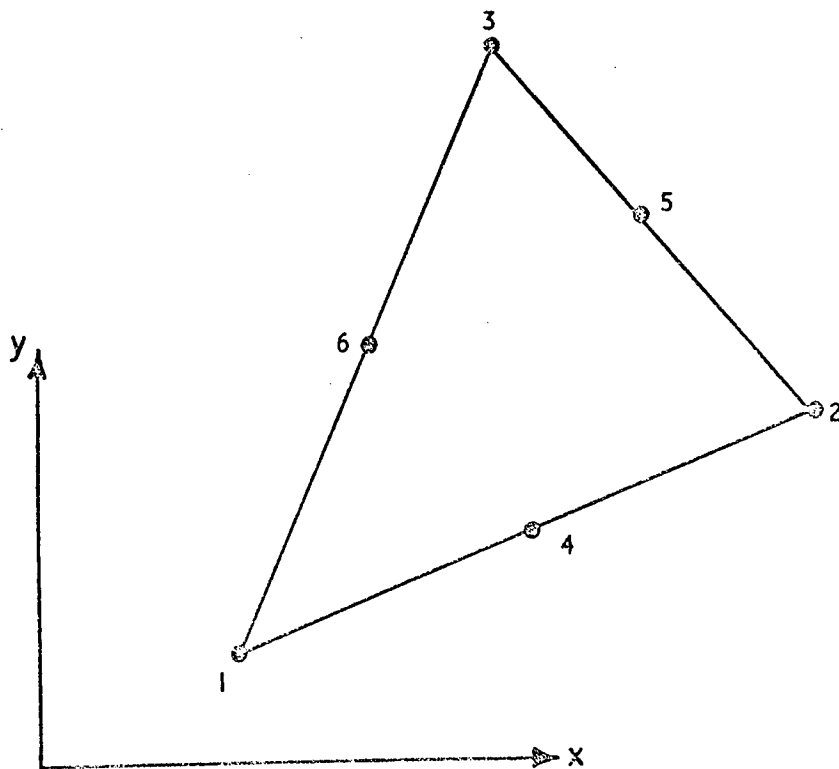


FIG. 3.7 LOCAL NODAL NUMBERING FOR SIX-NODE TRIANGULAR ELEMENT.

unity at i . The scaling factor is given by :

$$C = \frac{n+1}{1} \quad (3.14)$$

where 1 is the number of the layer in which node i lies below the base $2^* 3^*$. While the above process will not directly yield shape functions for points on the base these can be obtained by a simple transposition of indices or alternatively by redrawing Fig. 3.6 such that the base of the larger triangle becomes $1^* 3^*$ or $1^* 2^*$.

The relation between the two sets of area coordinates may be found from Fig. 3.6

$$L_2^n = \frac{\text{Area } P13}{\text{Area } 123}, \quad L_2^{n+1} = \frac{\text{Area } P13^*}{\text{Area } 12^*3^*}$$

therefore,

$$\begin{aligned} L_2^n &= \frac{\text{Area } P13}{\text{Area } P13^*} \frac{\text{Area } 12^*3^*}{\text{Area } 123} L_2^{n+1} \\ &= \left(\frac{n}{n+1} \right) \left(\frac{n+1}{n} \right)^2 L_2^{n+1} \\ &= \left(\frac{n+1}{n} \right) L_2^{n+1} \end{aligned} \quad (3.15a)$$

$$\text{Similarly } L_3^n = \left(\frac{n+1}{n} \right) L_3^{n+1} \quad (3.15b)$$

and by the fact that $L_1 + L_2 + L_3 = 1$, equation 3.3,

$$L_1^n = \frac{1}{n} \left[(n+1) L_1^{n+1} - 1 \right] \quad (3.15c)$$

For the first order triangle ($n = 1$), Fig. 3.4, the shape functions are given by equation 3.11. Suppose it is required to find the shape functions of the quadratic or six node triangle ($n = 2$), Fig. 3.5.

From equations 3.15,

$$L_1^n = (2 L_1^{n+1} - 1), \quad L_2^n = 2 L_2^{n+1}, \quad L_3^n = 2 L_3^{n+1}$$

and by equations 3.13 and 3.14, with a cyclic permutation of the indices,

$$\begin{aligned} \text{the shape functions are for corner nodes } N_1 &= (2 L_1 - 1) L_1, \\ N_2 &= (2 L_2 - 1) L_2, \quad N_3 = (2 L_3 - 1) L_3 \text{ and mid-side nodes } N_4 = 4 L_1 L_3, \\ N_5 &= 4 L_2 L_3, \quad N_6 = 4 L_3 L_1. \end{aligned} \quad (3.16)$$

The numbering sequence for the nodes is shown in Fig. 3.7.

Writing equations 3.16 in the form of equation 3.11 yields :

$$\{\phi\} = [N]^t \{\delta\}_e \quad (3.17)$$

where $\{\phi\}^t = \{u \quad v\}$

$$[N]^t = \begin{bmatrix} (2L_1 - 1)L_1 & 0 \\ 0 & (2L_1 - 1)L_1 \\ (2L_2 - 1)L_2 & 0 \\ 0 & (2L_2 - 1)L_2 \\ (2L_3 - 1)L_3 & 0 \\ 0 & (2L_3 - 1)L_3 \\ 4L_1L_3 & 0 \\ 0 & 4L_1L_3 \\ 4L_2L_3 & 0 \\ 0 & 4L_2L_3 \\ 4L_3L_1 & 0 \\ 0 & 4L_3L_1 \end{bmatrix} \quad (3.18)$$

and

$$\{\delta_e\}^t = \{u_1 \quad v_1 \quad u_2 \quad v_2 \quad u_3 \quad v_3 \quad u_4 \quad v_4 \quad u_5 \quad v_5 \quad u_6 \quad v_6\}$$

It will prove more convenient in later sections to re-write equations 3.17 and 3.18 in the alternative but equivalent form :

$$u = N_1u_1 + N_2u_2 + N_3u_3 + N_4u_4 + N_5u_5 + N_6u_6 = \{N\}^t \{\delta_u\}_e \quad (3.19)$$

$$v = N_1v_1 + N_2v_2 + N_3v_3 + N_4v_4 + N_5v_5 + N_6v_6 = \{N\}^t \{\delta_v\}_e$$

$$\text{where } \{N\}^t = \left\{ (2L_1-1)L_1 \quad (2L_2-1)L_2 \quad (2L_3-1)L_3 \quad 4L_1L_3 \quad 4L_2L_3 \quad 4L_3L_1 \right\} \quad (3.19a)$$

3.3.1 Use of Shape Functions for Element Mapping

It is required to find a suitable transformation function which maps a curved element from the physical x,y plane to the ξ, η plane in which it becomes a regular figure, Fig. 2.5. A convenient method of establishing the coordinate transformations is to use the shape functions previously derived, equations 3.16, to represent the variation of the

unknown function. The transformation is :

$$\begin{aligned}x &= N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 + N_5x_5 + N_6x_6 \\y &= N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4 + N_5y_5 + N_6y_6\end{aligned}\tag{3.20}$$

where the N_i are identical to those of equation 3.19a.

Note that since the parametric forms for u,v (equation 3.19) and x,y (equation 3.20) are identical, the elements are termed 'iso-parametric'.

It can easily be shown^(10,11) that these shape functions used simultaneously to represent element displacement and geometry satisfy the conditions of constant element strain, rigid body modes and compatible inter-element displacement required for convergence of the final solution as discussed in the introduction, Chapter 2.

3.4 CONSTITUTIVE RELATIONS

In keeping with one of the primary objectives of the project; to develop a general two-dimensional plane stress/strain finite element program, this section deals with the development of the coefficients of the elasticity matrix for various cases of material anisotropy.

There are many examples of structural materials, both naturally occurring and synthetically produced which exhibit anisotropic behaviour, i.e. in which resistance to mechanical actions is different for different directions. The group of materials known collectively as fibre reinforced composites, which by their very nature are grossly anisotropic, present probably the greatest single challenge to the stress analyst/designer. He has the freedom to design and stress the component in the normal way and also 'mathematically' design a suitable composite material to fulfill the requirements!

The finite element method, in which the stiffness matrices are formed for each individual element, is particularly suited to the analysis of anisotropic materials. The elastic properties of each individual element may be assigned at will and such properties may vary in an arbitrary manner without affecting the normal assembly and solution process.

3.4.1 Elasticity Matrix

The fundamental assumption of the linear theory of elasticity is that each of the six components of strain at every point must be a linear function of the six components of stress at that point, this assumption leads to the matrix equation :

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{xz} \\ \tau_{xy} \end{Bmatrix} \quad (3.21)$$

where symbols $\epsilon_x, \epsilon_y, \dots, \sigma_x, \sigma_y, \dots$, etc. have their usual connotations.

These equations contain 36 coefficients a_{ij} , the so-called elastic constants. It may be shown⁽¹⁸⁾, using the Principle of Conservation of Energy that the coefficients of the above matrix must be symmetrical and that 21 independent elastic constants are sufficient to describe any material. The elastic constants a_{ij} described above are related⁽¹⁷⁾ to the more common 'technical constants', E_x, E_y , etc.

The number of elastic constants for any given material may be further reduced by consideration of planes or axes of elastic symmetry. The technique is illustrated by developing the two-dimensional stress/strain relationships for a general orthotropic material. Results are then stated for the isotropic case and for the more practical transversely isotropic material⁽¹⁹⁾, Fig. 3.8.

3.4.2 Orthotropic Material

An orthotropic material is a material which has different properties in three orthogonal directions. If the coordinate axes O_x, O_y, O_z are taken in these directions then the material has the property that any plane perpendicular to one of these axes is a plane of elastic symmetry.

It follows from the material symmetry that precisely the same stress/strain relations must be obtained for coordinate system O_x^1, O_y^1, O_z^1 , Fig. 3.9, which represents a rotation of the original system to π^c about O_z . Referred to the new axes,

$$\epsilon_x^1 = a_{11}\sigma_x^1 + a_{12}\sigma_y^1 + a_{13}\sigma_z^1 + a_{14}\tau_{yz}^1 + a_{15}\tau_{xz}^1 + a_{16}\tau_{xy}^1 \quad (3.22)$$

where primed components refer to the new coordinate system.

Using the equations for the transformation of stress and strain components⁽¹⁷⁾, for a rotation of π^c about O_z^1 ,

$$\begin{aligned} \epsilon_x^1 &= \epsilon_x, \quad \sigma_x^1 = \sigma_x, \quad \sigma_y^1 = \sigma_y, \quad \sigma_z^1 = \sigma_z, \quad \tau_{yz}^1 = -\tau_{yz}, \quad \tau_{zx}^1 = -\tau_{zx}, \\ \tau_{xy}^1 &= \tau_{xy} \end{aligned} \quad (3.23)$$

Upon substituting the relations of equation 3.20 into equation 3.22,

$$\epsilon_x = a_{11}\sigma_x + a_{12}\sigma_y + a_{13}\sigma_z - a_{14}\tau_{yz} - a_{15}\tau_{xz} + a_{16}\tau_{xy} \quad (3.24)$$

Comparing coefficients of equation 3.24 with the first of equations 3.21,

$$a_{14} = a_{15} = 0$$

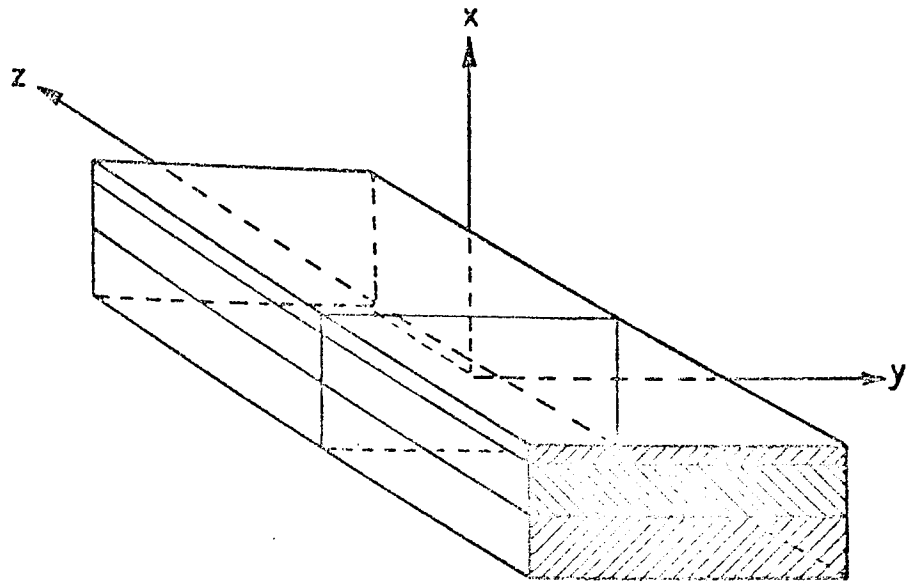


FIG 3.8 TRANSVERSELY ISOTROPIC (LAYERED) MATERIAL

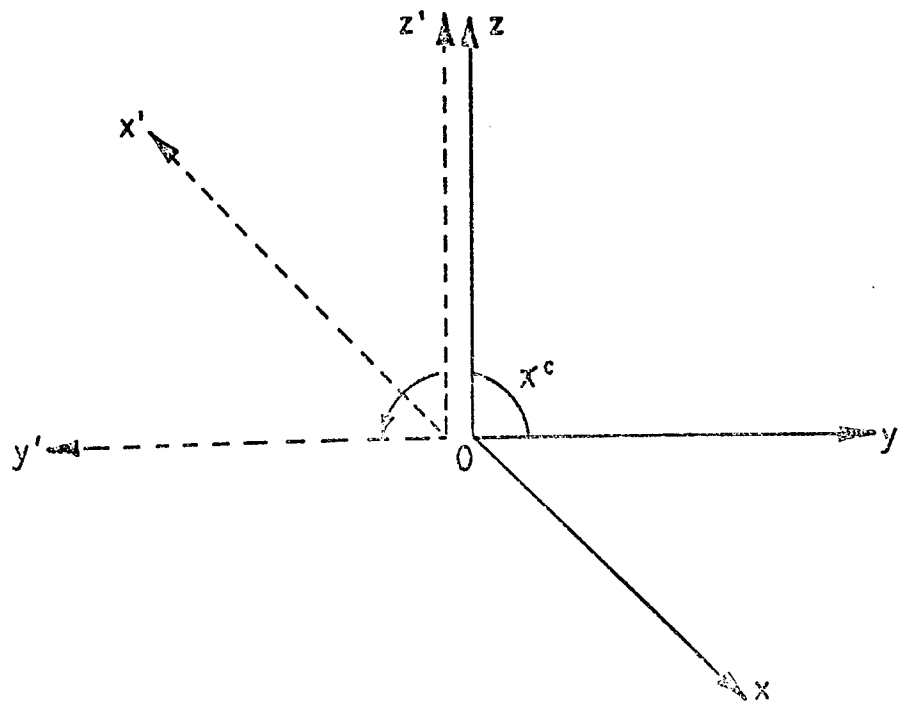


FIG 3.9 ROTATION OF COORDINATE AXES FOR PLANES OF ELASTIC SYMMETRY.

Treating the other five equations corresponding to equation 3.22 in a similar manner and repeating the complete process for a rotation of π^c about σ_x and σ_y , gives :

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 \\ & a_{22} & a_{23} & 0 & 0 & 0 \\ & & a_{33} & 0 & 0 & 0 \\ & & & a_{44} & 0 & 0 \\ \text{SYMMETRICAL} & & & & a_{55} & 0 \\ & & & & & a_{66} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{xz} \\ \tau_{xy} \end{Bmatrix} \quad (3.25)$$

Considering plane strain conditions, Appendix B, in surfaces parallel to the xy-plane,

$$\epsilon_z = \gamma_{yz} = \gamma_{xz} = 0 \quad (3.26)$$

Substituting equation 3.26 into equation 3.22 gives :

$$\sigma_z = -\frac{1}{a_{33}} (a_{13}\sigma_x + a_{23}\sigma_y) \quad (3.27)$$

Finally, inserting equation 3.27 into equation 3.25, gives :

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & 0 \\ & A_{22} & 0 \\ \text{SYMM.} & & A_{33} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \quad (3.28)$$

where,

$$A_{11} = \frac{a_{11}a_{33} - a_{13}^2}{a_{33}} ; \quad A_{12} = \frac{a_{12}a_{33} - a_{13}a_{23}}{a_{33}} ;$$

$$A_{22} = \frac{a_{22}a_{33} - a_{23}^2}{a_{33}} ; \quad A_{33} = a_{66}.$$

For plane stress problems, Appendix B,

$$\sigma_x = \tau_{zy} = \tau_{xz} = 0 \quad (3.29)$$

Substituting equation 3.29 into equation 3.25 gives :

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ & a_{22} & 0 \\ & & a_{66} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \quad (3.30)$$

In the solution by stiffness methods, it is more convenient to solve equations 3.28 and 3.30 in order to express stresses in terms of strains. Performing the required inversions and introducing the technical constants, the following equations are obtained ;

(a) for plane strain,

$$\begin{aligned} \epsilon_x &= \frac{1}{K} \left[\frac{(E_y - \nu_{yz}^2 E_z)}{E_y^2} \epsilon_x + \frac{(\nu_{xy} E_y + \nu_{xz} \nu_{yz} E_z)}{E_x E_y} \epsilon_y \right] \\ \epsilon_y &= \frac{1}{K} \left[\frac{(\nu_{xy} E_y + \nu_{xz} \nu_{yz} E_z)}{E_x E_y} \epsilon_x + \frac{(E_x - \nu_{xz}^2 E_z)}{E_x^2} \epsilon_y \right] \end{aligned} \quad (3.31)$$

$$\tau_{xy} = G_{xy} \gamma_{xy}$$

and

$$K = \frac{1}{E_x E_y} \left(-\frac{\nu_{xy}^2 E_y}{E_x} - \frac{2 \nu_{xy} \nu_{xz} \nu_{yz} E_z}{E_x} + 1 - \frac{\nu_{xz}^2 E_z}{E_x} - \frac{\nu_{yz}^2 E_z}{E_y} \right)$$

and

(b) for plane stress,

$$\begin{aligned} \sigma_x &= \frac{E_x^2 E_y}{E_x - E_y \nu_{xy}^2} \left(\frac{1}{E_y} \epsilon_x + \frac{\nu_{xy}}{E_x} \epsilon_y \right) \\ \sigma_y &= \frac{E_x E_y}{E_x - E_y \nu_{xy}^2} \left(\nu_{xy} \epsilon_x + \epsilon_y \right) \end{aligned} \quad (3.32)$$

$$\tau_{xy} = G_{xy} \gamma_{xy}$$

3.4.3 Isotropic Material

An isotropic material is one that has point symmetry; that is, every plane in the body is a plane of symmetry of material behaviour. The number of independent elastic constants is two, and the stress/strain relations have the form :

(a) for plane strain,

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \nu/(1-\nu) & 0 \\ \nu/(1-\nu) & 1 & 0 \\ 0 & 0 & (1-2\nu)/2(1-\nu) \end{bmatrix} \quad (3.33)$$

and

(b) for plane stress,

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \quad (3.34)$$

3.4.4 Transversely Isotropic

This is a material of considerable engineering interest, particularly in the study of laminates or stratified rock. In such materials perfect symmetry is obtained about any axis perpendicular to the plane of stratification, Fig. 3.8. The elastic properties remain invariant for any arbitrary rotation of the y and z axes about x, Fig. 3.10. The stress strain relations may be derived in an analogous manner to the orthotropic case discussed in Section 3.4.2. In the transversely isotropic case, however, when comparing coefficients, equations of the type :

$$a_{12} = a_{12} \cos^2 \alpha + a_{13} \sin^2 \alpha - a_{14} \sin \alpha \cos \alpha$$

arise and it should be remembered that this equation is valid for all α , therefore :

$$a_{12} = a_{13} \quad \text{and} \quad a_{14} = 0$$

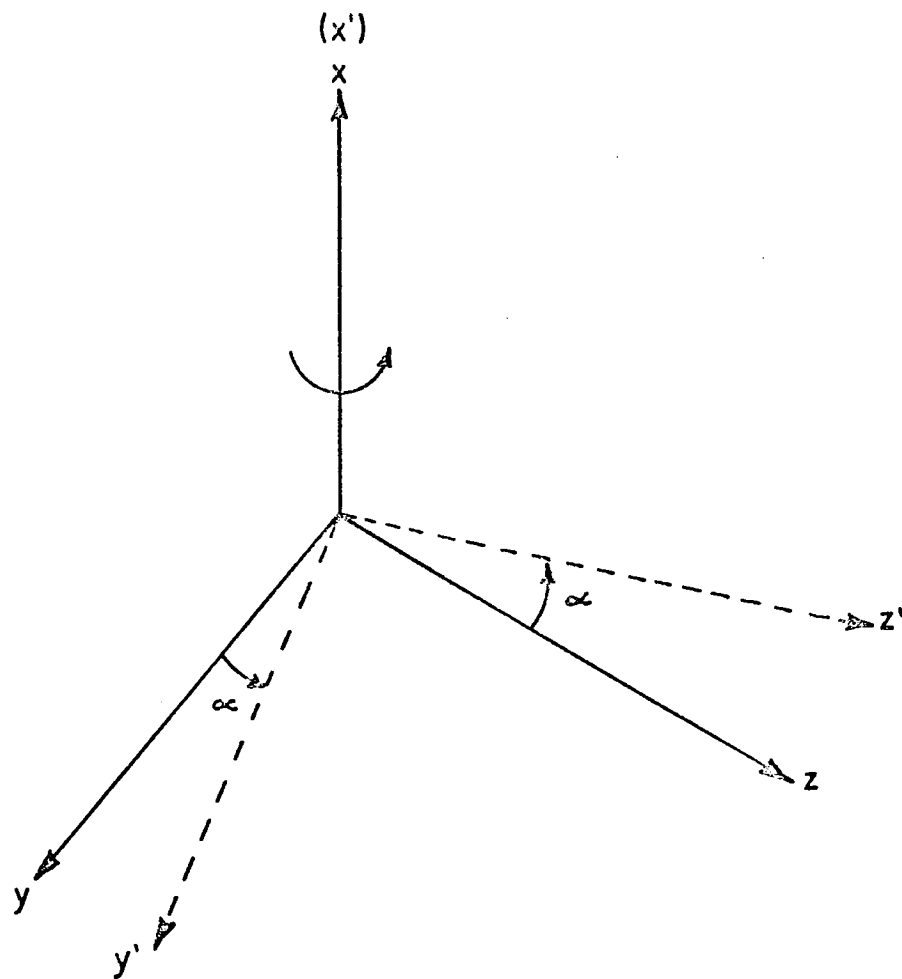


FIG 3.10 ELASTIC SYMMETRY ABOUT x-AXIS.

Continuing this procedure for the remaining strain components :

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{yz} \\ \gamma_{zx} \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{12} & 0 & 0 & 0 \\ & a_{22} & a_{23} & 0 & 0 & 0 \\ & & a_{33} & 0 & 0 & 0 \\ & & & a_{44} & 0 & 0 \\ \text{SYMM.} & & & & 2(a_{22}-a_{23}) & 0 \\ & & & & & a_{44} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{zx} \\ \tau_{xy} \end{Bmatrix} \quad (3.35)$$

Proceeding in a similar manner as in the orthotropic case, equations 3.26 to 3.30 and upon inserting the 'technical constants',⁽¹⁷⁾, the stress strain relations take the form ;

(a) for plane strain ;

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = K \begin{bmatrix} (1 - \nu_{zy}^2) & \frac{E_{yy}}{E_{xx}} \nu_{yx} (1 + \nu_{zy}) & 0 \\ & \frac{E_{yy}}{E_{xx}} (1 - \frac{E_{yy}}{E_{xx}} \nu_{yx}^2) & 0 \\ \text{SYMM.} & & \frac{1}{K} G_{yz} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (3.36)$$

where

$$K = \frac{E_{xx}}{(1 + \nu_{zy}) (1 - \nu_{zy} - 2 \frac{E_{yy}}{E_{xx}} \nu_{yx}^2)}$$

and,

(b) for plane stress ;

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = K^1 \begin{bmatrix} 1 & \frac{E_{yy}}{E_{xx}} \nu_{yx} & 0 \\ & \frac{E_{yy}}{E_{xx}} & 0 \\ & & \frac{1}{K^1} G_{yz} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (3.37)$$

where

$$K^1 = \frac{E_{xx}}{\left(1 - \frac{E_{yy}}{E_{xx}}\right) \nu_{yx}^2}$$

NOTE: In equations 3.36 and 3.37 the elastic constants ν_{zy} and E_{yy} may be associated with the plane of the strata while ν_{yx} , E_{xx} and G_{yz} may be associated with the behaviour normal to the plane of the strata.

3.4.5 Transformation of the Elasticity Matrix Under a Transformation of the Coordinate System

In cases where the direction of the strata does not coincide with the global axes, Fig. 3.11, it is necessary to transform the elasticity matrix developed for the element in the local coordinate system x^1y^1 to the global coordinate system xy .

If primed symbols are taken to represent the element coordinate system, then the equation relating stress and strain components for the element may be written,

$$\{\sigma^1\} = [c^1] \{\epsilon^1\} \quad (3.38)$$

If the angle between the coordinate systems is θ^0 , then⁽²⁰⁾

$$\begin{aligned} \{\sigma\} &= \begin{bmatrix} \cos^2 \theta & \sin^2 \theta & -2 \sin \theta \cos \theta \\ \sin^2 \theta & \cos^2 \theta & 2 \sin \theta \cos \theta \\ \sin \theta \cos \theta & -\sin \theta \cos \theta & (\cos^2 \theta - \sin^2 \theta) \end{bmatrix} \{\sigma^1\} \\ &= [T] \{\sigma^1\} \end{aligned} \quad (3.39)$$

Since the strain energy is invariant with respect to coordinate transformation ,

$$\{\sigma^1\}^T \{\epsilon^1\} = \{\sigma\}^T \{\epsilon\} \quad (3.40)$$

or from equation 3.34 :

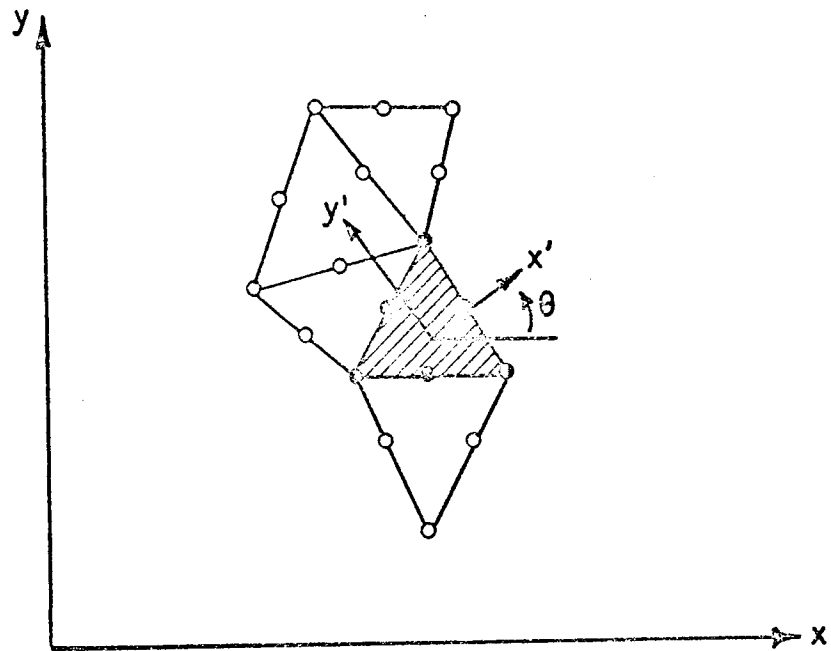


FIG 3.11 ELEMENT IN A LAYERED MATERIAL.

$$\{\sigma^1\}^T \{\epsilon^1\} = \{\sigma^1\}^T [T]^T \{\epsilon\}$$

$$\text{or} \quad \{\epsilon^1\} = [T]^T \{\epsilon\} \quad (3.41)$$

Substituting equations 3.39 and 3.37 into equation 3.36 and pre-multiplying by $[T]$

$$\{\sigma\} = [T] [c^1] [T]^T \{\epsilon\}$$

therefore,

$$[C] = [T] [c^1] [T]^T \quad (3.42)$$

where $[C]$ is the elasticity matrix referred to the global coordinate system.

3.5 GENERATION OF ELEMENT AND OVERALL STIFFNESS MATRICES AND LOAD VECTORS

It is the purpose of this section to describe, in symbolic form, the steps involved in formulating the element and overall system equations. Their numerical implementation is discussed in the following chapter.

The displacement finite element method of structural analysis is based on the Principle of Stationary Total Potential Energy using a modified form of the Rayleigh Ritz procedure for obtaining approximate solutions. In the conventional Ritz method the total potential energy is minimised with respect to trial displacement functions over the entire structure. The technique is restricted to simple shapes and is not suitable for automatic programming. In the modified Ritz procedure for the finite element method the potential energy is minimised with respect to undetermined nodal displacement functions for each element in the discretised structure. The above procedure will later be shown to result in a system of linear algebraic equations of the form :

$$[K] \{\delta\} = \{F\}$$

where $[K]$ is the overall stiffness matrix of dimensions $2n \times 2n$, where
 n is the number of nodes
 $\{\delta\}$ is the vector of nodal displacement components
 and $\{F\}$ is the generalised force vector.

The above equations can be formed by summing contributions from all the elements of the body. It is convenient, therefore, to derive the individual element stiffness properties and later to sum these over the body in an appropriate manner. Element equations are derived in section 3.5.2 whereas the assembly rules are discussed in section 3.5.3.

Before proceeding to the formal derivation in section 3.5.2, the coefficients of the matrix relating element strains to element nodal displacement components will be derived. This matrix is used in the element stiffness formulation and for clarity of presentation is derived separately, section 3.5.1

3.5.1 Derivation of the Strain-Element Nodal Displacement Matrix

In the section the matrix equation relating element strains to element nodal point displacements will be developed in a form suitable for its subsequent numerical implementation.

Referring to Section 3.2, it can be seen that three coordinates, L_1 , L_2 and L_3 are necessary to specify the position of a point within an element. Only two of these coordinates are independent, however, and it will prove convenient to replace the natural coordinates by a new set ξ , η , using equation 3.3, as follows :

$$\xi = L_1, \quad \eta = L_2 \quad \text{and} \quad L_3 = 1 - \xi - \eta \quad (3.43)$$

With these coordinates the shape functions of equation 3.19a become :

$$\{N\} = \begin{Bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \end{Bmatrix} = \begin{Bmatrix} \xi(2\xi - 1) \\ \eta(2\eta - 1) \\ 1 - 3\xi - 3\eta + 4\xi\eta + 2\xi^2 + 2\eta^2 \\ 4\xi\eta \\ 4(\eta - \xi\eta - \eta^2) \\ 4(\xi - \xi\eta - \xi^2) \end{Bmatrix} \quad (3.44)$$

For two-dimensional problems of plane stress or plane strain there are three components of the strain vector at any point.

$$\{\epsilon\}^T = \{\epsilon_x \quad \epsilon_y \quad \gamma_{xy}\} \quad (3.45)$$

where

$$\epsilon_x = \partial u / \partial x$$

$$\epsilon_y = \partial v / \partial y \quad (3.45a)$$

$$\text{and } \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

These equations can be expressed in terms of the nodal values of displacements by using the chain rule of differentiation as follows :

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial u}{\partial \eta} \frac{\partial \eta}{\partial x} \quad (3.46)$$

and from equations 3.19 and 3.44

$$\frac{\partial u}{\partial \xi} = \frac{\partial N_1}{\partial \xi} u_1 + \frac{\partial N_2}{\partial \xi} u_2 + \frac{\partial N_3}{\partial \xi} u_3 + \dots \frac{\partial N_6}{\partial \xi} u_6 = N_{j,\xi} u_j$$

and (3.47)

$$\frac{\partial u}{\partial \eta} = \frac{\partial N_1}{\partial \eta} u_1 + \frac{\partial N_2}{\partial \eta} u_2 + \frac{\partial N_3}{\partial \eta} u_3 + \dots \frac{\partial N_6}{\partial \eta} u_6 = N_{j,\eta} u_j$$

The repeated subscript, j , is summed from 1 to 6 and subscripts ξ, η , on N_j denote partial differentiation.

Equation 3.46 may now be written, using the results of equation 3.47

as :

$$\frac{\partial u}{\partial x} = \frac{\partial \xi}{\partial x} N_{j,\xi} u_j + \frac{\partial \eta}{\partial x} N_{j,\eta} u_j$$

An analogous procedure is followed for the remaining derivatives, ∂_v/∂_y , ∂_u/∂_y and ∂_v/∂_x .

The equations relating strains to element nodal displacements, u_i , can now be constructed as :

$$\{\epsilon\} = [B] \{\delta\}_e \quad (3.48)$$

where $[B]$ is the matrix,

$$[B] = \begin{bmatrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 \end{bmatrix}$$

with $B_j =$

$$\begin{bmatrix} \frac{\partial \xi}{\partial x} N_{j,\xi} + \frac{\partial \eta}{\partial x} N_{j,\eta} & 0 & 0 \\ 0 & \frac{\partial \xi}{\partial y} N_{j,\xi} + \frac{\partial \eta}{\partial y} N_{j,\eta} & 0 \\ \frac{\partial \xi}{\partial y} N_{j,\xi} + \frac{\partial \eta}{\partial y} N_{j,\eta} & \frac{\partial \xi}{\partial x} N_{j,\xi} + \frac{\partial \eta}{\partial x} N_{j,\eta} & 0 \end{bmatrix} \quad (3.48a)$$

It is necessary now to form the partial derivatives $\partial \xi / \partial x$, $\partial \xi / \partial y$, etc. To facilitate these calculations the coordinate transformation matrix or Jacobian matrix as it is known, is constructed as follows.

Using the chain rule of differentiation,

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \xi}$$

$$\frac{\partial N_i}{\partial \eta} = \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \eta}$$

or

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} \quad (3.49)$$

$$\text{where } [J] = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (3.49a)$$

The inverse of $[J]$ is

$$\frac{\partial(\xi,\eta)}{\partial(x,y)} = \left[\frac{\partial(x,y)}{\partial(\xi,\eta)} \right]^{-1}$$

or

$$\begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix} = \frac{1}{[J]} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \quad (3.50)$$

where $[J]$ is the determinant of the Jacobean matrix, known simply as the Jacobian.

$$[J] = \det \frac{\partial(x,y)}{\partial(\xi,\eta)} = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \quad (3.50a)$$

The derivatives, $\partial x / \partial \xi$, etc. may be found from equations 3.20 in the same way as used for the displacements in equation 3.47. This illustrates the use of the same shape functions to describe both element geometry and element displacement patterns : the Isoparametric Concept.

Consequently from equation 3.50,

$$\frac{\partial \xi}{\partial x} = \frac{1}{[J]} \frac{\partial y}{\partial \eta} = \frac{1}{[J]} N_{k,\eta} y_k$$

similarly for the others in the set.

With these derivatives equation 3.48a may be written

$$B_j = \frac{1}{[J]} \begin{bmatrix} q_j & 0 \\ 0 & p_j \\ p_j & q_j \end{bmatrix} \quad (3.51)$$

$$\text{where } \begin{Bmatrix} p_j \end{Bmatrix} = \begin{pmatrix} -N_{k,\eta} & N_{j,\xi} & + & N_{k,\xi} & N_{j,\eta} \end{pmatrix} x_k \quad (3.51a)$$

$$q_j = \begin{pmatrix} N_{k,\eta} & N_{j,\xi} & - & N_{k,\xi} & N_{j,\eta} \end{pmatrix} y_k$$

Calculation of the coefficients of the strain-nodal displacement matrix $[B]$ has been reduced to one involving only data for the coordinates of the element nodal points and the derivatives appearing in equation 3.51a. This matrix together with $/J/$ has to be evaluated at several points within each element and so, for ease of programming, it is written as a separate procedure. This procedure will be described in the subsequent chapter on Numerical Procedures.

3.5.2 Formulation of Element Stiffness Matrix and Load Vector

The procedure described is valid for small strains and rotations, using a linearised form of the strain-displacement equations.

For any two-dimensional problem of plane stress or plane strain the constitutive law takes the form :

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (3.52)$$

or

$$\{\sigma\} = [c] \{\epsilon\} \quad (3.52a)$$

The particular form of the matrix $[C]$ for the material under consideration may be found from section 3.4.

The applied external loads acting on the element, fig. 3.12, may be :

(a) distributed,

$$\{p\}^t = \{p_x \quad p_y\} \quad (3.53)$$

where p_x and p_y are surface tractions per unit area, or

(b) concentrated nodal loads,

$$\{f\}^t = \{f_{x1} \quad f_{y1} \quad \dots \quad f_{xo} \quad f_{yo}\} \quad (3.54)$$

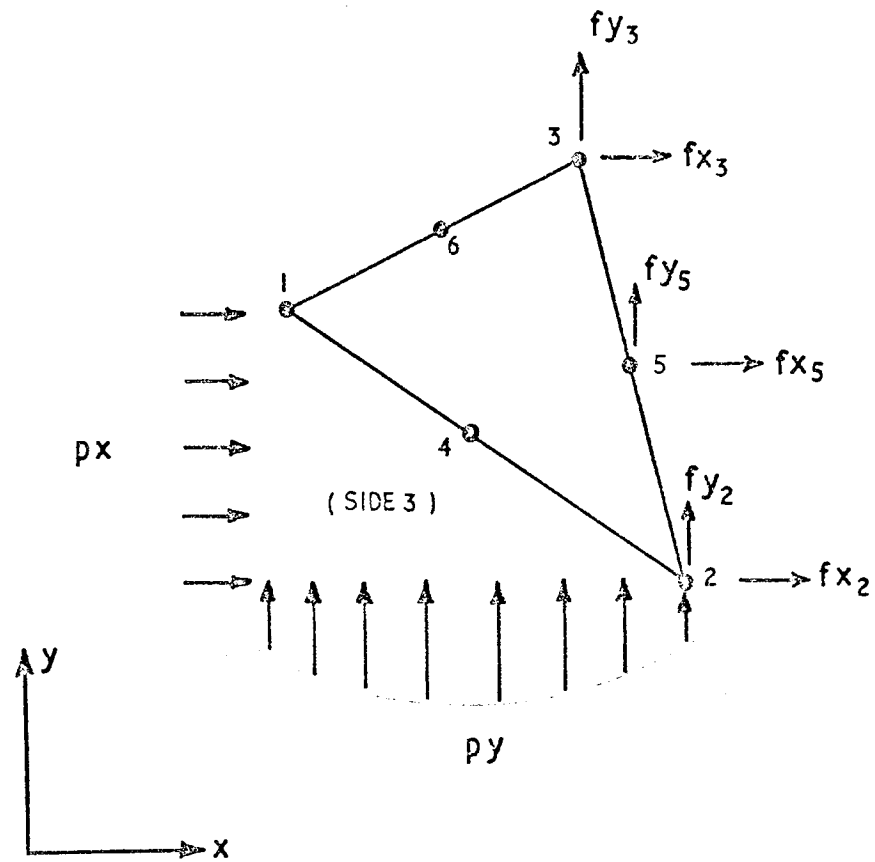


FIG 3.12 IN-PLANE CONCENTRATED AND DISTRIBUTED LOADING ON SIX-NODE TRIANGLE.

The total potential energy of the element is :

$$V = U + \Omega = W_i - W_e$$

where U is the strain energy, calculated from the strain field associated with the displacement field,

Ω is the potential energy of the prescribed loads, and

W_e, W_i are the external and internal work done respectively.

From equations 3.19, 3.48 and 3.52, the total potential energy of the element may be written :

$$\begin{aligned} V_e &= \frac{1}{2} \int_{\substack{\text{Element} \\ \text{Volume}}} \{\epsilon\}^t \{\sigma\} d(\text{vol}) - \int_{\substack{\text{Element} \\ \text{Side}}} \{p\} \{\phi\} dS - \{f\}^t \{\delta\}_e \\ &= \frac{1}{2} \{\delta e\}^t \int_{\text{Vol}} [B]^t [C] [B] d(\text{Vol}) \{\delta\}_e - \int_{\text{length}} \{p\}^t [N] dS \{\delta\}_e - \{f\}^t \{\delta\}_e \\ &= \frac{1}{2} \{\delta\}_e^t [K]_e \{\delta\}_e - \{F_e\}^t \{\delta\}_e \end{aligned} \quad (3.55)$$

where

$$[K]_e = \int_{\text{Vol}} [B]^t [C] [B] d(\text{Vol}) \quad (3.56)$$

is the element stiffness matrix,

$$\{F_e\} = \left(\int_{\text{length}} \{p\}^t [N] dS + \{f\}^t \right) \quad (3.57)$$

is the generalised force vector, statically equivalent to the prescribed surface forces.

For equilibrium, V must be stationary, so that setting the first variation equal to zero, from equation 3.55,

$$dV_e = 0 = \frac{1}{2} (\{d\delta_e\}^t [K_e] \{\delta_e\} + \{\delta_e\}^t [K_e] \{d\delta_e\}) - \{d\delta_e\}^t \{F_e\}$$

or since $[K_e]$ is symmetrical

$$\{d\delta_e\}^t [K_e] \{\delta_e\} = \{\delta_e\}^t [K_e] \{d\delta_e\}$$

and hence

$$\{\delta e\}^t ([K_e] \{\delta e\} - \{F_e\}) = 0 \quad (3.58)$$

Since the variations $\{\delta e\}$ are arbitrary, equation 3.58 yields the usual stiffness equilibrium equations,

$$[K_e] \{\delta e\} = \{F_e\} \quad (3.59)$$

Writing the element volume as $dVol = h dA$, where h is the element thickness, equation 3.56 yields the following expression for the element stiffness matrix,

$$[K_e] = \int_{Area} [B]^t [C][B] h dA \quad (3.60)$$

in the case of the six-node triangle this is a 12×12 symmetric matrix, the coefficients being derived by integrating over the area of the element.

The integration implied in equation 3.60 is to be performed with respect to the global Cartesian x, y axes. Since the matrix $[B]$ is a function of the natural coordinates it is necessary to transform the integral into the natural coordinates. To transform the region with respect to the natural coordinates, a standard process is used which involves the Jacobian, $\det [J]$.

$$dA = dx dy = \det [J] dL_1 dL_2$$

A justification of this statement is given in Appendix A.

With due regard to the limits of integration over the triangle, equation 3.60 becomes,

$$[K_e] = \int_0^1 \int_0^{1-L_2} [B]^t [C][B] h \det [J] dL_1 dL_2 \quad (3.61).$$

In practice the integrations implied in equation 3.61 are extremely cumbersome, if not completely intractable algebraically, and it is normal to resort to numerical integration for their evaluation.

3.5.3 Assembly of Overall Stiffness Matrix and Load Vector

The variational concept of the finite element method was previously applied to obtain the element stiffness relations, equation 3.61. The same variational principle will be re-applied here to establish the assembly rules for the complete structure.

Consider a continuum idealised into E elements interconnected at N nodes. Then with two degrees of freedom per node, the total number of undetermined displacement components is $2N$ and the order of the overall system stiffness matrix will be $2N \times 2N$.

Recalling equation 3.55 in which the total potential energy stored in an element was found in terms of the local numbering system of Fig. 3.7, it is convenient to expand the element stiffnesses and loads as $2N \times 2N$ square matrices $[K_e]$ and $2N \times 1$ vectors $\{F_e\}$ respectively. The procedure consists of a renumbering of the element matrices, changing local numbers to global numbers and locating the coefficients in their appropriate positions in the enlarged $[K_e]$ matrix.

The total potential energy for the element can be written :

$$V_e = \frac{1}{2} \begin{matrix} \{ \delta \}^t & [K_e] & \{ \delta \} & - & \{ \delta \}^t & \{ F_e \} \\ 1 \times 2N & 2N \times 2N & 2N \times 1 & & 1 \times 2N & 2N \times 1 \end{matrix}$$

The total potential energy for the assemblage is obtained by summing contributions from all the elements :

$$\begin{aligned} V &= \sum_{e=1}^E V_e = \frac{1}{2} \sum_{e=1}^E \{ \delta \}^t [K_e] \{ \delta \} - \sum_{e=1}^E \{ \delta \}^t \{ F_e \} \\ &= \frac{1}{2} \{ \delta \}^t \left(\sum_{e=1}^E [K_e] \right) \{ \delta \} - \{ \delta \}^t \left(\sum_{e=1}^E \{ F_e \} \right) \end{aligned}$$

For equilibrium, V must be stationary so that :

$$dV = 0 = \{d \delta\}^t \left(\sum_{e=1}^E [K_e] \right) \{\delta\} - \sum_{e=1}^E \{F_e\} \quad (3.62)$$

Since the variations $\{d \delta\}$ are arbitrary, equation 3.62 yields the overall equilibrium stiffness equations :

$$[K] \{\delta\} = \{F\} \quad (3.63)$$

The assembly rules for the overall stiffness matrix and load vector may then be stated,

$$[K] = \sum_{e=1}^E [K_e] \quad (3.64)$$

$$\{F\} = \sum_{e=1}^E \{F_e\} \quad (3.65)$$

The numerical implementation of equations 3.64 and 3.65 is discussed more fully in the following Chapter.

4. NUMERICAL PROCEDURES

4.1 INTRODUCTION

In Chapter 3 the basic equations necessary for the implementation of the finite element method have been derived, in symbolic form, for two-dimensional problems of plane stress and plane strain. The element employed was the six-node triangle. It is the purpose of this Chapter to develop appropriate numerical procedures for the operations described in the previous chapter in a form suitable for a digital computer. Most computer languages allow for the substructuring of a program, where procedures or subroutines are written separately to perform a given operation and may be called either once or several times later in the main program. This provides a conceptually simple program structure and permits adaptation to other element types within the same program.

In section 4.2 the program steps are presented for calculation of the elasticity matrix, equation 3.52. Section 4.3 deals with the input data, nodal coordinates and element nodal connections. The strain-element nodal displacement matrix, derived in section 3.5.1, and used extensively in the derivation of element stiffness equations and calculation of stresses and strains is programmed in section 4.4.

For a reasonable solution accuracy a large number of elements are required in the mesh subdivision, consequently computers of large storage capacity are required for storage of the overall stiffness array $[K]$. In order to minimise storage, certain properties of this array are utilised. Firstly, the matrix is symmetrical and so only the upper or lower triangle need be generated and stored. Secondly, due to the element numbering, the matrix is banded (coefficients outside the

band being zero) hence only coefficients of either the upper or lower band need be retained. Finally, by examining the nodal connections of each element it is possible to calculate the first non-zero coefficient occurring in any row of the overall stiffness matrix. Using a scheme, developed by Jennings and Tuff⁽²²⁾, the overall stiffness matrix may be stored as a one-dimensional array where only coefficients occurring between the first non-zero term in any row and the diagonal are retained. This method uses an address sequence to reference the positions of the coefficients in the one-dimensional array to their positions in the conventional two-dimensional array. The method represents a considerable improvement over the previous methods discussed and is described in more detail, together with the procedure for generating the address sequence, in section 4.5.

Section 4.6 is concerned with the assembly of the overall stiffness equations. The principles of numerical integration are discussed in detail with reference to the evaluation of the element stiffness properties.

The boundary conditions, namely applied loads and prescribed displacements are discussed in section 4.7.

A procedure for solving the set of linear simultaneous equations where the matrix is stored as a one-dimensional array is presented in section 4.8.

Finally, in section 4.9 the element and nodal point values of stresses and strains are found from the nodal point displacements.

These procedures represent all the basic operations necessary for the finite element program which is discussed, in completed form, in Chapter 5. It should be noted that all the procedures described in this Chapter perform the necessary operations automatically without any interference from the program user.

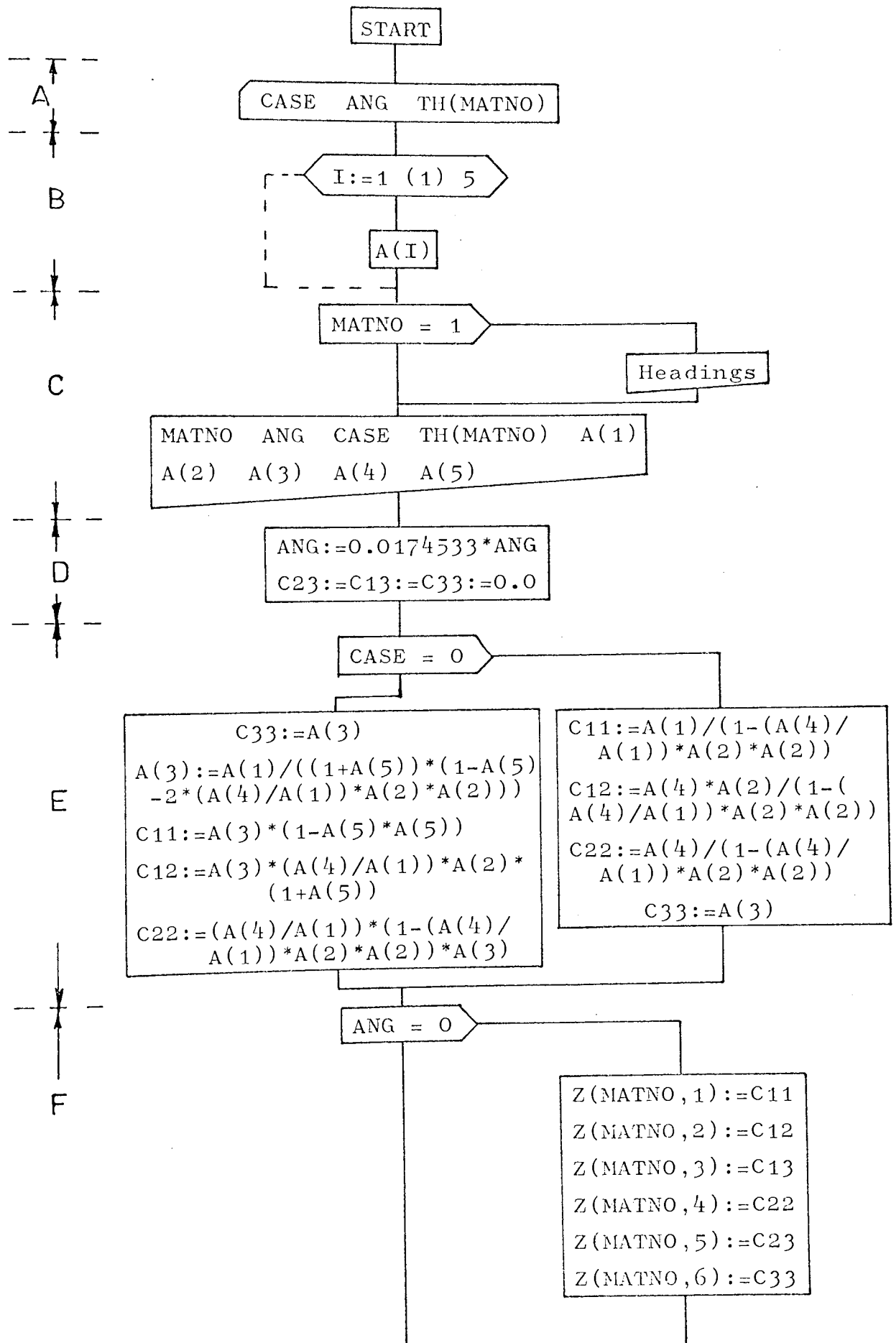
4.2 GENERATION OF ELASTICITY MATRIX

The following numerical procedure will construct stress/strain relations for the material cases, isotropic and transversely isotropic, equations 3.33, 3.34 and 3.36 and 3.37. In the case of isotropic materials there are two independent elastic constants, E and ν , whereas in the transversely isotropic case there are five, E_x , ν_{yx} , G_{yz} , E_{yy} and ν_{zy} . The angle of strata in the transversely isotropic case may be inclined to the global axes.

A flowchart and computer listing for the procedure follow and the various steps involved are, with reference to the flowchart :-

- A. - Case is read, 0 for plane stress and 1 for plane strain.
Angle of strata (degrees) to the global axes is read, θ as in fig. 3.11.
Thickness of element or material is read.
- B. - Elastic constants are read.
- C. - Input data, as read in steps A and B is output to the line printer.
- D. - Angle of strata is converted to radians.
- E. - For either plane stress or plane strain, coefficients of the stress/strain matrix is calculated according to equations 3.36 and 3.37. Note that equations 3.33 and 3.34 are identical to equations 3.36 and 3.37 with $E_{xx} = E_{yy} = E$ and $\nu_{zy} = \nu_{yx} = \nu$.
- F. - If the angle of the strata does not coincide with the global axes then coefficients of step E are modified according to equation 3.42.

PROCEDURE : FECMATRIX(Z,CASE,A,MATNO,TH,ANG)



S:=SIN(ANG) C:=COS(ANG)

S2:=S*S C2:=C*C

S3:=S2*S C3:=C2*C

S4:=S2*S2 C4:=C2*C2

Z(MATNO,1):=C11*C4+2*(C12+2*C33)*C2*S2-4*(C13*C2+C23*S2)*C*S+C22*S4

Z(MATNO,2):=(S4+C4)*C12+(C11+C22-4*C33)*C2*S2+2*(C13-C23)*C3*S+2*(C23-C13)*C*S3

Z(MATNO,3):=(C11-C12-2*C33)*C3*S+(C12-C22+2*C33)*C*S3+3*(C23-C13)*S2*C2+C13*C4-C23*S4

Z(MATNO,4):=C11*S4+C22*C4+4*C13*C*S3+4*C23*C3*S+2*(C12+2*C33)*C2*S2

Z(MATNO,5):=(C11-C12-2*C33)*S3*C+(C12-C22+2*C33)*C3*S+3*(C13-C23)*C2*S2+C23*C4-C13*S4

Z(MATNO,6):=(C11+C22-2*C12-2*C33)*S2*C2+2*(C13-C23)*C3*S+2*(C23-C13)*S3*C+C33*(S4+C4)

END

F

```

PROCEDURE FICMATRIX(Z,CASE,A,MATNO,TH,ANG);
VALUE MATNO; INTEGER MATNO,CASE; REAL ANG; ARRAY A,7,TH;
BEGIN
  INTEGER I; REAL S,C,S2,C2,S3,C3,S4,C4,C11,C12,C22,C23,C33,C13;
  COMMENT 'CALCULATION OF ELASTIC CONSTANTS WHERE SEVERAL MATERIALS
  ARE PRESENT EACH WITH VARYING ANGLES OF ANISOTROPY;
CASE:=READ; ANG:=READ; TH[MATNO]:=READ;
FOR I:=1 STEP 1 UNTIL 5 DO A[I]:=READ;
IF MATNO=1 THEN
  BEGIN
    WRITE TEXT('('('C')' MATERIAL PROPERTIES,=('('C')'(NOTE,IF%CASE=
0,THEN%PLANE%STRESS%ELSE%('CAS')'THE%PROBLEM%IS%OF%PLANE%STRAIN.)
('2C')' MATERIAL%ANGLE%OF%CASE%THICKNESS('21S')' PLASTIC
%PROPERTIES('CS')' NUMBER%ANISOTROPY('30S')' EXX('11S')' EYY
('9S')' VXX('7S')' VYY('9S')' GXY'))';
  END;
  ARLINE(2); SPACE(1);
PRINT(MATNO,2,0); SPACE(7); PRINT(ANG,0,2); SPACE(7); PRINT(CASE,1,0);
SPACE(3); PRINT(TH[MATNO],0,4); SPACE(2); PRINT(A[1],0,3); SPACE(2);
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);
ANG:=0.0174533*ANG; C23:=C33:=0.0;
IF CASE=0 THEN
  BEGIN
    C11:=A[1]/(1-(A[4]/A[1])+A[2]*A[2]);
    C12:=A[4]*A[2]/(1-(A[4]/A[1])+A[2]*A[2]);
    C22:=A[4]/(1-(A[4]/A[1])*A[2]*A[2]);
    C33:=A[3];
  END;
  ELSE
  BEGIN
    C33:=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:=A[3]*(A[4]/A[1]+A[2]*(1+A[5]));
    C22:=(A[4]/A[1])*(1-(A[4]/A[1])*A[2]*A[2])*A[3];
  END;
  IF ANG NE 0.0 THEN
  BEGIN
    S:=SIN(ANG); C:=COS(ANG); S2:=S*S; C2:=C*C; S3:=S2*S; C3:=C2*C;
    S4:=S2*S2; C4:=C2*C2;
    Z[MATNO,1]:=C11*C4+2*(C12+2*C33)*C2*S2-4*(C13*C2+C23*S2)*C*S+C22*S4;
    Z[MATNO,2]:=(S4+C4)*C12+(C11+C22-4*C33)*C2*S2+2*(C13-C23)*C3*S+2*(C23
-C13)*C*S3;
    Z[MATNO,3]:=(C11-C12-2*C33)*C3*S+(C12-C22+2*C33)*C*S3+3*(C23-C13)*S2*C2
+C13*C4-C23*S4;
    Z[MATNO,4]:=C11*S4+C22*C4+4*C13*C*S3+4*C23*C3*S+2*(C12+2*C33)*C2*S2;
    Z[MATNO,5]:=(C11-C12-2*C33)*S3*C+(C12-C22+2*C33)*C3*S+S*(C13-C23)*C2*S2
+C23*C4-C13*S4;
    Z[MATNO,6]:=(C11+C22-2*C12-2*C33)*S2*C2+2*(C13-C23)*C3*S+2*(C23-C13)*S3*
C+C33*(S4+C4);
  END;
  ELSE
  BEGIN
    Z[MATNO,1]:=C11; Z[MATNO,2]:=C12;
    Z[MATNO,3]:=C13; Z[MATNO,4]:=C22;
    Z[MATNO,5]:=C23; Z[MATNO,6]:=C33;
  END;
END OF PROCEDURE CMATRIX;

```

4.3 FORMAT FOR INPUT DATA

The objective of this procedure is to read and print the input data necessary to describe the mesh topology. Such data includes the coordinates of the nodes and the element nodal connections.

If the element boundaries are curved then the user must specify the coordinates of all nodes on such boundaries. In many cases, however, the element sides are straight and rather than have the user specifying the coordinates of midside nodes, the computer will automatically interpolate for missing midside nodes from the element corner nodes. In order that the computer can identify midside nodes where interpolation is required, the x-coordinates of all nodes are set, by the computer to 0.00001 before any data is read. This value is overwritten for nodes whose coordinates are specified and by checking x-coordinates after reading the input data the computer can identify those midside nodes at which interpolation is required.

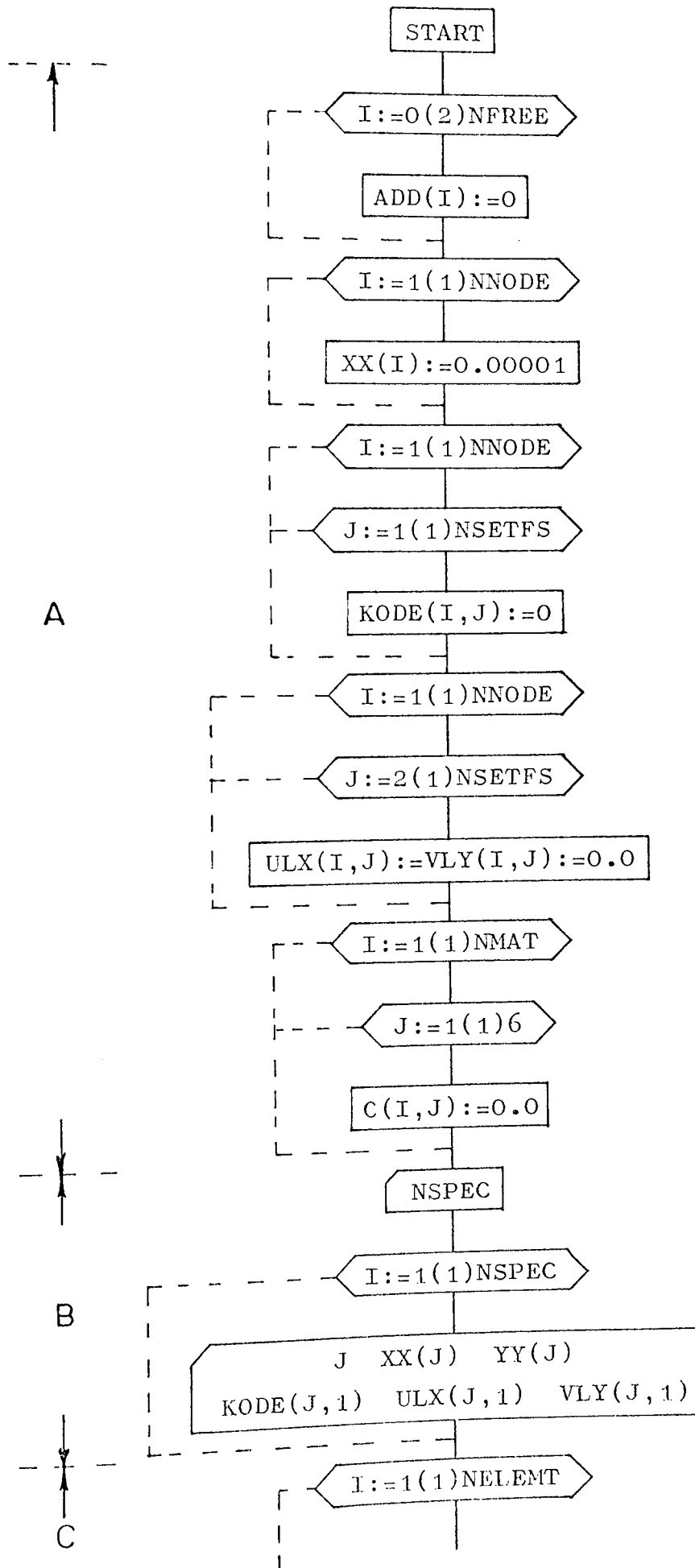
A flowchart and program listing follow this text and the various steps involved are described, with reference to the flow chart as :

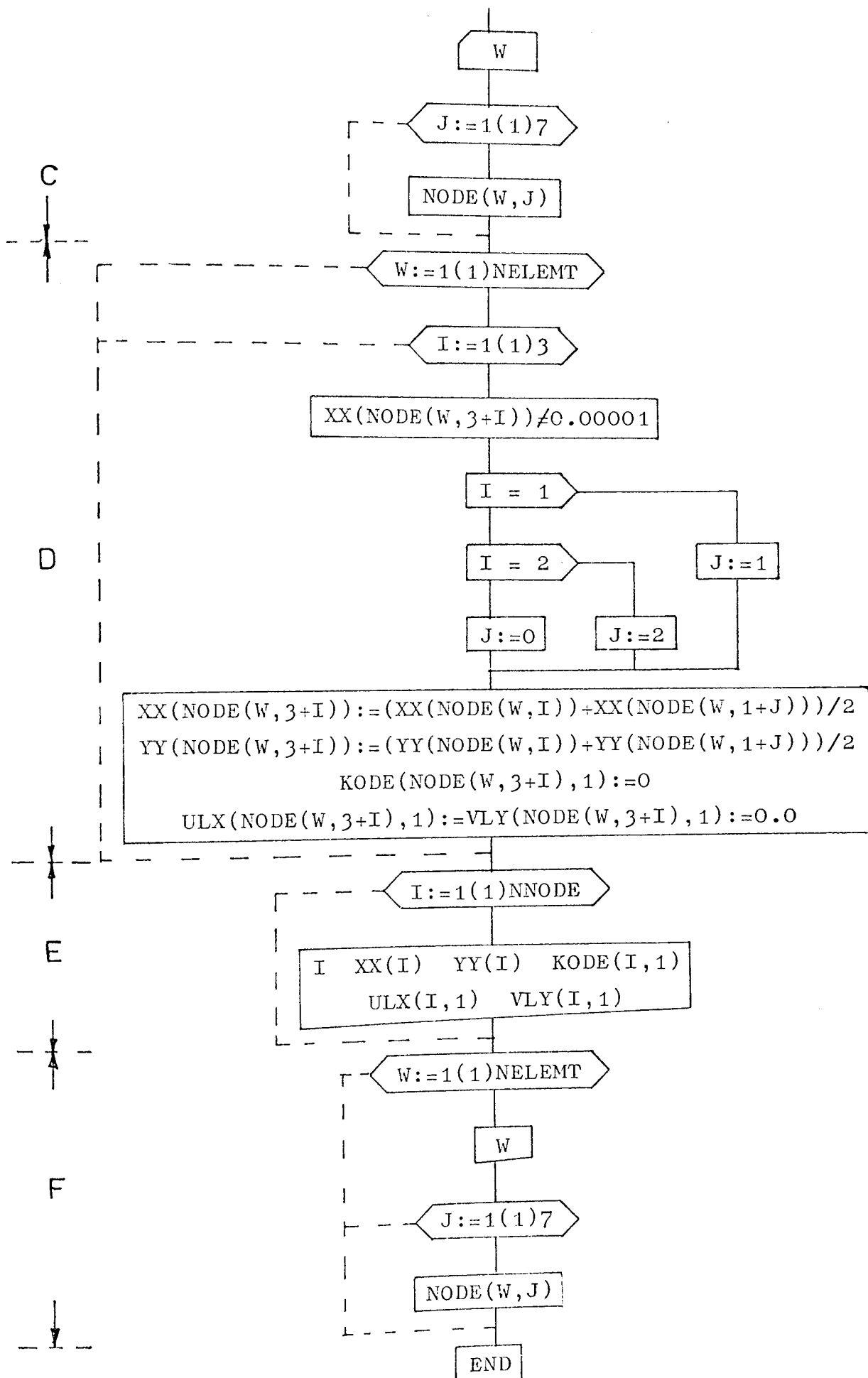
- A. - Various arrays used later in the program are initialised and all x-coordinates are set at 0.00001.
- B. - Node numbers and global X and Y coordinates are specified for corner nodes and such midside nodes as necessary. Where a coordinate is specified it overwrites the X value set in step A.
- C. - Element nodal connections and material number for each element are read. Each element may be of different material and those of the same material share the same material number, e.g. elements 1, 2, 3 material number 1, elements 4, 5, 6 material number 2, etc.
- D. - For midside nodes where the X-coordinate still equals 0.00001, then interpolation begins to calculate the coordinates of such nodes.

99
E. - Nodal coordinates are output.

F. - Element nodal connections are output.

PROCEDURE : FEINPUT(C,ADD,XX,YY,NFREE,NNODE,NSETF,KODE,NSPEC,
NSETFS,ULX,VLY,NELEMT,NODE,NMAT)





```

      'PROCEDURE' FFIINPUT(C,ADD,XX,YY,NERFF,NMODE,NSSETF,KODE,NSPEC,NSSETS,
      ULX,VLY,NELFNT,NODE,NMAT):
'VALUE' NERFF,NMODE,NSSETF,NSPEC,NELFNT,NSSETS,NMAT:
'INTEGER' NERFF,NMODE,NSSETF,NSPEC,NELFNT,NSSETS,NMAT:
'INTEGER' 'ARRAY' ADD,KODE,NODE: 'REAL' 'ARRAY' XX,YY,ULX,VLY,C:
'BEGIN' 'INTEGER' I,J,W:
'FOR' I:=0 'STEP' 2 'UNTIL' NERFF 'DO' ADD[I]:=0:
'FOR' I:=1 'STEP' 1 'UNTIL' NMODE 'DO' XX[I]:=0.00001:
'FOR' J:=1 'STEP' 1 'UNTIL' NMODE 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSSETS 'DO' KODE[I,J]:=0:
'FOR' I:=1 'STEP' 1 'UNTIL' NMODE 'DO'
'FOR' J:=2 'STEP' 1 'UNTIL' NSSETS 'DO'
ULX[I,J]:=VLY[I,J]:=0.0:
'FOR' I:=1 'STEP' 1 'UNTIL' NMAT 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO' C[I,J]:=0.0:
NSPEC:=READ:
'FOR' I:=1 'STEP' 1 'UNTIL' NSPEC 'DO'
'BEGIN'
  J:=READ:  XX[J]:=READ:  YY[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' I:=1 'STEP' 1 'UNTIL' NELFNT 'DO'
'BEGIN'
W:=READ:
'FOR' J:=1 'STEP' 1 'UNTIL' 7 'DO' NODE[W,J]:=READ:
'END':
WRITETEXT('(('('2C')'NODAL%CONNECTIONS%HAVE%BEEN%READ'))');
'FOR' W:=1 'STEP' 1 'UNTIL' NELFNT '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:
KODE[NODE[W,3+I],1]:=0:
ULX[NODE[W,3+I],1]:=VLY[NODE[W,3+I],1]:=0.0:
'END':
'END':
WRITETEXT('(('('4C')'NODAL%POINT%DATA'('2C4S')'NODE'('5S')'X%COORD
('5S')'Y%COORD'('5S')'TYPE'('7S')'X-DISP'('6S')'Y-DISP'('648S')'
OR%LOAD'('5S')'OR%LOAD'))');
'FOR' I:=1 'STEP' 1 'UNTIL' NMODE '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')'ELEMENT%DATA'('2C')'ELEMENT'('18S'))');
NODAL%CONNECTIONS'('17S')'MATERIAL'))');
'FOR' J:=1 'STEP' 1 'UNTIL' NELFNT '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 FFIINPUT:

```

4.4 GENERATION OF STRAIN-ELEMENT NODAL DISPLACEMENT MATRIX

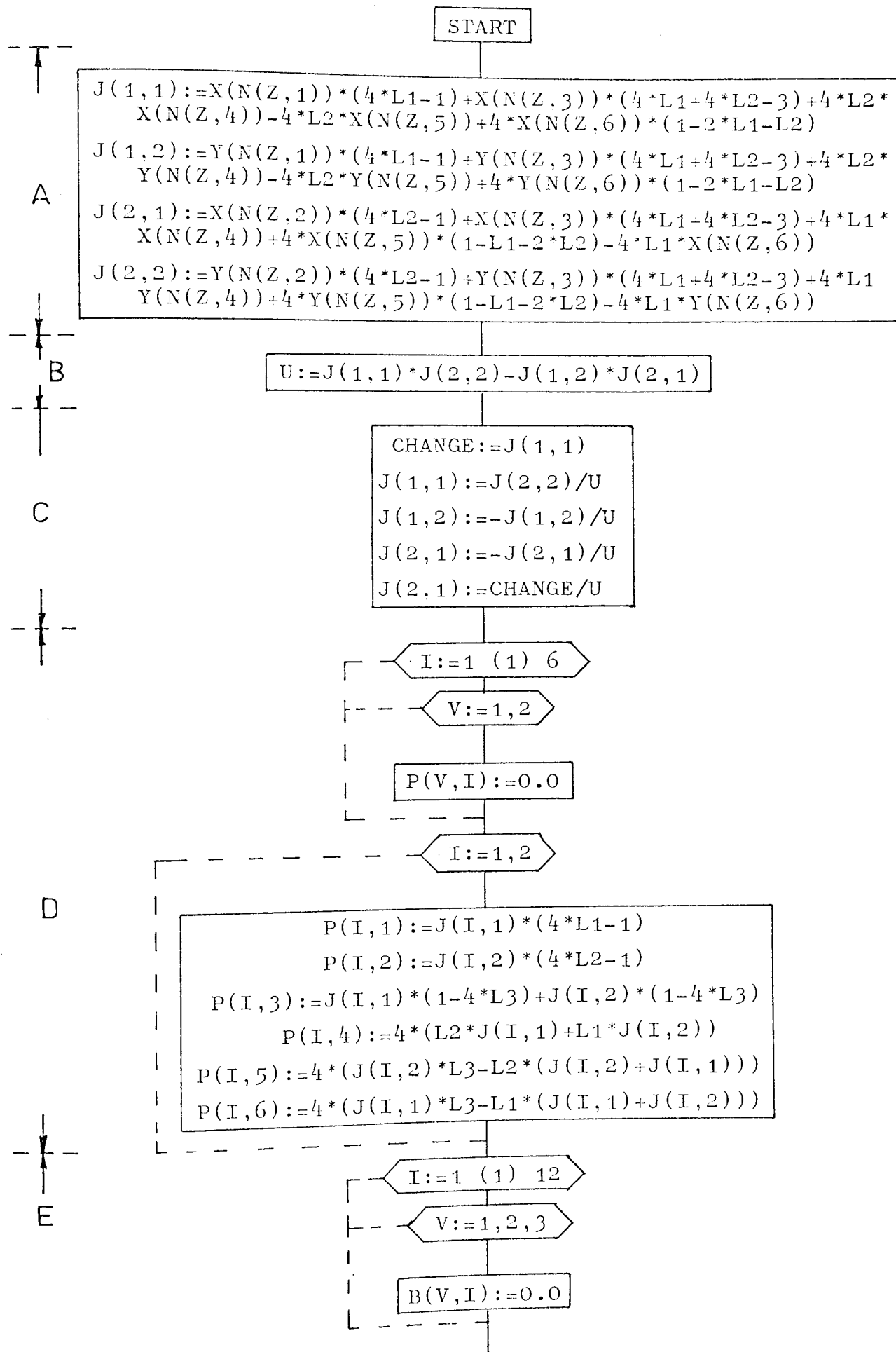
In order to formulate the element stiffness matrices, the values of the coefficients of the strain-element nodal displacement matrix, $[B]$ and the value of the Jacobian, $/J/$, are required at various points within each element. The number and coordinates of such points are fixed by the integration formula chosen to evaluate the element stiffness coefficients and will be described in detail in Section 4.6.

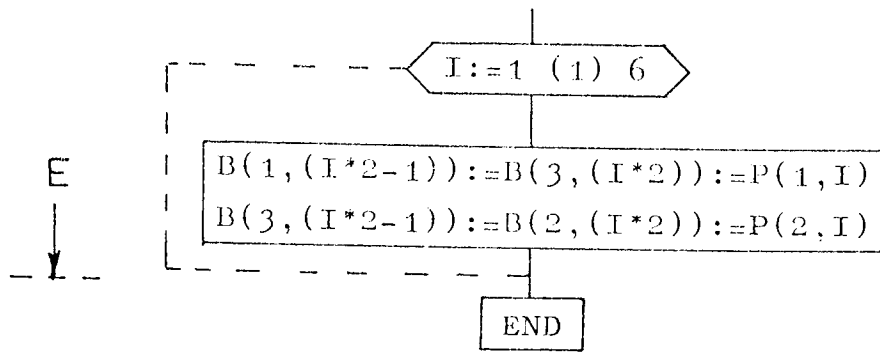
For simplicity of the final program structure a procedure AUX has been prepared specifically to evaluate both $/J/$ and the coefficients of $[B]$ at certain points within an element. At input to the procedure, the natural coordinates of the point of interest are specified by the computer, together with details of the geometry of the element concerned, i.e. nodal coordinates and element connections. Upon exit from the procedure, the values of the coefficients of $[B]$ have been calculated together with the value of the Jacobian $/J/$.

A flowchart and program listing for this procedure follows and the steps involved are, with reference to the flowchart,

- A. - Coefficients of $[J]$ are evaluated according to equation 3.49 (a); using equations 3.44 and 3.47 with u and v replaced by x and y respectively.
- B. - $/J/$ is calculated from equation 3.50 (a).
- C. - $[J]$ is inverted to give $[J]^{-1}$, equation 3.50.
- D. - Coefficients of $[B]$ are formed in procedure array $[P]$ as in equation 3.48 (a).
- E. - Coefficients from array $[P]$ are located in their correct positions in strain-displacement array $[B]$. Step D is necessary purely for ease of programming.

PROCEDURE : AUX(L1,L2,L3,B,X,Y,U,N,Z)





```

'PROCEDURE' AUX(L1,L2,L3,B,X,Y,U,M,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];
'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,5]]*(4*L1+4*L2-5)+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,5]]*(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]]*(1-1-2*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]]*(1-1-2*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;
'COMMENT' THE P ARRAY REPS THE DIFF COEFFS OF N[I] 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[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,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 'DO' B[V,I]:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN'
B[1,(I*2-1)]:=B[3,(I*2)]:=P[1,I];
B[3,(I*2-1)]:=B[2,(I*2)]:=P[2,I];
'END';
'END' OF AUX;

```

4.5 STORAGE OF THE OVERALL STIFFNESS MATRIX

The method of storing the overall stiffness matrix as a one-dimensional array, with a suitable address sequence linking the two, has been introduced in section 4.1. This technique will be elaborated in this section and a procedure will be presented for the derivation of the address sequence from the element nodal connections.

As an example of the storage scheme, consider the matrix $[K]$.

Row	Column	1	2	3	4	5	6
1		1.5					
2		0.2	1.2				
3		-1.1	0	2.2			
4		0	0	5.1	10.6		
5		0	0	0	0	2.6	
6		0	0	-1.2	0	0	6.1

which would be stored as,

Location	1	2	3	4	5	6	7	8	9	10	11	12	13
K	1.5	0.2	1.2	-1.1	0	2.2	5.1	10.6	2.6	-1.2	0	0	6.1

The address sequence becomes ,

Row No.	0	1	2	3	4	5	6
ADD	0	1	3	6	8	9	13

where the row numbers refer to rows of the 2 x 2 matrix $[K.]$

and the coefficients of the ADD array locate the position of the diagonal elements within the one-dimensional $[K]$ array.

Element $K(i,j)$ in the two-dimensional array would be accessed in the equivalent one-dimensional array by the call,

$$K(i,j) = K[ADD[i] - i + j] \quad (4.1)$$

for example,

$$K(4, 3) = K[ADD[4] - 4 + 3] = K(7) = 5.1$$

It is essential that the coefficients of the address array be determined before assembly of the overall stiffness matrix commences in order that stiffness coefficients can be located directly in their appropriate positions in the one-dimensional stiffness array according to equation 4.1. The address array may be determined from the element nodal numbering and the following scheme for its determination represents the most economical in terms of computer time, of the several different approaches attempted.

Consider the example of Fig. 4.1, where a triangular plate is divided into four elements and 15 nodes as shown. The form of the overall stiffness matrix is also shown in Fig. 4.1 where the positions of the first non-zero coefficient in any row is delineated by a dotted line. Positions of these non-zero coefficients can be identified from the assembly rules of equation 3.64 where nodes connected through a common element have a connection in the overall stiffness matrix, thus node 6 is connected to node 1 through element number 1 whereas node 7 is not connected to node 1 across any common element and hence $K [7, 1] = K [1, 7] = 0$.

Note that even in this trivial example there is a 20% reduction in storage required by the present row by row scheme as compared with the banded scheme. The bandwidth is shown by the chain-dotted line, Fig. 4.1.

The procedure is divided into basic steps :-

1. The number of columns occurring between the first non-zero coefficient and the leading diagonal term is found for each row of $[K]$. In the row corresponding to Node 10 of Fig. 4.1 for example, there are five columns between the first non-zero coefficient and the leading diagonal. The number of columns in any row is calculated by examining the difference in node numbers across each element. In Fig. 4.1, Node 10 appears in element number 4 where the lowest node number = 6, and so

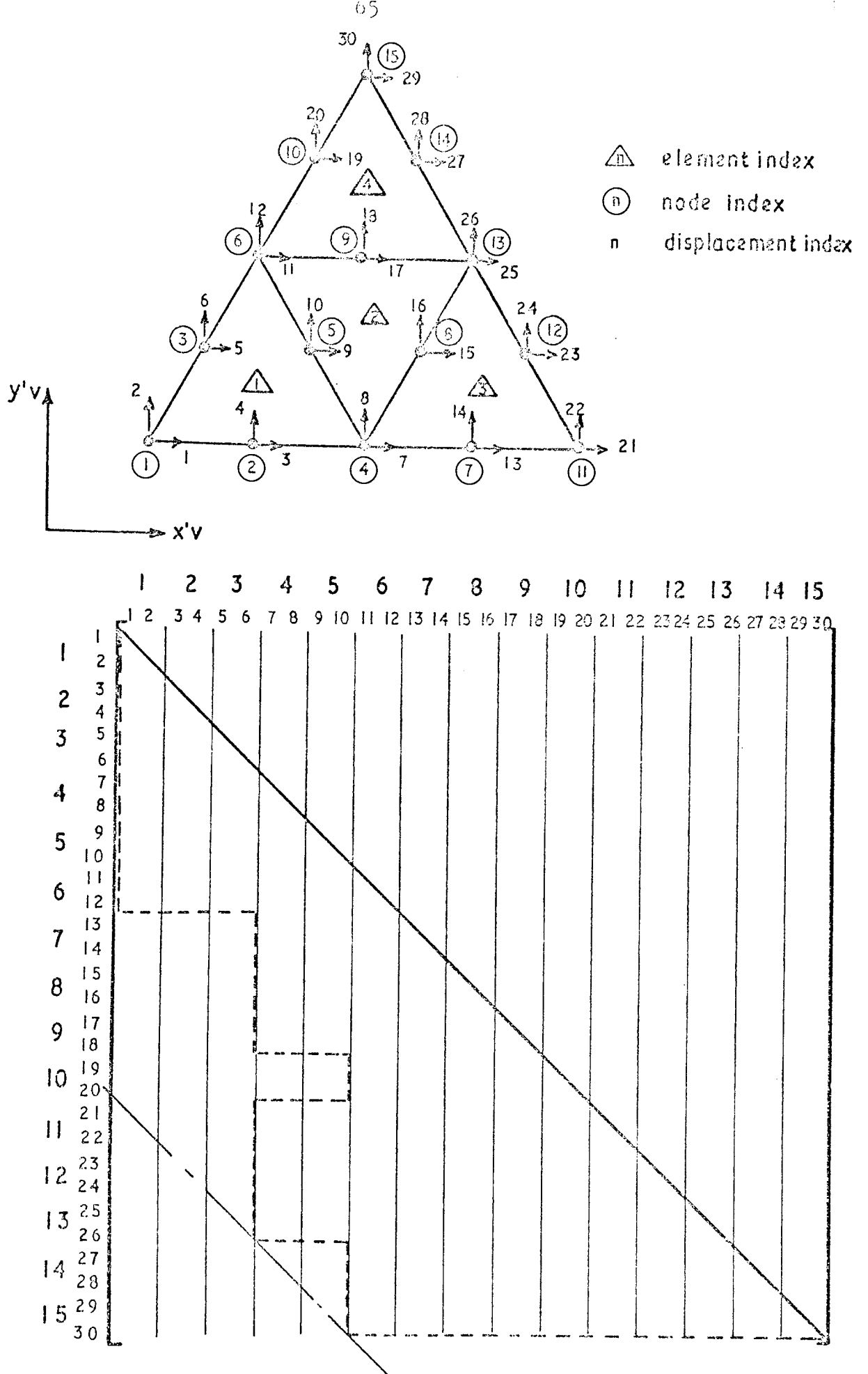


FIG.4.1 SCHEMATIC REPRESENTATION OF OVERALL STIFFNESS MATRIX.

the number of columns appearing in row 10 may be determined as $10 - 6 + 1 = 5$. This process is repeated for all elements appearing in the mesh.

2. The complete address array is now constructed using the information obtained from step 1. This is accomplished by summing the number of columns in each row, continuously from the first degree of freedom to the last, in order to find the number of coefficients occurring in $[K]$ up to and including the leading diagonal coefficient in each row.

A flowchart and program listing for the procedure ADDARRAY follows. With the coefficients of the address array ADD having been initialised elsewhere in the program, the various steps are, with reference to the flowchart :

A. - for the first element,

- (i) Integer CH is set to the first node number from nodal connection array $NODE$.
- (ii) All node numbers in this element are scanned to find and set CH to the lowest node number.
- (iii) Number of columns present in the overall stiffness matrix for each node (row) occurring in this element is calculated (see Step 1 above). If this number exceeds the value already present in the array ADD for the row under consideration then it is overwritten in order that the maximum number of columns is obtained.

Steps A(i) to A(iii) are repeated for all elements.

B. - Address array coefficients are determined from the number of columns in each row (see Step 2 above).

4.6 ASSEMBLY OF STIFFNESS MATRICES

In this section a procedure is described which forms the individual element stiffness matrices and assembles them in the overall one-dimensional stiffness array. The integrations necessary to obtain element stiffnesses are carried out numerically and are discussed separately.

4.6.1 Numerical Integration

It is required to calculate the values of integrals of the form :

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

In all the available methods for numerical integration (quadrature) the integral is replaced by the sum :

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

where

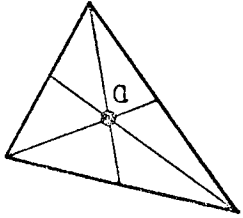
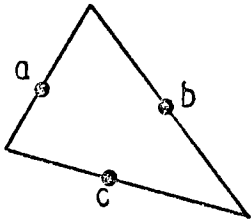
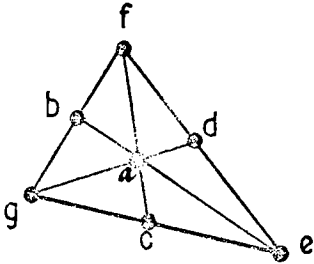
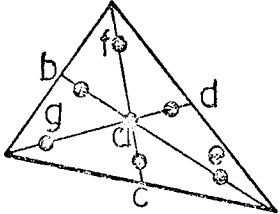
n is the number of points at which the function is to be evaluated

such points are referred to as integrating points.

w_i are the weight coefficients

and $f(L_{1i}, L_{2i}, L_{3i})$ is the value of the function evaluated at the integrating point L_{1i}, L_{2i}, L_{3i} .

The implementation of the procedure for determining the value of the integral I is now self-evident, once the coordinates and weighting coefficients have been determined. There are several quadrature formulas used in finite element analysis, details of which are given in Ref. 9. The one used in this analysis is known as Gauss-Legendre Quadrature and Table 4.1 gives details of the weighting coefficients and triangular coordinates.

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	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$ $\frac{1}{2}, \frac{1}{2}, 0$ $0, \frac{1}{2}, \frac{1}{2}$ $\frac{1}{2}, 0, \frac{1}{2}$ $1, 0, 0$ $0, 1, 0$ $0, 0, 1$	$\frac{27}{120}$ $\frac{8}{120}$ $\frac{3}{120}$
QUINTIC		$R=O(h^6)$	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

WITH $\alpha_1 = 0.05971587$
 $\beta_1 = 0.47014206$
 $\alpha_2 = 0.79742699$
 $\beta_2 = 0.10128651$

TABLE 4.1 NUMERICAL INTEGRATION FORMULAE FOR TRIANGLES (REF.9.)

Having established the integration formula the question arises as to what order of integration should be chosen for the analyses? From considerations of computer time and accuracy, an 'adequate minimum number' of integrating points may be determined.

This problem has been discussed in some detail elsewhere⁽⁹⁾. A detailed study of convergence and computing times was beyond the scope of this project and apart from a few simple trials described in Chapter 5 to ensure accuracy, the author was guided by the results of others^(9,15). In the light of their experience the three-point Gauss rule^{*} was chosen for performing the integrations and is used exclusively in all programs henceforth.

4.6.2 Element Stiffness Matrix

The element stiffness matrix was formulated in equation 3.61 as :

$$[K_e] = h \int_0^1 \int_0^{1-L_2} [B]^t [C][B] / J \, dL_1 \, dL_2$$

with the integration formula of section 4.6.1 the above equation becomes :

$$[K_e] = h \sum_{i=1}^n w_i ([B]^t [C][B] / J)_{L_{1i}, L_{2i}, L_{3i}} \quad (4.2)$$

where n is the number of integrating points

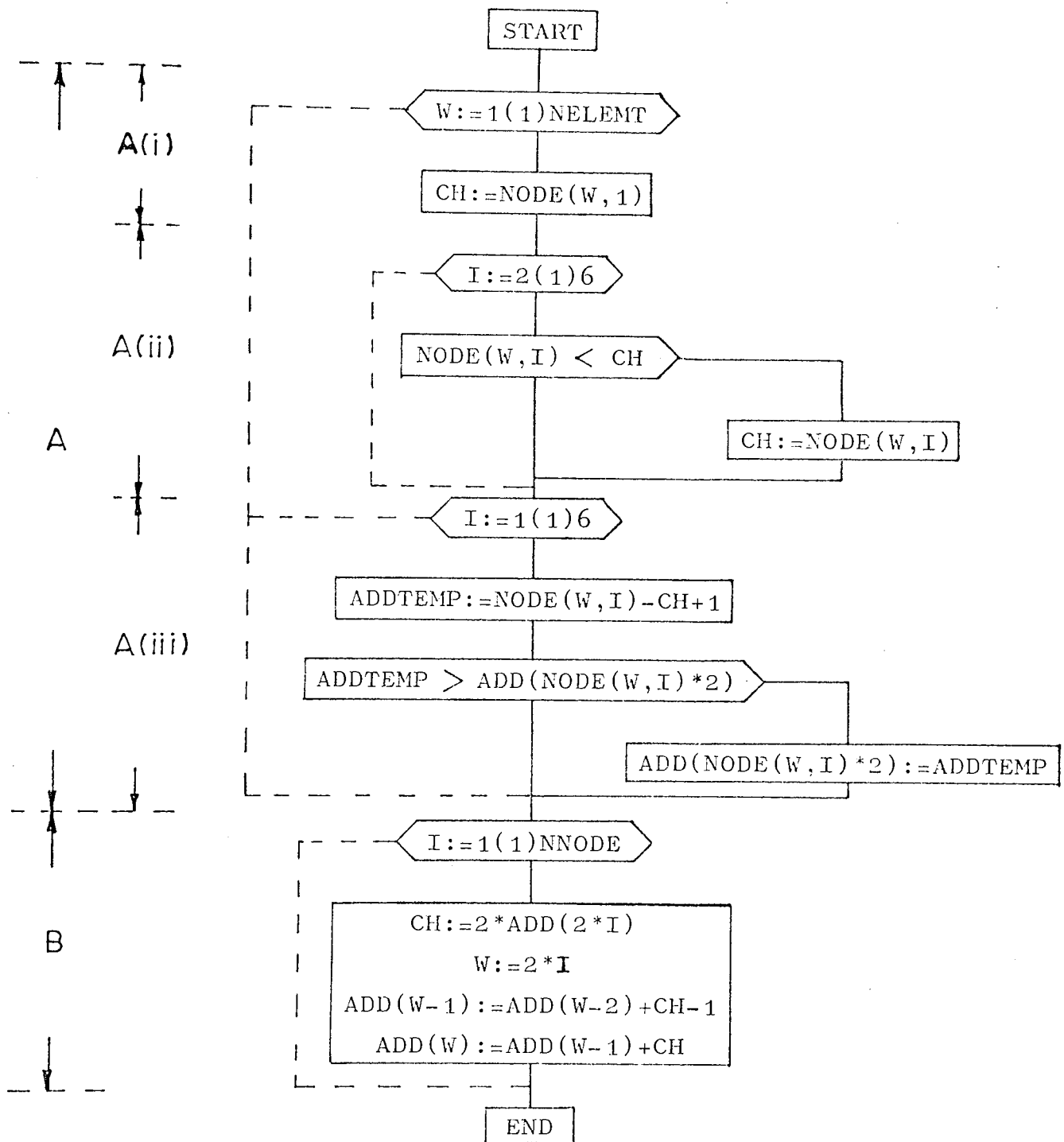
w_i are the weight coefficients

and $([B]^t [C][B] / J)_{L_{1i}, L_{2i}, L_{3i}}$ is understood to mean the evaluation of $([B]^t [C][B] / J)$ at integrating point i .

The numerical implementation of equation 4.2 is discussed at the end of the section where a program listing is presented.

** Quadratic in Table 4.1*

PROCEDURE : ADDARRAY (NELEMT, NNODE, ADD, NODE)



```

'PROCEDURE' ADDARRAY(NELEMENT,NNODE,ADD,NODE);
'INTEGER' NELEMENT,NNODE;
'INTEGER' 'ARRAY' NODE,ADD;
'BEGIN' 'INTEGER' W,CH,I,ADDTMP;
'FOR' W:=1 'STEP' 1 'UNTIL' NELEMENT 'DO'
'BEGIN' CH:=NODE[W,1];
'FOR' I:=2 'STEP' 1 'UNTIL' 6 'DO'
'IF' NODE[W,I] < CH 'THEN' CH:=NODE[W,I];
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN' ADDTMP:=NODE[W,I]-CH+1;
'IF' ADDTMP > 6 '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;

```


4.6.3 Overall Stiffness Matrix

The numerical procedure for constructing the overall stiffness matrix will be illustrated using the example of Fig. 4.2. The element stiffness equations for element number 1 are shown where the element stiffness coefficients k_{ij} are 2 x 2 sub-matrices referring to a particular node. Thus :

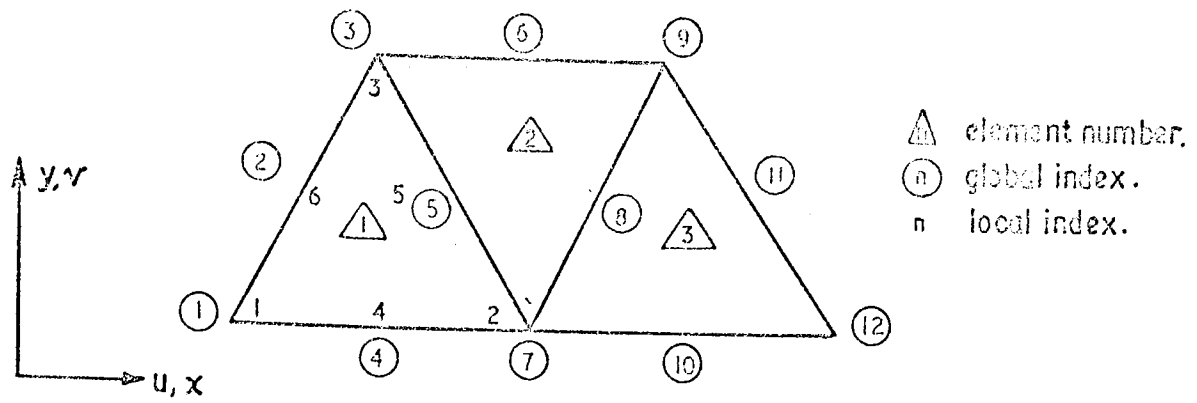
$$\delta_{e1} = \begin{Bmatrix} \delta_{1x} \\ \delta_{1y} \end{Bmatrix} \quad \text{and} \quad F_{e1} = \begin{Bmatrix} F_{1x} \\ F_{1y} \end{Bmatrix}$$

with δ_{1x} and δ_{1y} representing displacements of node number 1 (local numbering) in the x and y directions respectively. Similarly, F_{1x} and F_{1y} represent applied loads at node number 1.

The overall stiffness matrix $[K]$ and load vector $\{F\}$ are obtained, using equations 3.64 and 3.65, by considering each element in turn, converting the element equation numbering from local to global numbering as in Fig. 4.2 and locating the coefficients in their correct positions in the overall matrix as shown. The process is repeated for all the elements, summing contributions from each one into the overall system equations as required.

With the address sequence for the problem it is then a simple matter to relate element local numbering to the global node numbers and position the element stiffness coefficients in the overall one-dimensional array. The subsequent procedure is constructed such that an element stiffness matrix is formed and its coefficients are inserted directly into the overall one-dimensional stiffness array.

NOTE : By using the address array together with equation 4.1, element stiffness coefficients are located straight into the one-dimensional array without intermediate storing in an overall stiffness matrix of the form shown in Fig. 4.2.



		global	1	7	3	4	5	2				
		local	1	2	3	4	5	6				
1	1		k_{11}	k_{21}	k_{31}	k_{41}	k_{51}	k_{61}	$\left\{ \begin{matrix} \delta_{e1} \\ \delta_{e2} \\ \delta_{e3} \\ \delta_{e4} \\ \delta_{e5} \\ \delta_{e6} \end{matrix} \right\} = \left\{ \begin{matrix} F_{e1} \\ F_{e2} \\ F_{e3} \\ F_{e4} \\ F_{e5} \\ F_{e6} \end{matrix} \right\}$	$\left\{ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} \right\}$	$\left\{ \begin{matrix} 1 \\ 7 \\ 3 \\ 4 \\ 5 \\ 2 \end{matrix} \right\}$	
7	2	k_{21}	k_{22}	k_{32}	k_{42}	k_{52}	k_{62}					
3	3	k_{31}	k_{32}	k_{33}	k_{43}	k_{53}	k_{63}					
4	4	k_{41}	k_{42}	k_{43}	k_{44}	k_{54}	k_{64}					
5	5	k_{51}	k_{52}	k_{53}	k_{54}	k_{55}	k_{65}					
2	6	k_{61}	k_{62}	k_{63}	k_{64}	k_{65}	k_{66}					

global local

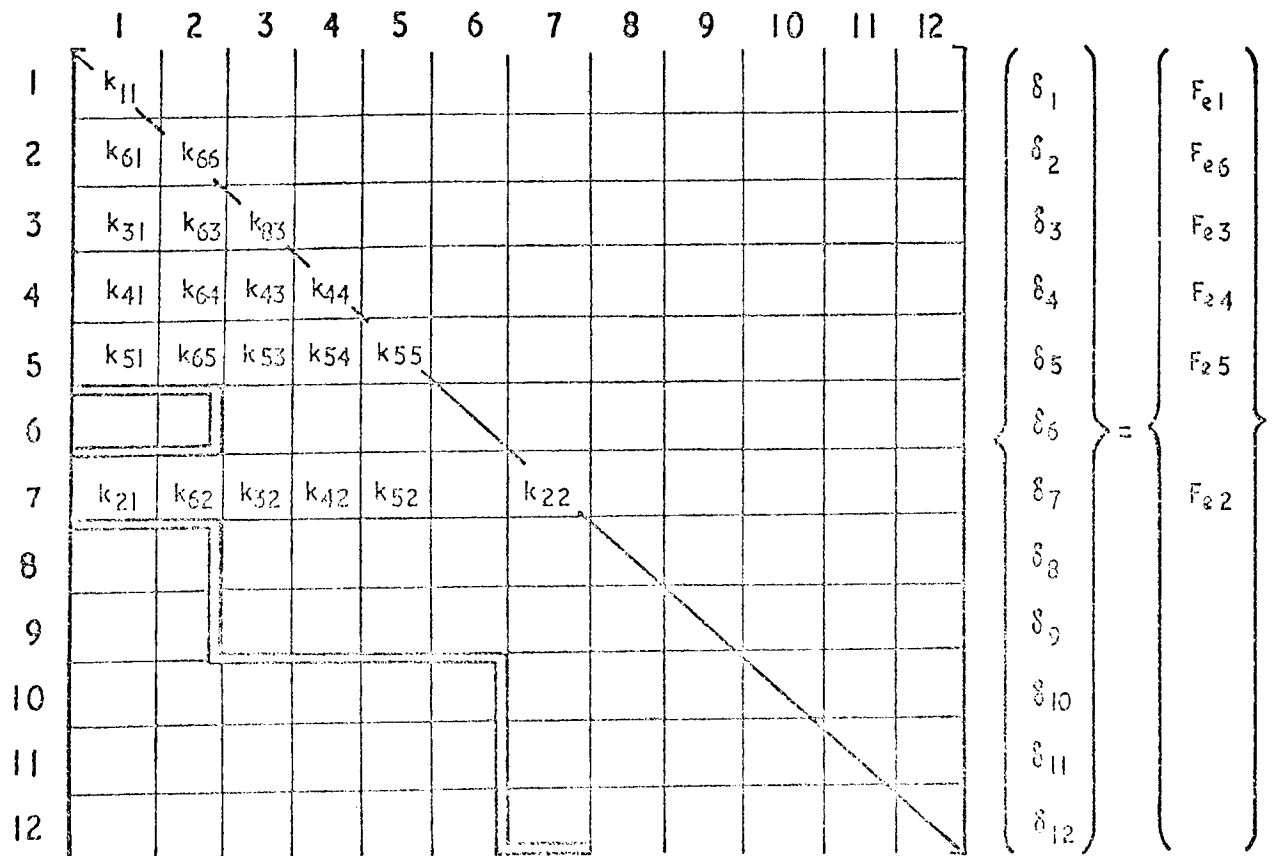


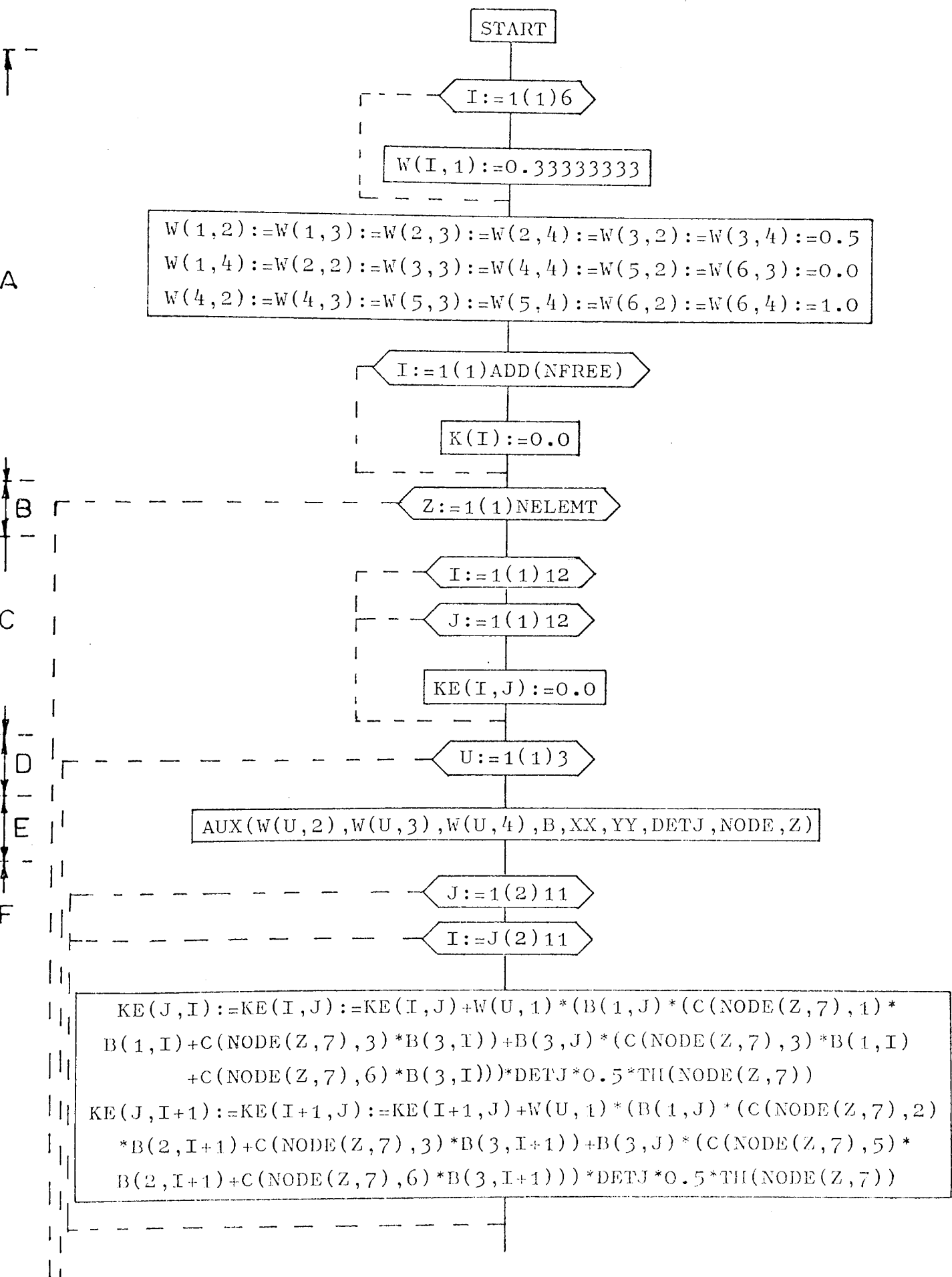
FIG 4.2 ASSEMBLY OF OVERALL STIFFNESS MATRIX .

Fig. 4.2 also shows the method for assembling the overall load vector from the individual load vectors. In the final application of the program however, loads are applied globally to each node without first sharing each between all elements contributing to a given node. This is discussed in more detail in subsequent sections.

The following procedure flowchart and program listing constructs each element stiffness matrix and assembles the overall stiffness matrix, the steps are described with reference to the flowchart as :

- A. - Coefficients of array W are set to weighting coefficients and coordinates of the integrating points.
- B. - Loop is constructed round the number of elements in the mesh, this loop encompasses all the remaining operations in the procedure.
- C. - Coefficients of the element stiffness array KE are initialised.
- D. - Loop on the number of integrating points is constructed; in this case a three-point rule is employed.
- E. - Procedure AUX is called to evaluate the coefficients of array [B] and $/J/$ at the first integrating point.
- F. - Matrix multiplications indicated by equation 4.2 are performed using the coefficients of the elasticity matrix previously found.
- G. - Element stiffness coefficients are located in their appropriate positions in the overall stiffness matrix (as a one-dimensional array) using the nodal connection matrix $NODE$, the address array ADD and equation 4.1.

PROCEDURE : FEASSEMBLY(NELEMT,K,XX,YY,DETJ,NODE,C,TH,ADD,NFREE,
AUX,SKEWEDCON,NSKEW,NSETF,ANGSK,NOSK)



J:=2 (2) 12

I:=J (2) 12

KE(J,I):=KE(I,J):=KE(I,J)+W(U,1)*(B(2,J)*(C(NODE(Z,7),4)*
B(2,I)+C(NODE(Z,7),5)*B(3,I))+B(3,J)*(C(NODE(Z,7),5)*B(2,I)
+C(NODE(Z,7),6)*B(3,I)))*DETJ*0.5*TH(NODE(Z,7))

I = 12

L6

KE(J,I+1):=KE(I+1,J):=KE(I+1,J)+W(U,1)*(B(2,J)*(C(NODE(Z,7),2)*
B(1,I+1)+C(NODE(Z,7),5)*B(3,I+1))+B(3,J)*(C(NODE(Z,7),3)*
B(1,I+1)+C(NODE(Z,7),6)*B(3,I+1)))*DETJ*0.5*TH(NODE(Z,7))

L6

NSKEW \neq 0

SKEWEDCON(Z,NODE,ANGSK,NOSK,
NSKEW,KE,NSETF)

I:=1 (1) 6

J:=1 (1) 6

V:=1,0

SUB1:=NODE(Z,I)*2-1
SUB2:=NODE(Z,J)*2-V
SUB3:=NODE(Z,I)*2

SUB1 < SUB2

LABA

K(ADD(SUB1)-SUB1+SUB2):=K(ADD(SUB1)-SUB1+SUB2)
+KE(I*2-1,J*2-V)

LABA

SUB3 < SUB2

LABB

K(ADD(SUB3)-SUB3+SUB2):=K(ADD(SUB3)-SUB3+SUB2)+KE(I*2,J*2-V)

LABB

END

F

G

```

'PROCEDURE' FASSEMBLY(X,ELEMT,K,XX,YY,DETJ,NODE,C,TH,ADD,NREF,
                     AUX,SKEVEDCON,NSKEW,NSSETF,ANGSK,NOSK);
'VALUE' NFELEMT,NREF,NSKEW,NSSETF;
'INTEGER' NFELEMT,NREF,NSKEW,NSSETF; 'REAL' DETJ;
'REAL' 'ARRAY' K,XX,YY,C,TH,ANGSK; 'INTEGER' 'ARRAY' NODE,ADD,NOSK;
'PROCEDURE' AUX,SKEVEDCON;
'REGIN' 'INTEGER' I,J,U,Z,V,SUB1,SUB2,SUB3;
'REAL' 'ARRAY' B[1:3,1:12],KE[1:12,1:12],W[1:6,1:4];
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' W[1,1]:=0.733333333;
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[NREF] 'DO' K[I,1]:=0.0;
'FOR' Z:=1 'STEP' 1 'UNTIL' NFELEMT 'DO'
'REGIN'
'FOR' I:=1 'STEP' 1 'UNTIL' 12 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 12 'DO' KE[I,J]:=0.0;
'COMMENT' THE LOOP FOR THE NUMBER OF INT PTS IS CONSTRUCTED;
'FOR' U:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
AUX(W[U,2],W[U,3],W[U,4],B,XX,YY,DETJ,'ODE,Z');
'FOR' J:=1 'STEP' 2 'UNTIL' 11 'DO'
'FOR' I:=J 'STEP' 2 'UNTIL' 11 'DO'
'BEGIN'
KE[I,1]:=KE[I,J]:=KE[I,J]+W[U,1]*(B[1,J]*(C[NODE[Z,7],1]*B[1,1]+
C[NODE[Z,7],3]*B[3,1])+B[3,J]*(C[NODE[Z,7],3]*B[1,1]+C[NODE[Z,7],6]*
B[3,1]))*DETJ*0.5*TH[NODE[Z,7]];
KE[I,1+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]*(B[1,J]*(C[NODE[Z,7],2]*
B[1,1+1]+C[NODE[Z,7],5]*B[3,1+1])+B[3,J]*(C[NODE[Z,7],5]*B[1,1+1]+
C[NODE[Z,7],6]*B[3,1+1]))*DETJ*0.5*TH[NODE[Z,7]];
'END';
'FOR' J:=2 'STEP' 2 'UNTIL' 12 'DO'
'FOR' I:=J 'STEP' 2 'UNTIL' 12 'DO'
'REGIN'
KE[I,1]:=KE[I,J]:=KE[I,J]+W[U,1]*(B[2,J]*(C[NODE[Z,7],4]*B[2,1]+
C[NODE[Z,7],5]*B[3,1])+B[3,J]*(C[NODE[Z,7],5]*B[2,1]+C[NODE[Z,7],6]*
B[3,1]))*DETJ*0.5*TH[NODE[Z,7]];
'IF' I=12 'THEN' 'GOTO' L6 'ELSE'
KE[I,1+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]*(B[2,J]*(C[NODE[Z,7],2]*
B[1,1+1]+C[NODE[Z,7],5]*B[3,1+1])+B[3,J]*(C[NODE[Z,7],3]*B[1,1+1]+
C[NODE[Z,7],6]*B[3,1+1]))*DETJ*0.5*TH[NODE[Z,7]];
L6: 'END';
'END';
'IF' NSKEW.NF=0 'THEN' SKEVEDCON(Z,NODE,ANGSK,NOSK,NSKEW,KE,NSSETF);
'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'
'REGIN' SUB1:=NODE[Z,I]*2-1;
SUB2:=NODE[Z,J]*2-V;
SUB3:=NODE[Z,I]*2;
'IF' SUB1 'LT' SUB2 'THEN' 'GOTO' LABA;
KIADD(SUB1)-SUB1+SUB2:=KIADD(SUB1)-SUB1+SUB2+KE[I*2-1,J*2-V];
LABA: 'IF' SUB3 'LT' SUB2 'THEN' 'GOTO' LABB;
KIADD(SUB3)-SUB3+SUB2:=KIADD(SUB3)-SUB3+SUB2+KE[I*2,J*2-V];
LABB: 'END';
'END';
'END' OF PROCEDURE FASSEMBLY;

```

4.7 APPLICATION OF BOUNDARY CONDITIONS

There are two types of boundary conditions which are considered, applied loading and prescribed displacements. The prescribed displacements may be non-zero and can be applied at any angle to the global axes. Each type is treated separately in the following subsections.

4.7.1 Applied Concentrated and Distributed Boundary Loading

In order to apply point loadings to a structure, it is merely necessary to establish the degree of freedom number in the direction of the required load at the relevant node and set the corresponding right hand side of the equation $[K] \{\delta\} = \{F\}$ to that load. For example, if a load of 100 lbf is required at node number 57 in the x-direction, then the term in the 113th row of $\{F\}$ is set to 100. (Details of how the user does this are given later where the presentation of input data is discussed, Section 5.2).

In the case of distributed load acting on an element it is possible to average the loads between the nodes on the boundary along which the load acts. However, this is not a very accurate method. A better method is to compute a set of nodal loads which do the same amount of work on the structure as the known distributed load.

Recalling equation 3.57 where the element force vector was derived for the element as :

$$\{F_e\} = \int_{\text{length}} \{p\}^t [N] dS$$

It is more usual to treat each side of an element separately and for demonstration purposes a distributed load is assumed to act on side 3 of the element shown in Fig. 3.12. In this illustration nodal loads are derived in the x-direction only, the procedure is the same for the y-direction and other faces.

Along the boundary of side 3 the displacements are expressed in terms of the nodal values on that side by the interpolation formula :

$$\begin{aligned} \{u(L_1)\} &= \begin{bmatrix} L_1(2L_1 - 1) & 4L_1(1 - L_2) & (1 - L_1)(1 - 2L_1) \end{bmatrix} \begin{Bmatrix} u_1 \\ u_4 \\ u_2 \end{Bmatrix} \quad (4.3) \\ &= [N_3]^t \{\delta_u\}_3 \quad (4.3a) \end{aligned}$$

Equation 4.3 is obtained by extracting terms corresponding to nodes 1, 4 and 2 from $[N]^t$ in equation 3.18 and setting $L_3 = 0$ with $L_2 = 1 - L_1$. Note also that the shape of this side is determined by the same interpolation formula as in equation 4.3.

$$\begin{aligned} x(L_1) &= [N_3]^t \begin{Bmatrix} x_1 \\ x_4 \\ x_2 \end{Bmatrix} \\ y(L_1) &= [N_3]^t \begin{Bmatrix} y_1 \\ y_4 \\ y_2 \end{Bmatrix} \end{aligned} \quad (4.4)$$

The limits of integration along side 3 are from 0 to 1 with dS_3 expressed in terms of dL_1 as follows :

$$\begin{aligned} dS_3 &= \sqrt{dx^2 + dy^2} = \sqrt{\left(\frac{\partial x}{\partial L_1} dL_1\right)^2 + \left(\frac{\partial y}{\partial L_1} dL_1\right)^2} \\ &= \sqrt{x_{,L_1}^2 + y_{,L_1}^2} dL_1 = G dL_1 \end{aligned} \quad (4.5)$$

where $x_{,L_1}$ and $y_{,L_1}$ are formed from equation 4.4.

For straight sided elements :

$$dS_3 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} dL_1 = l_3 dL_1$$

where l_3 is the length of side 3.

The distributed load $\{p\}$ of equation 3.57 may be expressed in terms of the nodal values of distributed loads using the appropriate interpolation functions as :

$$\{p_x\} = [N_3]_p^t \begin{Bmatrix} p_{x1} \\ p_{x4} \\ p_{x2} \end{Bmatrix} = [N_3]_p^t \{p_x\}_3 \quad (4.6)$$

Substituting equations 4.3, 4.5 and 4.6 into equation 3.57,

$$\begin{aligned} F_{x1} &= \{p_x\}_3^t Gh \int_0^1 [N_3]_p L_1 (2L_1 - 1) dL_1 \\ F_{x4} &= \{p_x\}_3^t Gh \int_0^1 [N_3]_p L_1 (1 - L_1) dL_1 \\ F_{x2} &= \{p_x\}_3^t Gh \int_0^1 [N_3]_p (1 - L_1)(1 - 2L_1) dL_1 \end{aligned} \quad (4.7)$$

where h is the element thickness

Similar expressions may be obtained for $\{F_y\}$ by replacing x by y in equations 4.7. Expressions for the other two sides of the element follow from a cyclic permutation of the indices.

Results are now derived for two load cases to demonstrate the generation of $[N_3]_p^t$.

(1) Linear variation of pressure along side 3, where distributed load is defined by p_{x1} and p_{x2} , magnitudes of distributed load at nodes 1 and 2 respectively.

$$p_x = [L_1 \ (1 - L_1)] \begin{Bmatrix} p_{x1} \\ p_{x2} \end{Bmatrix} = [N_3]_p^T \{p_x\} \quad (4.8)$$

Note: $[N_3]_p^t$ represents a linear variation of p_x where L_2 has been replaced by $(1 - L_1)$.

Substitution of equation 4.8 in equation 4.7 and performing the integrations, the result for a straight sided triangle is :

$$\begin{Bmatrix} F_{x1} \\ F_{x4} \\ F_{x2} \end{Bmatrix} = \frac{hl_3}{6} \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} p_{x1} \\ p_{x2} \end{Bmatrix} \quad (4.9)$$

If $p_x = p = \text{constant}$, equation 4.9 reduces to :

$$F_{x1} = F_{x2} = \frac{p l_3 h}{6}, \quad F_{x4} = \frac{2 p l_3 h}{3}$$

which is recognised as Simpson's Integration Rule, with two strips.

(2) Parabolic variation of $p_x (L_1)$ defined by :

$$\{p_x\}^T = \{p_{x_1} \quad p_{x_4} \quad p_{x_2}\}$$

In this case $[N_3]_p = N_3$ as defined in equation 4.3. Substituting in equation 4.7 and integrating gives,

$$\begin{Bmatrix} F_{x_1} \\ F_{x_4} \\ F_{x_2} \end{Bmatrix} = \frac{l_3 h}{30} \begin{bmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{bmatrix} \begin{Bmatrix} p_{x_1} \\ p_{x_4} \\ p_{x_2} \end{Bmatrix} \quad (4.10)$$

When the distributed load extends over more than one element then nodal loads on each element are determined and added together at the common nodes. When the distributed load implies an equivalent load at a node which is restrained in the direction of the load, then this portion of the load is ignored since a load and displacement may not both be prescribed at the same degree of freedom. These two points are illustrated in Fig. 4.3.

The load sharing is performed manually by the user in accordance with equation 4.9 or equation 4.10, and loads are input as nodal concentrated values. The following procedure locates these loads in their appropriate positions in the overall load vector.

A flowchart and listing of the procedure follows; the operations are as previously described with regard to concentrated loads, complicated slightly by the provision for prescribed displacements.

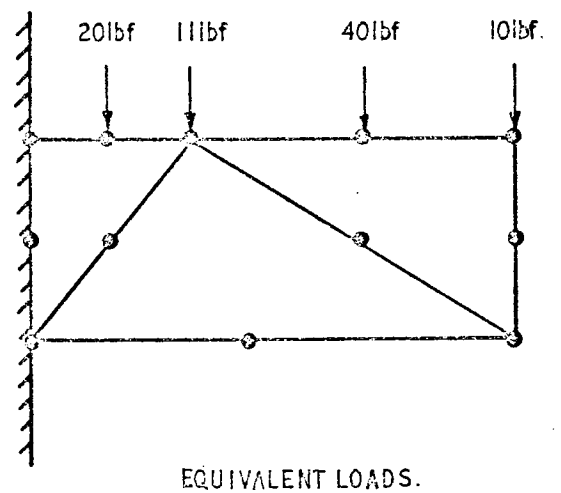
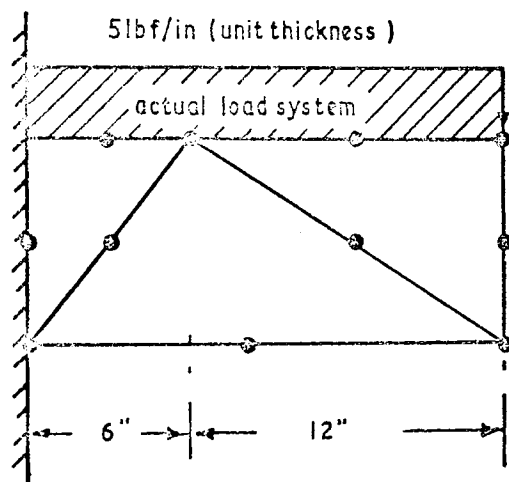
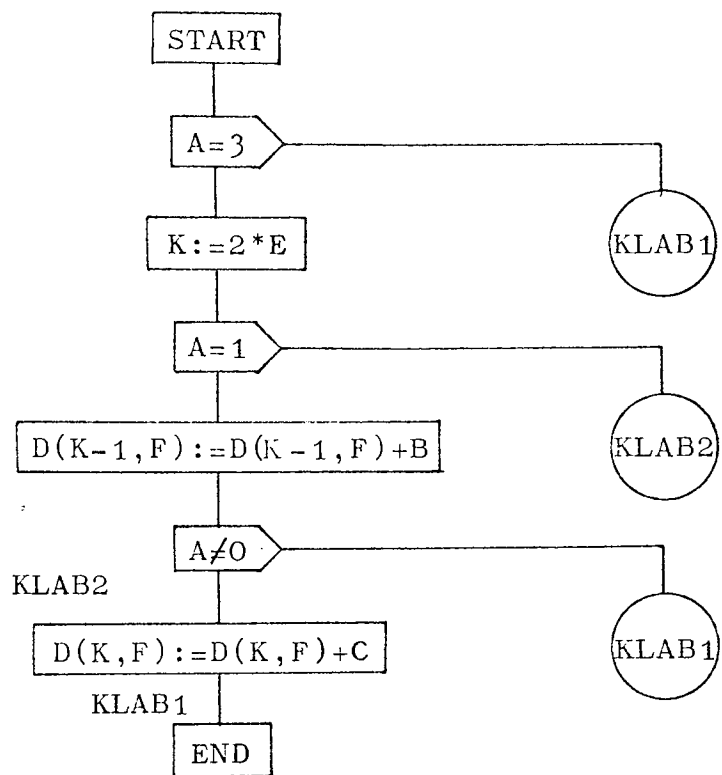


FIG 4.3 LOAD SHARING.

PROCEDURE : LOADAPP(A,B,C,D,E,F)



```

'PROCEDURE' LOADAPP(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' KLAR1;
K:=2*B;
'IF' A=1 'THEN' 'GOTO' KLAR2;
D[K-1,F]:=D[K-1,F]+B;
'IF' A 'NE' 0 'THEN' 'GOTO' KLAR1;
KLAR2: D[K,F]:=D[K,F]+C;
KLAR1: 'END' OF LOADAPP;

```

4.7.2 Prescribed Nodal Displacements

Once the overall stiffness matrix for the structure has been assembled the kinematic boundary conditions (prescribed nodal displacements) must be applied before a solution can be obtained. This requirement may be appreciated physically by considering unconstrained rigid body modes where displacements cannot be uniquely determined for the applied forces. The mathematical interpretation is that the matrix $[K]$ is singular, i.e. does not possess an inverse if the system is not constrained against rigid body displacements.

In some instances the prescribed displacements may not be in the direction of the global coordinates. The application of such skewed boundary conditions requires a transformation of the system matrices for the nodes where skewed conditions are specified.

Adopting the notation that prime indicates skewed coordinates, the transformation of displacements from the skewed to the unskewed coordinates at node i (fig. 4.4) may be written,

$$\{\delta_i\} = [S_i] \{\delta_i'\} \quad (4.11)$$

where

$$\{\delta_i\}^t = \{\delta_{x_i} \quad \delta_{y_i}\}$$

and $[S_i]$ is the transformation matrix.

The transformation matrix for the entire nodal displacement vector may be constructed as :

$$\{\delta\} = [S] \{\delta'\} \quad (4.12)$$

where $[S]$ is of the form,

$$[S] = \begin{bmatrix} [I] & & & \\ & [I] & & \\ & & [I] & \\ & & & [S_i] \\ & 0 & & & [I] \end{bmatrix}$$

$[I]$ is the identity matrix, order 2×2 (number of degrees of freedom/node) and the number of diagonal submatrices equals the number of nodes in the structure.

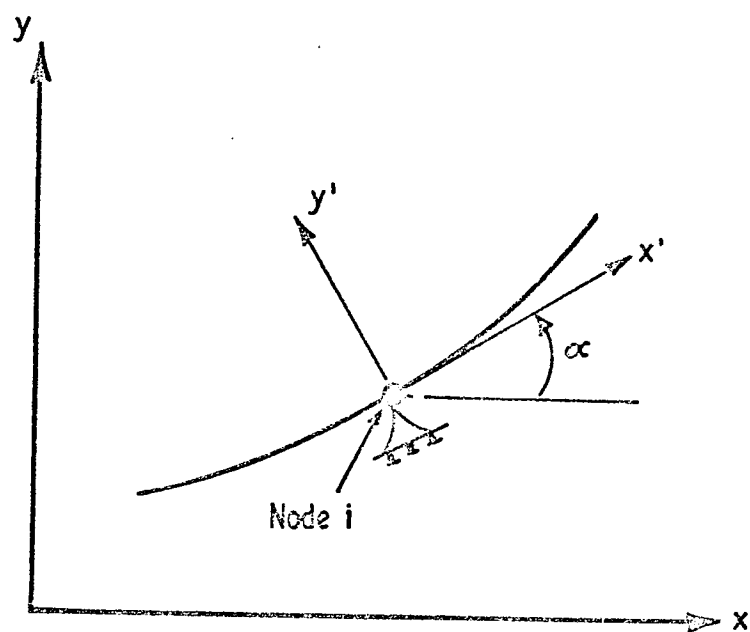


FIG4.4 SKEWED KINEMATIC CONSTRAINT.

The transformations for the overall system matrices may now be obtained as follows :-

Equating work done by each force system, which must be the same,

$$\{F\}^t \{\delta\} = \{F^1\}^t \{\delta^1\}$$

Substituting equation 4.12 in the above and comparing coefficients gives :

$$\{F^1\} = [S]^t \{F\}$$

Pre-multiplying the equilibrium equations ($[K]\{\delta\} = \{F\}$) for the unconstrained system by $[S]^t$ and substituting for $\{\delta\}$ from equation 4.12 :

$$[S]^t [K] [S] \{\delta^1\} = [S]^t \{F\}$$

from which the equilibrium equations for the skewed system may be constructed,

$$[K^1] \{\delta^1\} = \{F^1\} \quad (4.13)$$

where $[K^1] = [S]^t [K] [S]$

and $\{F^1\} = [S]^t \{F\}$

The procedure described above for the overall system equations can be applied to the individual element stiffness matrices before they are assembled, in such a case the overall stiffness matrix and load vector are replaced by the element stiffness matrices and load vectors respectively in the previous derivation. In the program described in this thesis, the matrix operations for transforming displacements are applied to the individual element stiffness matrices immediately after they are assembled. The procedure which performs these operations to nodes, as specified in the data, at which skewed constraints are imposed, is known as **SKENEDCON**. For the load vector, however, it proves more convenient to apply the transformation of equation 4.13 to the overall load vector. To carry out this transformation, procedure **SKETLOAD** has been written. After the transformations to the overall stiffness matrix have been carried out, the values of prescribed displacements are applied using procedure **GEONBC**.

The overall system equations can now be solved to determine values for the nodal point displacements. Since some of these displacements are in the skewed coordinate directions, the procedure SKEWLOAD is recalled to convert such skewed displacements back to the global coordinate system. The final displacement components which are output are all in the global coordinate directions.

4.7.2.1 Procedure SKEWEDCON

The transformation matrix for node i, equation 4.11, has the form :

$$\begin{Bmatrix} \delta_{xi} \\ \delta_{yi} \end{Bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{Bmatrix} \delta_{xi}^1 \\ \delta_{yi}^1 \end{Bmatrix} \quad (4.14)$$

where α is the angle of skew, Fig. 4.3

If the element stiffness matrix is considered as composed of 2 x 2 sub-blocks then the matrix multiplication is simplified considerably, i.e. from equation 4.13,

$$\begin{bmatrix} K_{ij}^1 \end{bmatrix}_e = \begin{bmatrix} S_i \end{bmatrix}^t \begin{bmatrix} K_{ij} \end{bmatrix}_e \begin{bmatrix} S_j \end{bmatrix} \quad i, j = 1, 2, \dots 6 \quad (4.15)$$

where $\begin{bmatrix} K_{ij}^1 \end{bmatrix}_e$ is a 2 x 2 submatrix of the new element stiffness matrix.

$\begin{bmatrix} S_i \end{bmatrix}^t$ is the transpose of the transformation matrix for node i.

$\begin{bmatrix} K_{ij} \end{bmatrix}_e$ is a 2 x 2 submatrix of the original element stiffness matrix,

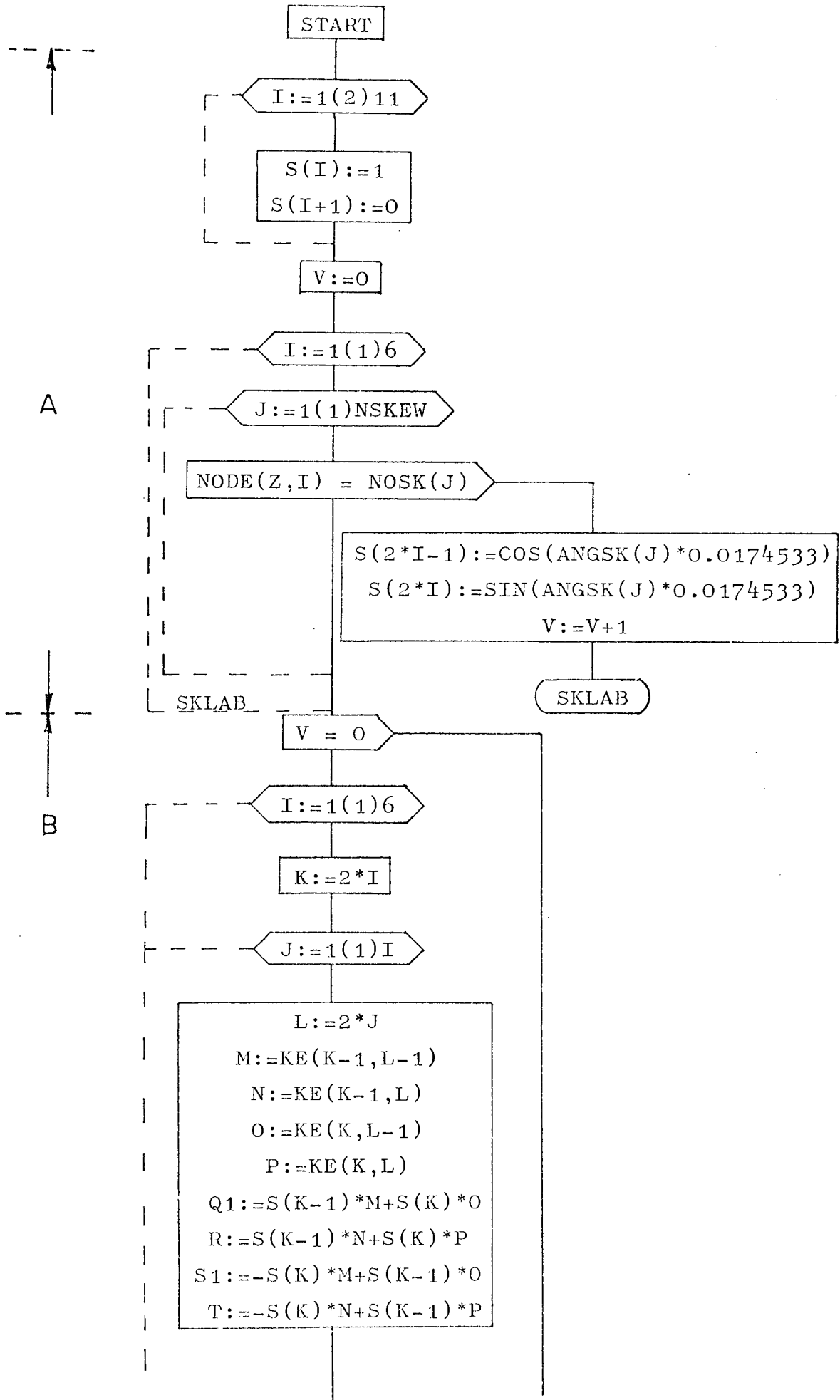
and $\begin{bmatrix} S_j \end{bmatrix}$ is the transformation matrix for node j.

The skew-symmetry of the nodal transformation matrices means that only two of the four coefficients need be stored, accordingly, the six submatrices for an element are stored in a one-dimensional array S by means of the following procedure.

A flowchart and program listing of the procedure follows and the steps involved are, with reference to the flowchart :-

- A. - Coefficients of the transformation array S are set according to equation 4.14 for the skewed node.
- B. - Matrix multiplication of equation 4.15 is performed on the element stiffness matrix.

PROCEDURE : SKEWEDCON(Z,NODE,ANGSK,NOSK,NSKEW,KE,NSETF)



B

$$KE(K-1, L-1) := KE(L-1, K-1) := Q1 * S(L-1) + R * S(L)$$

$$KE(K-1, L) := KE(L, K-1) := -Q1 * S(L) + R * S(L-1)$$

$$KE(K, L-1) := KE(L-1, K) := S1 * S(L-1) + T * S(L)$$

$$KE(K, L) := KE(L, K) := -S1 * S(L) + T * S(L-1)$$

END

```

'PROCEDURE' SKWEDCON(Z,NODE,ANGSK,NOSK,NSKEW,KE,NSETF):
'VALUE' Z,NSKEW,NSETF: 'INTEGER' Z,NSKEW,NSETF:
'INTEGER' 'ARRAY' NODE,NOSK: 'REAL' 'ARRAY' ANGSK,KE:
'REGIN' 'INTEGER' I,J,V,K,L: 'REAL' M,N,O,P,O1,R,S1,T:
'REAL' 'ARRAY' S[1:12]:
'FOR' I:=1 'STEP' 2 'UNTIL' 11 'DO' 'BEGIN' S[I]:=1;S[I+1]:=0; 'END':
V:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'REGIN' 'FOR' J:=1 'STEP' 1 'UNTIL' NSKEW 'DO'
'IF' NODE[Z,I]=NOSK[J] 'THEN'
'REGIN' S[2*I-1]:=COS(ANGSK[J]*0.0174533):
S[2*I]:=SIN(ANGSK[J]*0.0174533):
V:=V+1; 'GOTO' SKLAB;
'END':
SKLAB: 'END':
'IF' V 'NE' 0 'THEN'
'REGIN' 'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'REGIN' K:=2*I:
'FOR' J:=1 'STEP' 1 'UNTIL' I 'DO'
'REGIN' L:=2*J:
M:=KE[K-1,I-1]; N:=KE[K,I];
O:=KE[K,L-1]; P:=KE[K,L];
O1:=S[K-1]*M+S[K]*O; R:=S[K-1]*N+S[K]*P;
S1:=-S[K]*M+S[K-1]*O; T:=-S[K]*R+S[K-1]*P;
KE[K-1,I-1]:=KE[L-1,K-1]:=O1*S[L-1]+R*S[L];
KE[K-1,L]:=KE[L,K-1]:=-O1*S[L]+R*S[L-1];
KE[L,I-1]:=KE[I-1,K]:=S1*S[L-1]+T*S[L];
KE[K,L]:=KE[L,K]:=-S1*S[L]+T*S[L-1];
'END':
'END':
'END':
'END' OF PROCEDURE SKWEDCON;

```

4.7.2.2 Procedure SKEWLOAD

The modified overall load vector for a typical skewed node i is obtained from equation 4.13 as :

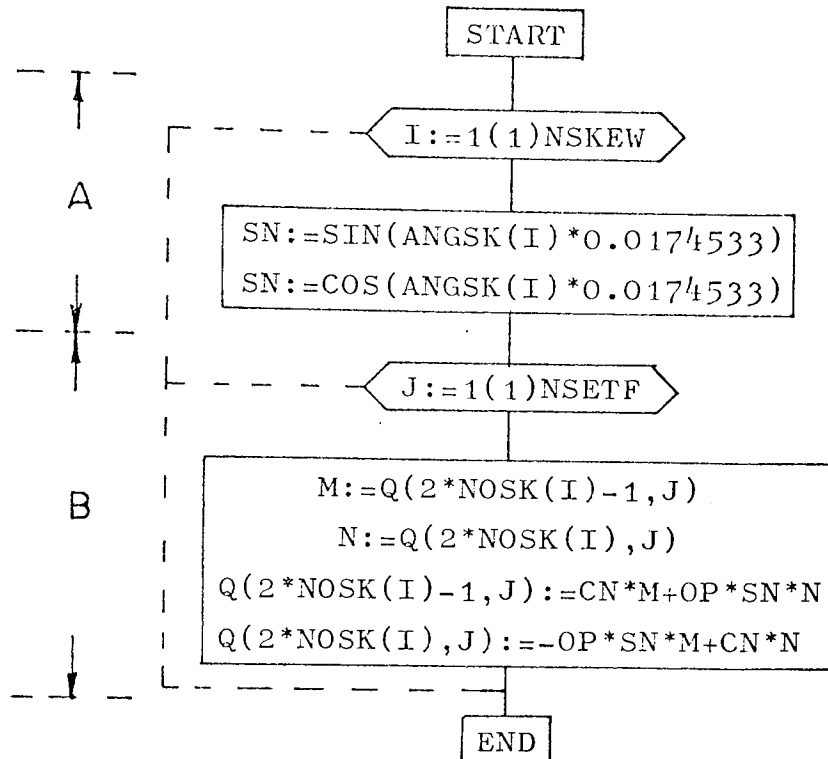
$$\{F_i^1\} = [S]^t \{F_i\} \quad (4.16)$$

In this case $\{F_i\}$ is a 2×2 sub-block of the overall load vector corresponding to node i .

The flowchart and program listing for procedure SKEWLOAD follow and the steps involved, with reference to the flowchart, are :

- A. - The two independent coefficients of the transformation matrix for the skewed node i are calculated.
- B. - Appropriate terms of the overall load vector are modified according to equation 4.16.

PROCEDURE : SKEWLOAD(NSETF,Q,NOSK,ANGSK,NSKEW,OP)



```

'PROCEDURE' SKEWLOAD('NSETF,Q,MOSK,ANGSK,NSKEW,OP):
'VALUE' NSETF,NSKEW,OP; 'INTEGER' NSETF,NSKEW,OP;
'INTEGER' 'ARRAY' MOSK; 'REAL' 'ARRAY' Q,ANGSK;
'BEGIN' 'INTEGER' I,J; 'REAL' SN,CN,'N;
'FOR' I:=1 'STEP' 1 'UNTIL' NSKEW 'DO'
'BEGIN' SM:=SIN(ANGSK[I]*0.0174533);
      CN:=COS(ANGSK[I]*0.0174533);
      'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO'
      'BEGIN' M:=Q[2*MOSK[I]-1,J];
            N:=Q[2*MOSK[I],J];
            Q[2*MOSK[I]-1,J]:=CN*M+OP*SN*N;
            Q[2*MOSK[I],J]:=-OP*SN*M+CN*N;
      'END';
    'END';
'END' OF PROCEDURE SKEWLOAD;

```


4.7.2.3 Procedure GEOMBC

The application of prescribed displacements to the overall equilibrium equations is performed after the transformations for any skewed boundary conditions have been applied. The principle for the application of such kinematic displacements can be explained by partitioning and re-arranging the overall system equations as :

$$\begin{bmatrix} [K_{11}] & [K_{21}]^t \\ [K_{21}] & [K_{22}] \end{bmatrix} \begin{Bmatrix} \{\delta_1\} \\ \{\delta_2\} \end{Bmatrix} = \begin{Bmatrix} \{F_1\} \\ \{F_2\} \end{Bmatrix} \quad (4.17)$$

where $\{\delta_1\}$ is a vector of unconstrained or free nodal displacements and $\{\delta_2\}$ is a vector containing values of specified displacements.

The problem is then reduced to finding the solution of the system of equations,

$$[K_{11}] \{\delta_1\} = \{F_1\} - [K_{21}]^t \{\delta_2\}$$

for the vector $\{\delta_1\}$, and the subsequent resubstitution of this vector to find the reactions at the constrained nodes, i.e.

$$\{F_2\} = [K_{12}] \{\delta_1\} + [K_{22}] \{\delta_2\}$$

For the particular case where the values of the specified displacements are zero, the procedure is considerably simplified. The rows and columns of the system equations corresponding to the constrained degrees of freedom are eliminated and the solution proceeds as usual.

For the general case where the displacements may be non-zero, it is more convenient to rearrange equation 4.17 in the form,

$$\begin{bmatrix} [K_{11}] & [0] \\ [0] & [I] \end{bmatrix} \begin{Bmatrix} \{\delta_1\} \\ \{\delta_2\} \end{Bmatrix} = \begin{Bmatrix} \{F_1\} - [K_{21}] \{\delta_2\} \\ \{\delta_2\} \end{Bmatrix} \quad (4.18)$$

In order to preserve the address sequence for $[K]$ the above process, equation 4.18, is carried out without the reordering implied by the partitioning. Rather, the contributions to the subvector $(\{F_1\} - [K_{12}] \{\delta_2\})$ are first constructed for each prescribed displacement, then

the row and column of $[K']$ corresponding to this degree of freedom are made zero, with the exception of the diagonal element which is set to unity. Finally, the prescribed value of the displacement is inserted in the load vector.

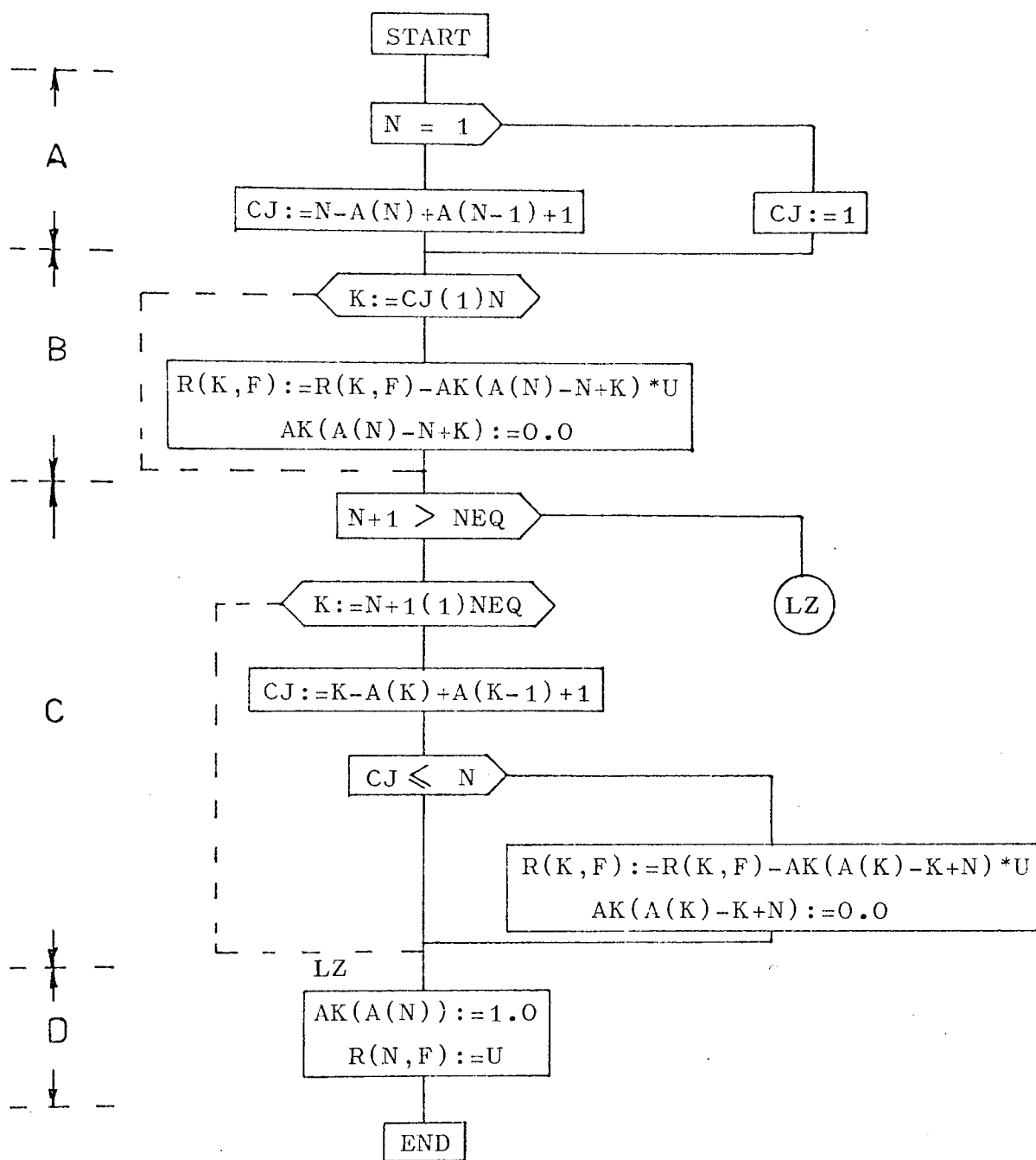
The above procedure may be summarised for a specified displacement δ_j at the j^{th} degree of freedom, as :

$$\begin{aligned} F_i &= F_i - K_{ij} \delta_j & i &= 1, 2, \dots, N \\ K_{jm} &= K_{mj} = 0 & m &= 1, 2, \dots, N \\ K_{jj} &= 1 \\ \text{and } F_j &= \delta_j \end{aligned} \quad (4.19)$$

The subsequent flowchart and procedure listing represents the application of these matrix operations where the matrix $[K]$ is stored as a one-dimensional array. Equation 4.1 is employed to relate equations 4.19 to the one-dimensional array. The steps involved are, with reference to the flowchart :

- A. - Integer N represents the degree of freedom at which the prescribed displacement is applied. Column number of the first non-zero coefficient in row N is calculated, CJ.
- B. - The load vector and rows of the stiffness matrix are modified according to the first two equations of equation 4.19, i.e. from column CJ to the leading diagonal term.
- C. - The load vector and columns of the stiffness matrix are modified according to the first two equations of equation 4.19. In this case a check must be made by the procedure to ensure that the first non-zero coefficient stored in any row is not beyond the degree of freedom N under consideration, hence CJ is computed and compared with N as shown.
- D. - The diagonal term is set to unity and the value of the prescribed displacement is inserted into the load vector according to the last two equations of equation 4.19.

PROCEDURE : GEOMBC(U,N,R,AK,NEQ,F,A)



```

'PROCEDURE' GLOMBC(U,N,R,AK,N=Q,F,A);
'VALUE' U,N,N=Q,F;
'REAL' U;
'INTEGER' N,N=Q,F;
'ARRAY' R,AK;
'INTEGER' 'ARRAY' A;
'REGION' 'INTEGER' M,K,CJ;
'IF' M=1 'THEN' CJ:=1 'ELSE' CJ:=N-(A[N]-A[N-1])+1;
'FOR' K:=CJ 'STEP' 1 'UNTIL' N 'DO'
'REGION'
    R[K,F]:=R[K,F]-AK[A[N]-N+K]*U;
    AK[A[N]-N+K]:=0.0;
'END';
'IF' M+1 'GT' N=Q 'THEN' 'GOTO' L2;
'FOR' K:=M+1 'STEP' 1 'UNTIL' N=Q 'DO'
'REGION' CJ:=K-(A[K]-A[K-1])+1;
    'IF' CJ 'LE' N 'THEN'
        'REGION'
            R[K,F]:=R[K,F]-AK[A[K]-K+N]*U;
            AK[A[K]-K+N]:=0.0;
        'END' 'ELSE'
'END';
L2: A[A[N]]:=1.0;
    R[N,F]:=U;
'END';

```

4.8 SOLUTION OF SYSTEM EQUATIONS

The method of storage and solution is due to Jennings and Tuff⁽²²⁾. In the method the equations are stored using the variable bandwidth technique discussed in section 4.5 and Choleski triangular factorisation is used for their solution.

During the reduction of a set of sparse equations by any variant of Gaussian Elimination, the zeroes before the first non-zero terms in any row remain zero if there is no row or column interchange. The variable band storage scheme makes use of this property by storing, for each row, only those terms occurring between the first non-zero term and the leading diagonal.

Choleski factorisation of the equations $[K] \{\delta\} = \{F\}$ yields a matrix of the form :

$$[L][L]^t = [K] \quad (4.20)$$

where $[L]$ is a lower triangular matrix with positive diagonal terms.

Substituting equation 4.20 into $[K]\{\delta\} = \{F\}$ gives,

$$[L][L]^t \{\delta\} = \{F\}$$

$$\text{or } [L][Y] = \{F\}$$

$$\text{where } [Y] = [L]^t \{\delta\}$$

The variables in $[Y]$ are the modified right hand side coefficients after elimination. The back substitution process to complete the solution is the determination of $\{\delta\}$ from

$$[L]^t \{\delta\} = [Y]$$

The matrix $[L]$ overwrites $[K]$ in the store by use of recursive relations,

$$L_{ii} = \left\{ K_{ii} - \sum_{k=1}^{i-1} L_{ik}^2 \right\}^{\frac{1}{2}} \quad \text{for diagonal terms, and}$$

```

'PROCEDURE' SYMVRSDI(A,I,S,B)DIMENSIONS:(B,R)FAILURE EXIT:(FAIL);
'VALUE' N,R; 'ARRAY' A,I,P; 'INTEGER' 'ARRAY' S; 'INTEGER' N,R;
'LABEL' FAIL;
'REGION'
'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'
    I:=I+H-S[I]*1; G:=H+1;
    P:=N-I-1;
    'FOR' J:=T 'STEP' 1 'UNTIL' I-1 'DO'
      'BEGIN'
        Q:=P+1; H:=H+1;
        P:=S[Q]; V:=J+Q-P;
        V:=H-P; U:=G;
        Y:=A[U];
        'IF' V<'G' T 'THEN' U:=U+K-T;
        'FOR' H:=U 'STEP' 1 'UNTIL' H-1 'DO'
          Y:=Y-L[U]*I[U-V];
          Y:=Y/I[H-V]; I[H]:=Y;
        'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'
          B[I,M]:=B[I,M]-P[J,M]*Y;
        'END' J;
        Y:=A[H+1];
        'FOR' U:=G 'STEP' 1 'UNTIL' H 'DO'
          Y:=Y-L[U]*2;
        'IF' Y<'E' 0 'THEN' 'GOTO' FAIL;
        H:=H+1; V:=SORT(Y);
        I[H]:=Y;
        'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'
          B[I,M]:=R[I,M]/Y;
        'END' I;
'COMMENT' REDUCTION COMPLETE;
'FOR' I:=N 'STEP' -1 'UNTIL' 1 'DO'
  'BEGIN'
    Y:=L[I];
    'FOR' M:=1 'STEP' 1 'UNTIL' R 'DO'
      B[I,M]:=R[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,H]*Y;
        'END' H;
        R:=P;
COMPLETE: 'END' I;
'END' SYMVRSDI;

```

$$L_{ij} = \left\{ K_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} \right\} / L_{jj} \quad \text{for off-diagonal terms.}$$

These relations show that to form any coefficient L_{ij} only rows i and j of the stiffness matrix are required to be stored. This facilitates the use of backing store for the solution of large problems and is discussed more fully by Jennings and Tuff⁽²²⁾.

A program listing is given but no flowchart is included as the program was taken direct from (22).

4.9 CALCULATION OF ELEMENT AND NODAL POINT STRESSES AND STRAINS

Two procedures are presented in this section for the evaluation of stresses and strains. In the first, the stresses and strains are calculated at the centroid of each element and in the second they are calculated at the nodal points by averaging contributions from adjoining elements sharing a common node. In each case, however, the basic theory is the same and will be discussed prior to a detailed description of the respective procedures. The program user has a choice, therefore, of element centroidal and/or nodal values of stresses and strains. This is discussed in Section 5.2 where the input data instructions for the programs are given.

The method of obtaining the stresses and strains is extremely simple once the nodal point displacements have been determined. Firstly, the coefficients of the strain-element nodal point displacement array $[B]$ are evaluated at the point of interest. The strains at that point are determined from equation 3.48 as :

$$\{\epsilon\} = [B] \{\delta_e\}$$

Finally, the stresses at the same point may be calculated from the strains using the equations of section 3.4 in the form of equation 3.52a as

$$\{\sigma\} = [c] \{\epsilon\}$$

The two procedures which follow employ the above equations for the calculation of centroidal values of stress and strain, procedure FEELSTR and nodal values of stress and strain, procedure FENSTR.

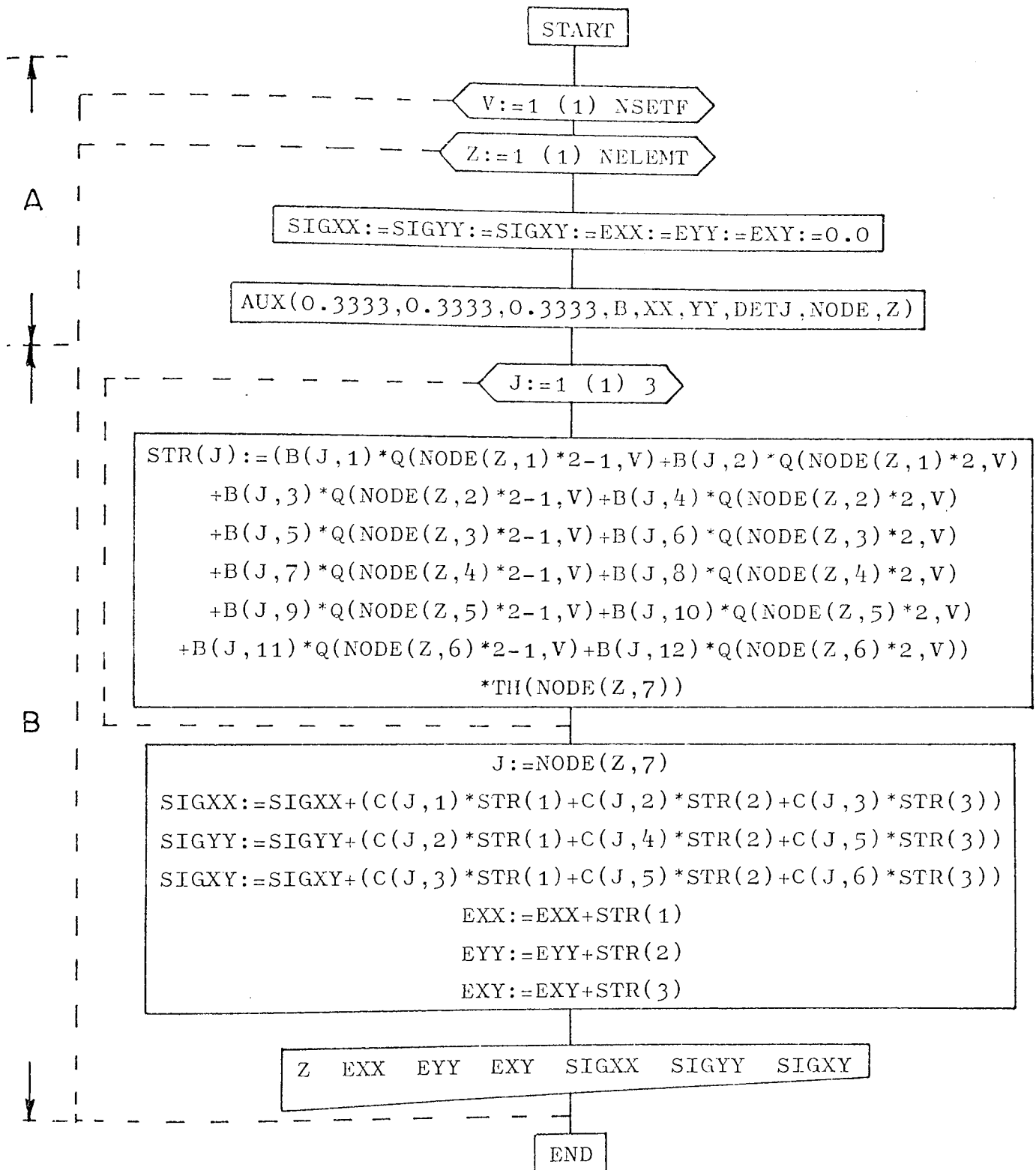
4.9.1 Procedure FEELSTR

The procedure flowchart and program listing follow and the relevant steps involved are described, with reference to the flowchart as :

- A. - Procedure AUX is called to calculate the coefficients of $[B]$ at the centroid of the first element, natural coordinates $\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$.
- B. - Matrix operations, equations 3.48 and 3.52a are carried out to determine values of stresses and strains.

Steps A and B are repeated for each element in turn.

PROCEDURE : FEELSTR(NSETF,Q,NELEMT,NODE,XX,YY,DETJ,TH,C,AUX)



```

'PROCEDURE' FEELSTR(NSETF, N, NELEMT, NODE, XX, YY, DETJ, TH, C, AUX);
'VALUE' NSETF, NELEMT; 'INTEGER' NSETF, NELEMT;
'REAL' DETJ; 'INTEGER' 'ARRAY' NODE;
'REAL' 'ARRAY' XX, YY, 0, 0, TH; 'PROCEDURE' AUX;
'BEGIN' 'COMMENT' CALCULATION OF ELEMENT STRESSES AND STRAINS;
'INTEGER' V, Z, J;
'REAL' SIGXX, SIGYY, SIGXY, FXX, FYY, FXY;
'REAL' 'ARRAY' STR[1:3], P[1:3, 1:12];
'FOR' V:=1 'STEP' 1 'UNTIL' NSETF 'DO'
'BEGIN'
WRITETEXT('('('5020S')'STRAINS'('43S')'STRESSES'('C')'ELEM'('3S')'
FXX'('10S')'FYY'('10S')'EXY'('21S')'SIGMA-X'('6S')'
SIGMA-Y'('6S')'SIGMA-XY')');
'FOR' Z:=1 'STEP' 1 'UNTIL' NELEMT 'DO'
'BEGIN'
SIGXX:=SIGYY:=SIGXY:=FXX:=FYY:=FXV:=0.0;
AUX(0.3333, 0.3333, 0.3333, F, XX, YY, DETJ, NODE, Z);
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
STR[J]:=(B[J, 1]*O[NODE[Z, 1]*2-1, V]+B[J, 2]*O[NODE[Z, 1]*2, V]
+P[J, 3]*O[NODE[Z, 2]*2-1, V]+P[J, 4]*O[NODE[Z, 2]*2, V]
+P[J, 5]*O[NODE[Z, 3]*2-1, V]+P[J, 6]*O[NODE[Z, 3]*2, V]
+P[J, 7]*O[NODE[Z, 4]*2-1, V]+P[J, 8]*O[NODE[Z, 4]*2, V]
+P[J, 9]*O[NODE[Z, 5]*2-1, V]+P[J, 10]*O[NODE[Z, 5]*2, V]
+P[J, 11]*O[NODE[Z, 6]*2-1, V]+P[J, 12]*O[NODE[Z, 6]*2, V])
*TH[NODE[Z, 7]];
'END';
J:=NODE[Z, 7];
SIGXX:=SIGXX+(C[J, 1]*STR[1]+C[J, 2]*STR[2]+C[J, 3]*STR[3])
SIGYY:=SIGYY+(C[J, 2]*STR[1]+C[J, 4]*STR[2]+C[J, 5]*STR[3])
SIGXY:=SIGXY+(C[J, 3]*STR[1]+C[J, 5]*STR[2]+C[J, 6]*STR[3])
FXX:=FXX+STR[1];
FYY:=FYY+STR[2];
FXY:=FXY+STR[3];
NEWLINE(1);
PRINT(Z, 2, 0);
PRINT(FXX, 0, 4);
PRINT(FYY, 0, 4);
PRINT(FXY, 0, 4); SPACE(10);
PRINT(SIGXX, 0, 4);
PRINT(SIGYY, 0, 4);
PRINT(SIGXY, 0, 4);
'END';
'END';
'END' OF PROCEDURE FEELSTR;

```

↑*****

4.9.2 Procedure FENOSTR

By means of this procedure values of nodal point stresses and strains are calculated by averaging contributions from all elements adjoining a given node. The procedure flowchart and program listing follow and the various steps involved are described, with reference to the flowchart, as

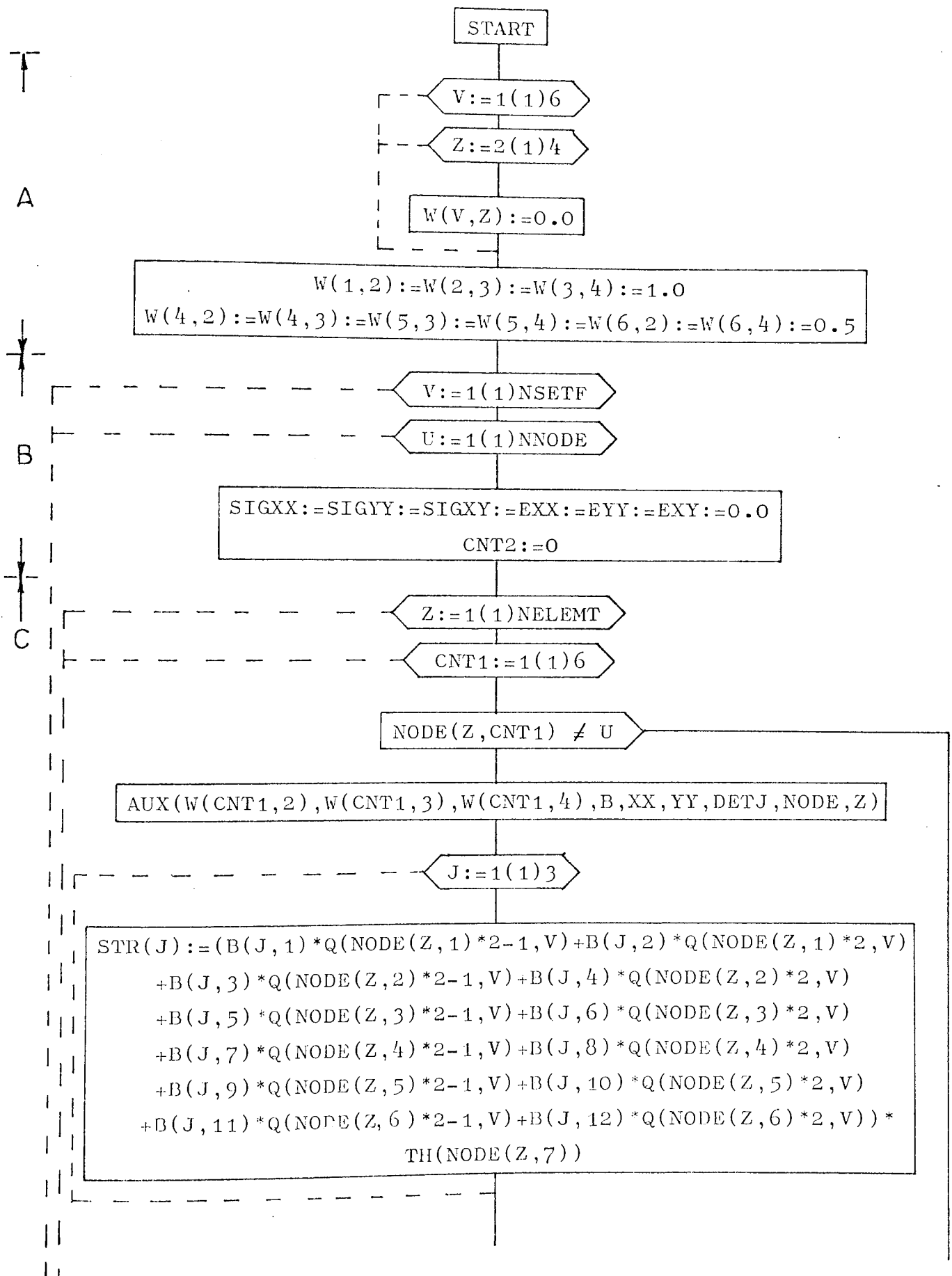
- A. - The coefficients of the array W are set to the natural coordinates of the triangle nodal points.
- B. - A loop is constructed round the number of sets of forces, headings are printed and a loop is constructed round all the nodes.
- C. - For the first node, each element is searched to find if it contains this node. If an element is found to contain this node, procedure AUX is called with the appropriate natural coordinates to evaluate the coefficients of the matrix [B] . With these coefficients the matrix operations of equations 3.48 and 3.52a are executed to find values of stresses and strains.
- D. - Values of stresses and strains are divided by the number of elements found to contain this node.
- E. - Values found in step D are printed.

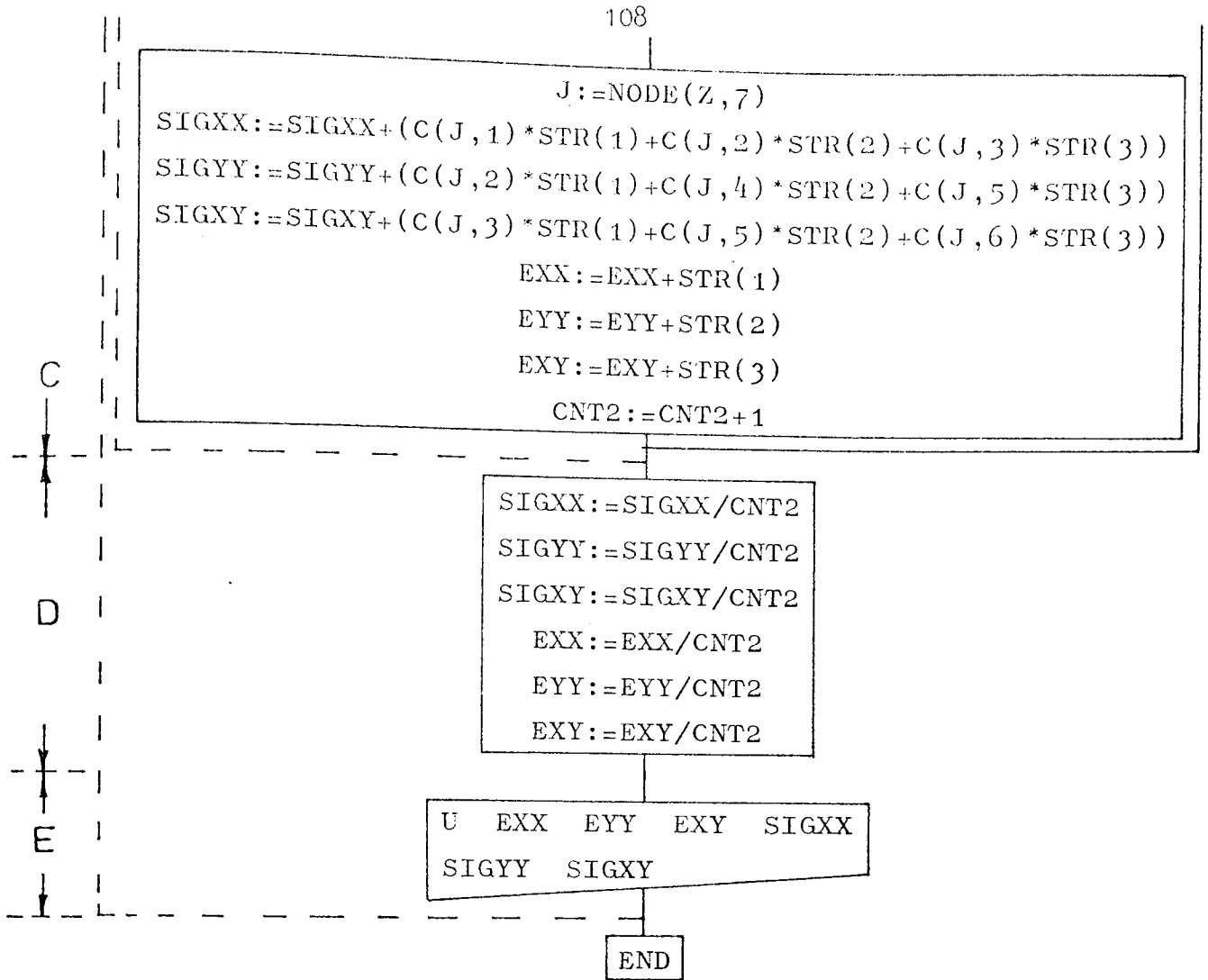
Steps C to E are carried out for each node in turn.

```

PROCEDURE : FENOSTR(NSETF,NNODE,NELENT,NODE,XX,YY,DETJ,Q,
                  TH,C,AUX)

```





```

'PROCEDURE' FEPOSTR(NSETF, NNODE, NELEMT, NODE, XY, YY, DETJ, Q, TH, C, AUX);
'VALUE' NSETF, NNODE, NELEMT; 'INTEGER' NSETF, NNODE, NELEMT;
'REAL' DETJ; 'INTEGER' 'ARRAY' NODE;
'REAL' 'ARRAY' XX, YY, Q, C, TH; 'PROCEDURE' AUX;
'BEGIN' 'COMMENT' CALCULATION OF STRAINS AND STRESSES AT NODAL
                    POINTS BY AVERAGING CONTRIBUTIONS FROM
                    ADJACENT ELEMENTS;
'INTEGER' CNT1, CNT2, V, Z, H, J;
'REAL' SIGXX, SIGYY, SIGXY, EXX, EYY, EXY;
'REAL' 'ARRAY' STR[1:3], W[1:6, 1:4], R[1:3, 1:12];
'FOR' V:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' Z:=2 'STEP' 1 'UNTIL' 4 'DO' W[V, Z]:=0.0;
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('('('SC20S')'STRAINS'('43S')'STRESSES'('C')'NODE'('3S')'
          EXX'('10S')'EYY'('10S')'EXY'('21S')'SIGMA-X'('6S')'
          SIGMA-Y'('6S')'SIGMA-XY'(')'));
'FOR' H:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'REGIN' SIGXX:=SIGYY:=SIGXY:=EXX:=EYY:=EXY:=0.0;
CNT2:=0;
'FOR' Z:=1 'STEP' 1 'UNTIL' NELEMT 'DO'
'BEGIN'
'FOR' CNT1:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN' 'IF' NODE[Z, CNT1]=U 'THEN'
'BEGIN'
AUX(W[CNT1, 2], W[CNT1, 3], W[CNT1, 4], R, XY, YY, DETJ, NODE, Z);
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
STR[J]:=(R[J, 1]*Q[NODE[Z, 1]*2-1, V]+R[J, 2]*Q[NODE[Z, 1]*2, V]
+R[J, 3]*Q[NODE[Z, 2]*2-1, V]+R[J, 4]*Q[NODE[Z, 2]*2, V]
+R[J, 5]*Q[NODE[Z, 3]*2-1, V]+R[J, 6]*Q[NODE[Z, 3]*2, V]
+R[J, 7]*Q[NODE[Z, 4]*2-1, V]+R[J, 8]*Q[NODE[Z, 4]*2, V]
+R[J, 9]*Q[NODE[Z, 5]*2-1, V]+R[J, 10]*Q[NODE[Z, 5]*2, V]
+R[J, 11]*Q[NODE[Z, 6]*2-1, V]+R[J, 12]*Q[NODE[Z, 6]*2, V])*
TH[NODE[Z, 7]];
'END';
J:=NODE[Z, 7];
SIGXX:=SIGXX+(C[J, 1]*STR[1]+C[J, 2]*STR[2]+C[J, 3]*STR[3])
SIGYY:=SIGYY+(C[J, 2]*STR[1]+C[J, 4]*STR[2]+C[J, 5]*STR[3])
SIGXY:=SIGXY+(C[J, 3]*STR[1]+C[J, 5]*STR[2]+C[J, 6]*STR[3])
EXX:=EXX+STR[1];
EYY:=EYY+STR[2];
EXY:=EXY+STR[3];
CNT2:=CNT2+1;
'END';
'END';
'END';
SIGXX:=SIGXX/CNT2; SIGYY:=SIGYY/CNT2; SIGXY:=SIGXY/CNT2;
EXX:=EXX/CNT2; EYY:=EYY/CNT2; EXY:=EXY/CNT2;
NEWLINE(1);
PRINT(H, 2, 0);
PRINT(EXX, 0, 4);
PRINT(EYY, 0, 4);
PRINT(EXY, 0, 4); SPACE(10);
PRINT(SIGXX, 0, 4);
PRINT(SIGYY, 0, 4);
PRINT(SIGXY, 0, 4);
'END';
'END';
'END' OF PROCEDURE FEPOSTR;

```

5. APPLICATIONS OF THE SIX-NODE ISOPARAMETRIC TRIANGULAR FINITE ELEMENT

5.1 INTRODUCTION

A flowchart and listing for the two-dimensional plane stress/plane strain finite element program using a six-node isoparametric triangle as element is presented. Data instructions necessary for running the program are given together with an example to demonstrate a typical data set with the resulting output.

The objective in writing any finite element computer program is its eventual use to find stresses, strains and displacements for problems where an exact mathematical solution does not exist. To this end, the programmer must, at some stage, verify his program, test its accuracy and state its range of applicability. The usual method of verification is to compare results for a few cases where exact solutions or experimental results are available. This method may give good comparisons for the few problems attempted covering a limited range of parameters but it is often questionable whether the program will be best for another range of parameters⁽²³⁾.

Melosh⁽³⁾ identified the possible sources of error associated with approximate solutions as; idealisation, discretisation and manipulation errors. Idealisation errors are those involved in formulating a mathematical model of the physical system. Examples of which are, using flat surfaces to simulate curved ones or using constant thickness elements to represent varying thickness pieces. Discretisation errors are those associated with replacing the continuous structure by one composed of finite elements, in other words selecting a suitable numerical method to solve the mathematical model. Finally, manipulation errors which result

from programming the discretised system and its solution. In this case the mathematical model and its numerical solution are generally well accepted with accuracy bounds and convergence rates defined⁽³⁾. It is, therefore, in programming the solution that the greatest possibility of error arises. If the program is assumed to be corrected for errors in logic and syntax any errors are then caused in the numerical execution. The programmer has little knowledge of the effect of round-off errors or the degree of conditioning of the final stiffness equations. He can only check solutions of standard problems and perhaps resubstitute results to check the equation solving procedure.

Several examples are presented in section 5.3 which provide some numerical verification of the program. It is emphasised, however, that these examples are not as exhaustive as they might be and several questions still linger as to a general application of the program. Firstly, the accuracy has not been examined in regions of high stress gradients and secondly, more attention needs to be given to the calculation of nodal stresses at nodes where several different materials abut. However, the program appears to work well and with regard to the problems discussed above, one can be guided by the experience of other programmers and suppose that similar conclusions to theirs can be drawn^(6, 15, 21).

The effect of order of integration on the accuracy and convergence of the solution is examined in Section 5.3.3.

5.2 PROGRAM FLOWCHART, LISTING AND INPUT DATA

A program listing and flowchart are given for the complete finite element program together with instructions for the user. For brevity, the procedure declarations have been removed from the program listing as these are presented in detail in Chapter 4. The steps involved are described, with reference to the flowchart, as :

A. - Mesh details are read where ;

NELEMT = Number of Elements

NNODE = Number of Nodes

NSETFS = Number of Sets of Forces

PRNT = 1, 2, 3, depending on whether strains and stresses are required.

NSKEW = Number of skewed nodes

NMAT = Number of materials

NSETC = Number of sets of constraints.

Note. The provision of a number of sets of forces and constraints mean that a number of stressing problems can be solved in one program run for a given structure.

The address array is formed and the elasticity matrices are calculated.

B. - Loads are applied to the overall load vector.

C. - Element stiffness matrices and overall stiffness matrix is assembled. Procedure SKEWEDCON is called within procedure FEASSEMBLY to modify the stiffness coefficients for any skewed displacements which may be present.

D. - Procedure SKEWLOAD is called to modify the overall load vector for any skewed displacements.

E. - Kinematic boundary conditions are applied.

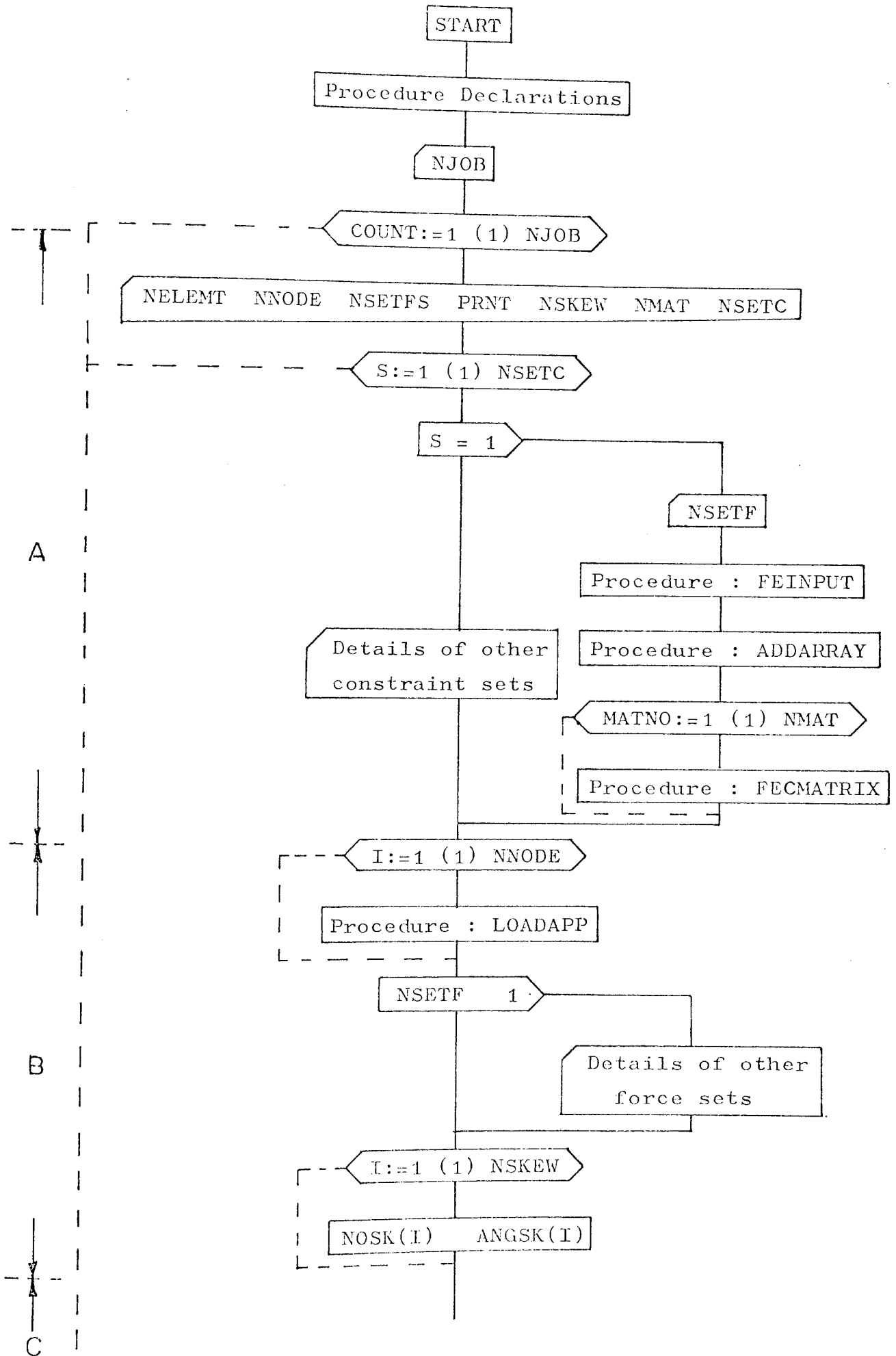
F. - Equilibrium equations are solved for nodal point displacements.

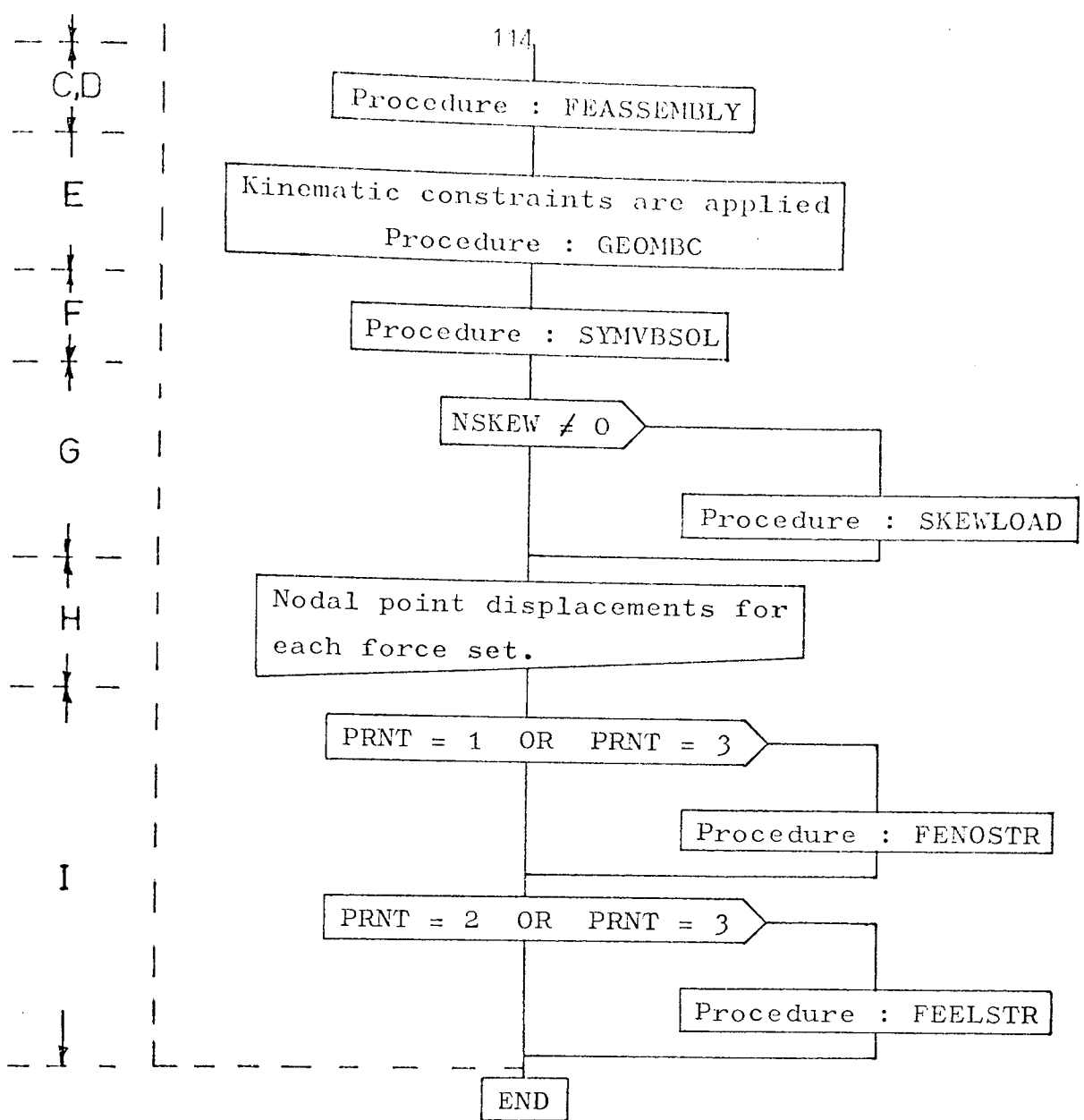
G. - Procedure SKEWLOAD is recalled to convert skewed displacements back to the global coordinate system.

H. - Nodal point displacements are printed.

I. - Element or Nodal point stresses and strains are calculated if required.

Finite Element Program Flowchart.





↑*****

#LISTING OF :EAXXX.ARTHESIS(1/) PRODUCED ON 24JUL75 AT 16.56.37

#OUTPUT ON U.M.R.C.C. 1906A UNIT 176 BY JOB :EAXXX.ARFFP' ON 24JUL75

DOCUMENT ARTHESIS

'BEGIN' 'COMMENT' THIS PROGRAM EMPLOYS THE ISOPARAMETRIC
FORMULATION FOR 2-D PROBLEMS OF PLANE STRESS/STRAIN
THE ELEMENT USED IS A SIX-NODE TRIANGLE;

'COMMENT' CHANGES IN INPUT:-

NODAL CONNECTIONS, MATERIAL NO.

EL. PROPS AND EL. THICKNESS;

'INTEGER' NELEMT, NNODE, NSETF, NREFE, I, J, K, V, W, U, CASE, NSETC, NNENC, S,
NJOB, COUNT, PRNT, NMAT, MATNO, NSKEW;

'REAL' DELTA, DETJ, ANG;

'PROCEDURE' AUX(L1, L2, L3, R, X, Y, U, N, Z);

'PROCEDURE' GEOMBC(U, N, R, AK, NEQ, F, A);

'PROCEDURE' LOADAPP(A, R, C, D, E, F);

'PROCEDURE' SYMVBSC(I, L, S, R) DIMENSIONS: (N, R) FAILURE EXIT: (FAIL);

'PROCEDURE' CMATRIX(Z, CASE, A, MATNO, TH, ANG);

'PROCEDURE' ADDARRAY(NELEMT, NNODE, ADD, NODE);

'PROCEDURE' SKEWEDCON(Z, NODE, ANGSK, NOSK, NSKEW, KE, NSETF);

'PROCEDURE' SKEWLOAD(NSETF, O, NOSK, ANGSK, NSKEW, OP);

'PROCEDURE' FFINPUT(C, ADD, XX, YY, NREFE, NNODE, NSETF, KODE, NSPEC, NSETES,
ULX, VLY, NELEMT, NODE, NMAT);

'PROCEDURE' FFASSEMBLY(NELEMT, K, XX, YY, DETJ, NODE, C, TH, ADD, NREFE,
AUX, SKEWEDCON, NSKEW, NSETF, ANGSK, NOSK);

'PROCEDURE' FFNOSTR(NSETF, NNODE, NELEMT, NODE, XX, YY, DETJ, O, TH, C, AUX);

'PROCEDURE' FFELSTR(NSETF, O, NELEMT, NODE, XX, YY, DETJ, TH, C, AUX);

NJOB:=READ;

'FOR' COUNT:=1 'STEP' 1 'UNTIL' NJOB 'DO'

'BEGIN'

WRITETEXT('('('20')'JOB#JANEX-----')');

```

COPYTEXT('('END%OF%TITLE')');
NELEMT:=READ;
NNODE:=READ;
WRITETEXT('('('20')'NO%OF%ELEMENTS%---')');
PRINT(NELEMT,3,0);
WRITETEXT('('('20')'NO%OF%NODES%-----')');
PRINT(NNODE,3,0);
NREFE:=NNODE*2;
NSETFS:=READ;
PRNT:=READ; NSKEW:=READ;
NMAT:=READ;
'BEGIN'
'INTEGER' BAND,NSPEC,7,COMPA,NSETF;
'REAL' 'ARRAY' XX,YY[1:NNODE],ULX,VLY[1:NNODE,1:NSETFS],
      Q[1:NREFE,1:NSETFS],C[1:NMAT,1:6],A[1:5],TH[1:NMAT],
ANGSK[1:NSKEW+1];
'INTEGER' 'ARRAY' NODE[1:NELEMT,1:7],KODE[1:NNODE,1:NSETFS],ADD[0:NREFE],
NOSK[1:NSKEW+1];
NSETC:=READ;
'FOR' S:=1 'STEP' 1 'UNTIL' NSETC 'DO'
'BEGIN'
'IF' S=1 'THEN' 'BEGIN'
NSETF:=READ;
FEINPUT(C,ADD,XX,YY,NREFE,NNODE,NSETF,KODE,NSPEC,NSETFS,
      ULX,VLY,NELEMT,NODE,NMAT);
ADDARRAY(NELEMT,NNODE,ADD,NODE);
'FOR' MATNO:=1 'STEP' 1 'UNTIL' NMAT 'DO'
CMATRIX(C,CASE,A,MATNO,TH,ANG);
'END' 'ELSE'
'BEGIN' NNEWC:=READ;
      NSETF:=READ;
WRITETEXT('('('100')'CONSTRAINT%SET---')');
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('('('20')'NODAL%POINT%DATA'('204S')'NONE'('5S')'X%COORD
('5S')'Y%COORD'('5S')'TYPE'('5S')'X-DISP'('5S')'Y-DISP'('046S')'
OR%LOAD'('4S')'OR%LOAD')');
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'BEGIN' NELLINE(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';
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' NREFE 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
      LOADAPP(KODE[I,1],ULX[I,1],VLY[I,1],C,1,1);
'IF' NSETF 'GT' 1 'THEN'
'BEGIN'
'FOR' I:=2 'STEP' 1 'UNTIL' NSETF 'DO'
'BEGIN'
NSPEC:=READ;
WRITETEXT('('('20')'FORCE%SET---')');
PRINT(I,3,0);
WRITETEXT('('('20')'NODE'('5S')'TYPE'('8S')'X-DISP'('7S')'Y-DISP
('021S')'OR%LOAD'('6S')'OR%LOAD')');

```

```

'FOR' J:=1 'STEP' 1 'UNTIL' NSPEC 'DO'
'BEGIN'
  K:=READ; KODE[K,1]:=READ; ULY[K,1]:=READ; VLY[K,1]:=READ;
  LOADAPP(KODE[,1],ULX[K,1],VLY[K,1],0,K,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';
'FOR' I:=1 'STEP' 1 'UNTIL' NSKEW 'DO'
'BEGIN'
  NOSK[I]:=READ;
  ANGSK[I]:=READ;
'END';
'BEGIN' 'REAL' 'ARRAY' K[1:ADD[HEREF]1;
FEASSEMBLY(ELEMT,K,XX,YY,DETJ,MODE,C,TH,ADD,HEREF,AUX,SKWEDCON,
  NSKEW,NSETF,ANGSK,NOSK);
'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;
  GEOMBC(ULY[I,1],2*1-1,0,K,HEREF,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: GEOMBC(VLY[I,1],2*1,0,K,HEREF,1,ADD);
  'FOR' J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]:=Q[I,1];
  KC1: 'END';
  WRITETEXT('(('('4C')'SYMBVSOL%$BEGINS'))');
  SYMBVSOL(K,K,ADD,0,HEREF,NSETF,FAIL);
  'IF' NSKEW.NF.0 'THEN' SKEWLOAD(NSETF,Q,LOSK,ANGSK,NSKEW,-1);
  'FOR' I:=1 'STEP' 1 'UNTIL' NSETF 'DO'
  'BEGIN' WRITETEXT('(('('4C')'NODAL%$DISPLACEMENTS%FOR% FORCE%SET'))');
    PRINT(I,2,0);
    WRITETEXT('(('('3CS')'NODE'('5S')'Y-DIRECTION'('8S')'Y-DIRECT
      JCN'('1KS')'NODE'('5S')'X-DIRECTION'('8S')'Y-DIRECTION'))');
    W:=2*(NNODE/'2);
  'FOR' J:=2 'STEP' 2 'UNTIL' W 'DO'
    'BEGIN' NEWLINE(2); W:=2*(J-1);
    PRINT((J-1),3,0); SPACE(2); PRINT(Q[V-1,1],0,8); SPACE(2);
    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'
FEOSTR(NSETF,NNODE,ELEMT,MODE,XX,YY,DETJ,0,TH,C,AUX);
'IF' PRNT=2 'OR' PRNT=3 'THEN'
FEELSTR(NSETF,Q,ELEMT,MODE,XX,YY,DETJ,TH,C,AUX);
'END';
'END' OF CONSTRAINT LOOP;
'END';
FAIL: 'END' OF JOB LOOP;
'END' OF PROGRAM;
***

```

DATA INPUT INSTRUCTIONS

- (A) Number of Jobs to be Analysed.
 - (B) for first job
 - (1) JOB TITLE, followed by END OF TITLE
 - (2) NUMBER OF ELEMENTS
 - (3) NUMBER OF NODES
 - (4) NUMBER OF SETS OF FORCES^{(1)*}
 - (5) STRESSES AND STRAINS⁽²⁾
 - (6) NUMBER OF NODES WHERE SKEWED BOUNDARY CONDITIONS ARE APPLIED
 - (7) NUMBER OF MATERIALS⁽³⁾
 - (8) NUMBER OF SETS OF CONSTRAINTS.
 - (C) for first set of constraints
 - (1) NUMBER OF SETS OF FORCES FOR THIS SET OF CONSTRAINTS
 - (2) NUMBER OF NODAL POINTS WHOSE COORDINATES ARE SPECIFIED⁽⁴⁾
 - (3) for the number of points specified in C(2), input
 - (a) NODE NUMBER
 - (b) X-COORDINATE
 - (c) Y-COORDINATE
 - (d) KODE⁽⁵⁾ (prescribed load or displacement)
 - (e) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN X-DIRECTION
 - (f) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN Y-DIRECTION
 - (4) NODAL CONNECTIONS AND MATERIAL NUMBER ARE READ FOR EACH ELEMENT⁽⁶⁾
 - (5) for elements with material number 1, input
 - (a) TYPE OF PROBLEM (0, plane stress - 1, plane strain)
 - (b) ANGLE OF ORTHOTROPY (measured clockwise from + ve x-axis)
 - (c) THICKNESS
 - (d) ELASTIC CONSTANTS⁽⁷⁾
- (repeat 5(a) - (d) for material numbers 2, 3, etc.)

- (6) If the number of sets of forces for this set of constraints is greater than 1, then input 6(a) - (e), else go to (D)
- (a) NUMBER OF SPECIFIED FORCES
for the number specified in 6(a), input
 - (b) NODE NUMBER
 - (c) KODE
 - (d) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN X-DIRECTION
 - (e) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN Y-DIRECTION
- (7) for the number of skewed nodes, B(6), input
- (a) NODE NUMBER
 - (b) ANGLE OF SKEW⁽⁸⁾
- (D) for second set of constraints
- (1) NUMBER OF NEW CONSTRAINTS⁽⁹⁾
 - (2) NUMBER OF SETS OF FORCES IN THIS SET OF CONSTRAINTS
 - (3) for the number of new constraints, which must include the first set of forces for this set of constraints, input
 - (a) NODE NUMBER
 - (b) KODE
 - (c) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN X-DIRECTION
 - (d) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN Y-DIRECTION
 - (4) If D(2) is greater than 1, then repeat C(6) (a-e)
 - (5) REPEAT C(7) (a-b)
- (E) For subsequent sets of constraints repeat from (D).
- (F) For next Job repeat from (B).

* Superscripts refer to notes which follow the Data Input Instructions

NOTES

- (1) The number of sets of forces must be the maximum number for any set of constraints as this number fixes array bounds.
- (2) Put, 1 if Nodal Stresses are required
2 if Element stresses are required
3 if both Nodal and Element Stresses are required
- (3) A change in the angle of orthotropy from element to element is treated as a new material in each case.
- (4) Most elements are straight sided and in these cases it is necessary to specify the coordinates of the vertices only.
The program will automatically interpolate to calculate the coordinates of the missing nodes. If there is some prescribed load or displacement at a midside node then the coordinate of that node must be specified.
- (5) If $KODE =$
0, this represents prescribed loads in the X and Y direction
1, this represents a prescribed displacement in X and load in Y direction
2, this represents a prescribed load in X and displacement in Y direction
3, this represents prescribed displacements in X and Y directions.

Note. If a node is constrained in a direction other than the global then the KODE number refers to the loads/displacements for the skewed axes.

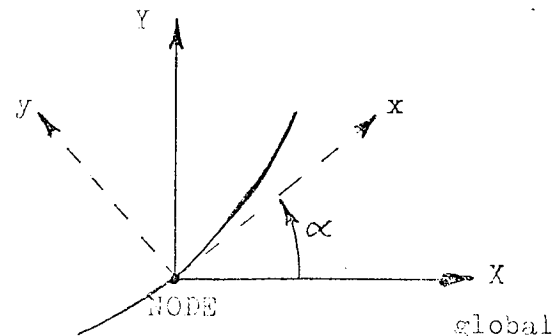
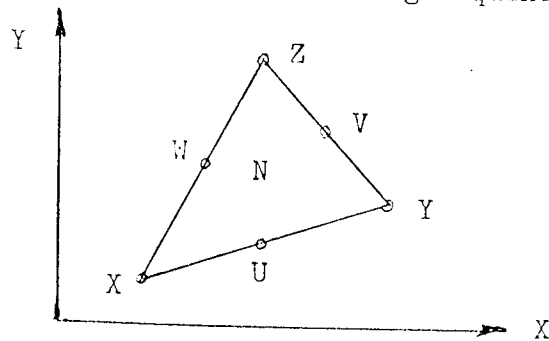


Fig. (1)

- (6) The nodal connections and material number for each element must be specified using the following sequence :



If N is the element number, M is the material number, and X, Y, Z, U, V, W are the node numbers, then input N X Y Z U V W M for each element.

- (7) If the material is

(a) isotropic

$$E \quad \nu \quad G \quad E \quad \nu \quad \text{where } G = \frac{E}{2(1+\nu)}$$

(b) transversely isotropic

$$E_x \quad \nu_{xy} \quad G_{xy} \quad E_{xy} \quad \nu_{yz}$$

- (8) Specified in degrees and taken in a clockwise direction from the global X-axis, See Fig. (1).

- (9) If any constraints have been applied previously, then they must be removed by including them in C(1).

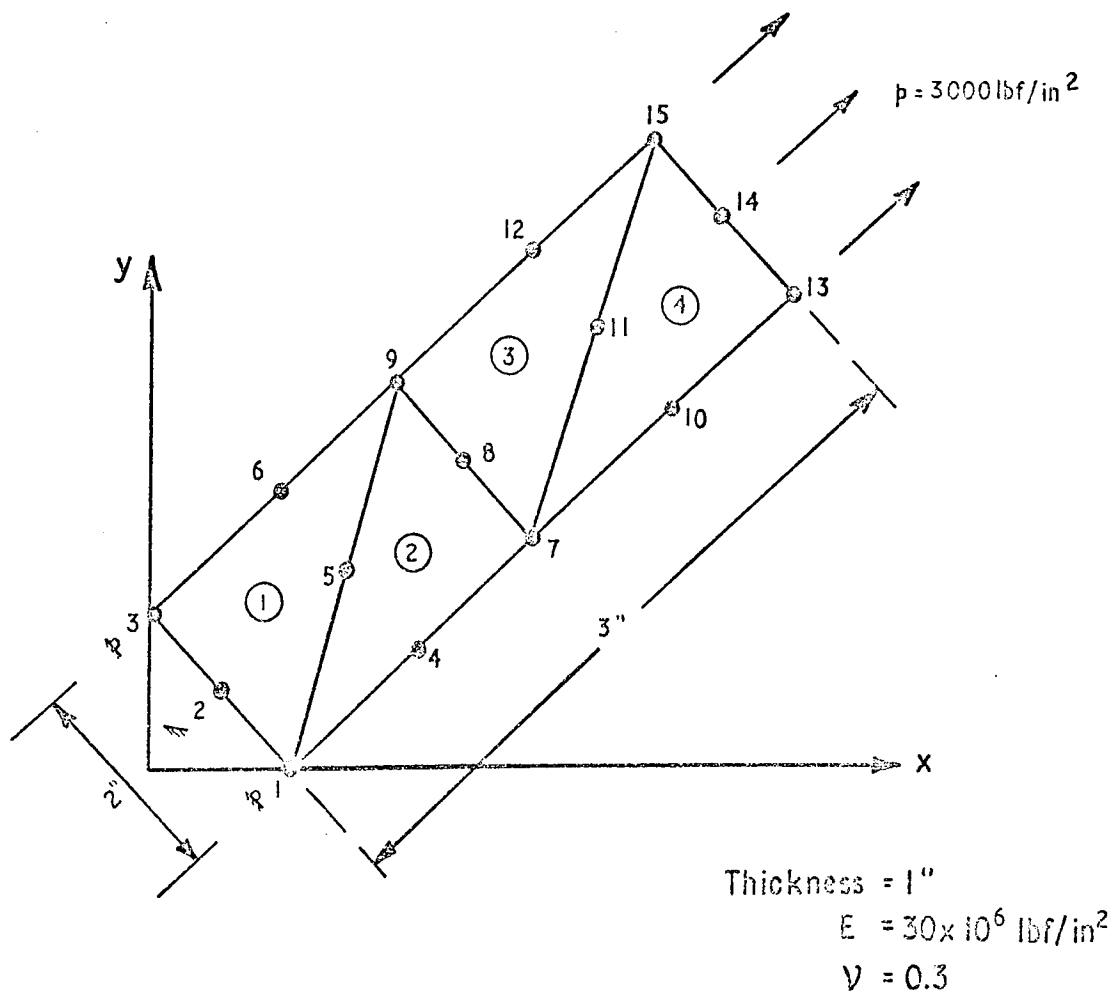
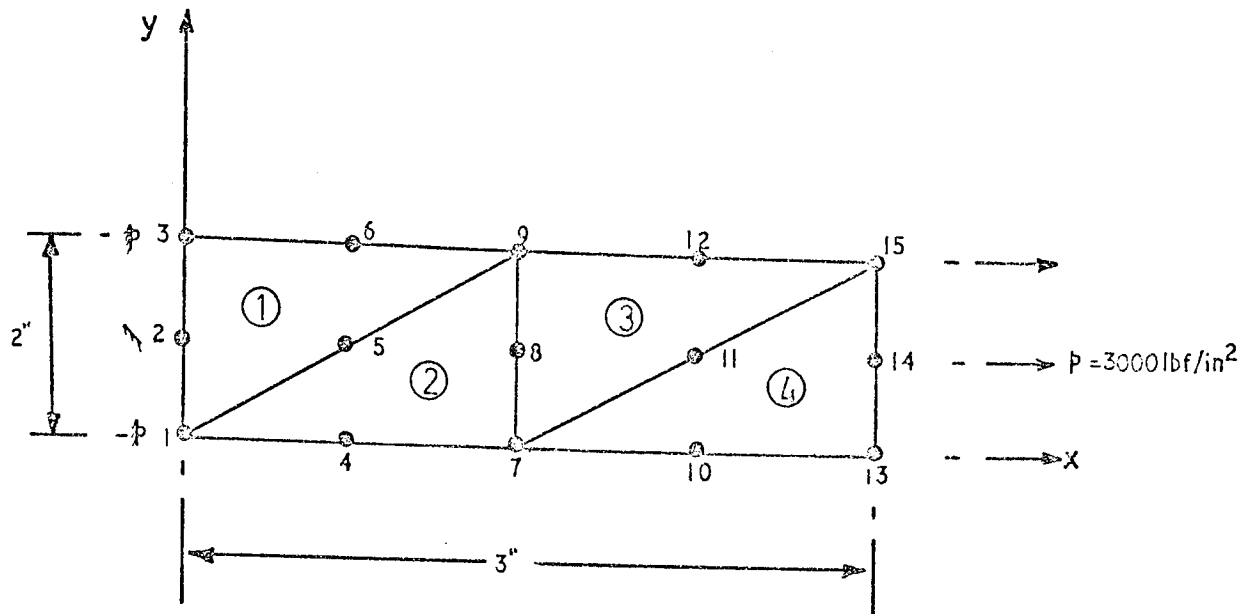


FIG.5.1 SAMPLE PROBLEMS TO DEMONSTRATE INPUT DATA.

SAMPLE PROBLEMS

Consider the two problems shown in Fig. 5.1, the data input instructions necessary for running these two cases are :

2

UNSKWEVED END OF TITLE

4	15	1	3	0	1	1	1	1	8
1	0	0	1	0	0				
2	0	1	3	0	0				
3	0	2	1	0	0				
7	1.5	0	0	0	0				
9	1.5	2	0	0	0				
13	3	0	0	1000	0				
14	3	1	0	4000	0				
15	3	2	0	1000	0				
1	1	9	3	5	6	2		1	
2	1	7	9	4	8	5		1	
3	7	15	9	11	12	8		1	
4	7	13	15	10	14	11		1	
1	0	1	30000000		0.3	12000000	30000000		0.3

SKEWED END OF TITLE

4	15	1	3	6	1	1	1	8
1	1.4142	0	2	0	0			
2	0.7071	0.7071	3	0	0			
3	0	1.4142	2	0	0			
7	2.4749	1.0607	0	0	0			
9	1.0607	2.4749	0	0	0			
13	3.5355	2.1213	0	1000	0			
14	2.8284	2.8284	0	4000	0			
15	2.1213	3.5355	0	1000	0			

1	3	1	9	2	5	6	1
2	1	7	9	4	8	5	1
3	9	7	15	8	11	12	1
4	7	13	15	10	14	11	1
1	0	1	300000000	0.3	120000000	300000000	0.3

1	135
---	-----

2	135
---	-----

3	135
---	-----

13	45
----	----

14	45
----	----

15	45
----	----

JOB NAME -----UNSKewed

NO OF ELEMENTS ---- 4

NO OF NODES ----- 15

NODAL CONNECTIONS HAVE BEEN READ

NODAL POINT DATA

NODE	X COORD	Y COORD	TYPE	X-DISP OR LOAD	Y-DISP OR LOAD
1	0.0000 00	0.0000 00	1	0.0000 00	0.0000 00
2	0.0000 00	1.0000 00	3	0.0000 00	0.0000 00
3	0.0000 00	2.0000 00	1	0.0000 00	0.0000 00
4	2.5000-01	0.0000 00	0	0.0000 00	0.0000 00
5	2.5000-01	1.0000 00	0	0.0000 00	0.0000 00
6	2.5000-01	2.0000 00	0	0.0000 00	0.0000 00
7	1.5000 00	0.0000 00	0	0.0000 00	0.0000 00
8	1.5000 00	1.0000 00	0	0.0000 00	0.0000 00
9	1.5000 00	2.0000 00	0	0.0000 00	0.0000 00
10	2.2500 00	0.0000 00	0	0.0000 00	0.0000 00
11	2.2500 00	1.0000 00	0	0.0000 00	0.0000 00
12	2.2500 00	2.0000 00	0	0.0000 00	0.0000 00
13	3.0000 00	0.0000 00	0	1.0000 03	0.0000 00
14	3.0000 00	1.0000 00	0	4.0000 03	0.0000 00
15	3.0000 00	2.0000 00	0	1.0000 03	0.0000 00

ELEMENT DATA

ELEMENT	NODAL CONNECTIONS	MATERIAL
1	9 3 5	1
2	1 7 4	1
3	7 15 11	1
4	7 13 15 10	1

MATERIAL PROPERTIES:=
 (NOTE,IF CASE=0,THEN PLANE STRESS ELSE
 THE PROBLEM IS ONE OF PLANE STRAIN.)

MATERIAL NUMBER	ANGLE OF ANISOTROPY	CASE	THICKNESS	FXX	ELASTIC EYY	PROPERTIES VX	VY	GXY
1	0.008 00	1	1.00008 00	3.0008 07	3.0008 07	0.30	0.30	1.2008 07

SYNVSOL REGINS

NODAL DISPLACEMENTS FOR FORCE SET 1

NODE	X-DIRECTION	Y-DIRECTION	NODE	X-DIRECTION	Y-DIRECTION
1	0.00000008 00	3.900000078-05	2	0.000000008 00	0.000000008 00
3	0.000000008 00	-3.9000000078-05	4	6.825000128-05	3.9000000078-05
5	6.825000128-05	-6.061438148-15	6	6.825000138-05	-3.9000000078-05
7	1.365000038-04	3.9000000068-05	8	1.365000028-04	-3.703169668-15
9	1.365000038-04	-3.9000000078-05	10	2.047500048-04	3.9000000068-05
11	2.047500048-04	-2.994896098-14	12	2.047500048-04	-3.9000000088-05
13	2.730000068-04	3.900000108-05	14	2.730000058-04	-1.577980448-13
15	2.730000068-04	-3.900000008-05			

NODE	STRAINS			FX	FXY	STRESSES		
	FXX	FYY	SIGMA-X			SIGMA-Y	SIGMA-XY	
1	9.10008-05	-3.90002-05	-4.00688-14	3.00008 03	4.15802-07	-4.80828-07		
2	9.10008-05	-3.90002-05	-2.74458-14	3.00008 03	2.01978-07	-3.29348-07		
3	9.10008-05	-3.90002-05	-1.77548-14	3.00008 03	-2.84058-07	-2.13058-07		
4	9.10008-05	-3.90002-05	-2.02348-13	3.00008 03	3.72728-06	-2.42808-06		
5	9.10008-05	-3.90002-05	9.48568-14	3.00008 03	-1.26348-07	1.13838-06		
6	9.10008-05	-3.90002-05	6.75038-14	3.00008 03	4.30038-07	8.10038-07		
7	9.10008-05	-3.90002-05	-3.74588-13	3.00008 03	6.57948-06	-4.49508-06		
8	9.10008-05	-3.90002-05	1.01888-14	3.00008 03	1.37308-06	1.22268-07		
9	9.10008-05	-3.90002-05	2.06068-13	3.00008 03	4.18518-07	2.47278-06		
10	9.10008-05	-3.90002-05	-8.45478-13	3.00008 03	-7.64998-07	-1.01468-05		
11	9.10008-05	-3.90002-05	2.08318-13	3.00008 03	1.68898-06	2.40078-06		
12	9.10008-05	-3.90002-05	3.33228-13	3.00008 03	2.52308-06	3.99868-06		
13	9.10008-05	-3.90002-05	-9.91848-13	3.00008 03	-1.58708-05	-1.19028-05		
14	9.10008-05	-3.90002-05	4.78158-14	3.00008 03	-1.19978-05	5.73788-07		
15	9.10008-05	-3.90002-05	7.97658-13	3.00008 03	-2.83598-06	9.57188-06		

ELET	STRAINS			FXX	FXY	STRESSES		
	FXX	FYY	SIGMA-X			SIGMA-Y	SIGMA-XY	
1	9.09828-05	-3.89928-05	3.26178-14	2.99948 03	5.21828-07	3.91418-07		
2	9.09828-05	-3.89928-05	-3.26188-14	2.99948 03	1.65138-06	-3.91418-07		
3	9.09828-05	-3.89928-05	2.01148-13	2.99948 03	1.04378-06	2.41378-06		
4	9.09828-05	-3.89928-05	-2.01178-13	2.99948 03	-3.21768-06	-2.41408-06		

JOB NAME -----
 SAVED

NO OF ELEMENTS ---- 4

NO OF NODES ----- 15

NODAL CONNECTIONS HAVE BEEN READ

NODAL POINT DATA

NODE	X COORD	Y COORD	TYPE	X-DISP OR LOAD	Y-DISP OR LOAD
1	1.4148 00	0.0008 00	2	0.0008 00	0.0008 00
2	7.0718-01	7.0718-01	3	0.0008 00	0.0008 00
3	0.0008 00	1.4148 00	2	0.0008 00	0.0008 00
4	1.0458 00	5.3048-01	0	0.0008 00	0.0008 00
5	1.2378 00	1.2378 00	0	0.0008 00	0.0008 00
6	5.3048-01	1.0458 00	0	0.0008 00	0.0008 00
7	2.4758 00	1.0618 00	0	0.0008 00	0.0008 00
8	1.7688 00	1.7688 00	0	0.0008 00	0.0008 00
9	1.0618 00	2.4758 00	0	0.0008 00	0.0008 00
10	3.0058 00	1.5018 00	0	0.0008 00	0.0008 00
11	2.2988 00	2.2988 00	0	0.0008 00	0.0008 00
12	1.5918 00	3.0058 00	0	0.0008 00	0.0008 00
13	3.5368 00	2.1218 00	0	1.0008 03	0.0008 00
14	2.8288 00	2.8288 00	0	4.0008 03	0.0008 00
15	2.1218 00	3.5368 00	0	1.0008 03	0.0008 00

ELEMENT DATA

ELEMENT	3	1	9	2	5	6	MATERIAL
1							1
2	1	7	9	4	8	5	1
3	9	7	15	8	11	12	1
4	7	13	15	10	14	11	1

MATERIAL PROPERTIES: =

(NOTE, IF CASE=0, THEN PLANE STRESS ELSE

THE PROGRAM IS ONE OF PLANE STRAIN.)

MATERIAL NUMBER	ANGLE OF ANISOTROPY	CASE	THICKNESS	FXX	ELASTIC EYV	PROPERTIES VX	VY	GXY
1	0.008 00	1	1.00008 00	3.0008 07	3.0008 07	0.30	0.30	1.2008 07

SYNV8SOL BEGINS

NODAL DISPLACEMENTS FOR FORCE SET 1

NODE	X-DIRECTION	Y-DIRECTION	NODE	X-DIRECTION	Y-DIRECTION
1	-2.580942618-05	2.580937408-05	2	0.000000008 00	0.000000008 00
3	2.580941048-05	-2.580935828-05	4	2.112702658-05	7.274600198-05
5	4.693636138-05	4.693648308-05	6	7.274565548-05	2.112703268-05
7	6.806340328-05	1.196826568-04	8	9.387263928-05	9.387306238-05
9	1.196818428-04	6.806351338-05	10	1.14952528-04	1.666149008-04
11	1.408044418-04	1.408052678-04	12	1.666136048-04	1.149956508-04
13	1.610270628-04	2.135471408-04	14	1.877362168-04	1.877374958-04
15	2.135453488-04	1.619278558-04			

NODE	EXX	STRANS EYX	FXV	SIGMA-X	STRESSFS SIGMA-Y	SIGMA-XY
1	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
2	2.60008-05	2.60008-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
3	2.60003-05	2.60008-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
4	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
5	2.60008-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
6	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
7	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
8	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
9	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
10	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
11	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
12	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
13	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
14	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03
15	2.60002-05	2.60002-05	1.25008-04	1.50008 03	1.50008 03	1.50008 03

FLIT	EXX	STRANS EYX	FXV	SIGMA-X	STRESSES SIGMA-Y	SIGMA-XY
1	2.50958-05	2.50958-05	1.24988-04	1.49978 03	1.49978 03	1.49978 03
2	2.50958-05	2.50958-05	1.24988-04	1.49978 03	1.49978 03	1.49978 03
3	2.50958-05	2.50958-05	1.24988-04	1.49978 03	1.49978 03	1.49978 03
4	2.50958-05	2.50958-05	1.24988-04	1.49978 03	1.49978 03	1.49978 03

5.3 NUMERICAL EXAMPLES

The following examples are included to demonstrate the accuracy of the finite element program for several two-dimensional problems of plane stress or plane strain.

5.3.1 Circular Hole in a Tension Plate

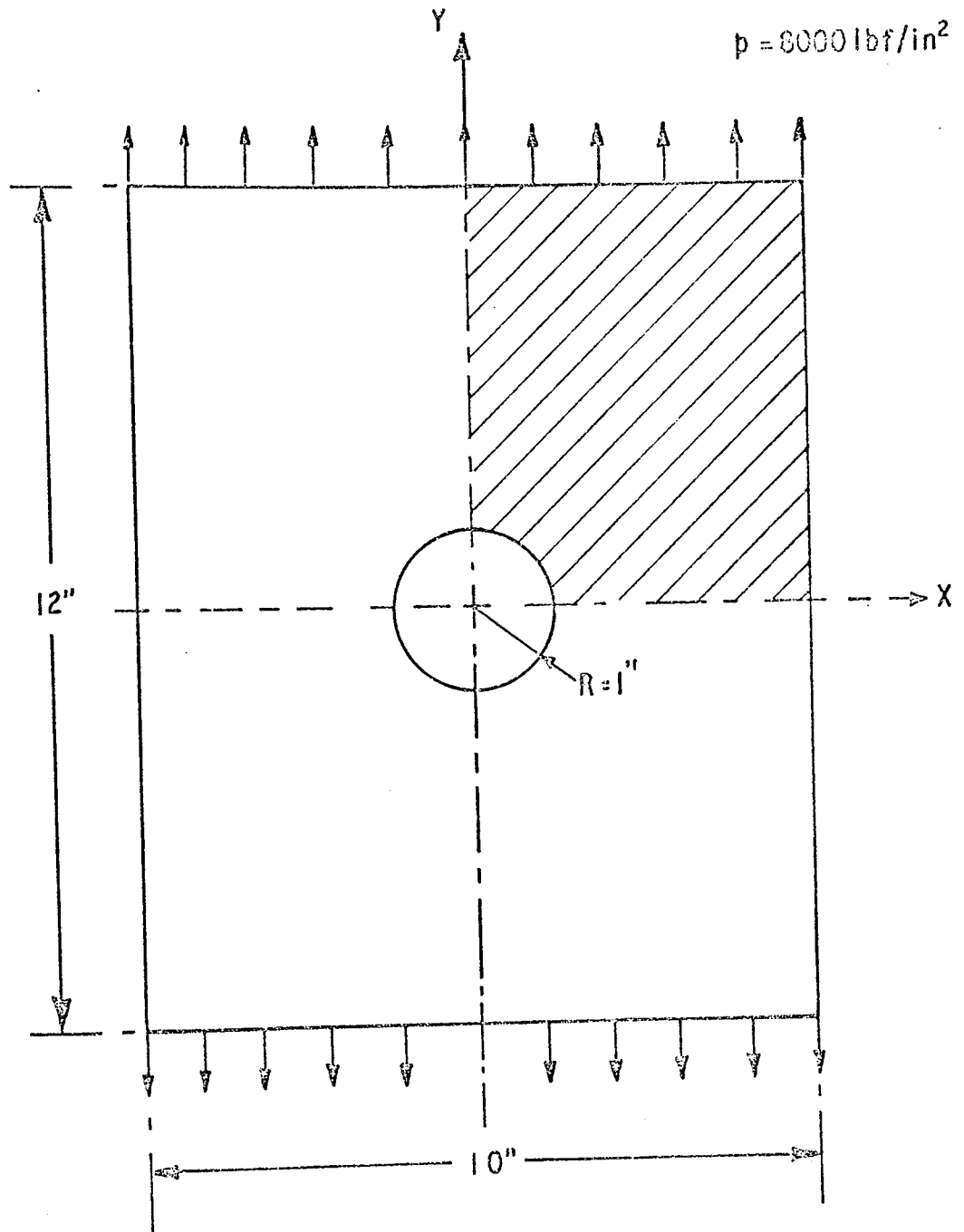
This example is included to test the performance of the curved element for a problem of stress concentration due to geometric irregularity. The plate and loading system are shown in Fig. 5.2. Taking advantage of symmetry, it was only necessary to analyse one quadrant of the plate. The finite element idealisations are shown in Fig. 5.3. The problem was solved using elements with both straight and curved sides on the hole boundary.

Graphs of σ_y/p at $y = 0$ for the two meshes are compared with Howland's analytic solution⁽²⁴⁾ for a plate of finite width in Fig. 5.4. In each case a three point Gauss rule was used for the integration. Results show good convergence as the mesh is refined, the second mesh giving acceptable accuracy for most practical purposes. Fig. 5.5 shows a graph of nodal point displacements in the y -direction at $x = 0$. The curves show clearly the benefit of the curved element boundaries permitted by the isoparametric formulation.

5.3.2 Circular Hole in Rectangular Plate, Material Anisotropy

This example was chosen to check the program facility for anisotropic materials and is taken from reference (19). Results were compared with the analytical solutions presented by Savin⁽²⁵⁾. The geometry and material properties of the plate are shown in Fig. 5.6 and the mesh for the quadrant analysed contained 29 elements with 74 nodes.

The nodal stresses obtained from the program are shown in Figs. 5.7(a) and 5.7 (b) for the isotropic and orthotropic cases respectively. Results show reasonably good agreement even for this relatively crude idealisation.

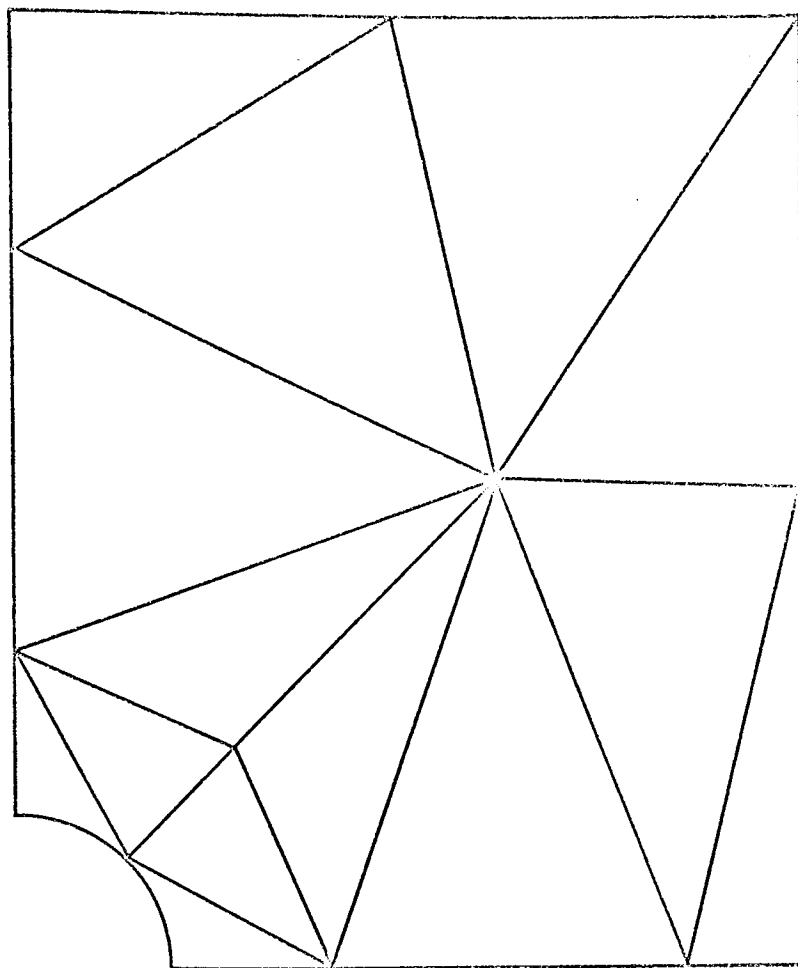


$$E = 30 \times 10^6 \text{ lbf/in}^2$$

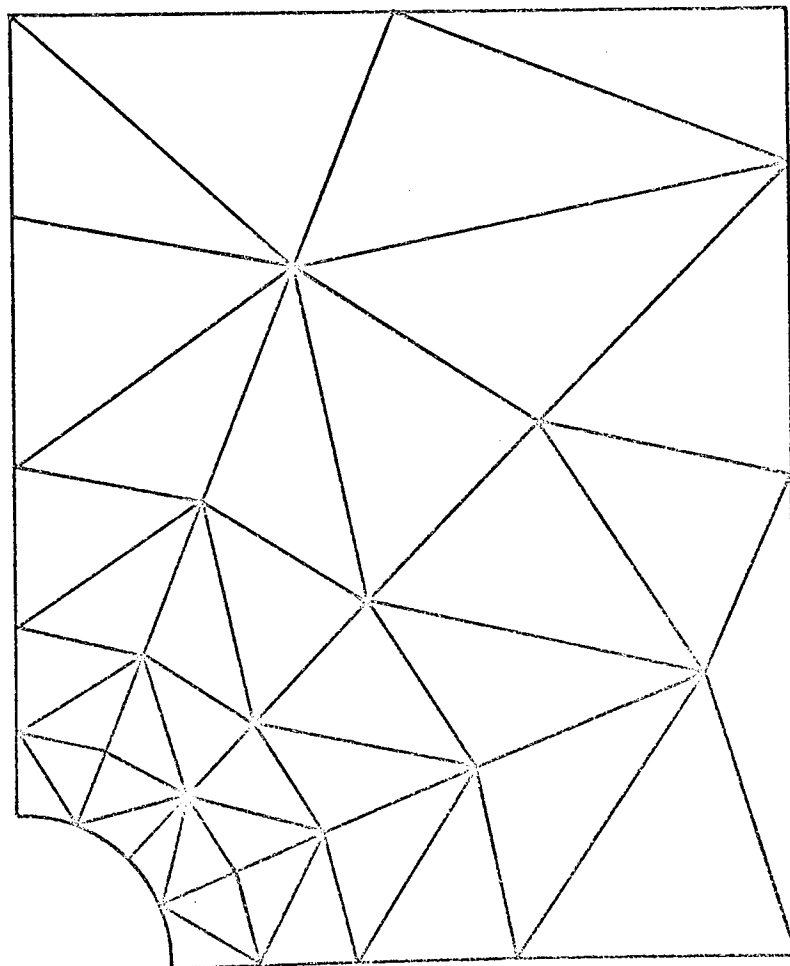
$$\nu = 0.25$$

$$\text{Thickness} = 1"$$

FIG 5.2 CIRCULAR HOLE IN A TENSION PLATE.



MESH No.1.
14 ELEMENTS.
41 NODES.



MESH No.2.
40 ELEMENTS
99 NODES

FIG 5.3 PLATE WITH CIRCULAR HOLE-FINITE ELEMENT IDEALISATION .

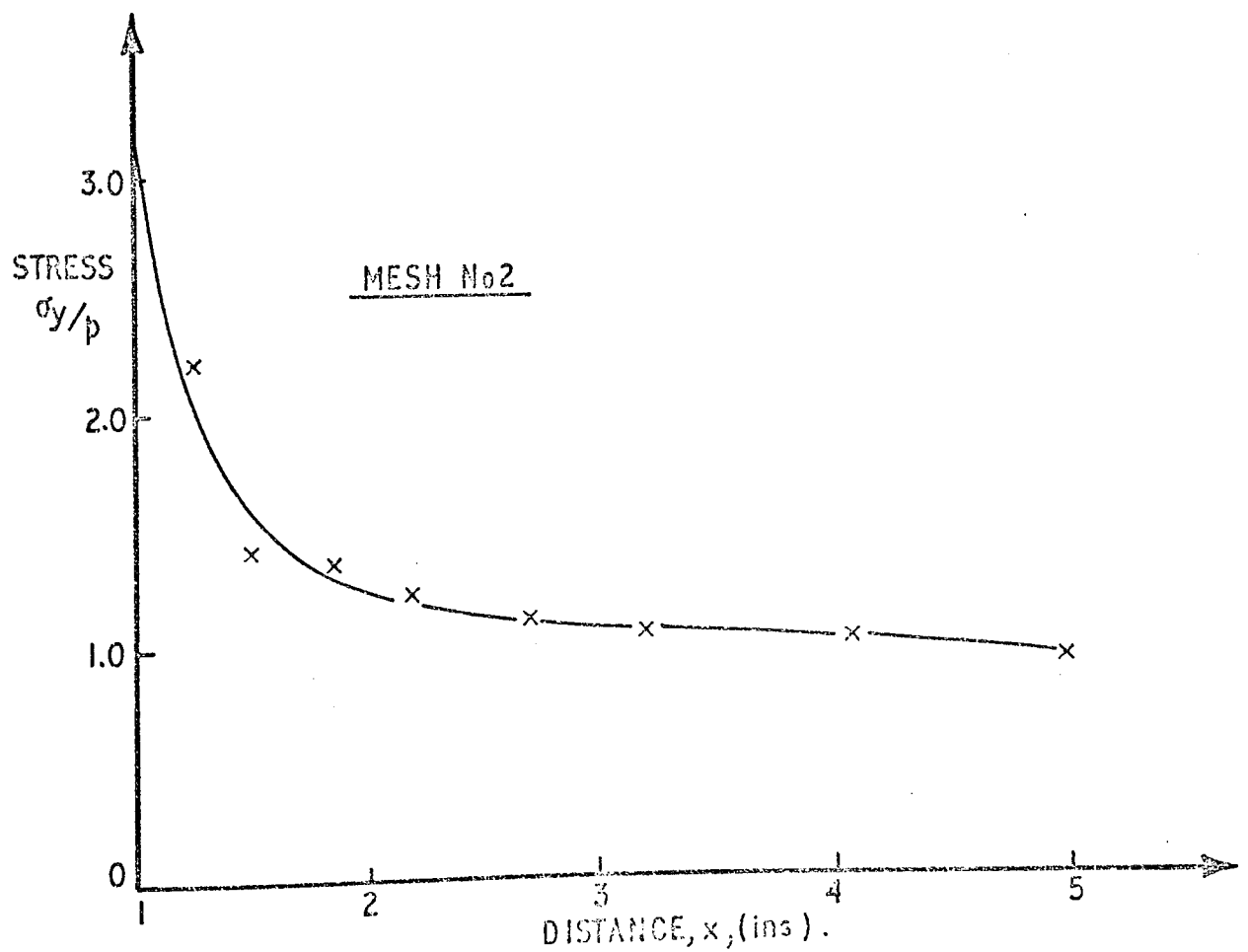
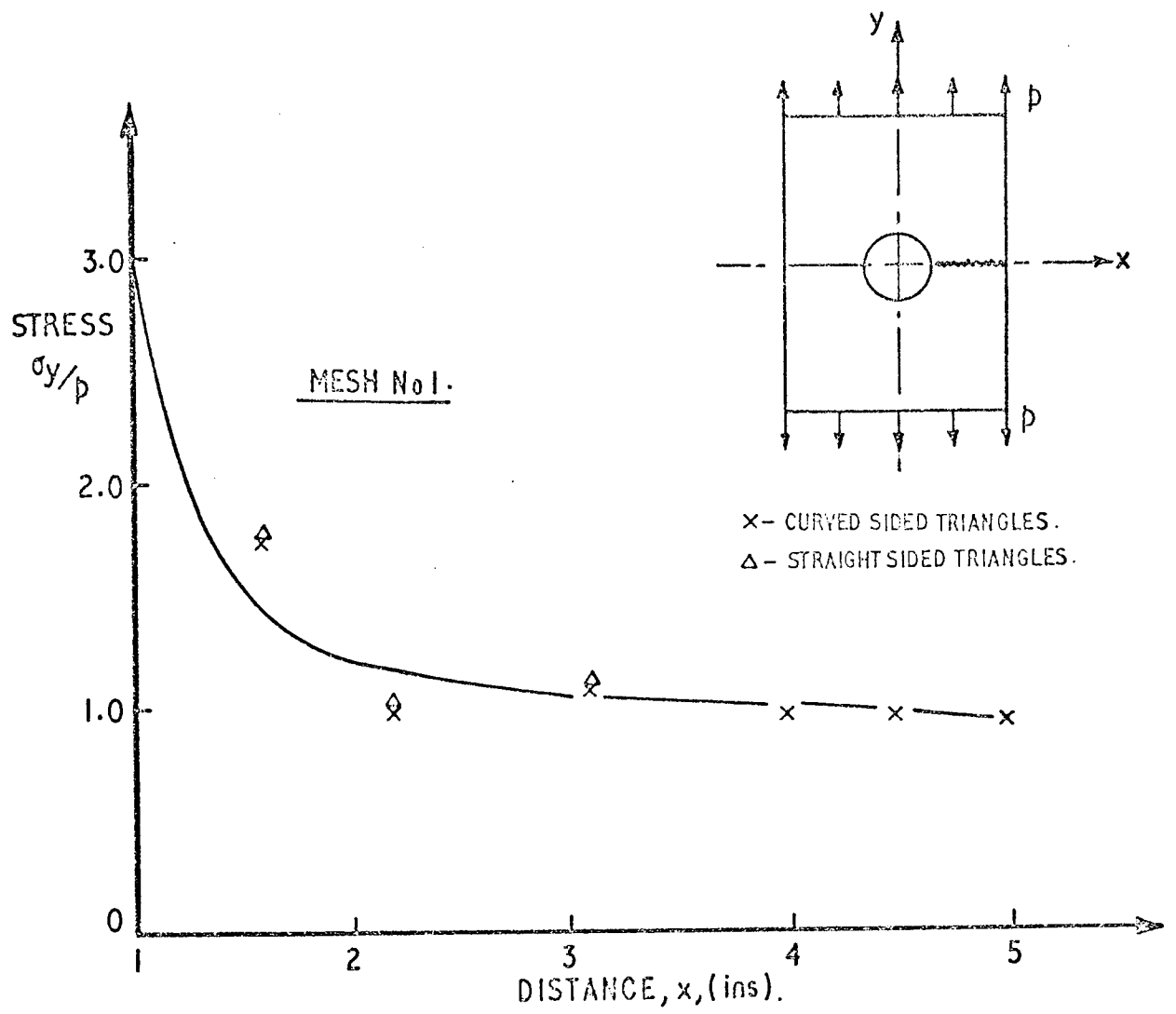


FIG 5.4. PLOTS OF σ_y/p AT THE SECTION $Y=0$.

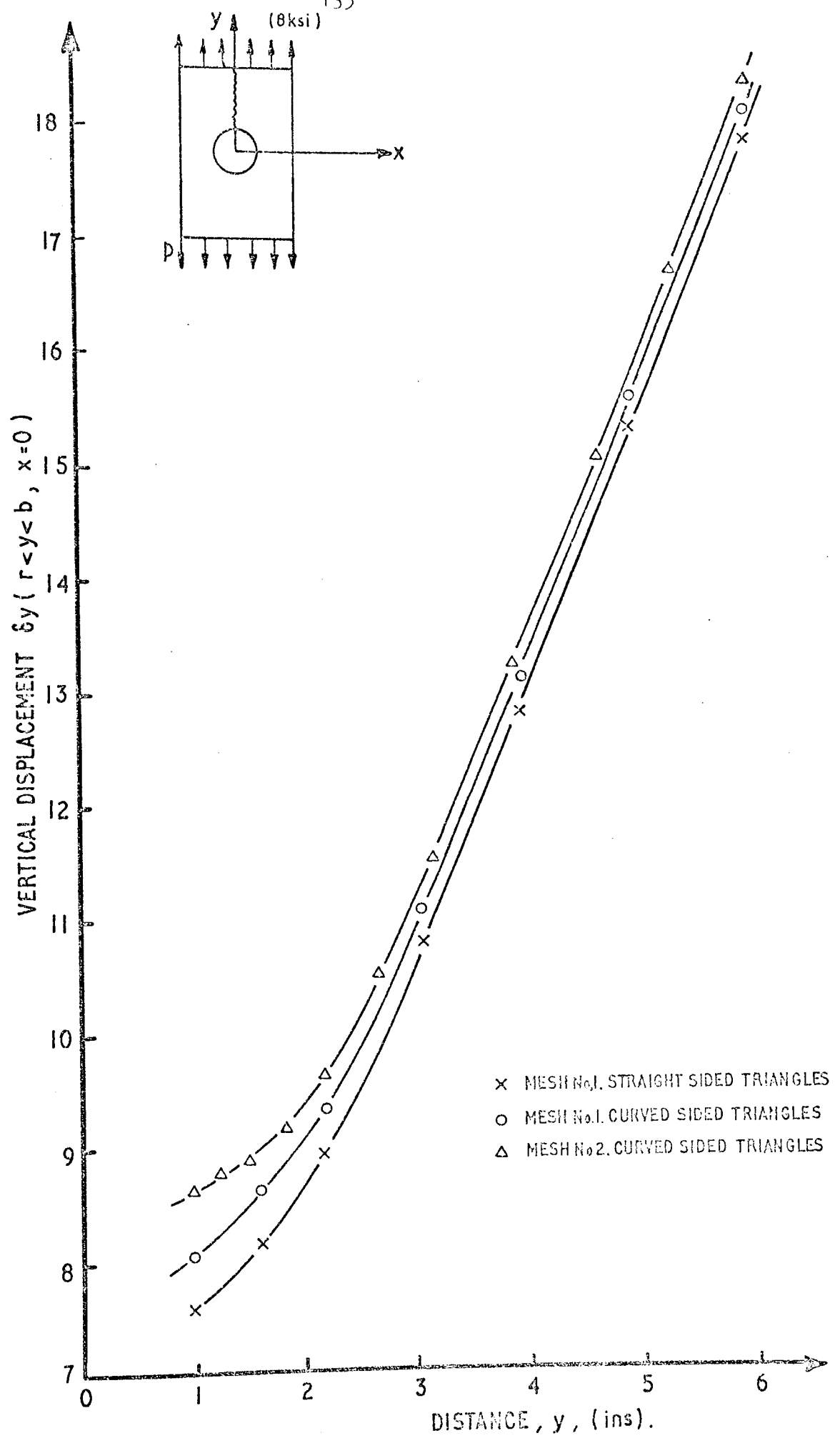
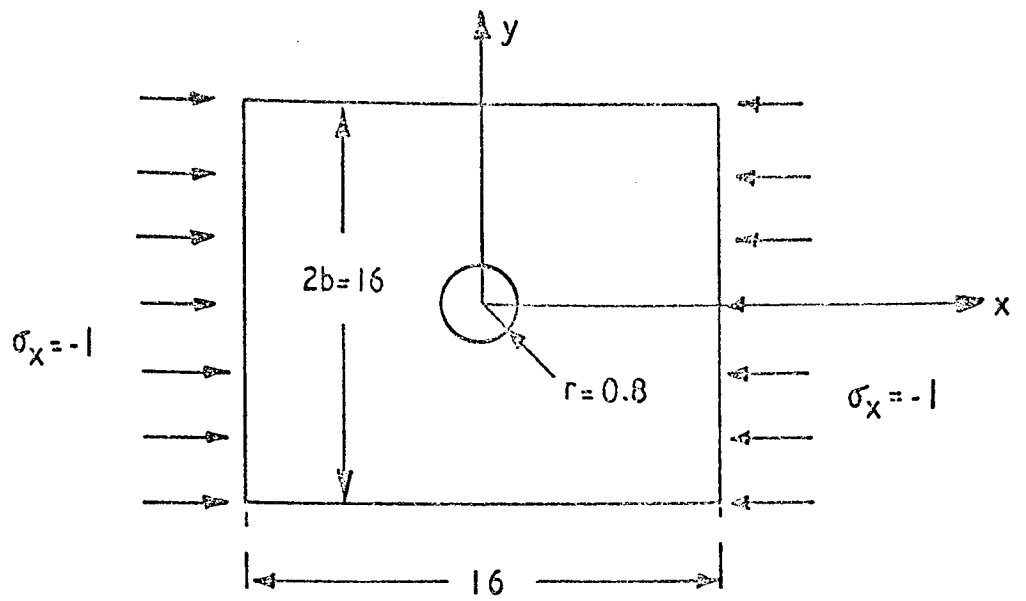
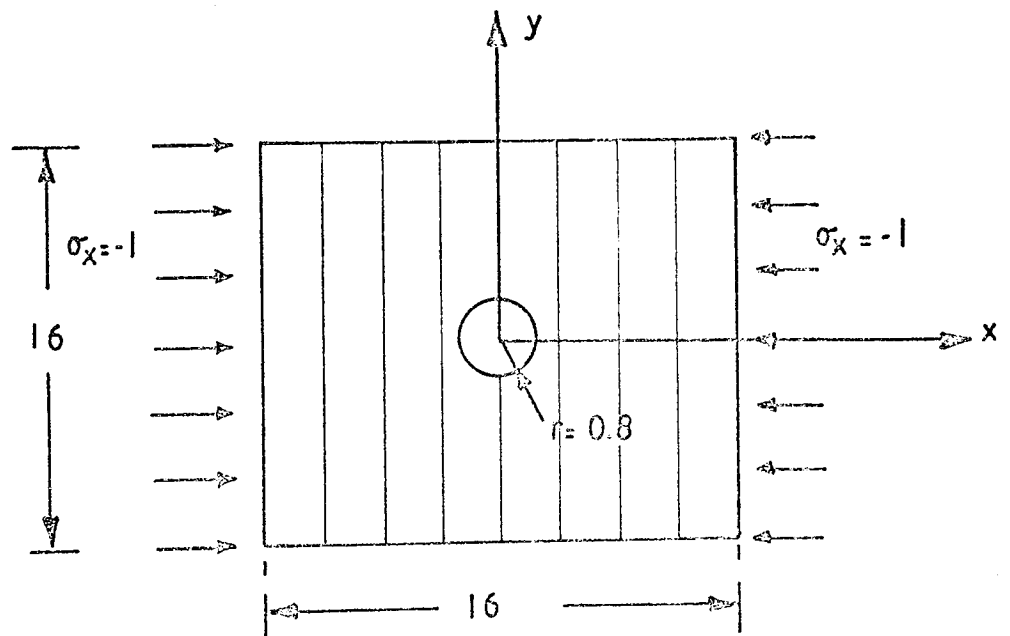


FIG. 5.5 PLOT OF δy AT SECTION $x = 0$



ISOTROPIC: $E_x = E_y = 1$, $\nu_x = \nu_y = 0.1$

$$G = \frac{E}{2(1+\nu)} \quad \text{Thickness} = 1$$



ORTHOTROPIC: $E_x = 3$ $E_y = 1$ $\nu_x = 0.0$ $\nu_y = 0.1$

$$G_{xy} = 0.42 \quad \text{Thickness} = 1$$

FIG 5.6 GEOMETRY AND MATERIAL PROPERTIES.

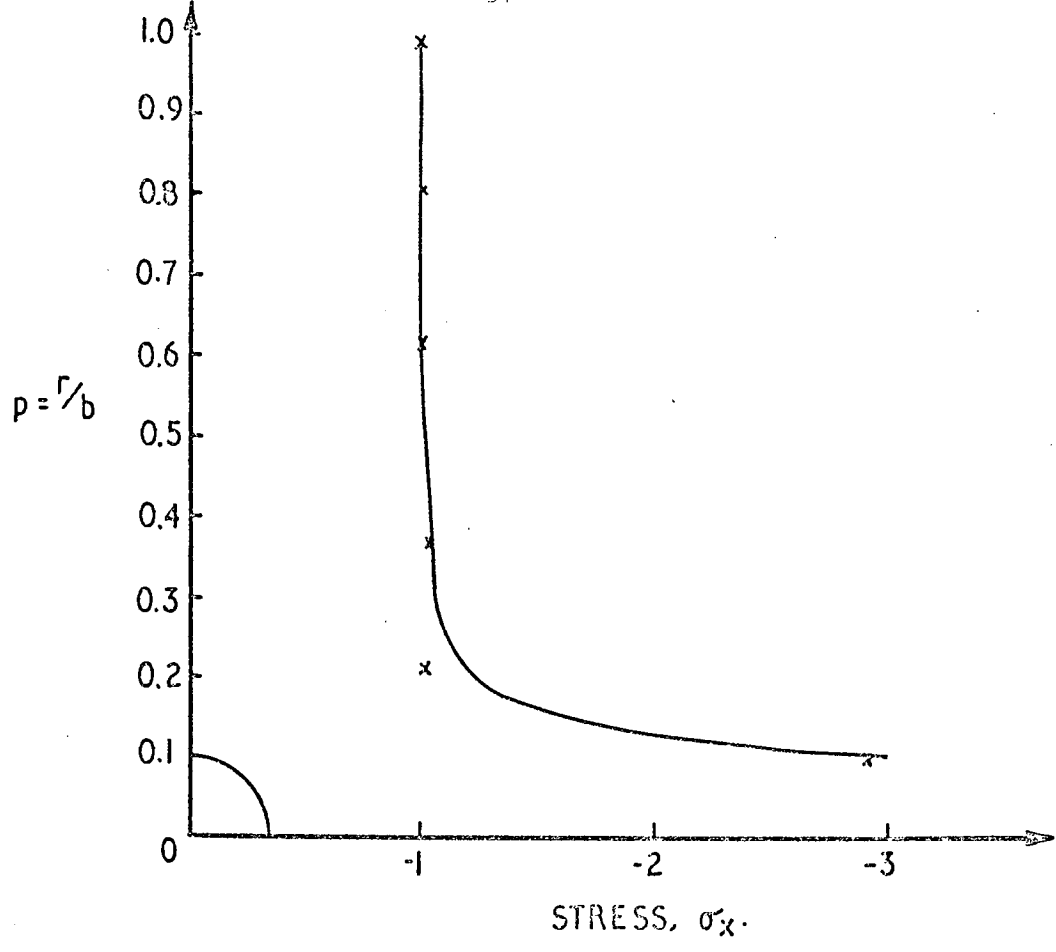


FIG 5.7(a) GRAPH OF NODAL STRESSES ALONG AXIS $X=0$
(ISOTROPIC)

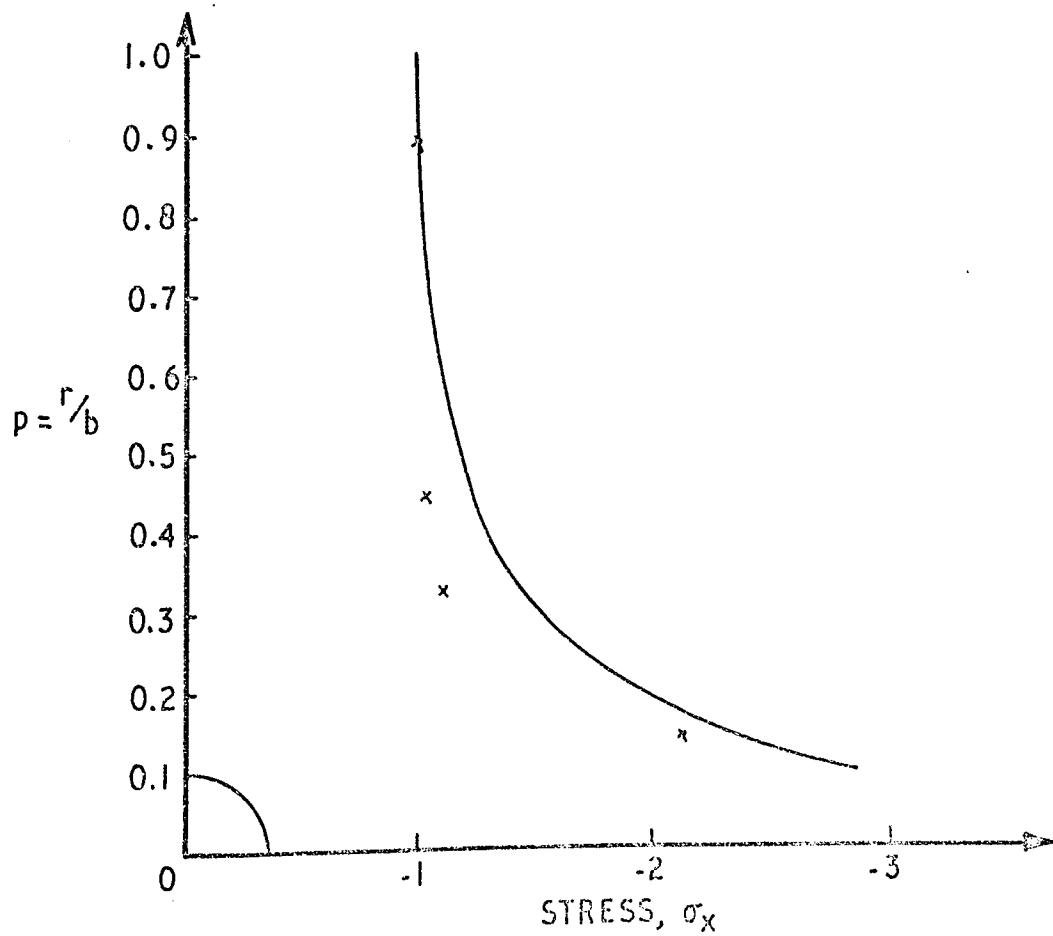


FIG 5.7(b) GRAPH OF NODAL STRESSES ALONG AXIS $X=0$
(ORTHOTROPIC)

5.3.3 Thick Cylinder Under Internal Pressure

This example serves the dual role of providing a further check on the finite element program and also of checking the order of integration i.e. the number of Gaussian integrating points, and its effect on accuracy. Fig. 5.8 shows a quarter of the cylinder used in the analysis, the analytical solution was obtained from Reference (20). Fig. 5.9 shows the idealisations which are referred to hereafter as Mesh Nos. 1, 2, 3 and 4 corresponding to those in the figure. The integration orders used were, linear (1 Gauss Point, G.P.), quadratic (3 G.P.s), cubic (7 G.P.s) and quintic (7 G.P.s) in the manner of table 4.1.

Referring to Figs. 5.10 and 5.11 where graphs of radial and tangential stress are plotted across the section $\theta = 0$. The results are self-explanatory, showing convergence to the exact solution as the mesh is refined. It is apparent from these graphs that there is some degree of uncertainty associated with the stresses at the inner radius. This is to be expected due to the distribution of loading, especially for Meshes 1 and 2.

Referring to Figs. 5.12 and 5.13 where graphs of radial and tangential stress are again plotted across the section $\theta = 0$, using the results of mesh 1 and varying the order of integration. No results were obtained using the linear (1 G.P.) integration formula as this rule produced a singular stiffness matrix. This singularity problem arises from an inadequate number of integrating points and is not uncommon, a similar problem was experienced by Bond et al⁽¹⁵⁾. Again the graphs are self-explanatory, the cubic order being generally more stable than the quintic. For such a trivial problem the job cost changes little with integration order but for larger problems it is expected that there could be an appreciable increase in cost as coefficients of the strain-nodal point displacement array [B] must be evaluated four times more for each element.

The most significant feature which emerges from these results is that increasing the order of integration does not always increase the accuracy of the results. Of primary importance in the accuracy of solution is the number of nodes in the mesh and as such a minimum adequate integration order may be used; the 3 G.P. quadratic formula.

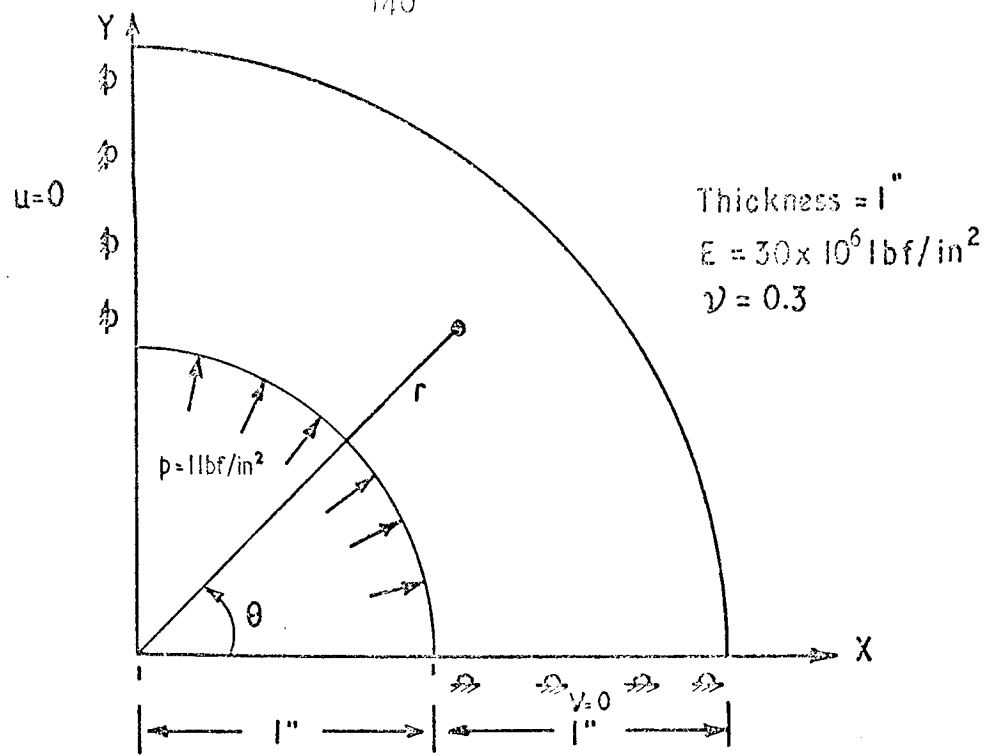


FIG 5.8 QUADRANT OF CYLINDER.

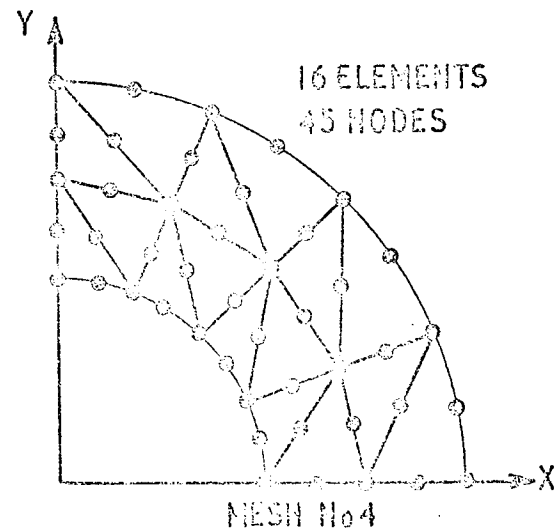
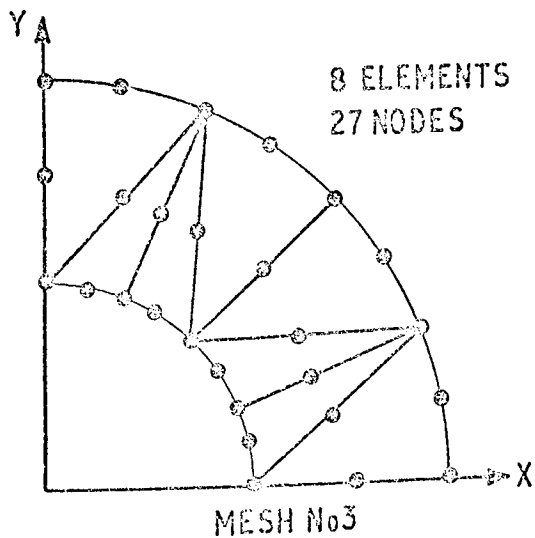
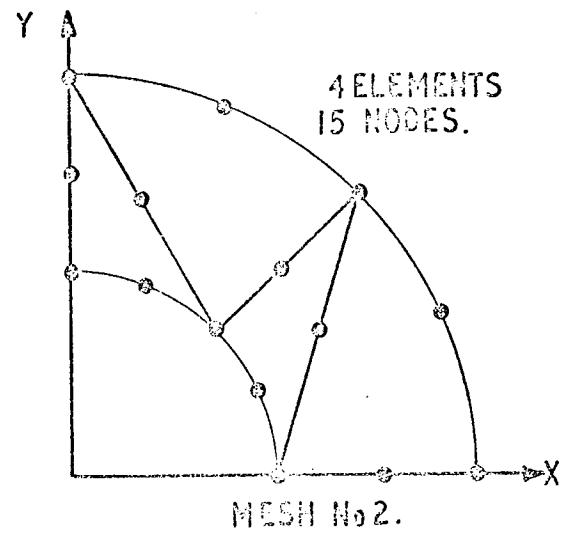
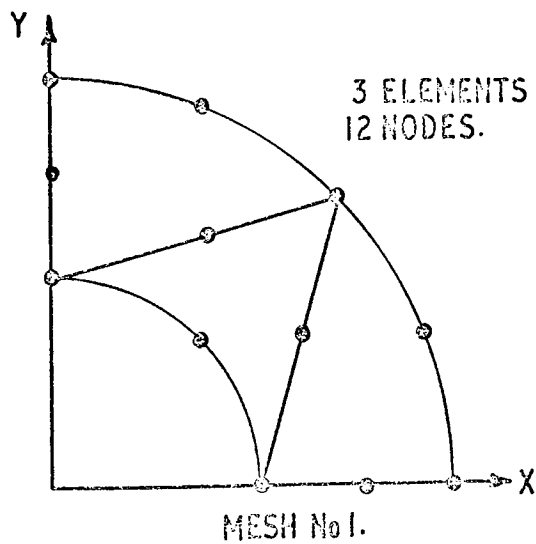


FIG 5.9 FINITE ELEMENT IDEALISATIONS.

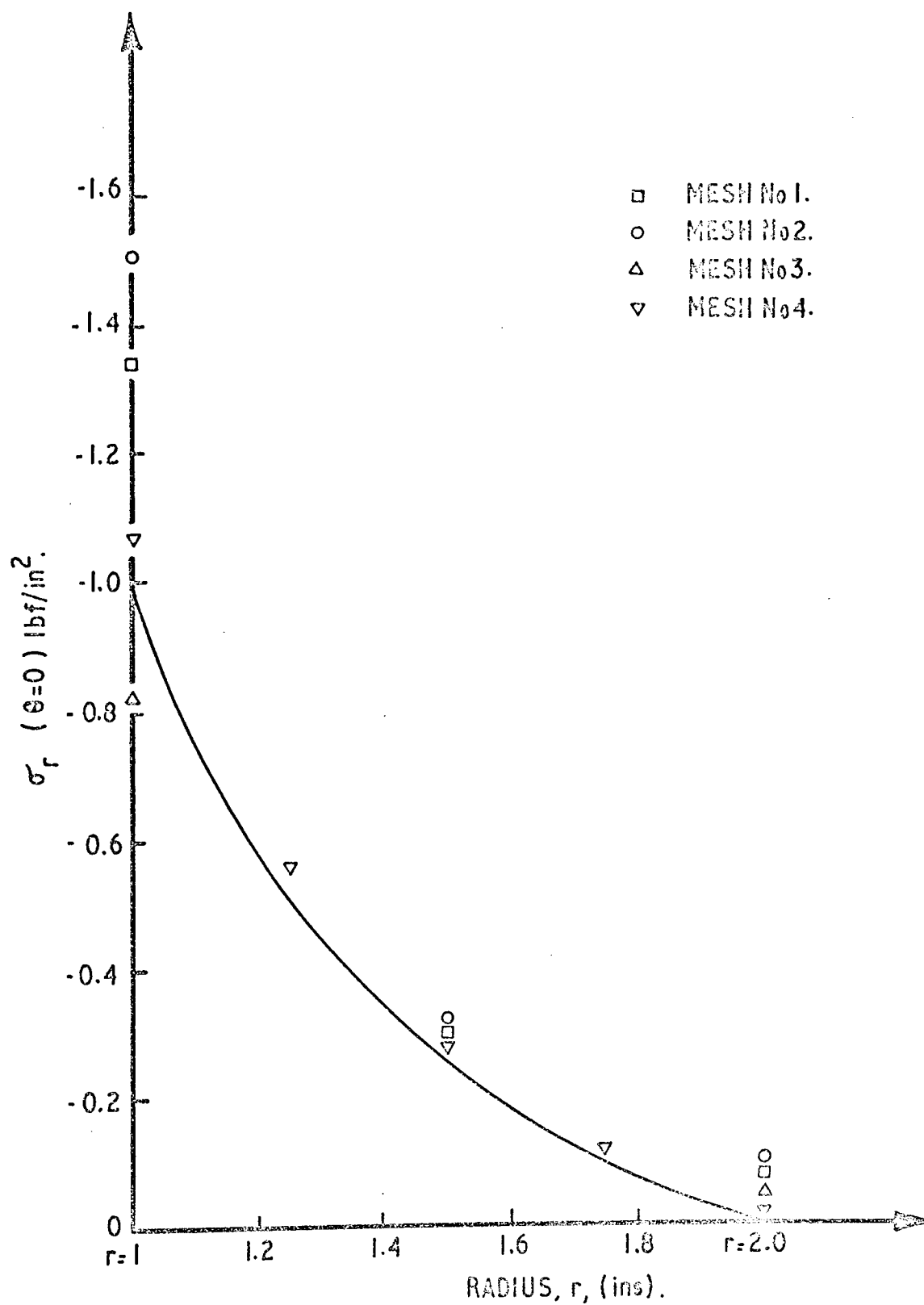


FIG 5.10 RADIAL STRESSES ACROSS CYLINDER SHOWING CONVERGENCE WITH AN INCREASING NUMBER OF ELEMENTS.

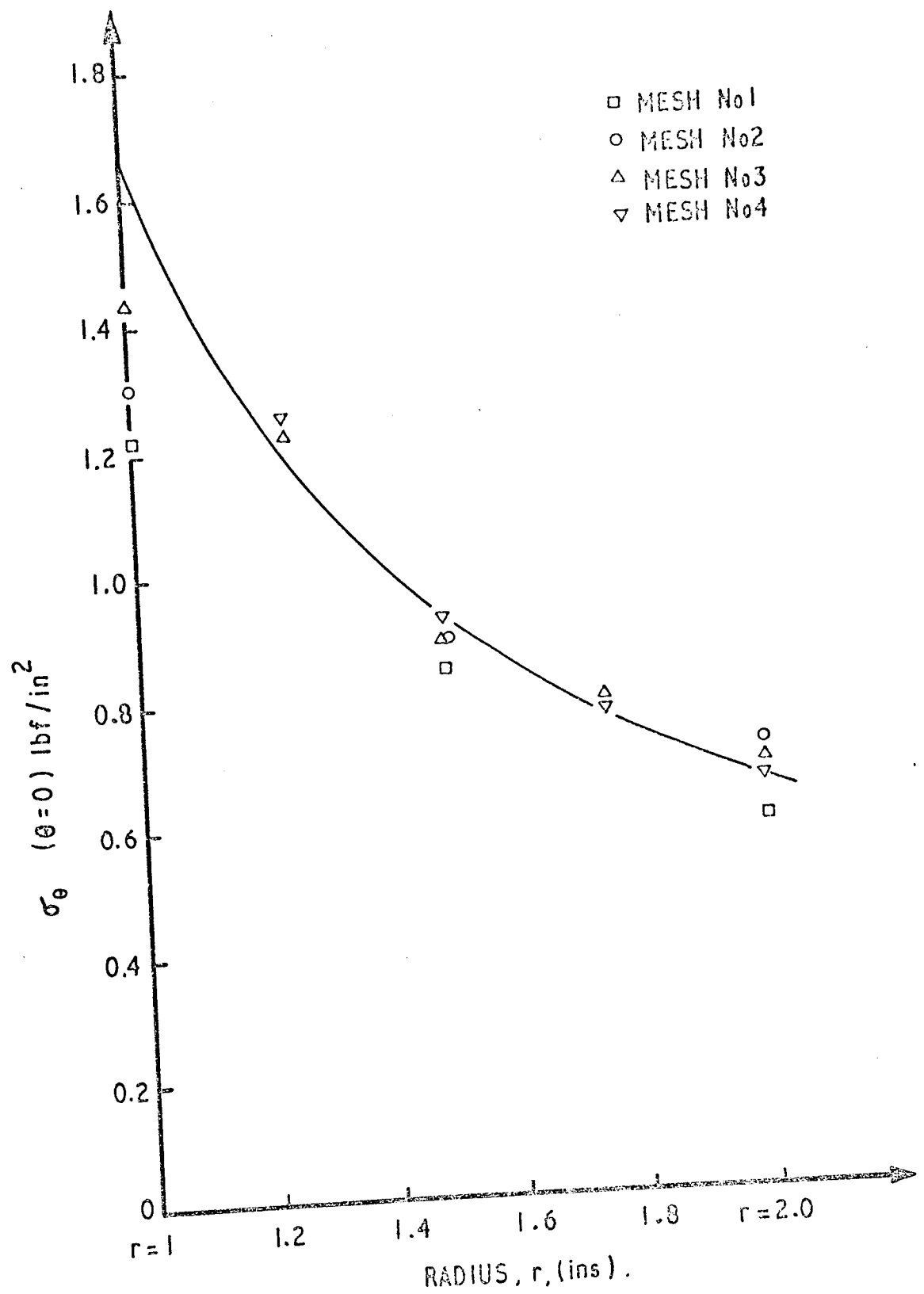


FIG 5.11 TANGENTIAL STRESSES ACROSS CYLINDER
SHOWING CONVERGENCE WITH AN INCREASING
NUMBER OF ELEMENTS.

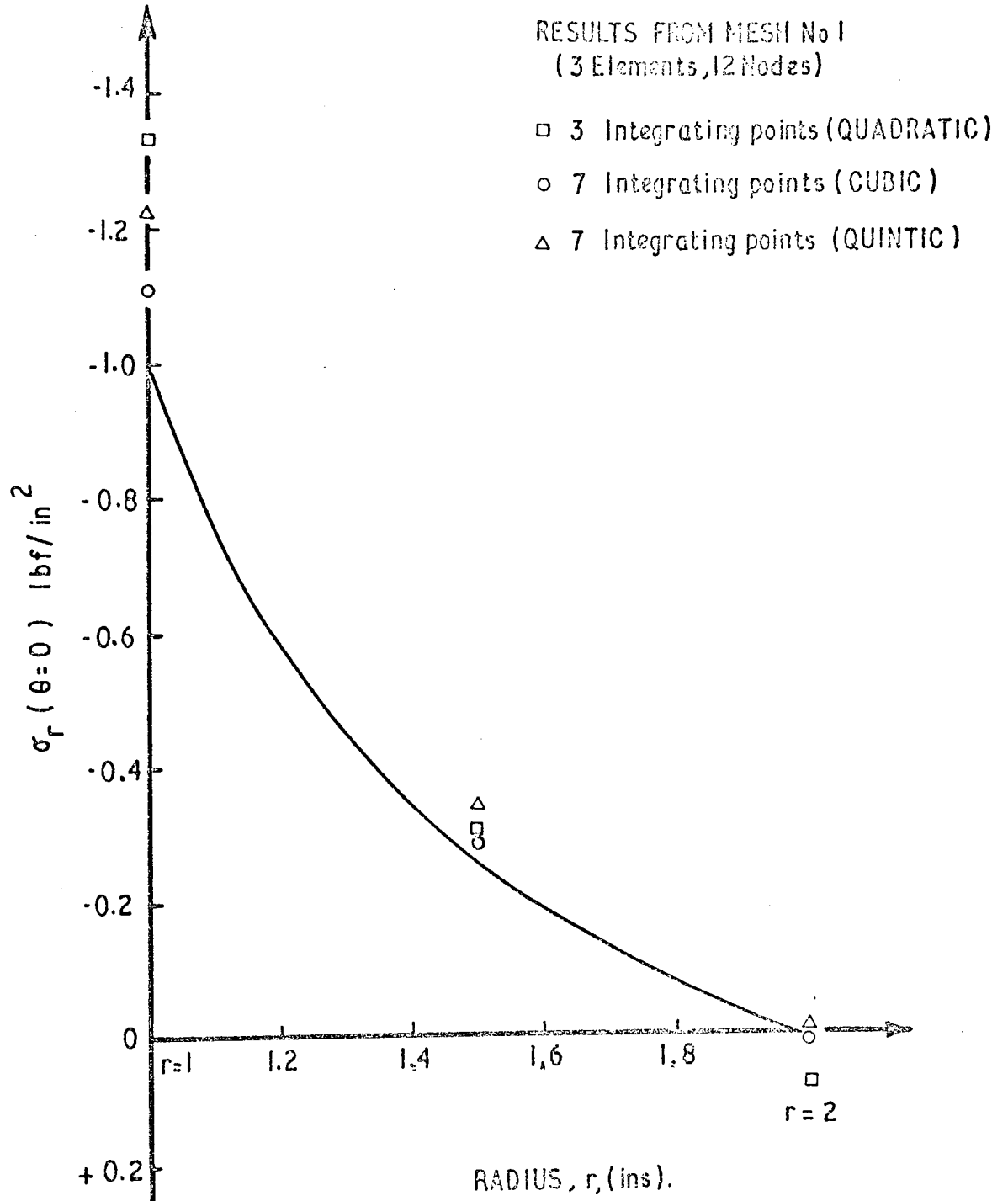


FIG 5.12

RADIAL STRESSES ACROSS CYLINDER SHOWING
CONVERGENCE OF SOLUTION WITH AN INCREASED
NUMBER OF INTEGRATING POINTS

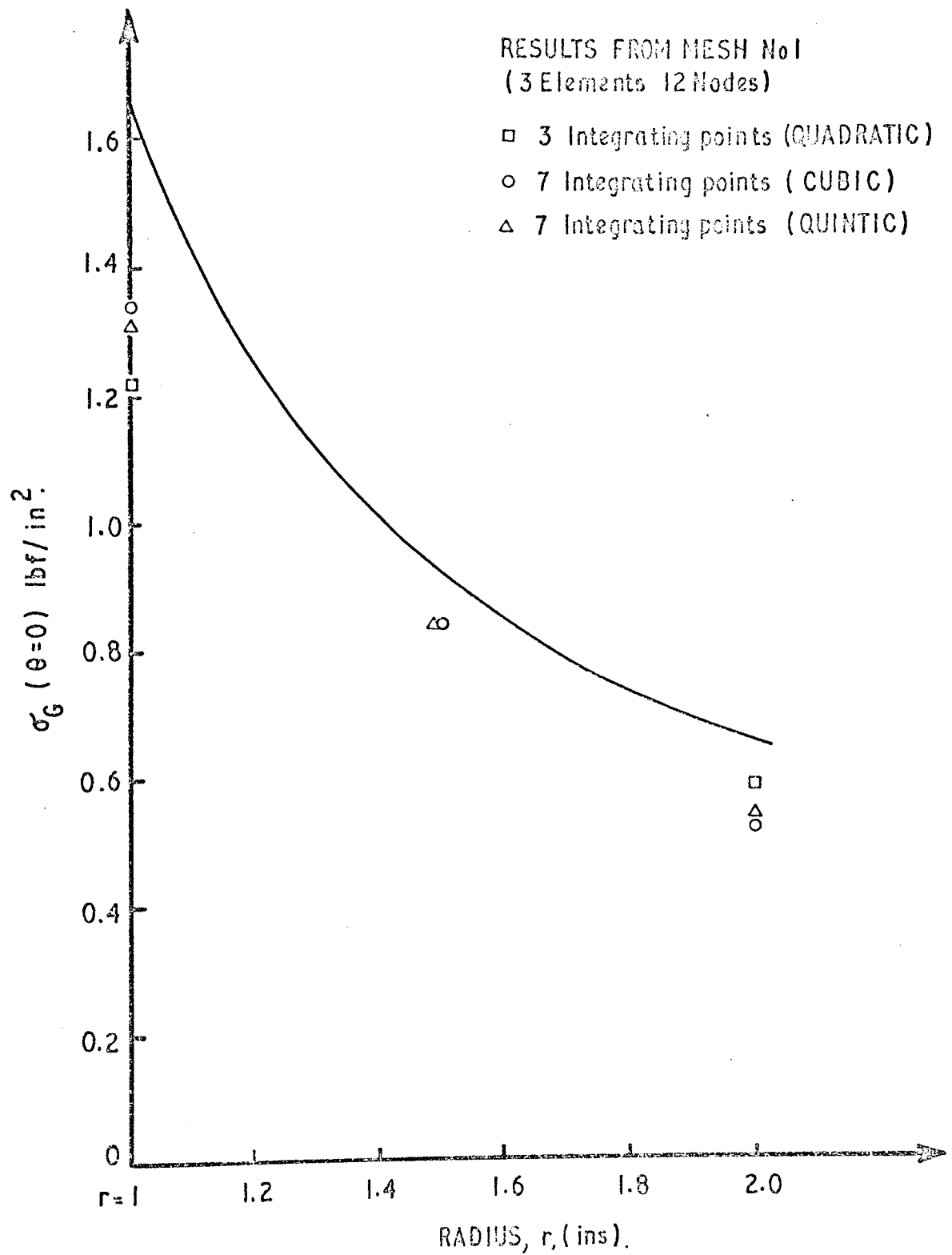


FIG 5.13 TANGENTIAL STRESSES ACROSS CYLINDER SHOWING CONVERGENCE OF SOLUTION WITH AN INCREASED NUMBER OF INTEGRATING POINTS.

6. DISCUSSION AND CONCLUSIONS

In this Section the full finite element program has been described and results obtained for several two-dimensional plane problems in elasticity. The program facilities include :

- (a) Several sets of forces may be accommodated per run.
- (b) Several sets of constraints may be accommodated per run.
- (c) Skewed boundary displacements and loads may be applied, i.e. specified nodal loads or displacements may be applied at any angle to the global axes.
- (d) Each element may be of different material.
- (e) Material may be isotropic or transversely isotropic as for example in laminated sheets, where the elastic properties are orthotropic in the x-y plane. The angle of orthotropy may be arbitrarily inclined to the global axes.
- (f) Option of nodal or element centroidal values of stresses and strains.

The isoparametric formulation, using the natural coordinate system was found to be particularly simple both from a conceptual viewpoint and with regard to its numerical implementation. Use of numerical integration relieves the programmer of tedious algebraic manipulations which provide ample opportunity for errors. From the experience gained herein the advantages and possibilities for mathematical programming as discussed by Irons⁽⁸⁾ can be clearly seen. Other element shapes can easily be derived, using the existing finite element code as a skeleton on which to build.

There is scope for further work on numerical integration formulae. Results obtained here were for a simple example and inconclusive as regards stability in the case of highly distorted elements or elements in regions of high stress gradients. It may well be found most economical in terms of computer time and accurate in terms of the final solution, to specify the integration order for each element individually during mesh subdivision.

The method of storage of the overall stiffness matrix as a one-dimensional array represents a considerable reduction in storage requirements as compared with the more normal banded storage schemes. It may be possible to refine this technique even further where, by using a more complex addressing system with a corresponding equation solving routine even fewer zeroes are generated and retained in the overall stiffness matrix. This problem is discussed and alternative schemes are presented by Schrem⁽²⁷⁾. In considering more elaborate addressing schemes there will be a break even point where benefits accruing in terms of storage are balanced by an increase in computer time and program complexity.

The most tedious aspect of any finite element analysis lies in the sub-division of the structure and in the preparation of input data. A graphical display of input data, such as that presented here, is a useful aid for correcting data errors but there is considerable scope for automatic mesh generation schemes to relieve the user of this task. This problem of mesh generation has received and continues to receive a great deal of attention from researchers, as evidenced by the numerous recent publications on the subject.

There are considerable advantages in the method of programming adopted here, where standard operations are written separately as procedures and called, when required, by the main program. Firstly, it leads to a clear understanding of the main program steps where procedures perform certain well defined operations. Secondly, the program may be easily overlaid, a scheme whereby procedures are held on magnetic tape in compiled form and only called to core when required by the main program. This represents a large core saving considering that the program alone with procedures occupies approximately 17K words of core before arrays are declared. Finally, it allows changes to be made to the procedures with the knowledge that these changes will not affect the main flow. For example, a user may decide to use the program for a new material case, it would then be

necessary to change only the procedure FECMATRIX, the remainder of the program would be unaffected.

As noted previously, tests on accuracy of the program are not as comprehensive as they might have been, particularly with regard to the transversely isotropic material case. The program does, however, work well for isotropic materials and as it was designed for ultimate application to isotropic fracture problems it was considered adequate in this form.

The isotropic finite element program forms the basis for the various programs for fracture analysis to be discussed in Section II. It will be shown in this subsequent Section that excellent results are obtained in the analysis of certain fracture problems, this also serves to provide further confirmation on the accuracy of the original finite element program.

7. REFERENCES

1. Argyris, J.H., 'Energy Theorems and Structural Analysis', Butterworth Scientific Publications, 1960.
2. Turner, M.J., Clough, R.W., Martin, H.C. and Topp, L.J., 'Stiffness and Deflection Analysis of Complex Structures', J. Aero. Sci., 23, 1956.
3. Melosh, R.J., 'Basis for Derivation of Matrices for the Direct Stiffness Method', A.I.A.A. Jnl., Vol. 1, No. 7, 1963.
4. Herrmann, L.R., 'A Bending Analysis of Plates', Proc. Conf. Matrix Methods in Struct. Mechs., Wright-Patterson A.F.B. Ohio, 1965.
5. Pian, T.H.H., 'Derivation of Element Stiffness Matrices by an Assumed Stress Distribution', A.I.A.A. Jnl., 2, 1964.
6. Zienkiewicz, O.C., 'The Finite Element Method: From Intuition to Generality', App. Mechs. Revs., 1971.
7. Irons, B.M.R. and Draper, K.J., 'Inadequacy of Nodal Connections in a Stiffness Solution for Plate Bending', A.I.A.A. Jnl., Vol. 3, No. 5, 1965.
8. Irons, B.M.R. and Barlow, 'Comment on Matrices for the Direct Stiffness Method', A.I.A.A. Jnl., 2, 1964.
9. Zienkiewicz, O.C., 'The Finite Element Method in Engineering Science', McGraw Hill, London, 1971.
10. Irons, B.M.R., 'Numerical Integration Applied to Finite Element Methods', Int. Symp. on Use of Electronic Digital Computers in Structural Engineering, Univ. of Newcastle, Paper No. 19, 1966.
11. Ergatoudis, I., Irons, B.M.R. and Zienkiewicz, O.C., 'Curved Iso-parametric "Quadrilateral" Elements for Finite Element Analysis', Int. J. Solids Struct., Vol. 4, 1968.
12. Irons, B.M.R., 'Engineering Applications of Numerical Integration in Stiffness Methods', A.I.A.A. Jnl., Vol. 4, No. 11, 1966.

13. Argyris, J.H. 'The LUMINA Element for the Matrix Displacement Method', Aeronautical Jnl., Vol. 72, 1968.
14. Zienkiewicz, O.C., 'Isoparametric and Allied Numerically Integrated Element - A Review', Numerical and Computer Methods in Structural Mechanics, Ed. Fenves, S.J., Perrone, N., Robinson, A.R. and Schnobrich, W.C., Academic Press, 1973.
15. Bond, T.J., Swannell, J.H., Henshell, R.D. and Warburton, G.B., 'A Comparison of Some Curved Two-Dimensional Finite Elements', J. Strain Analysis, Vol. 8, No. 3, 1973.
16. Zienkiewicz, O.C., Irons, B.M.R., Ergatoudis, J., Ahmed, S, and Scott, F.C., 'Isoparametric and Associated Element Families for Two- and Three-Dimensional Analysis', (Chapter 13 of "Finite Element Method in Stress Analysis") Holand, I. and Bell, K., ed. Tapir, Trondheim, Norway, 1969.
17. Lekhnitskii, S.G., 'Theory of Elasticity of an Anisotropic Elastic Body', Holden-Day, 1963.
18. Jaeger, J.C., 'Elasticity, Fracture and Flow', Methuen and Co. Ltd.,
19. Zienkiewicz, O.C., Cheung, Y.K. and Stagg, K.G., 'Stresses in Anisotropic Media with Particular Reference to Problems of Rock Mechanics', J. of Strain Analysis, Vol. 1, No. 2, 1966.
20. Timoshenko, S. and Goodier, J.N., 'Theory of Elasticity', McGraw-Hill, 2nd Ed., 1951.
21. Felippa, C.A. 'Refined Finite Element Analysis of Linear and Non-Linear Two-Dimensional Structures', Univ. of California, Berkeley, Ph.D., 1966.
22. Jennings, A. and Tuff, A.D., 'A Direct Method of the Solution of Large Sparse Symmetric Simultaneous Equations', Int. Conf. of Large Sparse Sets of Linear Equations, Oxford, 1970.

23. Griffin, D.S., 'The Verification and Acceptance of Computer Programs for Design Analysis', A.S.M.E. 91st Winter Annual Meeting, New York, 1970, (Pressure Vessels and Piping Div. On General Purpose Finite Element Computer Programs, Morcal, P.V. ed.)
24. Howland, R.C.J., Trans. Roy. Soc. London, Series A, 221, 1921.
25. Savin, G.N., 'Stress Concentration Around Holes', Pergamon Press, 1961.
26. Murnaghan, F.D., 'Finite Deformation of an Elastic Solid', Wiley, 1951.
27. Schrem, E., 'Computer Implementation of the Finite Element Procedure', Ref. contained in 'Numerical and Computer Methods in Structural Mechanics' by Fenves et al., Ref. 14.

APPENDIX 8.A. CHANGE OF VARIABLES FOR AREA INTEGRALS

The substitution method for the evaluation of ordinary definite integrals, is well known and the substitution is normally made with the aim of producing a more recognisable integrand. A change of variables for multiple integrals, which is similar in principle to the substitution method, has a second objective in that the region of integration may be transformed, through substitution, to a more straightforward one.

In the integral

$$I = \iint_R f(x, y) dA \quad (A.1.)$$

the variables may be transformed from the x, y coordinates to new curvilinear coordinates ξ and η by the functions ;

$$\begin{aligned} x &= x(\xi, \eta) \\ y &= y(\xi, \eta) \end{aligned} \quad (A.2.)$$

and their conjugates ;

$$\begin{aligned} \xi &= \xi(x, y) \\ \eta &= \eta(x, y) \end{aligned} \quad (A.3.)$$

Equations A.2 and A.3 define the mapping of the plane, i.e. to each point in the $\xi\eta$ -plane there corresponds a point in the xy -plane and vice versa, Fig. A.1. Similarly to a region R in the xy -plane, there corresponds a region R^1 in the $\xi\eta$ -plane. Choosing a subregion dA^1 in the $\xi\eta$ -plane, Fig. A.1,

$$dA^1 = d\xi d\eta \quad (A.4.)$$

It is required to relate the two elementary areas dA and dA^1 .

The coordinates of the points KLMN in the xy -plane may be written⁽²⁶⁾,
from the definition of the derivative;

$$K(x, y), \quad L\left(x + \frac{\partial x}{\partial \xi} d\xi, y + \frac{\partial y}{\partial \xi} d\xi\right),$$

$$M\left(x + \frac{\partial x}{\partial \xi} d\xi + \frac{\partial x}{\partial \eta} d\eta, y + \frac{\partial y}{\partial \xi} d\xi + \frac{\partial y}{\partial \eta} d\eta\right) \text{ and } N\left(x + \frac{\partial x}{\partial \eta} d\eta, y + \frac{\partial y}{\partial \eta} d\eta\right)$$

Introducing the vectors,

$$\begin{aligned} \overline{KL} &= \frac{\partial x}{\partial \xi} d\xi \hat{i} + \frac{\partial y}{\partial \xi} d\xi \hat{j} \\ \overline{KN} &= \frac{\partial x}{\partial \eta} d\eta \hat{i} + \frac{\partial y}{\partial \eta} d\eta \hat{j} \end{aligned} \quad (\text{A.5.})$$

and forming the cross product of these two vectors to give the area dA as,

$$\begin{aligned} \overline{dA} &= \left| \overline{KL} \times \overline{KN} \right| \\ &= \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{\partial x}{\partial \xi} d\xi & \frac{\partial y}{\partial \xi} d\xi & 0 \\ \frac{\partial x}{\partial \eta} d\eta & \frac{\partial y}{\partial \eta} d\eta & 0 \end{vmatrix} \\ &= \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix} d\xi d\eta \hat{k} \end{aligned} \quad (\text{A.6.})$$

The determinant of equation A.6 is known as the Jacobian of x and y with respect to ξ and η and is denoted by $\frac{\partial(x,y)}{\partial(\xi,\eta)}$ or simply $/J/$.

Substituting from equation A.4,

$$dA = /J/ dA^1 \quad (\text{A.7.})$$

and from equation A1, the integral becomes

$$I = \iint_{R^1} f(\xi, \eta) /J/ dA^1 \quad (\text{A.8.})$$

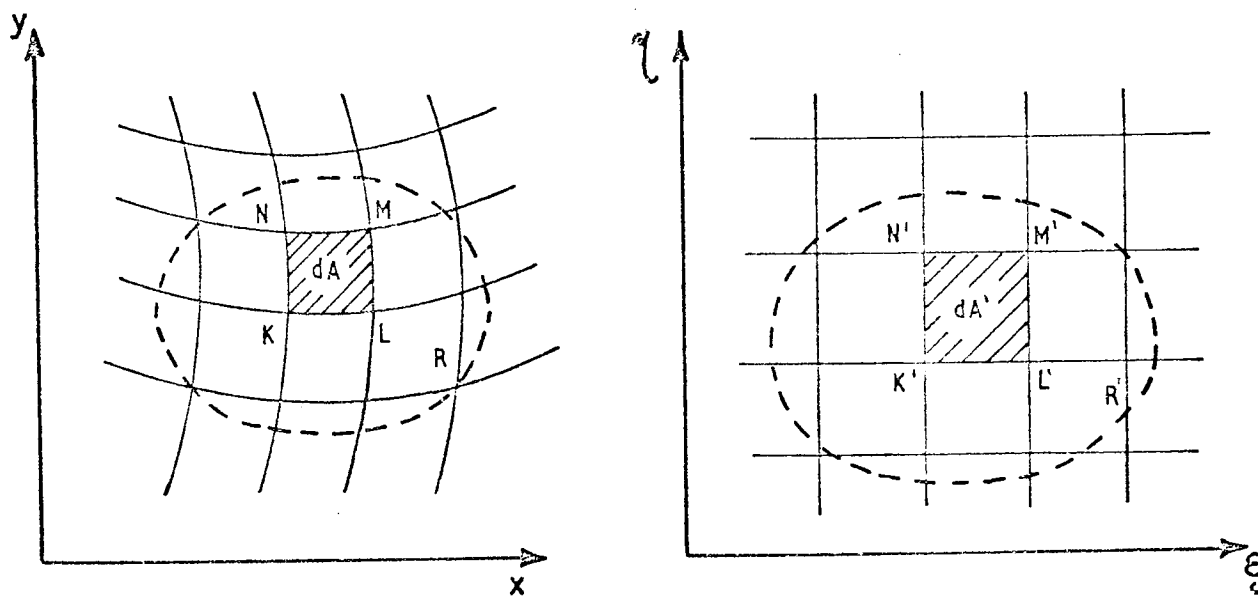


FIG A.I MAPPING FROM XY-PLANE TO $\xi\eta$ -PLANE.

APPENDIX 8.B. PLANE STRESS AND PLANE STRAIN

PLANE STRAIN

A body is in a state of plane strain, parallel to the x,y -plane, if the displacement component in the z -direction, w , is zero and if components of displacement in the x,y -plane, u,v , are functions of x and y only.

The strain-displacement relations appropriate to problems of plane strain reduce to the form,

$$\epsilon_x = \frac{\partial u}{\partial x}, \quad \epsilon_y = \frac{\partial v}{\partial y}, \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

$$\epsilon_z = \gamma_{xz} = \gamma_{yz} = 0$$

This strain state corresponds to that in a long cylindrical body with loads that are independent of the variable z , Fig. B.1.

PLANE STRESS

Consider a thin plate, Fig. B.2, of thickness $2h$ with the xoy -plane as the middle plane of the plate. Assume that external forces X_N and Y_N act on the side surfaces of the plate in such a way that the resultant of these forces for the entire height $2h$ is situated in the xOy -plane. The two faces of the plate, $z = \pm h$ are assumed free of external forces, i.e.

$$\sigma_z = \tau_{yz} = \tau_{xz} = 0 \text{ for } z = \pm h.$$

For a plate that is thin in the z -plane it can then be assumed that

$$\sigma_z = \tau_{yz} = \tau_{xz} = 0 \text{ everywhere inside the plate, or instead of the actual stress components } \sigma_x, \sigma_y \text{ and } \tau_{xy} \text{ their mean values are considered, i.e.}$$

$$\sigma_x^* = \frac{1}{2h} \int_{-h}^h \sigma_x \, dz, \quad \sigma_y^* = \frac{1}{2h} \int_{-h}^h \sigma_y \, dz,$$

$$\tau_{xy}^* = \frac{1}{2h} \int_{-h}^h \tau_{xy} \, dz$$

The latter stress state described above is known as one of 'generalised plane stress'. As the plate thickness approaches zero, a condition of 'plane stress' is approached; i.e. $\sigma_z = \tau_{yz} = \tau_{xz} = 0$ everywhere in the plate.

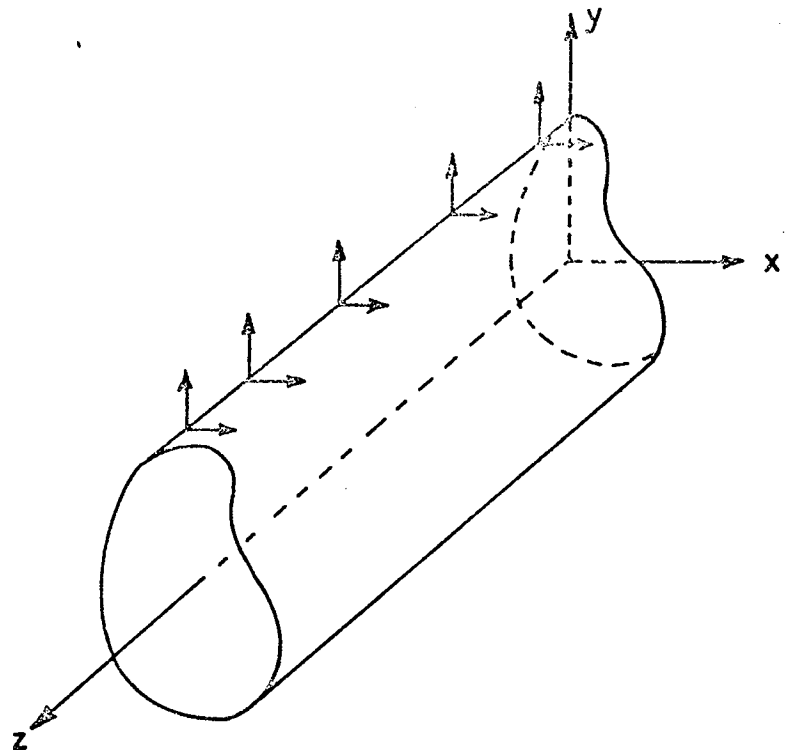


FIG B.1 CONDITIONS OF PLANE STRAIN.

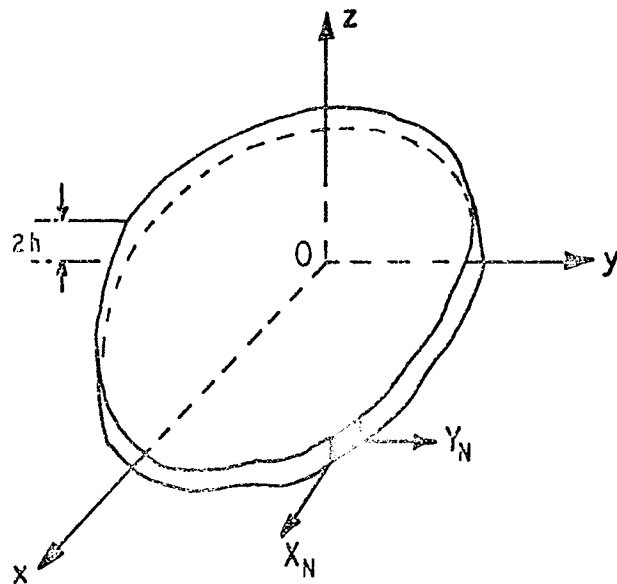


FIG B.2 CONDITIONS OF PLANE STRESS.

APPENDIX 8.C. PROGRAM FOR PLOTTING DEFINED FINITE ELEMENT MESH

The most tedious aspects of the use of the finite element method are the basic processes of subdividing the continuum and generating input data for the computer. Errors in the input data, which are extremely common, are often difficult to detect and if undetected can lead to erroneous results. Consequently, it is essential to provide checks to detect such errors and it is to this end that the following program has been written. The program produces a plot of the mesh and allows a visual check of the input data.

A listing of the program follows and its logic is based on the procedure FEINPUT, previously described in section 4.3 but includes several more print statements to facilitate the detection of errors. Data necessary for running the program is identical to that used for the finite element program.

```

      *****
      BEGIN 'COMMENT' GRAPHICAL DISPLAY OF INPUT DATA;
      REAL SCALE,X0,Y0,CHANGE;
      INTEGER NPLOT,R,START,NLEMT,NNODE,THICK,NREFE,NSEIFS,PRNT,NSEIC,
              S,NSEIF,I,J,NSPEC,W,NNEW,NG1,NOD,NCS,NCKEW,NMAT;
      INTEGER 'ARRAY' BCD1,BCD2,BCD3[1:10];
      REAL 'ARRAY' X,Z[1:7];
      NPLOT:=READ;
      FOR R=START:=1 STEP 1 UNTIL NPLOT 'DO'
      BEGIN
      OPENPLOT;
      WRTTEXT('('('2')'JOB%NAME%-----')');
      COPYTEXT('END%OXTITLE');
      NLEMT:=READ; NNODE:=READ; NSEIFS:=READ;
      PRNT:=READ; NCKEW:=READ; NMAT:=READ;
      WRTTEXT('('('2')'NO%OF%ELEMENTS%-----')');
      PRINT(NLEMT,3,0);
      WRTTEXT('('('2')'NO%OF%NODES%-----')');
      PRINT(NNODE,3,0);
      NREF:=NNODE*2;
      BEGIN
      REAL 'ARRAY' XX,VY[1:NNODE],ULX,VLY[1:NNODE,1:NSEIFS],
              Q[1:NREF,1:NSEIFS];
      INTEGER 'ARRAY' CODE[1:NLEMT,1:6],KODE[1:NNODE,1:NSEIFS];
      NSEIC:=READ;
      COMMENT PROG REQUIRES MODIFICATION TO HANDLE SEVERAL SETS OF CONCS
      FOR S:=1 STEP 1 UNTIL NSEIC 'DO'
      BEGIN 'IF' S=1 'THEN'
      BEGIN NSEIF:=READ;
      FOR I:=1 STEP 1 UNTIL NNODE 'DO' XX[I]:=0.00001;
      NSPEC:=READ;
      WRTTEXT('('('2')'COORDINATES%OF%SPECIFIED%NODES('204S')'
      NODE('5S')'X-COORD('5S')'Y-COORD('5S')'TYPE('7S')'X-DISP('6S')'

```

```

X=DISP('C48S')'OR%LOAD'('5S')'OR%LOAD')': NEWLINE(2);
'FOR' I:=1 'STEP' 1 'UNTIL' NSPEC 'DO'
'BEGIN'
  J:=READ; XX[I,1]:=READ; YY[I,1]:=READ;
  KODE[I,1]:=READ; UGX[I,1]:=READ; VLY[I,1]:=READ;
  NEWLINE(1); SPACE(3); PRINT(J,3,0); SPACE(3);
  PRINT(XX[I,1],0,3); PRINT(YY[I,1],0,3); SPACE(2); PRINT(KODE[I,1],3,0);
  SPACE(2); PRINT(UGX[I,1],0,3); PRINT(VLY[I,1],0,3);
'END';
WRITE TEXT('('('40')'ELEMENT%DATA'('20')'ELEMENT'('18S')'
  NODAL%CONNECTIONS'('10')'INATE-1A'('2');
'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN' W:=READ; NEWLINE(1); PRINT(W,3,0); SPACE(6);
  'FOR' J:=1 'STEP' 1 'UNTIL' 4 'DO'
  'BEGIN' NODE[W,J]:=READ;
    PRINT(NODE[W,J],3,0); SPACE(2);
  'END';
'END';
WRITE TEXT('('('20')'NODAL%CONNECTIONS%HAVE%BEEN%READ'('30')'
  INTERPOLATION%DETAILS'('20')'ELEMENT'('3S')'NODE'('4S')'
  Y=COORD'('7S')'Y=COORD')':
'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN' IF XX[NODE[W,I]]#0.00001 THEN
'BEGIN'
  NEWLINE(1); PRINT(W,3,0); SPACE(2);
  PRINT(NODE[W,3+1],3,0); SPACE(2);
  IF I#1 THEN J:=1 ELSE IF I#2 THEN J:=2 ELSE J:=0;
  XX[NODE[W,3+1]]:=(XX[NODE[W,I]]+XX[NODE[W,1+J]])/2;
  PRINT(XX[NODE[W,3+1]],0,3); SPACE(2);
  YY[NODE[W,3+1]]:=(YY[NODE[W,I]]+YY[NODE[W,1+J]])/2;
  PRINT(YY[NODE[W,3+1]],0,3);
  KODE[NODE[W,3+1],1]:=0;
  UGX[NODE[W,3+1],1]:=VLY[NODE[W,3+1],1]:=0.0;
'END';
'END';
WRITE TEXT('('('20')'NODAL%POINT%DATA'('204S')'NODE'('5S')'X=COORD
  '('5S')'Y=COORD'('5S')'TYPE'('7S')'X=DISP'('6S')'Y=DISP'('C48S')'
  OR%LOAD'('5S')'OR%LOAD')':
'FOR' J:=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(UGX[I,1],0,3);
  PRINT(VLY[I,1],0,3);
'END';
'END' ELSE
'BEGIN' NNEWC:=READ;
  NSEIF:=READ;
'FOR' I:=1 'STEP' 1 'UNTIL' NNEWC 'DO'
'BEGIN' J:=READ;
  KODE[J,1]:=READ; UGX[J,1]:=READ; VLY[J,1]:=READ;
'END';
'END';
'BEGIN'
  HCPLOT(-5,0,25,0,0,4);
  CHANGE:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'BEGIN'

```



```

      'IF' XX[W]>CHANGE 'THEN' CHANGE:=XX[W];
      'IF' YY[W]>CHANGE 'THEN' CHANGE:=YY[W];
'END';
SCALE:=2;
WRITE(1,XT('(','2,')' SCALE')): PRINT(SCALE,0,4);
'FOR' M:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
  X[1]:=X[7]:=XX[NODE[W,1]]*SCALE;
                                Y[1]:=Y[7]:=YY[NODE[W,1]]*SCALE;
  X[2]:=XX[NODE[W,4]]*SCALE;
                                Y[2]:=YY[NODE[W,4]]*SCALE;
  X[3]:=XX[NODE[W,2]]*SCALE;
                                Y[3]:=YY[NODE[W,2]]*SCALE;
  X[4]:=XX[NODE[W,5]]*SCALE;
                                Y[4]:=YY[NODE[W,5]]*SCALE;
  X[5]:=XX[NODE[W,3]]*SCALE;
                                Y[5]:=YY[NODE[W,3]]*SCALE;
  X[6]:=XX[NODE[W,6]]*SCALE;
                                Y[6]:=YY[NODE[W,6]]*SCALE;
  HGPLINE(X,Y,7,1);
'END' OF ELEMENT;

```

```

STRAPP(BCD1,6,1('XAXIS')));
STRAPP(BCD2,6,1('YAXIS')));
STRAPP(BCD3,23,1('STRUCTURAL IDEALISATION')));
HGPAXIS(0.0,0.0,BCD1,-6,14,0.0,0.0,0.0,1.0);
HGPAXIS(0.0,0.0,BCD2,6,13,0.90,0.0,0.0,1.0);
HGPSYMBL(7,11,0.25,BCD3,0.25);
'END' OF GRAPH PLOTTER PROCEDURES;
'END';
'END';
'END';
CLOSEPLOT;
'END';

```

```

XX          LENGTH 1674
EIS USED   38
#AXXX      EC

```

SECTION II

CRACKED STRUCTURES AND FRACTURE PROBLEMS

1. SUMMARY OF SECTION II

In this Section, on fracture problems, several cracked configurations are analysed by embedding into the finite element mesh a sub-region, containing the singularities and over which an analytic solution is used.

The modifications necessary to augment a standard finite element program, such as that developed in Section I, are discussed and complete programs for each cracked configuration are presented. Several examples are included to demonstrate the accuracy and flexibility of the technique.

2. INTRODUCTION AND LITERATURE REVIEW

It is perhaps complimentary to the designer and his analytical methods that only a relatively small percentage of engineering components actually fail by fracture during their normal service lives. Nevertheless, the few failures which do occur by this mechanism can often have disastrous consequences, as the bibliography of (1) will testify. A subsequent examination of structural components which have failed during service generally reveals that the fracture origin was a small crack or crack-like flaw. The initial flaw size may have been sufficiently large to cause fracture upon initial loading, in some cases the failure load being only a small percentage of the design load, or it may have been small enough such that the component withstood a number of load cycles before the flaw reached a sufficient size to cause failure. The fractured component is normally characterised by an absence of a large amount of plastically deformed or yielded material. This highlights the danger involved in assuming that the strength and operational life of structures can be estimated using conventional strength analysis where no flaws are considered and large amounts of plastic yielding are assumed to take place prior to fracture. Clearly, it must be recognised that all structural components contain defects and flaws of various kinds and that the lives of these components are controlled by them.

In an effort to provide designers with quantitative and easily applied information concerning the behaviour of cracked components, the general subject area of Fracture Mechanics has arisen. Fracture mechanics provides a method of characterising such fracture behaviour in terms of common structural parameters: e.g. applied stress and flaw size. As fracture mechanics is based on a stress analysis of the cracked component it is generally a relatively simple matter to translate laboratory results into practical design information.

Using this technique it is possible to account for the effects of temperature and loading rate on the behaviour of structural members having sharp cracks. Thus the results of a fracture mechanics analysis would yield the combinations of design stress and flaw size that would be required to cause sudden failure. In addition, fracture mechanics can be used to analyse the growth of small flaws, by fatigue loading or stress corrosion, to a critical size.

Of the several available methods of fracture analysis, a comprehensive survey of which may be found in Refs. (2, 3) only two of the major approaches will be discussed in detail. The first is the energy approach of Griffith⁽⁴⁾ and the second uses a detailed stress analysis of the crack tip region.

2.1 THE GRIFFITH CONCEPT

In his now classic paper on brittle fracture, Griffith postulated that a necessary condition for a crack to spread under the action of external loads is that the energy used in creating new fracture surfaces is supplied from the released energy in the elastic solid. Only if the total energy decreased would the crack extend spontaneously under the applied stress.

As an example, consider a centre cracked infinite plate of unit thickness loaded by a remote tensile stress σ as shown in Fig. 2.1. The changes in energy which occur as the crack extends an infinitesimal amount da are twofold :-

- (a) As the new crack surfaces are created, the energy increase during crack extension is simply the 'work to fracture', $2\gamma da$. Here γ is the surface energy of the material and the factor 2 relates to the number of surfaces created.

(b) As the crack extends, energy in the elastic solid is released.

Under 'fixed-grip' conditions (fixed load point displacement) an increase in crack length results in a decrease in stored elastic strain energy while under constant loading conditions ($\sigma = \text{constant}$) the total potential energy is decreased. It can be shown, however⁽¹⁰⁾, that for an infinitesimally small amount of crack extension, the decrease in stored elastic energy of a cracked body under fixed grip conditions is identical to the decrease in total potential energy under conditions of constant loading.

Griffith recognised that the driving force for crack extension was the difference between the energy which would be released if the crack were extended and that needed to create new surfaces. The stress analysis he used to calculate the elastic energy release was based upon the work of Inglis⁽⁵⁾.

According to the Griffith theory, therefore, a necessary condition for fracture to occur is that,

$$G \geq G_c$$

where G is the strain or potential energy release rate ($-\partial u / \partial a$) and $G_c = 2\gamma$, the work to fracture per unit crack extension.

G is conventionally defined as the 'strain energy release rate' or 'crack extension force'.

2.2 THE STRESS INTENSITY FACTOR CONCEPT

The stress intensity factor concept, as its title suggests, leads to a stress criterion for fracture. Stress intensity factors provide a measure of the near crack tip stress field and when this is great enough then the material fails.

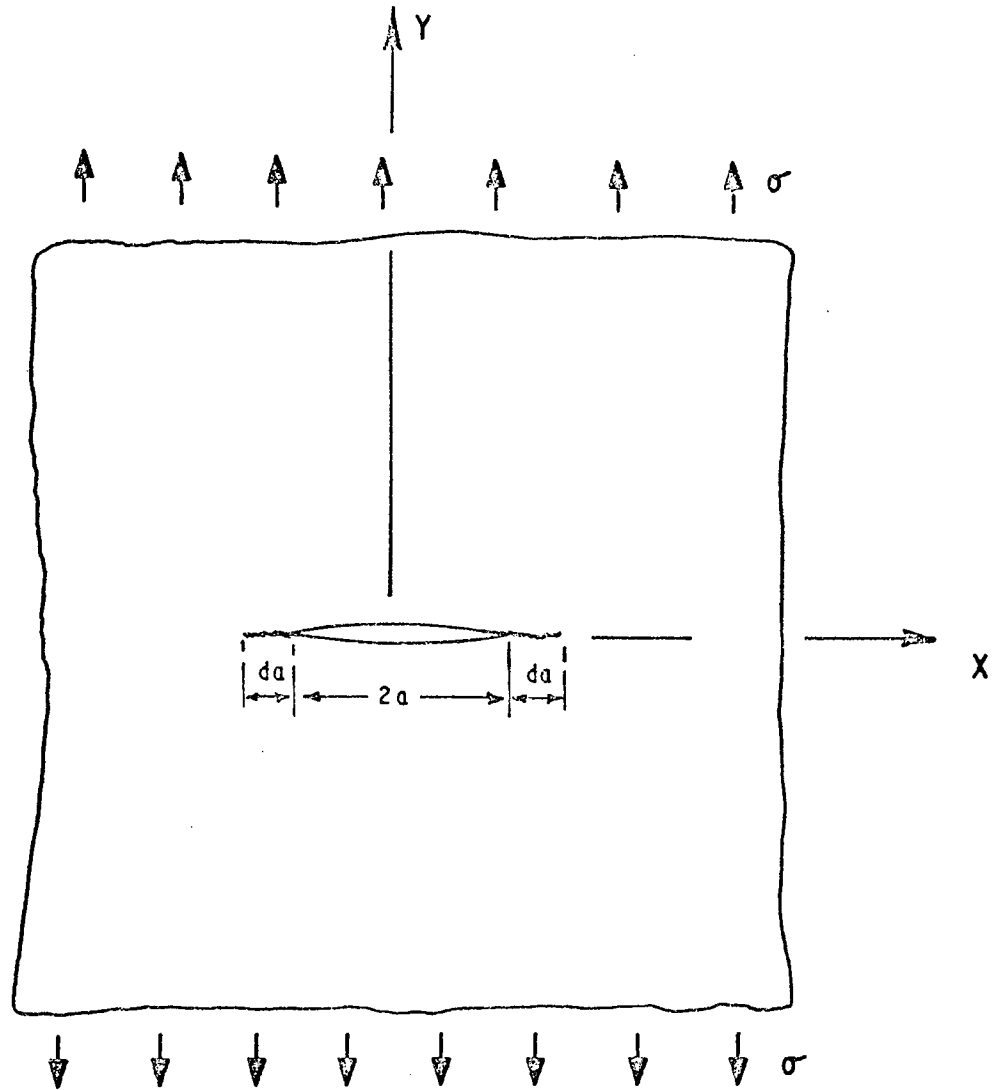


FIG 2.1 CENTRE CRACKED INFINITE PLATE .

The basic philosophy behind this approach will now be introduced with reference to the first mode of crack extension. For this mode, complete symmetry of loading and geometry prevails with respect to the crack which extends by opening along its own axis. The other possible modes of crack extension will be discussed in section 2.4, the analytical analyses for which are developed in an analogous manner to the following Mode I case.

Westergaard⁽⁶⁾, using the method of complex functions, derived singular equations for the elastic stress distribution close to crack tips. For the infinite plate of Fig. 2.1, Westergaard obtained for the component of stress in the y-direction

$$\sigma_y = \sigma \sqrt{\frac{a}{2r}} \cos \frac{\theta}{2} \left(1 + \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) + (\text{other terms bounded at } r=0) \quad (2.1)$$

where the origin of polar coordinates is at the crack tip as shown in Fig. 2.2.

Equation 2.1 may be rewritten as,

$$\sigma_y = \frac{K_I}{\sqrt{2r}} \cos \frac{\theta}{2} \left(1 + \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) + \dots \quad (2.2)$$

where K_I is a factor, given by $K_I = \sigma \sqrt{a}$ and known generally as the Mode I stress intensity factor⁽¹¹⁻¹³⁾.

In Appendix 10.B, equations are listed in terms of K_I for the other components of stress. The units of K_I are lbf in^{-3/2} (MNm^{-3/2}).

The near crack tip stress field equations have also been obtained by other authors⁽⁷⁻¹⁰⁾, using methods such as Muskhelishvili's complex function approach⁽²⁷⁾ or the method of eigenfunction expansions as devised by Williams⁽²⁶⁾.

Analysis of other loading configurations reveal that equations of the form of 2.2 are obtained in each case but the value of K_I is unique to a given situation. The stress intensity factor, therefore, characterises the intensity of the stress field ahead of a crack. For any cracked body

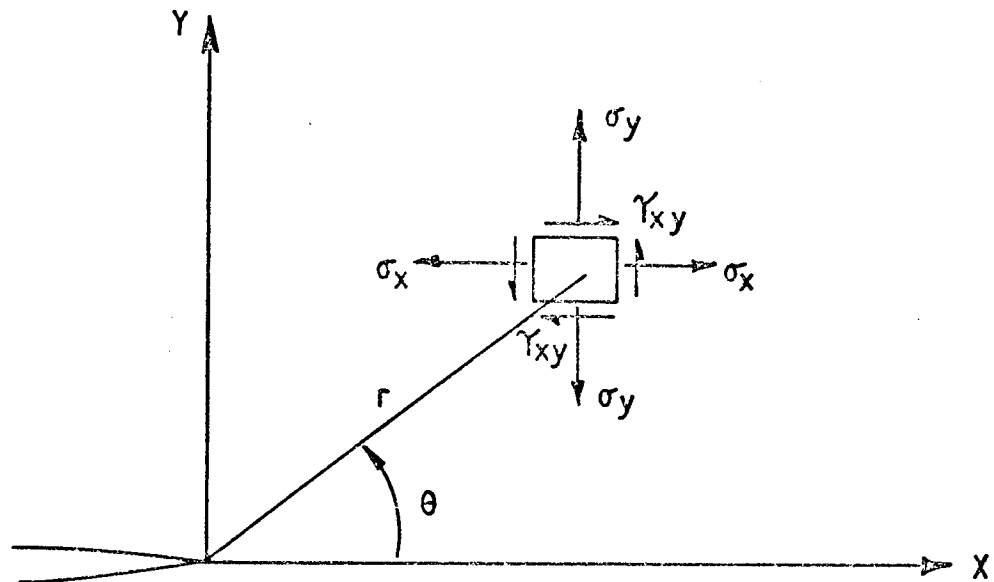


FIG 2.2 CO-ORDINATE SYSTEM AT CRACK TIP.

under load it is thus known that the local crack stresses will diminish as the inverse square root of the distance from the tip and that K_I gives the precise magnitude of the stress by embracing, in one convenient parameter, both the crack geometry and the contribution of the applied forces.

The stress intensity factor criterion for failure states that, fracture will take place when the stress intensity factor reaches some critical value for the material, $K_{IC}^{(14)}$, previously determined by experiment. This critical value is known as the fracture toughness and is taken as a material property. The difference between K_I and K_{IC} can be considered analogous to the difference between stress and yield strength in conventional stress analysis.

In summary, the utility of elastic stress analyses lies in the similarity of near crack tip stress distributions for all configurations and the stress intensity factor provides a convenient measure of the load applied to the crack tip region.

The form of the stress intensity factor was obvious from equation 2.1 for the particular example of Fig. 2.1 where a closed form solution is available. For more complex component shapes under arbitrary loading conditions, exact solutions are virtually impossible. In these cases, therefore, other methods must be used to obtain a suitable expression for K_I . These methods will be described in subsequent sections of this thesis.

2.3 EQUIVALENCE OF STRESS INTENSITY FACTOR AND GRIFFITH METHODS

For the linear-elastic case, the two approaches of sections 2.1 and 2.2 may be shown to be equivalent⁽¹⁵⁾. From a consideration of the work done in extending the crack by a small increment, it can be shown that, for plane strain

$$G_I = \frac{K_I^2}{E} (1 - \nu^2) \quad (2.3)$$

and for plane stress

$$G_I = \frac{K_I^2}{E} \quad (2.4)$$

where ν = Poisson's Ratio

and E = Young's Modulus.

2.4 MODES OF CRACK EXTENSION

In the preceeding sections only the opening mode of fracture was discussed; there, the crack extends by opening along the X-axis, Fig.

2.1. There are, however, two other possible modes by which a crack may extend; these are illustrated diagrammatically in Fig. 2.3. The modes are generally referred to as^(3, 10),

- (a) Opening Mode, K_I
- (b) Shear Mode, K_{II}
- (c) Antiplane Shear Mode, K_{III}

The Shear Mode II and the Antiplane Shear Mode III fracture mechanisms may be analysed in a similar manner⁽¹⁰⁾ to the opening Mode I case as in section 2.2. It is found from such analyses that each mode has the characteristic inverse square root singularity similar to that for K_I found from equation 2.2. K_{II} and K_{III} are, therefore, analogous to K_I in that they characterise the magnitude of the near crack tip stress field for these modes of crack extension.

For a general three-dimensional crack problem, the fracture criterion would have to incorporate all of the modes, K_I , K_{II} and K_{III} . Two theories of fracture have been advanced for a cracked component under combined loading; maximum stress criterion⁽⁶⁰⁾ and strain energy density factor theory^(21, 61). In the maximum stress concept, it is assumed that the crack extension occurs in a plane perpendicular to the direction of the greatest tension for a combined Mode I-II problem. In the strain energy density factor theory, it is assumed that a critical strain energy

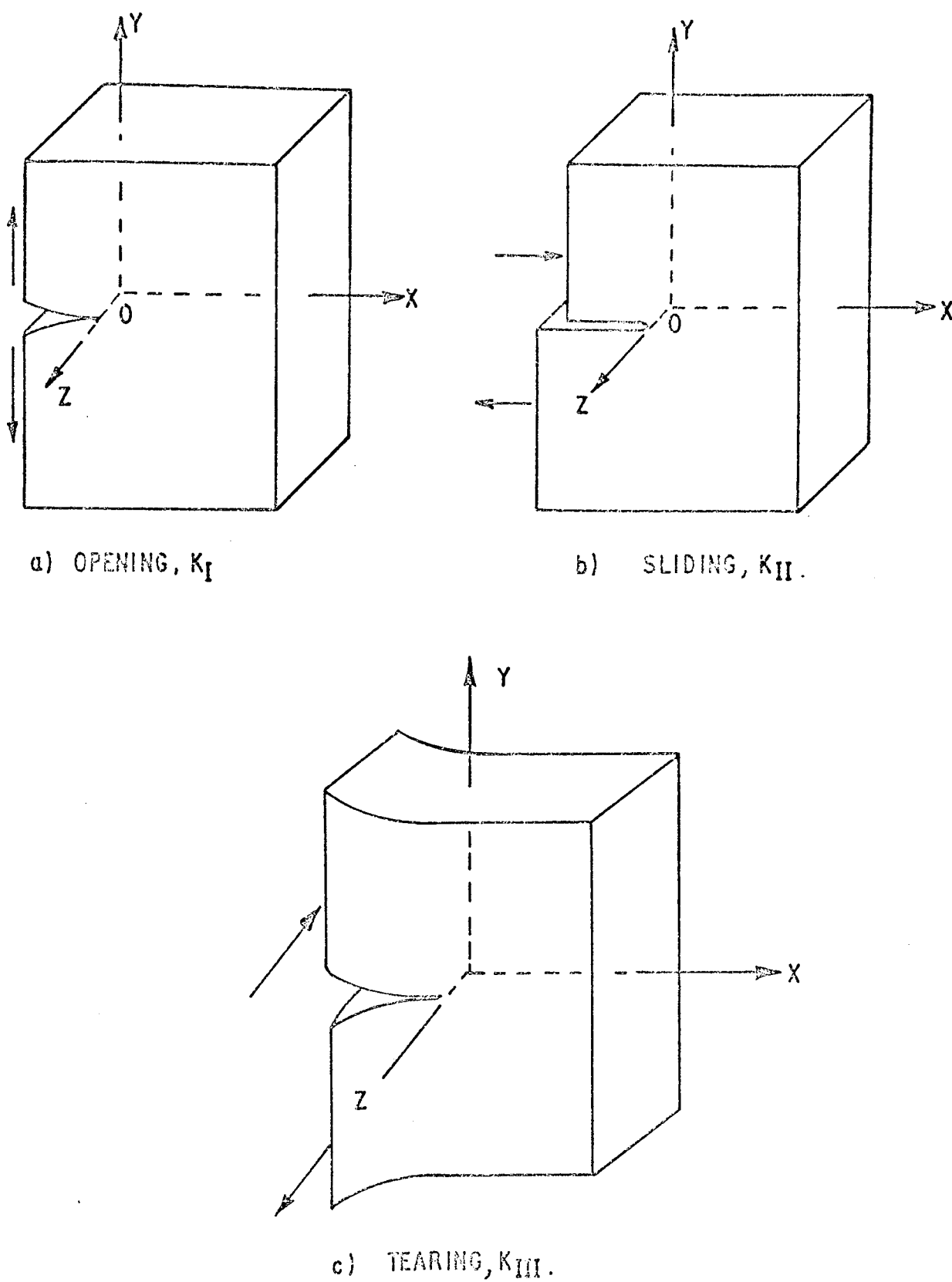


FIG. 2.3

BASIC MODES OF CRACK EXTENSION.

density factor, S_c , is an intrinsic material property independent of the loading conditions and crack configurations, and S_c governs the onset of crack propagation. The strain energy density factor, S_d ,

$$S_d = a_{11} K_I^2 + 2a_{12} K_I K_{II} + a_{22} K_{II}^2 + a_{33} K_{III}^2$$

where a_{ij} ($i, j = 1, 2, 3$) are known functions of the elastic constants of the material and the angle of crack extension, θ , measured with the crack plane. The value of θ is found by minimising the strain energy density factor with respect to θ ($\frac{dS_d}{d\theta} = 0$). These theories are examined in detail by Shah⁽⁶²⁾, who concludes from his experiments on plane problems (K_I and K_{II} only) that the presence of K_{II} can have a significant effect on the K_I at which fracture occurs and vice versa.

The separate evaluation of each stress intensity factor is, therefore, of considerable importance for the application of the principles of linear elastic fracture mechanics to general cracked configurations. This thesis is concerned with plane problems where the relevant stress intensity factors are K_I and K_{II} . A technique based on a standard finite element program will later be presented which allows the separate determination of both K_I and K_{II} .

2.5 ROLE OF PLASTICITY

Thus far it has been assumed that a material behaves in an elastic/brittle manner with no plastic flow; a physically inappropriate model considering the singular equation 2.2. The most useful of all structural materials, metals and their alloys, have the ability to flow plastically in regions of high stress concentration such as a crack tip. In such regions plastic flow is almost certain to take place although this flow may be restricted to a small area round the crack tip. Indeed, it is this limited ductility which is responsible for the toughness of certain high strength steels. The theories of fracture must be modified to some extent to allow for such plastic deformation.

Consider firstly the Griffith theory; Irwin⁽¹⁶⁾ and Orowan⁽¹⁷⁾ independently analysed the way in which the work of plastic deformation could be incorporated into the Griffith formula for crack propagation. Essentially, they proposed that an additional term, U_p , representing the plastic work done during the actual loading path, should be included in the expression for potential energy, i.e. Total Potential Energy = $P_\sigma + U_e + U_p$.

where P_σ is the potential energy of the applied loads and

U_e is the stored internal elastic strain energy.

Also, the surface energy term γ must be modified to include the work done against plastic deformation

$$\gamma = \gamma_e + \gamma_p,$$

where γ_p , the work done in plastic deformation/unit area of crack extension, can often greatly exceed the true surface energy γ of unit area of separated surface.

Secondly, considering the stress intensity factor approach: plastic flow around the tip of a crack would tend to relax the singular nature of the stress field upon which the stress intensity factor concept is based. In order that linear elastic stress intensity factors still form a useful measure of the intensity of the surrounding elastic field, the yielded zone at the tip must be small compared to characteristic geometric dimensions such as crack length, uncracked ligament width, etc. The approach is generally referred to as 'Small-Scale Yielding Fracture Mechanics' where the elastic stress intensity factor still controls the local deformation field immediately beyond the yielded zone.

Irwin⁽¹⁸⁾, proposed a simple model to allow for yielding at a 'crack tip'. He visualised the plastic zone as consisting of a plastic cylinder surrounding the crack tip, the radius of which for Mode I loading is proportional to ;

$$\left(\frac{K_I}{\sigma_{ys}} \right)^2 \quad (2.5)$$

where σ_{ys} is the material yield stress.

The crack was assumed to be extended by an amount equal to the radius of this plastic zone giving a fictitious crack length

$$a' = a + r_y \quad (2.6)$$

where r_y , the plasticity correction factor is obtained from an equation similar to equation 2.5.

Actual plastic zone sizes and shapes can be vastly different from this simple model^(3, 9, 10) and their influence in the fracture process is still not fully understood. However, in spite of the non-linearity caused by plastic flow, the small scale yielding solutions have been found to be highly accurate approximations up to substantial fractions of the general yielding loads⁽⁶⁴⁾.

2.6 DETERMINATION OF STRESS INTENSITY FACTORS

The basic principles of linear elastic fracture mechanics have been laid in the preceeding sections where the stress intensity factor has been identified in the form ;

$$K_N = Y_N \sigma \sqrt{a} \quad (2.7)$$

where $N = I, II, III$ refer to the modes of crack extension

Y_N ($N = 1, 2, 3$) is a geometrical factor which accounts for such factors as the orientation and shape of a crack, the proximity of boundary surfaces or other cracks.

It is with the determination of Y_N that most crack analysis methods are concerned. Once the values of K_N have been determined they can be used to calculate critical stress levels or critical defect sizes from a knowledge of the material toughness K_{Nc} or to give a measure of the reserve of strength in a structure with some pre-existing crack.

The work described in this thesis is concerned with two-dimensional plane problems where the only relevant fracture parameters are K_I and K_{II} .

Methods available for the determination of stress intensity factors can be classified under the general headings ;

1. Analytical
2. Experimental
- and 3. Approximate.

These three methods will be described below. The analytical and experimental methods will be touched on relatively briefly to highlight their relative merits or shortcomings. Approximate methods will, however, be treated in some detail since it is in this group that the finite element method falls.

2.6.1 Analytical Methods

Analytical methods are taken to be those which satisfy all the relevant equations and boundary conditions and result in closed form solutions. Such methods have the advantage of leading to explicit expressions for stress intensity factors but they require considerable mathematical expertise for a successful solution and are usually restricted to fairly idealised configurations. An extensive review of these methods is beyond the scope of this thesis and such reviews may be found in Refs. (3, 19-23). However, two of the more notable analytical methods are discussed briefly here. These methods are based on the idea of a stress function introduced by G.B. Airy.

Airy's Stress Function :

For problems of plane strain with respect to the xoy plane, the displacement components are

$$u_1 = u_1(x, y), \quad u_2 = u_2(x, y) \text{ and } u_3 = 0$$

The equilibrium and compatibility equations of elasticity theory⁽⁶³⁾, reduce to

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} = 0$$

$$\frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} = 0$$

(2.8)

$$2 \frac{\partial^2 \epsilon_{xy}}{\partial x \partial y} = \frac{\partial^2 \epsilon_{xx}}{\partial y^2} + \frac{\partial^2 \epsilon_{yy}}{\partial x^2}$$

Equilibrium is satisfied automatically by writing the stress components in terms of the Airy stress function, $U = U(x, y)$;

$$\sigma_x = \frac{\partial^2 U}{\partial y^2}, \quad \sigma_y = \frac{\partial^2 U}{\partial x^2}, \quad \tau_{xy} = -\frac{\partial^2 U}{\partial x \partial y} \quad (2.9)$$

when body forces are absent.

Writing strains in terms of stresses from the material stress-strain relations, satisfaction of the compatibility conditions leads to the requirement that U is biharmonic. That is

$$\nabla^2 (\nabla^2 U) = 0 \quad (2.10)$$

The same equations result for generalised plane stress situations providing the stress and strain components are taken as through thickness averages and σ_z is assumed equal to zero.

Westergaard's Stress Functions:

Westergaard⁽⁶⁾ formulated an Airy stress function for cracked structures, which takes the form :

$$U = \operatorname{Re} \left[\bar{Z}_I \right] + y \operatorname{Im} \left[\bar{Z}_I \right] \quad (2.11)$$

where Z_I is the Westergaard stress function. \bar{Z}_I and $\bar{\bar{Z}}_I$ are defined by

$$\frac{d \bar{\bar{Z}}_I}{dz} = \bar{Z}_I \quad \text{and} \quad \frac{d \bar{Z}_I}{dz} = Z_I \quad (2.12)$$

where $z = x + iy$ is the complex variable.

In terms of the Westergaard stress function the opening mode stress intensity factor for a crack tip situated at $z = a$ becomes

$$K_I = \sqrt{2} \lim_{z \rightarrow a} \left\{ \sqrt{z - a} Z_I \right\} \quad (2.13)$$

The method can be extended to the other crack modes⁽¹⁹⁾. Application of the technique requires a suitable expression for the stress function Z_I selected with reference to the particular problem under analysis. Several stress functions are available for a variety of different problems⁽¹⁹⁾.

Complex stress functions :

The Airy stress function U may be written⁽²⁷⁾ in terms of two complex functions $\phi(z)$ and $\psi(z)$ as

$$U = \operatorname{Re} \left[\bar{z} \phi(z) + \int \psi(z) dz \right] \quad (2.14)$$

From equations 2.14 and 2.9,

$$\sigma_x + \sigma_y = 4 \operatorname{Re} \left[\phi'(z) \right] \quad (2.15)$$

Using equation 2.15 with the near crack tip field equations of Appendix 10.B, then,

$$K_I = \sqrt{\frac{r}{2}} \frac{\lim_{r \rightarrow 0} (\sigma_x + \sigma_y)}{\cos \frac{\theta}{2}} \quad (2.16)$$

$$K_{II} = -\sqrt{\frac{r}{2}} \frac{\lim_{r \rightarrow 0} (\sigma_x + \sigma_y)}{\sin \frac{\theta}{2}}$$

This method of determining stress intensity factors has the advantage over the Westergaard method in that it allows conformal mapping of cracks into circular holes⁽³⁾. It is then simpler to select suitable forms for the complex functions $\phi(z)$ and $\psi(z)$.

2.0.2 Experimental Methods

There are two major experimental techniques used for the determination of stress intensity factors: Compliance Calibration and Photoelastic Analysis.

Compliance Calibration :

When a structure supports a simple point load, P , the energy release rate G_I may be written in terms of the change in load point displacement δu , as

$$G_I = \frac{1}{2} \frac{P}{B} \frac{\delta u}{\delta a} \quad (2.17)$$

where B is the thickness of the plate and δa is the change in crack length, a ⁽¹⁰⁾.

For elastic problems,

$$u = C P \quad (2.18)$$

where C is the reciprocal of the slope of the load-deflection curve at a particular value of the crack length and is known as the compliance of the system. For constant loading, equation 2.17 becomes in the limit as

$$\delta a \rightarrow 0,$$

$$G_I = \frac{1}{2} \frac{P^2}{B} \left(\frac{dC}{da} \right)_a \quad (2.19)$$

The value of the stress intensity factor may then be calculated from equations 2.3 or 2.4.

For an experimental evaluation of K_I ; the compliance of a specimen is measured over a range of crack lengths, values of dC/da are found from a graph of compliance versus crack length and G_I and hence K_I may be calculated from equations 2.19 and 2.3 or 2.4 as appropriate.

Photoelastic Analysis :

In this well-known technique for experimental stress analysis, a transparent model made of a suitable birefringent material is examined in a polariscope. Details of the stress distribution around the crack tip may be obtained from the resulting fringe patterns. Stress intensity factors can then be found by comparing these stress values with those

obtained from the analytical expansions for the near crack tip stress field, Appendix 10.B.

The technique may be used for the analysis of mixed mode and dynamic fracture problems. An extensive bibliography of the various applications may be found in Ref. (20).

While these experimental techniques are valuable for particular cases, analysis is a tedious and difficult process involving the careful preparation of specimens and interpretation of results for each example.

2.6.3 Approximate Methods

Several approximate numerical methods are available for calculating stress intensity factors. These include Boundary Collocation, the numerical solution of integral equations and Finite Elements, details of others are discussed in Refs. (19, 25). Suitable approximations have been applied in order to solve problems having more complex component shapes than would be possible by analytical methods. Typical of the kind of approximations applied are the truncation of an infinite series in the case of Boundary Collocation or the idealisation of an infinite degree of freedom continuum into a finite degree of freedom system in the case of the Finite Element Method. The finite element method forms the basis of investigations described in this thesis and, naturally, its application to the determination of stress intensity factors is described here. However, many important results have been obtained using boundary collocation so that a brief account of the technique, for contrast with finite element work, is appropriate.

2.6 .3.1 Boundary Collocation

The method of analysis consists in finding an Airy stress function U that satisfies the biharmonic equation, equation 2.10, and also the necessary boundary conditions on the crack boundary surface. This stress function takes the form of an infinite series with unknown coefficients.

The series is truncated and the remaining coefficients are determined from the conditions that the stress function and its derivative must satisfy at a finite number of selected boundary stations. Boundary values of the stress function and its derivative are often well-known⁽⁶³⁾ although for most problems they can be determined by inspection. The number of undetermined coefficients can be equal to or less than twice the number of boundary stations. If it is less, then an overdetermined system of simultaneous equations results, and this set of equations is solved by a least squares best fit technique. This procedure has the advantage of introducing smoothing into the numerical analysis.

If the Williams' stress function⁽²⁶⁾ is used then the stress intensity factors K_I and K_{II} are found from the even and odd singular terms of the series respectively.

Stress intensity factors may also be determined from series representation of the complex functions $\phi(z)$ and $\psi(z)$ of Muskhelishvili⁽²⁷⁾.

The procedure has been employed extensively by Gross et al⁽²⁸⁻³⁰⁾ for edge cracks and in a modified form by Kobayashi et al⁽³¹⁾ for internal crack problems. There is no guarantee of numerical convergence and accuracy is judged largely by insensitivity of results to the inclusion of more terms in the series. The selection of the number and distribution of the boundary points also plays an important role in the accuracy of the final solution⁽²⁵⁾.

2.6.3.2 Finite Element Method

The finite element method of stress analysis has been discussed extensively in Section I, where the development of a program for two-dimensional plane stress or plane strain elastic analysis was described. The element employed was the six-node isoparametric triangle, formulated such that curved boundaries could be modelled very accurately. The method

is a powerful tool for stress analysis capable of handling arbitrary component geometries to which various boundary conditions are easily applied. Arbitrary crack configurations can be modelled in the finite element idealisation and various implementations have been developed for treating the singular conditions at the crack tip. The technique has been extended to three-dimensional problems, elastic/plastic interface problems, etc.⁽²⁵⁾

The finite element method is, therefore conceptually suited to fracture analysis where general specimen shapes and loading conditions must be considered. Indeed, although the classical approaches previously described have been and continue to be appropriate for the determination of stress intensity factors in the presence of simple geometries the finite element method has come to the fore for more complicated situations⁽²⁰⁾.

The principal techniques for applying the finite element method to fracture analysis are discussed below and there are a number of general survey papers available on similar topics⁽³²⁻³⁴⁾. It is convenient to divide the methods of application into two groups, depending on how the crack tip singularity is represented by the finite element mesh. In the first group, no special provision is made for the crack tip singularity in formulating the problem and stress intensity factors are calculated from the results of a standard finite element program. In the second group, some explicit provision is made for the singularity in the idealisation and usually these methods make use of results from analytical solutions for the singular stresses and displacements at a crack tip and incorporate these distributions into special crack tip elements.

Methods Based on a Standard Finite Element Program

(a) Crack Tip Stress

In this method a mesh of ordinary finite elements is constructed about the crack tip for which the stress intensity factors are required.

As no special provision has been made for the crack tip singularity, in the assumption of the element shape functions, a large number of such elements are required for a reasonable estimate of the near tip stress field. In order to determine the stress intensity factors, the nodal point stresses obtained from the finite element program are correlated with the crack tip stresses given by an equation of the form, Appendix 10.A,

$$\sigma_{ij}(r, \theta) = \frac{K_N}{\sqrt{2r}} f_{ij}(\theta) \quad i, j = 1, 2, \quad N = 1, 2, 3$$

The stress intensity factors may be obtained by a simple rearrangement of the above equation and substitution of the appropriate values of the stress components $\sigma_{ij}(r, \theta)$ obtained from the finite element results. By a proper choice of the stress component, σ_{ij} , and θ , Fig. 2.2, each stress intensity factor may be determined separately. For example, in equation 10.4 at $\theta = 0^\circ$, σ_x is a function of K_I only and at $\theta = 90^\circ$, σ_x is a function of K_{II} only.

The method was employed by Chan et al⁽³⁵⁾, Kobayashi et al⁽³⁶⁾ and Watwood⁽³⁷⁾ for the calculation of opening Mode I stress intensity factors. Results obtained were generally unsatisfactory when considering the large number of degrees of freedom required in the idealisation for any reasonable solution accuracy. A similar rationale to the above stress method, to be described subsequently, based on a near tip displacement solution provides more accurate results. This is understandable since the nodal point displacements are the primary unknowns and stresses are obtained by differentiation of these displacements which reduces accuracy.

(b) Crack Tip Displacement

This method is somewhat similar to the crack tip stress method previously described, in this case, however, the displacements yielded by an ordinary finite element approach are used. In order to determine values for the stress intensity factors, these displacements are

correlated with the crack tip displacement equations, Appendix 10.A,

$$u_i = \frac{K_N}{G} \sqrt{\frac{r}{2}} g(\theta, \mu) \quad N = 1, 2 \quad (2.20)$$

The stress intensity factors may be found by a simple re-arrangement of the above equation. Consider, for example, the determination of K_I from the symmetric displacement (u_y) of the crack at a point near the crack tip. Upon solving equation 2.20 for K_I ,

$$K_I = \frac{G u_y}{\sqrt{\frac{r}{2}} g(\theta, \mu)} \quad (2.21)$$

Following the procedure of Chan et al⁽³⁵⁾ a value of K_I is obtained by substituting a finite element method y displacement component at $\theta = \pi$, Fig. 2.2 into equation 2.21. If the displacements obtained from the finite element program were exact, then a single value of K_I would be obtained. As this is not realised, a curve of K_I^* versus r is obtained rather than a single exact value for K_I . In the examples considered by Chan et al such a curve appeared to become linear some distance away from the crack tip; the distance at which the curve became linear decreased with the fineness of the mesh. The linear portion of the curve was then extrapolated to the crack tip (note, there is no theoretical justification for this) and it was this extrapolated value which was taken as the stress intensity factor, K_I . For the finest mesh used (1500 degrees of freedom) the estimate obtained was approximately 5% below the theoretical value; for coarser meshes, estimates were about 9% below the theoretical value.

The procedure has also been applied by Watwood⁽³⁷⁾ and Anderson et al⁽³⁸⁾ who considered the influence of mesh design in the crack tip region. Kobayashi et al⁽³⁶⁾ calculated both Mode I and Mode II stress intensity factors by this procedure and Miyamoto et al⁽³⁹⁾ extended the principles to the three-dimensional problem of a semi-elliptical surface flaw in a finite thickness plate.

The method has received considerable attention, probably in part due to its conceptual simplicity, nevertheless it requires very fine meshes in the crack tip region for a reasonable accuracy to be obtained. Computer core requirements and execution times are correspondingly high.

Oglesby and Lomacky⁽³³⁾ recently considered a refinement of the technique by retaining more terms in the series expansions for the displacements in the crack tip region. The range of validity of the fracture mechanics formula is extended, therefore, into the region where the finite element displacements are more reliable, i.e. away from the high strain gradients near the crack tip. By taking displacements at pairs of adjacent nodes, the authors were able to solve for the series constants K_I , K_{II} , α_1 and α_2 (Appendix 10.A.). Estimates of stress intensity factors of the order of 1 to 5% below the theoretical values were obtained for meshes with much fewer degrees of freedom than were used in the studies of Chan et al.

(c) Energy Methods

The three main energy methods to be discussed below are :-

- (i) Strain Energy (Compliance),
- (ii) Line Integral, and
- (iii) Crack Opening Energy.

Energy methods have the advantage that they do not rely on the accuracy of the localised crack tip stresses, strains or displacements obtained from a finite element program. Consequently, for the same solution accuracy, the energy methods allow coarser grids to be used than were admissible for the displacement methods.

(c.i) Strain Energy (Compliance) Method

It was shown by Irwin⁽⁴⁰⁾ and has been discussed previously in section 2.3, that the crack tip stress intensity factor can be related to the strain energy release rate, G , by equations 2.3 and 2.4, for Mode I loading. The strain energy release rate is the amount of energy per

unit area extension of the crack made available at the crack tip for the crack extension process. From section 2.1 for the fixed-load condition ;

$$G = \frac{\delta U_e}{\delta a}$$

By obtaining finite element solutions for the cracked structure of interest at slightly different crack lengths, G and subsequently the crack tip stress intensity factor can be estimated. The strain energy stored in the body for each solution can be determined from the work done on the structure (movement of applied loads) or by the summation of the energy in all the elements⁽³⁷⁾.

This method has been developed and successfully used by Watwood⁽³⁷⁾, Kobayashi⁽³⁶⁾ and Anderson et al⁽³⁸⁾, good accuracy being obtained by the use of relatively coarse meshes. While the strain energy calculated for a particular crack length may not be very accurate for a coarse mesh, there is a cancelling of errors when taking the difference in strain energy at two different crack lengths. For this reason, as pointed out by Watwood, it is important that the coarseness of the mesh be approximately the same for each crack length. Watwood has considered bounds for calculating stress intensity factors by this method and also explains how modes may be separated by appropriately constraining crack surface nodes for energy computation.

As shown by Paris and Sih⁽³⁾ and discussed in section 2.6.2, the energy release rate can also be related to the rate of change of compliance, C (the inverse spring constant, defined in equation 2.18) with crack extension by the equation

$$G = \frac{P^2}{2} \frac{dC}{da} \quad (2.22)$$

where P is the applied load.

Dixon and Pook⁽⁴¹⁾ found the compliance of a centre cracked plate for a number of different crack lengths and then by means of equation 2.22 calculated the strain energy release rate. The approach is equivalent to the strain energy approach described above.

The major objection to the method is that it requires numerical differentiation of the strain energy (compliance) versus crack length curve, a process that can introduce large errors. For the calculation of K_I the square root of the derivative is required and the error in K_I is approximately one half the error in the derivative but still may be significant. Another restriction on the method's range of applicability arises when considering crack face loading, the accuracy of the energy calculation for such loading is now heavily dependent on displacements near the crack tip, where displacement gradients are high and the solution by the finite element method is less accurate.

(c.ii) Line Integral Method

Rice⁽⁴²⁾ introduced a path-independent integral, J , defined by :

$$J = \int_{\Gamma} \left(U \, dy - \bar{T} \cdot \frac{\partial \bar{u}}{\partial x} \, ds \right) \quad (2.23)$$

where Γ is an arbitrary contour surrounding the crack tip with outer normal \bar{n} (Fig. 2.4)

U is the strain energy density, for linear elastic behaviour,

$$U = \frac{1}{2} \sigma_{ij} \epsilon_{ij}$$

\bar{T} is the traction vector along the outward normal to the contour,

$$T_i = \sigma_{ij} n_j$$

\bar{u} is the displacement vector on an arc element ds along the arc s .

The line integral is evaluated in a counter-clockwise direction starting at the lower crack surface and continuing along the path Γ to the upper crack surface. In the linear elastic case, Rice has shown^(9, 42)

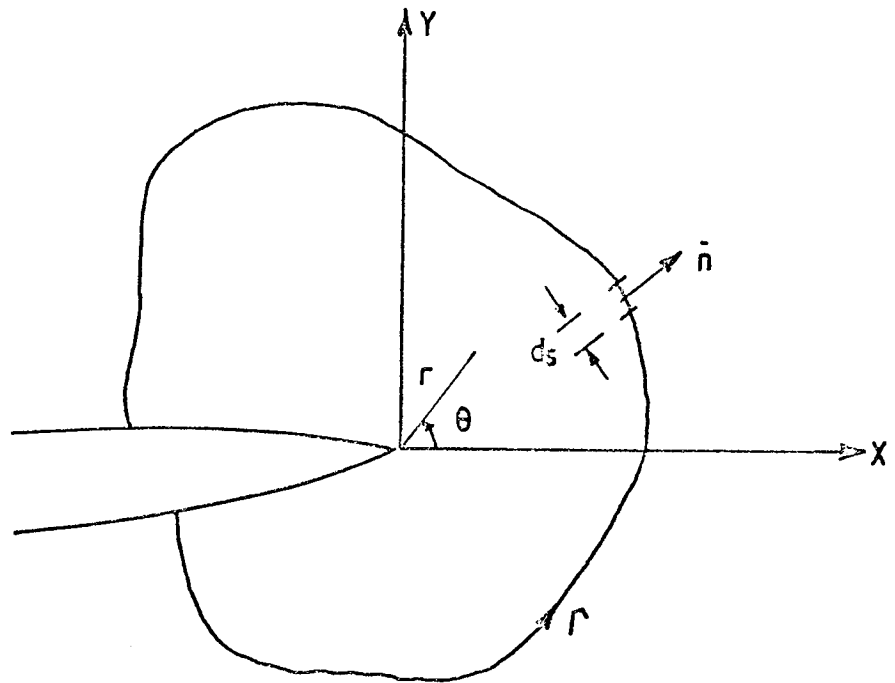


FIG 2.4 PATH OF INTEGRATION FOR DETERMINING THE J - INTEGRAL.

that J is equal to the strain energy release rate, G , hence for Mode I calculations K_I can be determined from equations 2.3 and 2.4.

The line integral method has been applied by Chan et al⁽³⁵⁾ and Leverenz⁽⁵³⁾ who evaluated Mode I stress intensity factors.

For combined mode situations, the relations between J and the stress intensity factors are⁽⁹⁾, for plane strain :

$$J = \frac{(1 - \nu^2)}{E} (K_I^2 + K_{II}^2) \quad (2.24)$$

and for plane stress,

$$J = \frac{1}{E} (K_I^2 + K_{II}^2) \quad (2.25)$$

It is seen that in mixed mode situations, the magnitudes of the individual components K_I and K_{II} cannot be calculated from the J value, only the sum of their squares.

(c.iii) Crack Opening Energy Method

Hayes⁽⁴³⁾ applied the energy integral of Bueckner⁽⁴⁴⁾ for the calculation of opening Mode I stress intensity factors. The method involves the evaluation of the energy, U , per unit thickness, released by introducing a crack, calculated from

$$U = -\frac{1}{2} \int_s \sigma_{ij} u_i ds \quad (2.26)$$

where σ_{ij} is the distribution of stress on the prospective crack surface,

u_i is the displacement of the crack face due to σ_{ij} acting on the crack face. All other boundaries are stress free.

s is an integration path over the entire crack surface.

The stress and displacement components σ_{ij} and u_i are determined from a double application of the finite element program :

- (a) to determine the distribution of stress σ_{ij} on the prospective crack surface due to the applied loads,
- and (b) to determine the crack surface displacements u_i resulting from the stresses found in step (a).

Finally, the energy released is calculated by evaluating the integral of equation 2.26, using the results of steps (a) and (b).

The stress intensity factor is determined by evaluating U at a number of crack lengths, numerically differentiating U with respect to crack length to get strain energy release rate, G , from which the stress intensity factors may be calculated.

The method can be applied to problems involving thermal stresses, body forces, residual stresses and mixed mode deformations. A standard finite element program may be employed but the procedure is cumbersome with the attendant errors of numerical differentiation.

To conclude this brief summary of methods based on a standard finite element program it is instructive to consider convergence of the finite element solution in the presence of geometric singularities such as a crack.

In general applications of the displacement finite element method, when compatibility and constant element strain conditions are satisfied, convergence can be proven and the rate of convergence can be established⁽⁴⁵⁾. Such convergence criteria are not applicable when the exact solution contains singularities, i.e. monotonic convergence is not assured. An analysis by Tong et al⁽⁴⁶⁾ reveals that the convergence rate is dominated by the nature of the solution near the points of singularity and errors in elements adjacent to the point of singularity are of the same order as those of the remainder of the elements. Therefore, the effect of the singularity is not restricted to elements immediately adjacent to it but

extends over a finite region around such a singularity. This author also concludes that higher order elements do nothing to improve the rate of convergence. These conclusions raise doubts as to the accuracy and even validity of the direct methods previously described for the analysis of the singular crack problem. It thus seems expedient to make special provision for a singularity to eliminate the problem of convergence encountered using conventional elements.

Methods Based on Singular Crack Tip Elements

The analytical form of the crack tip singularity is well known, Appendix 10.A. It is possible, therefore, to construct special crack tip finite elements which have the singular condition built into their imposed displacement patterns. Elements of this type eliminate the need for very fine meshes near the crack tip and are more elegant from a mathematical viewpoint when compared with conventional elements at the crack tip. Of the many proposals for special crack tip elements presented in the literature, a survey of which is contained in Ref.(19), only three of the most widely applied will be discussed here.

(a) Byskov Element

Byskov⁽⁴⁷⁾ developed an element of polygonal form which contains the crack tip; the element is suitable for both Mode I and Mode II deformation. The specific polygonal form he developed and applied was triangular in shape with a straight crack extending from one vertex of the triangle into its interior, Fig. 2.5. The displacement terms used for the element were obtained by expressing the complex stress functions of Muskhelishvili⁽²⁷⁾, equation 2.14, in terms of a power series. These singular trial functions satisfy the equilibrium conditions within the element as well as the vanishing of tractions on the crack faces but not the compatibility conditions on the boundaries with the surrounding constant strain elements.

The number of undetermined coefficients retained in the truncated power series is equal to the number of degrees of freedom of the element, i.e. eight coefficients for the triangle of Fig. 2.5.

The stiffness matrix of the cracked element is expressed in terms of the nodal displacements using the truncated power series and the values of the stress intensity factors K_I and K_{II} are obtained directly from the computer solution together with the values of the nodal displacement components for the mesh.

Byskov obtained solutions for a centre cracked and single-edge cracked plate under remote uniform and concentrated loading. For the finest mesh with 521 degrees of freedom, accuracy was of the order of 3 to 4 percent.

The method appears to be fairly accurate with the added computational convenience of obtaining K_I and K_{II} directly as output from the computer program. Crack face loading can also be handled by modifying the Muskhelishvili stress functions used by Byskov. With the Byskov element however, only three conventional constant strain triangular elements surround its boundary and the influence of interelement incompatibility cannot be controlled.

(b) Tracey Element

Tracey⁽⁴⁶⁾ proposed a method, based on isoparametric quadrilateral elements, for the calculation of Mode I stress intensity factors. Two types of element were employed in the analysis :

- (a) singular near crack tip elements which have only one nodal point at the crack tip. The crack tip is enclosed by a group of these special elements, Fig. 2.6.
- and (b) conventional quadrilateral isoparametric elements which surround the special elements of Fig. 2.6.

The essential aspects of the two types of elements may be described as follows. Fig. 2.7a shows a typical quadrilateral element located in the x - y plane. A set of natural coordinates ξ and η is chosen together with the transformation function connecting coordinates x , y and ξ , η , in the form :

$$\begin{aligned} x &= x_A (1-\xi) \eta + x_B (1-\xi)(1-\eta) + x_C \xi (1-\eta) + x_D \xi \eta \\ y &= y_A (1-\xi) \eta + y_B (1-\xi)(1-\eta) + y_C \xi (1-\eta) + y_D \xi \eta \end{aligned} \quad (2.27)$$

Here, x_A , y_A , denote the coordinates of node A in the x , y plane, etc. As shown in Figs. 2.7a and 2.7b, such a transformation maps either the quadrilateral A-B-C-D or the triangle A_1 B-C-D located in the physical x,y plane into rectangles in the ξ , η plane. For the triangle the edge A_1 -B maps into a single point A_1 B in the physical x,y plane with $x_A = x_B$ and $y_A = y_B$.

On the assumption that the crack is located at $x \leq 0$, the displacement function within the singularity element is chosen in the form,

$$\begin{aligned} u(\xi, \eta) &= b_1 + b_2 \sqrt{\xi} + b_3 \sqrt{\xi \eta} + b_4 \eta \\ v(\xi, \eta) &= b_1^1 + b_2^1 \sqrt{\xi} + b_3^1 \sqrt{\xi \eta} + b_4^1 \eta \end{aligned} \quad (2.28)$$

where $u(\xi, \eta)$, $v(\xi, \eta)$ are the displacement components within elements in the x - and y - directions respectively and b_i , b_i^1 ($i = 1, 2, 3, 4$) can be expressed in terms of the eight element nodal point displacement components. For the singularity elements located at the crack tip, the auxiliary constraint is introduced in the form $u_A = u_B$, $v_A = v_B$, to express the condition that the two distinct points A and B in the ξ , η plane correspond to the same point (the crack tip) in the physical x , y plane. The displacement field expressed in equation 2.28 reproduces the radial, r , dependence of equation 10.2 although the smooth angular, θ , variation is not duplicated. The constants b_i , b_i^1 of equation 2.28 are not directly expressible in terms of the stress intensity factors K_I and K_{II} .

The isoparametric type of element in the form of a general quadrilateral ABCD, Fig. 2.7a, has the displacement function specified in the form :

$$\begin{aligned} u(\xi, \eta) &= b_1 + b_2 \xi + b_3 \eta + b_4 \xi \eta \\ v(\xi, \eta) &= b_1^1 + b_2^1 \xi + b_3^1 \eta + b_4^1 \xi \eta \end{aligned} \quad (2.29)$$

Such isoparametric elements can be joined with the singular elements (having the displacement field described by equation 2.28) in such a way that there is no displacement incompatibility along the element boundary.

In an application of the technique, the singular elements are constructed around the crack tip as shown in Fig. 2.6, which are themselves surrounded by the quadrilateral elements. Values for the element stresses and nodal point displacements are obtained from a finite element program adapted to include these elements. Finally, stress intensity factors may be computed from these stresses and displacements using the direct methods previously described in section 2.6.3.2.

Tracey obtained solutions for two geometries: a circumferentially notched bar and a double edge notched rectangular bar under uniform remote tension. In the case of the edge notched rectangular bar an accuracy of 5% was obtained with a mesh having 248 degrees of freedom.

The method is similar to one due to Wilson⁽⁴⁹⁾ who analysed anti-plane mode III deformation and the principles have recently been extended⁽⁵⁰⁾ to three dimensional studies of combined modes I and II where each stress intensity factor is allowed to vary across the crack front.

Singular elements are employed only to increase the accuracy of the finite element solution, stress intensity factors are not produced as output from the program. It is, therefore, a fairly tedious procedure involving considerable program modification to include the singular elements.

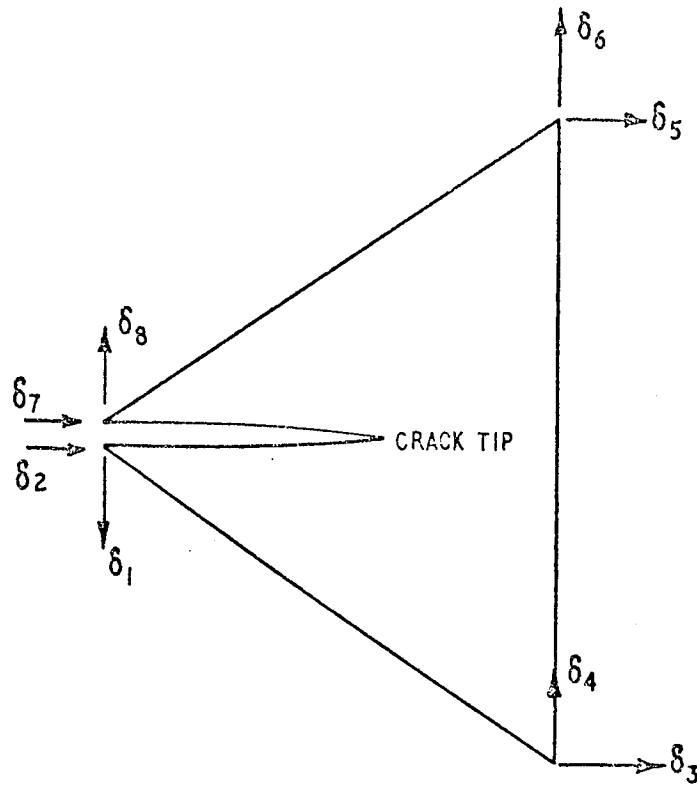


FIG.2.5 BYSKOV ELEMENT.

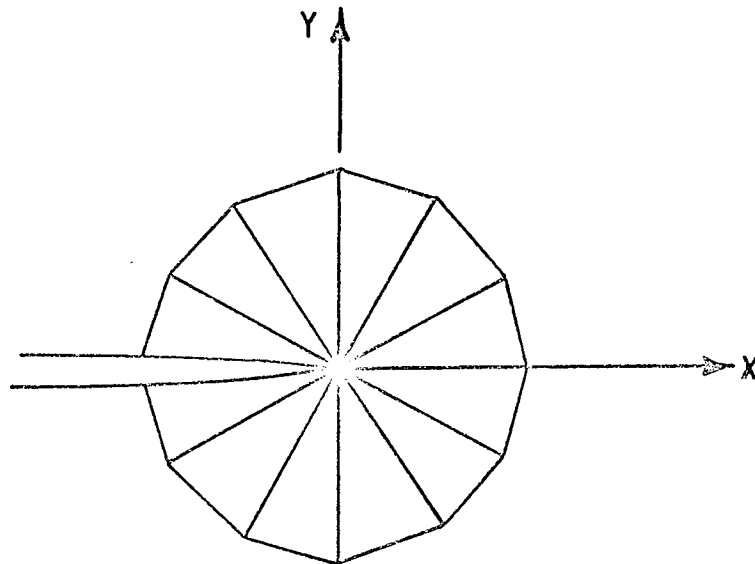


FIG.2.6 TRACEY ELEMENTS.

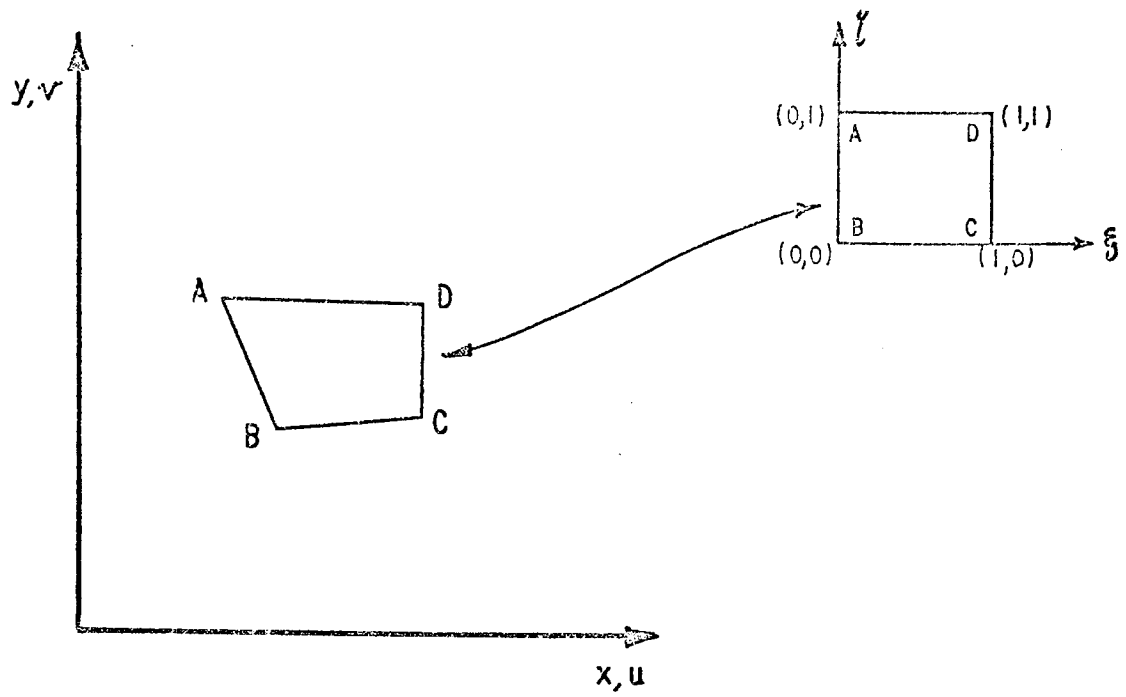


FIG.2.7a TYPICAL QUADRILATERAL ELEMENT MAPPED INTO A SQUARE.

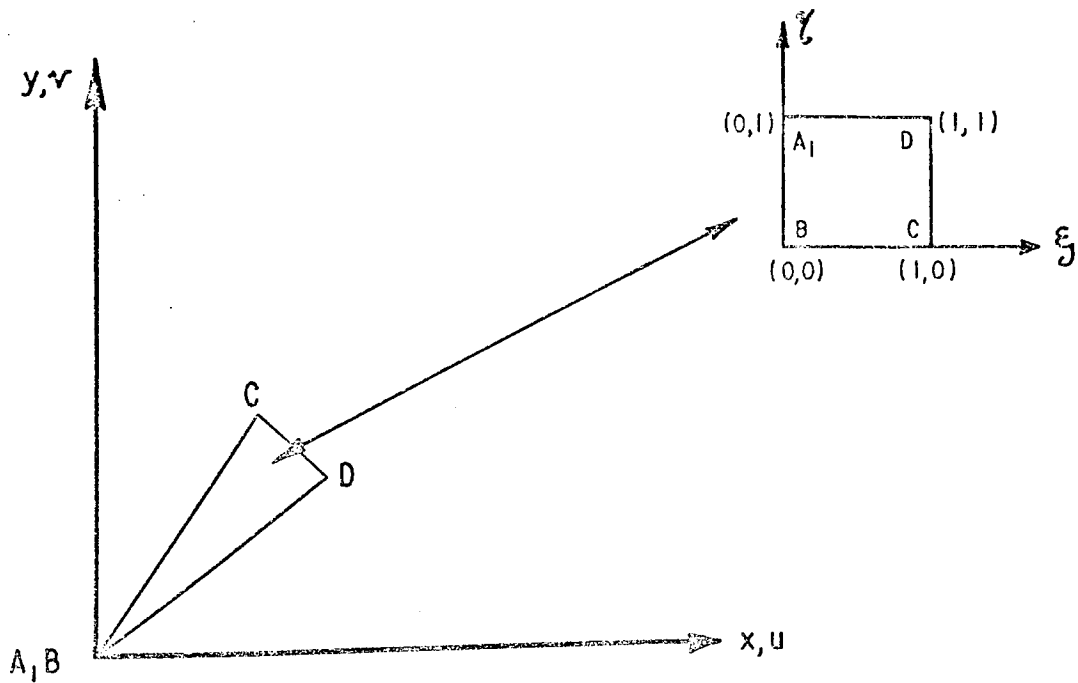


FIG 2.7b TYPICAL NEAR TIP TRIANGLE MAPPED INTO A SQUARE.

(c) Hilton and Hutchinson Element

The method of Hilton and Hutchinson⁽⁵¹⁾ was originally applied to the calculation of Mode I and Mode III plastic stress intensity factors and by Wilson⁽⁵²⁾ for problems of out of plane shear, Mode III. For the case of a centrally cracked infinite plate under uniform applied tension, elastic Mode I stress intensity factors were obtained with an accuracy of within 1% of the theoretical value.

Essentially the procedure involves the embedding of the singular solution, represented by equations 10.5, into a finite element grid. As an example of the procedure, consider a centrally notched plate under uniaxial Mode I loading, Fig. 2.8. By symmetry, only one quadrant of this plate need be analysed, Fig. 2.9. A circular core of radius R_c (defining the boundary Γ_1) is constructed around the crack tip, this is shown in Fig. 2.9 together with the first ring of triangular elements in position around the core.

The displacements within the core region are defined from equation 10.5 with $K_{II} = \alpha_1 = \alpha_2 = 0$, as :

$$u_x = \frac{K_I}{4G} \left(\frac{r}{2}\right)^{\frac{1}{2}} \left[(2K-1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \right] + \delta x \quad (2.30)$$

$$u_y = \frac{K_I}{4G} \left(\frac{r}{2}\right)^{\frac{1}{2}} \left[(2K+1) \sin \frac{\theta}{2} - \sin \frac{3\theta}{2} \right]$$

δx is the rigid body displacement of the core in the x-direction and constitutes one of the parameters to be determined from the solution of the problem.

In addition to specifying the displacement pattern within the core, equation 2.30 also represents constraints on those nodal points of the first ring of triangular elements that fall on the circumference Γ_1 . Next, the potential energy of the quadrant is written in the form :

$$PE = U_{\text{core}} + \sum_{el} SE + \int_s \bar{T} \cdot \bar{u} \, dS \quad (2.31)$$

Here, U_{core} is the strain energy stored in the core, the second term is the strain energy stored in all the elements outside of the core and the last term is the work done by the surface traction vector \bar{T} on the surface displacement vector \bar{u} . The strain energy stored in the core is computed using the singular crack tip stress, σ_{ij} and strain, ϵ_{ij} , components, equations 10.4 and 10.6, as :

$$U_{\text{core}} = \int_{\text{core}} \frac{1}{2} \sigma_{ij} \epsilon_{ij} \, d\text{vol} = \frac{K_I^2 R_c \pi}{32G} \quad (2K-1) \quad (2.32)$$

The stress intensity factor K_I and the rigid body displacement component δ_x therefore become generalised coordinates of the problem together with the nodal displacements u_i (i refers to the u_x or u_y displacement component at node i). Finally, the governing linear algebraic equations to be solved for u_i , K_I and δ_x are obtained from the minimisation of the potential energy, equation 2.31, as :

$$\begin{aligned} \frac{\partial(\text{PE})}{\partial K_I} &= 0 \\ \frac{\partial(\text{PE})}{\partial \delta_x} &= 0 \\ \frac{\partial(\text{PE})}{\partial u_i} &= 0 \quad i = 1, 2, \dots, N \end{aligned} \quad (2.33)$$

Here, N is the total number of unconstrained degrees of freedom, the nodes on Γ_1 are constrained by equation 2.30.

It will be shown in greater detail later in this thesis that the above procedure can be easily incorporated into a standard finite element program. Modification of such a program consists essentially of constraining the nodes on the first ring of elements surrounding the crack tip and adding equations which couple the stress intensity factors and rigid body modes of the core to the displacements of the second ring of elements and thus through the remainder of the elements. The stress intensity factors are obtained directly, together with the nodal displacements, as output from the program.

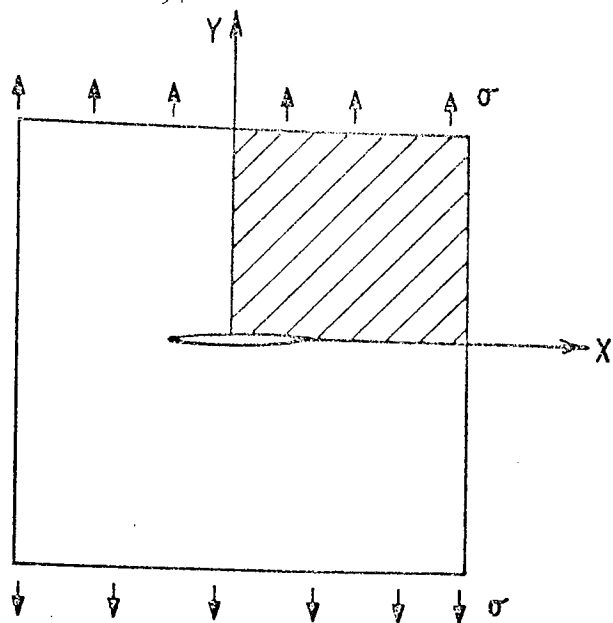


FIG.2.8 CENTRALLY CRACKED RECTANGULAR PLATE.

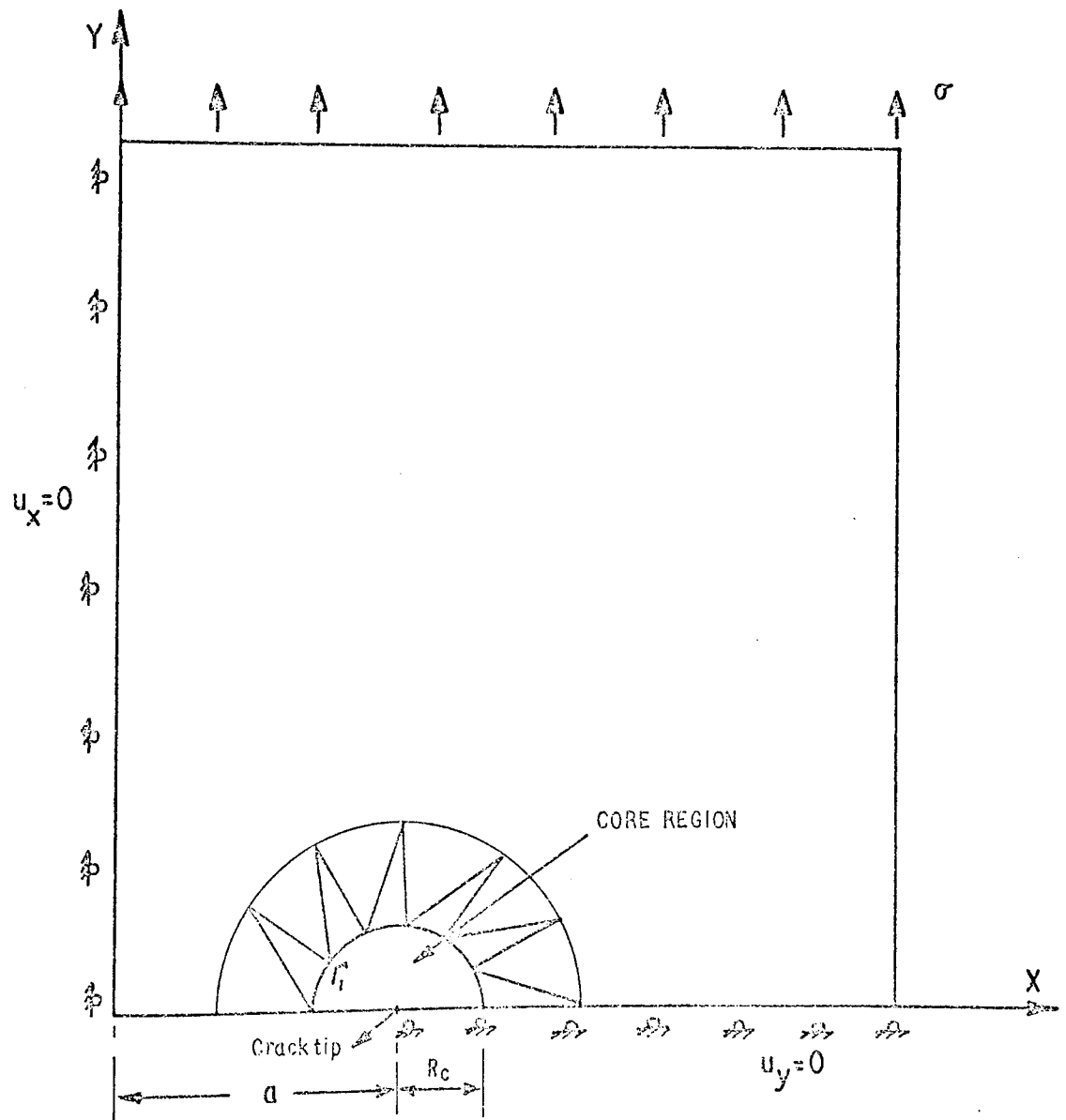


FIG.2.9. QUADRANT OF CENTRALLY CRACKED RECTANGULAR PLATE.

The compatibility of displacements between the nodes on the core/finite element mesh interface Γ_1 is violated since displacements along Γ_1 in the core region vary as trigonometric functions and the displacements in the triangular elements are described in terms of polynomial expansions. Monotonic convergence to the exact solution cannot, therefore, be guaranteed. However, as the number of elements on Γ_1 is increased, thus minimising the angular separation $d\theta$ between nodal points, the difference between the polynomial variation in the mesh and the trigonometric variation in the core will also be minimised. This is borne out by the results obtained in the work reported in this thesis and those of Oglesby⁽³³⁾ in his study of Mode I singularities. By changing the number of nodes on Γ_1 there is at least some control over the incompatibility whereas in the Byskov element there is no control since the element sides are fixed in advance.

The method of Hilton and Hutchinson is appealing for several reasons,

- (a) its conceptual simplicity and ease of implementation into a standard finite element program,
 - (b) separation of K_I and K_{II} for combined mode analyses,
 - (c) arbitrary crack face loading may be applied,
- and (d) its ease of extension to the analysis of plastic or elastic/plastic cores and a variety of material properties.

The technique is extended in this thesis for the analysis of a variety of crack configurations of single (Mode I) and mixed (Modes I and II) types. The elements used in the analysis are isoparametric six-node triangles with quadrilateral displacement functions as opposed to the linear functions of (51).

Several problems are considered in this thesis which have not been previously discussed in the literature. Firstly, the method is extended such that it is capable of analysing an array of cracks where each crack

tip has the opening Mode I singularity, K_I . Secondly, the problem of straight or curved internal cracks is considered. In this case Mode I and Mode II stress intensity factors are present. Finally, with regard to the internal crack problem, partial or full crack surface closure is considered and a numerical model is proposed to correct for this phenomenon.

3. CRACK BORDER STRESS/STRAIN AND DISPLACEMENT FIELDS

3.1 INTRODUCTION

It is the purpose of this chapter to summarise the equations describing the crack tip stress, strain and displacement fields for plane problems of elasticity.

Of the several different methods used for obtaining the stress distributions near the tip of a sharp crack, most rely on the suitable choice of an Airy stress function which satisfies the governing differential equations and the boundary conditions of the problem. Westergaard⁽⁶⁾ used a semi-inverse approach to obtain suitable stress functions, his method was used extensively by Paris and Sih⁽³⁾ in their classic paper on fracture. A particularly lucid description of the Westergaard method is given by Corten⁽⁵⁴⁾ who derives the field equations for all three modes of crack extension. A more direct method of finding the singular crack tip stresses was devised by Williams⁽²⁶⁾ who used eigenfunction expansions as his stress function. The method of Williams was later combined with the complex function theory of Muskhelishvili⁽²⁷⁾ by Sih and Liebowitz^(8, 55) who thus obtained the solutions of many different crack problems. The method employed in this chapter is the same as that of (8), but with extra terms included in the stress function series to give a better approximation to the near tip stress and displacement fields.

The equations of the plane theory of elasticity for the two cases of plane strain and generalised plane stress, have been discussed in Section I, Appendix 8.B. With reference to crack problems, a state of plane strain can be approximated in a specimen whose thickness is large compared with the crack length.

3.2 DERIVATION OF NEAR TIP FIELD EQUATIONS

According to Muskhelishvili⁽²⁷⁾, the state of stress and deformation in plane problems of isotropic elasticity is determined by two complex functions $\phi(z)$ and $\psi(z)$ and their derivatives. The components of stress and displacement are associated to these complex functions by the relations,

$$\sigma_x + \sigma_y = 2[\phi'(z) + \overline{\phi'(z)}] \quad (3.1)$$

$$\sigma_y - \sigma_x + 2i\tau_{xy} = 2[\bar{z} \phi''(z) + \psi'(z)] \quad (3.2)$$

$$2\mu(u_x + i u_y) = K \phi(z) - z \overline{\phi'(z)} - \overline{\psi(z)} \quad (3.3)$$

for Cartesian coordinates, Fig. 3.1, and

$$\sigma_r + \sigma_\theta = 2[\phi'(z) + \overline{\phi'(z)}] \quad (3.4)$$

$$\sigma_\theta - \sigma_r + 2i\tau_{r\theta} = 2e^{2i\theta}[\bar{z} \phi''(z) + \psi'(z)] \quad (3.5)$$

$$2\mu(u_r + i u_\theta) = e^{-i\theta}[K \phi(z) - z \overline{\phi'(z)} - \overline{\psi(z)}] \quad (3.6)$$

for Polar coordinates, Fig. 3.2.

$$\text{Here } K = 3 - 4\nu \text{ for plane strain} \quad (3.7)$$

$$\text{and } K = (3 - \nu)/(1 + \nu) \text{ for plane stress}$$

z refers to the complex number $x + iy = r e^{i\theta}$ ($i^2 = -1$)

The prime denotes the derivative with respect to z and the bar indicates the complex conjugate number. Thus $\bar{z} = x - iy$ and $\overline{\psi'(z)}$ means $\overline{\psi'(z)}$ with i replaced everywhere by $-i$. To construct $\overline{\phi'(z)}$ the derivative is taken as,

$$\phi'(z) = \frac{d}{dz} (\phi(z))$$

and the result is then converted to the conjugate form.

The next step in the solution is to assume suitable forms for the functions $\phi(z)$ and $\psi(z)$. A detailed discussion of the suitability of $\phi(z)$ and $\psi(z)$ in terms of analyticity is contained in the original work⁽²⁷⁾. For the purpose of this analysis, the Goursat functions are taken as suitable forms for the functions $\phi(z)$ and $\psi(z)$ in the same manner as

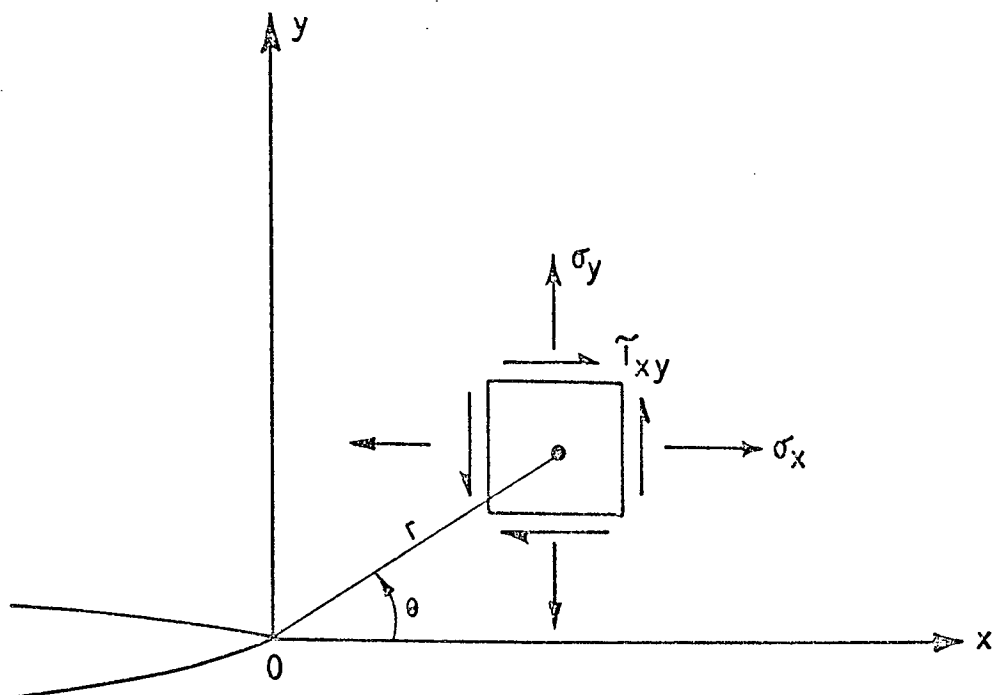


FIG 3.1 NOTATION FOR RECTANGULAR STRESS COMPONENTS NEAR A CRACK TIP.

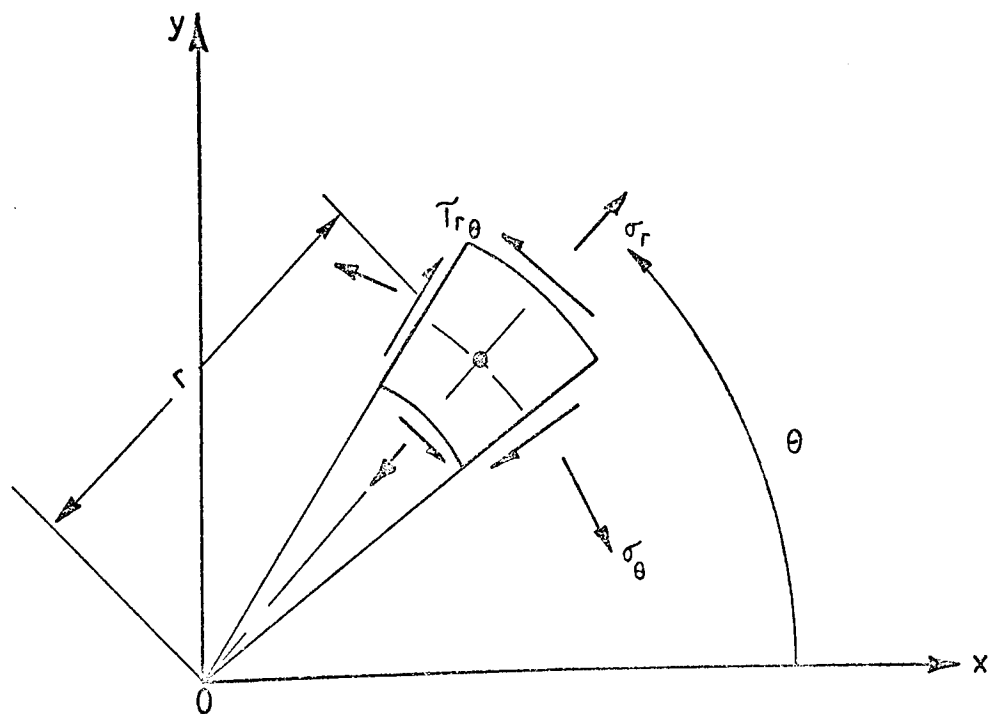


FIG 3.2 NOTATION FOR POLAR STRESS COMPONENTS NEAR A CRACK TIP.

Ref. (8), as :

$$\phi(z) = \sum_{n=0}^{\infty} A_n z^{\lambda_n}, \quad \gamma(z) = \sum_{n=0}^{\infty} B_n z^{\lambda_n + 1} \quad (3.8)$$

where A_i and B_i are complex constants to be determined from the boundary conditions and the λ_i are real eigenvalues.

Referring to Fig. 3.1, the upper and lower surfaces of the semi-infinite crack occupying the negative x-axis are free from tractions,

$$\text{i.e. } \sigma_{\theta} + i \tau_{r\theta} = 0, \quad \theta = \pm \pi$$

Summing equations 3.4 and 3.5

$$\sigma_{\theta} + i \tau_{r\theta} = \phi'(z) + \overline{\phi'(z)} + e^{2i\theta} [\bar{z} \phi^{11}(z) + \gamma^{11}(z)] \quad (3.9)$$

Substituting appropriate derivatives from equation 3.8,

$$\sum_{n=0}^{\infty} \lambda_n \tau^{\lambda_n-1} \left\{ e^{i\lambda_n\theta} \left[e^{i\theta} + (\lambda_n-1) e^{-i\theta} \right] A_n + e^{i\theta} \left[e^{-i\lambda_n\theta} + (\lambda_n+1) e^{i\lambda_n\theta} \right] B_n \right\} = 0 \quad (3.10)$$

From the two boundary conditions at $\theta = \pm \pi$, the characteristic equation

$$\sin 2\pi \lambda_n = 0 \quad (3.11)$$

is obtained and its solution yields the eigenvalues,

$$\lambda_n = n/2, \quad n = 0, 1, \dots \quad (3.12)$$

Negative eigenvalues are excluded as they lead to unbounded displacements at the origin. Hence equation 3.10 is satisfied if,

$$\lambda_n A_n + (-1)^n \bar{A}_n + (\lambda_n+1) B_n = 0, \quad n = 1, 2, \dots \quad (3.13)$$

Omitting rigid body contributions and retaining only the first two terms of equation 3.8,

$$\phi(z) = A_1 z^{\frac{1}{2}} + A_2 z \quad (3.14)$$

$$\gamma(z) = B_1 z^{3/2} + B_2 z^2 \quad (3.15)$$

and from equations 3.12 and 3.13,

$$\begin{aligned}\frac{A_1}{2} - \bar{A}_1 + \frac{3}{2} B_1 &= 0 \\ A_2 + \bar{A}_2 + 2 B_2 &= 0\end{aligned}\quad (3.16)$$

Introducing constants a_1 , a_2 and α_1 , α_2 as the real and imaginary parts of A_1 and A_2 , respectively,

$$A_1 = a_1 + i a_2 \quad A_2 = \alpha_1 + i \alpha_2 \quad (3.17)$$

Therefore, from equations 3.14 and 3.25,

$$\begin{aligned}B_1 &= \frac{1}{3} (a_1 - 3 i a_2) \\ B_2 &= -\alpha_1\end{aligned}\quad (3.18)$$

The derivatives of the functions $\phi(z)$ and $\chi(z)$ may be constructed from equations 3.14 and 3.15,

$$\phi^1(z) = \frac{1}{2} A_1 z^{-\frac{1}{2}} + A_2 \quad \chi^1(z) = \frac{3}{2} B_1 z^{\frac{1}{2}} + 2 B_2 z$$

$$\phi^{11}(z) = -\frac{1}{4} A_1 z^{-\frac{3}{2}} \quad \chi^{11}(z) = \frac{3}{4} B_1 z^{-\frac{1}{2}} + 2 B_2$$

or

$$\chi^{11}(z) = \frac{1}{2} \bar{z}^{\frac{1}{2}} \bar{A}_1 - \frac{1}{4} z^{-\frac{1}{2}} A_1 - A_2 - \bar{A}_2$$

Substitution of these results in equation 3.9 with A_1 and A_2 replaced according to equation 3.17 yields,

$$\begin{aligned}\sigma_\theta + i \tau_{r\theta} &= \frac{1}{2} r^{-\frac{1}{2}} a_1 \left(\frac{1}{2} e^{-i\theta/2} + \frac{1}{2} e^{3i\theta/2} + e^{i\theta/2} \right) + \frac{1}{2} r^{-\frac{1}{2}} i a_2 \\ &\quad \left(\frac{1}{2} e^{-i\theta/2} - \frac{3}{2} e^{3i\theta/2} - e^{i\theta/2} \right) + 2\alpha_1 (1 - e^{2i\theta})\end{aligned}$$

$$\text{but } e^{i\theta} = \cos \theta + i \sin \theta$$

$$\text{and } e^{-i\theta} = \cos \theta - i \sin \theta$$

Whereupon, after substituting and separating real and imaginary parts :

$$\sigma_\theta = \frac{1}{2\sqrt{r}} \left[a_1 (1 + \cos \theta) \cos \frac{\theta}{2} + a_2 (3 \sin \theta) \cos \frac{\theta}{2} \right] + 2\alpha_1 (1 - \cos 2\theta) \quad (3.18)$$

$$\tau_{r\theta} = \frac{1}{2\sqrt{r}} \left[a_1 \sin \theta \cos \frac{\theta}{2} - a_2 (3 \cos \theta - 1) \cos \frac{\theta}{2} \right] - 2\alpha_1 \sin 2\theta \quad (3.19)$$

From equations 3.14 and 3.15,

$$\sigma_r - i \tau_{r\theta} = \left[\phi^1(z) + \overline{\phi^1(z)} - e^{2i\theta} \bar{z} \phi^{11}(z) - e^{2i\theta} \bar{z} \overline{\phi^{11}(z)} \right]$$

Proceeding in a similar manner as above ;

$$\sigma_r = \frac{1}{2\sqrt{r}} \left[a_1 (3 - \cos \theta) \cos \frac{\theta}{2} - a_2 (3 \cos \theta - 1) \sin \frac{\theta}{2} \right] + 2\alpha_1 (1 + \cos 2\theta) \quad (3.20)$$

Finally, from equation 3.6, expressions for displacement components u_r and u_θ may be obtained.

The constants a_1 and a_2 are related to Irwin's definition of the stress intensity factors K_I and K_{II} as :

$$a_1 + i a_2 = \frac{1}{\sqrt{2}} (K_I - i K_{II}) \quad (3.21)$$

In many cases this expression is written in the form :

$$a_1 + i a_2 = \frac{1}{\sqrt{2\pi}} (K_I - i K_{II})$$

which leads to some confusion in the interpretation of the values of stress intensity factors. Most authors, however, list the forms of the equations for near tip stress and displacement used in their analyses and it is generally obvious from these if the $\frac{1}{\sqrt{\pi}}$ term has been included or not. Appropriate action may subsequently be taken either to include or eliminate this term from values of stress intensity factors. In the work reported here, the $\frac{1}{\sqrt{\pi}}$ term is excluded and combination of equation 3.21 with the equations for stress and displacement components yield the expansions, in polar coordinates, for stress and displacement components around the crack tip as :

$$\begin{aligned} \sigma_r &= \frac{1}{2\sqrt{2r}} \left[K_I (3 - \cos \theta) \cos \frac{\theta}{2} + K_{II} (3 \cos \theta - 1) \sin \frac{\theta}{2} \right] + 2\alpha_1 (1 + \cos 2\theta) \\ \sigma_\theta &= \frac{1}{2\sqrt{2r}} \left[K_I (1 + \cos \theta) \cos \frac{\theta}{2} - K_{II} (3 \sin \theta) \cos \frac{\theta}{2} \right] + 4\alpha_1 \sin^2 \theta \end{aligned} \quad (3.22)$$

$$\tau_{r\theta} = \frac{1}{2\sqrt{2r}} \left[K_I \sin \theta \cos \frac{\theta}{2} + K_{II} (3 \cos \theta - 1) \cos \frac{\theta}{2} \right] - 2\alpha_1 \sin 2\theta$$

$$u_r = \frac{K_I}{4G} \left(\frac{r}{2}\right)^{\frac{1}{2}} \left[(2K-1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \right] - \frac{K_{II}}{4G} \left(\frac{r}{2}\right)^{\frac{1}{2}} \left[(2K-1) \sin \frac{\theta}{2} - 3 \sin \frac{3\theta}{2} \right] \\ + \alpha_1 \left(\frac{r}{2G}\right) (K-1 + 2 \cos 2\theta)$$

$$u_\theta = \frac{K_I}{4G} \left(\frac{r}{2}\right)^{\frac{1}{2}} \left[\sin \frac{3\theta}{2} - (1+2K) \sin \frac{\theta}{2} \right] - \frac{K_{II}}{4G} \left(\frac{r}{2}\right)^{\frac{1}{2}} \left[(2K+1) \cos \frac{\theta}{2} - 3 \cos \frac{3\theta}{2} \right] \\ - \alpha_1 \left(\frac{r}{G}\right) \sin 2\theta + \alpha_2 \left(\frac{r}{2G}\right) (K+1)$$

Strain components ϵ_r , ϵ_θ and $\gamma_{r\theta}$ may be found from equations 3.22 using the strain displacement relations ,

$$\epsilon_r = \frac{\partial u_r}{\partial r} \\ \epsilon_\theta = \frac{1}{r} \frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} \\ \gamma_{r\theta} = \frac{1}{r} \frac{\partial u_r}{\partial \theta} + \frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r} \quad (3.23)$$

The equations are listed in Appendix 10.A together with their Cartesian counterparts which can be calculated in an analogous manner to the above from equations 3.1, 3.2 and 3.3 or simply by transforming equations 3.22 from Polar to Cartesian coordinates.

It is worth noting that the near field stress components σ_x , σ_y and τ_{xy} listed in Appendix 10.A are the same for both plane strain and generalised plane stress. The general states of stress and displacement for plane strain and generalised plane stress are different, however, for plane strain ;

$$\sigma_z = \nu(\sigma_x + \sigma_y) = 2\nu \frac{K_I}{\sqrt{2r}} \cos \frac{\theta}{2} - 2\nu \frac{K_{II}}{\sqrt{2r}} \sin \frac{\theta}{2} + \dots \quad (3.24)$$

$$\text{with } u_z = 0$$

for generalised plane stress,

$$u_z = -\frac{\nu}{E} \int (\sigma_x + \sigma_y) dz = -\frac{2\nu}{E} \frac{z k_I}{\sqrt{2r}} \cos \frac{\theta}{2} + \frac{2\nu}{E} \frac{z k_{II}}{\sqrt{2r}} + \dots \quad (3.25)$$

with $\sigma_z = 0$

Note that u_z becomes unbounded at the crack tip, $r = 0$. This condition is not generally permitted in elasticity problems and is one reason why generalised plane stress solutions are approximate in nature.

4. APPLICATION OF THE HILTON ELEMENT FOR DETERMINING STRESS INTENSITY FACTORS

4.1 INTRODUCTION

It is the purpose of this chapter to describe the modifications necessary to augment a standard finite element program to perform Modes I and II stress intensity calculations using the embedded singularity element of Hilton and Hutchinson⁽⁵¹⁾.

The basic principles of the technique have been discussed in Chapter 2 with reference to opening mode stress intensity factor K_I . These principles will be extended here to the completely general case where several unknowns may be present within the core element, e.g. K_I , K_{II} and various rigid body modes. The derivation follows that given in Chapter 2 but here the more concise matrix notation is employed. There are alternative methods available for synthesising the analytical and finite element solutions, for example the method of Lagrange Multipliers used by Richards⁽⁵⁶⁾ for general field problems containing singularities such as a crack.

4.2 MODIFYING A FINITE ELEMENT PROGRAM TO INCLUDE THE HILTON ELEMENT

Consider the cracked configuration of Fig. 4.1 where a 'core' region is identified surrounding the crack tip. The mesh subdivision is as shown in the figure. From chapter 3 and Appendix 10.A, series expansions are available for describing the near tip displacement components in the form :

$$u_x = K_I f(r, \theta) + K_{II} g(r, \theta) + \alpha_1 k(r, \theta) + \dots \quad (4.1)$$

$$u_y = K_I l(r, \theta) + K_{II} m(r, \theta) + \alpha_1 n(r, \theta) + \dots$$

Particular relations for functions $f(r, \theta)$, $g(r, \theta)$, etc. may be obtained from Appendix 10.A.

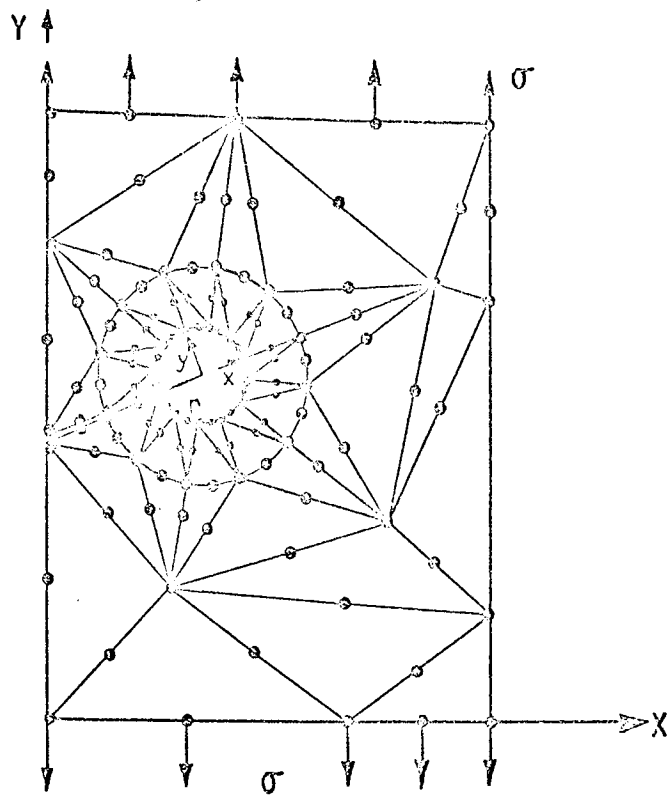


FIG.4.1. CORE ELEMENT AND FINITE ELEMENT IDEALISATION FOR SINGLE CRACK TIP PROBLEM

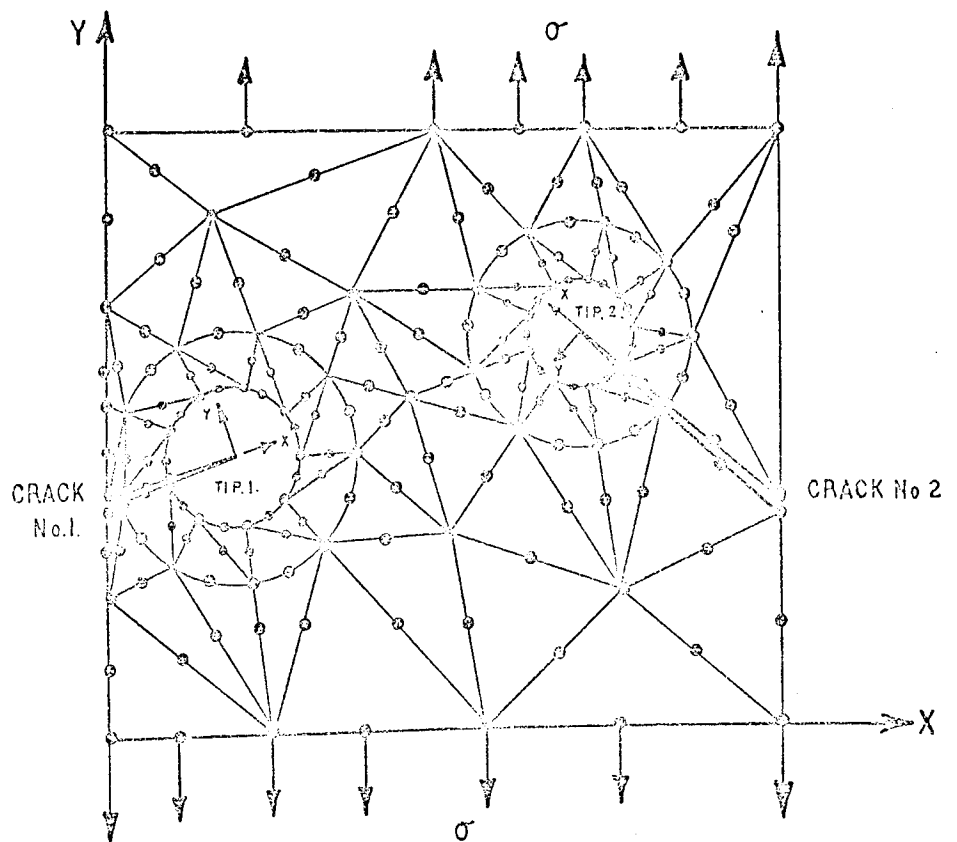


FIG.4.2. CORE ELEMENT AND FINITE ELEMENT IDEALISATION FOR DOUBLE TIP CRACK PROBLEM.

Equation 4.1 may be written :

$$\begin{Bmatrix} u_x \\ u_y \end{Bmatrix} = \{u_c\} = [N]_c \{\alpha\} \quad (4.2)$$

where $[N]_c$ is a matrix of functions $f(r, \theta)$, etc. and $\{\alpha\}$ is a vector of unknown parameters.

The matrix relating strain components to the vector $\{\alpha\}$ may be found, Appendix 10.A, in the form ;

$$\{\epsilon\} = [B]_c \{\alpha\} \quad (4.3)$$

so that the strain energy of the core region may be computed as ;

$$U_c = \frac{1}{2} \int_{\text{core}} \{\sigma\}^t \{\epsilon\} d(\text{Vol})$$

and using the stress-strain relation ($\{\sigma\} = [C]\{\epsilon\}$) with equation 4.3

$$U_c = \frac{1}{2} \{\alpha\}^t \left(\int_{\text{core}} [B_c]^t [C] [B]_c d(\text{Vol}) \right) \{\alpha\}$$

or

$$U_c = \frac{1}{2} \{\alpha\}^t [k_c] \{\alpha\} \quad (4.4)$$

The shape of the core region is arbitrary but since equation 4.1 is expressed in terms of polar coordinates centred at the crack tip, as in Fig. 4.1, then a circular core considerably eases the integrations necessary to form $[k_c]$.

The total potential energy of the plate is written in the form ;

$$V^* = U_c + U_e - \Omega \quad (4.5)$$

where U_e is the strain energy of the elements in the mesh and $(-\Omega)$ is the work done by the applied loads.

In the manner of Section 1, Chapter 3, equation 4.5 may be written ;

$$V^* = \frac{1}{2} \{\alpha\}^t [k_c] \{\alpha\} + \frac{1}{2} \{\delta\}^t [K] \{\delta\} - \{\delta\}^t \{F\} \quad (4.6)$$

It proves convenient for the minimisation of potential energy, V^* , that follows, to partition the vectors $\{\delta\}$, $\{F\}$ and the matrix $[K]$ to correspond to degrees of freedom on the core/finite element interface $\{\delta_1\}$ and degrees of freedom in the remainder of the finite element mesh $\{\delta_2\}$, as

$$\{\delta\} = \begin{Bmatrix} \{\delta_1\} \\ \{\delta_2\} \end{Bmatrix}, [K] = \begin{bmatrix} [K_{11}] & [K_{12}] \\ [K_{21}] & [K_{22}] \end{bmatrix} \quad \text{and} \quad \{F\} = \begin{Bmatrix} \{F_1\} \\ \{F_2\} \end{Bmatrix} \quad (4.7)$$

Note that careful mesh nodal numbering obviates the need for any rearrangement implied by the above partitioning. This point is of crucial importance in ease of programming and is discussed more fully in the subsequent chapter on Numerical Procedures.

The vector $\{\delta_1\}$ may be expressed as a function of the unknowns $\{\alpha\}$ from equation 4.2 in the form;

$$\{\delta_1\} = \begin{Bmatrix} \{u_1\} \\ \{v_1\} \\ \vdots \\ \vdots \end{Bmatrix} = [A] \{\alpha\} \quad (4.8)$$

where $[A]$ is a matrix of functions $f(r, \theta)$, etc. evaluated at nodes on the core boundary.

Substituting equations 4.7 and 4.8 in equation 4.6 and noting that

$$\{\delta_1\}^t = \{\alpha\}^t [A]^t$$

$$\begin{aligned} V^* = & \frac{1}{2} \{\alpha\}^t [K_c] \{\alpha\} + \frac{1}{2} \{\alpha\}^t [A]^t [K_{11}] [A] \{\alpha\} + \frac{1}{2} \{\alpha\}^t [A]^t [K_{12}] \{\delta_2\} \\ & + \frac{1}{2} \{\delta_2\}^t [K_{21}] [A] \{\alpha\} + \frac{1}{2} \{\delta_2\}^t [K_{22}] \{\delta_2\} - \{\alpha\}^t [A]^t \{F_1\} \\ & - \{\delta_2\}^t \{F_2\} \end{aligned}$$

Variation dV^* of V^* is found by treating δ 's and α 's as free variables which may receive arbitrary independent increments, so that for equilibrium,

$$\begin{aligned}
dV^* = 0 = & \{d\alpha\}^t [k_c] \{\alpha\} + \{d\alpha\}^t ([A]^t [K_{11}] [A]) \{\alpha\} \\
& + \frac{1}{2} \{d\alpha\}^t [A]^t [K_{12}] \{\delta_2\} + \frac{1}{2} \{\alpha\}^t [A]^t [K_{12}] \{d\delta_2\} \\
& + \frac{1}{2} \{d\delta_2\} [K_{21}] [A] \{\alpha\} + \frac{1}{2} \{\delta_2\} [K_{21}] [A] \{d\alpha\} \\
& + \{d\delta_2\}^t [K_{22}] \{\delta_2\} - \{d\alpha\}^t [A]^t \{F_1\} - \{d\delta_2\}^t \{F_2\}
\end{aligned}$$

but $[K]$ is symmetric so that $[K_{12}] = [K_{21}]^t$, also matrix products of the type $\frac{1}{2} \{\alpha\}^t [A]^t [K_{12}] \{d\delta_2\}$ are scalar so that

$$\begin{aligned}
\frac{1}{2} \{\alpha\}^t [A]^t [K_{12}] \{d\delta_2\} &= (\frac{1}{2} \{\alpha\}^t [A]^t [K_{12}] \{d\delta_2\})^t \\
&= \frac{1}{2} \{d\delta_2\}^t [K_{21}] [A] \{\alpha\}
\end{aligned}$$

Then,

$$\begin{aligned}
dV^* = 0 = & \{d\alpha\}^t [k_c] \{\alpha\} + ([A]^t [K_{11}] [A]) \{\alpha\} \\
& + [A]^t [K_{12}] \{\delta_2\} - [A]^t \{F_1\} + \{d\delta_2\}^t [K_{22}] \{\delta_2\} \\
& + [K_{21}] [A] \{\alpha\} - \{F_2\}
\end{aligned}$$

Since $\{d\alpha\}$ and $\{d\delta_2\}$ are arbitrary, the above equation yields two simultaneous matrix equations of equilibrium,

$$[k_c] \{\alpha\} + ([A]^t [K_{11}] [A]) \{\alpha\} + [A]^t [K_{12}] \{\delta_2\} - [A]^t \{F_1\} = 0 \quad (4.9)$$

$$[K_{22}] \{\delta_2\} + [K_{21}] [A] \{\alpha\} - \{F_2\} = 0$$

Reforming these two equations in one modified stiffness matrix and load vector in the form,

$$[K^*] \{\delta^*\} = \{F\}^* \quad (4.10)$$

$$\text{where } [K^*] = \begin{bmatrix} [K_{11}]^* & [K_{12}]^* \\ [K_{21}]^* & [K_{22}]^* \end{bmatrix}$$

$$\{\delta^*\} = \begin{Bmatrix} \{\delta_2\} \\ \{\alpha\} \end{Bmatrix}$$

$$\text{and } \{F\}^* = \begin{Bmatrix} \{F_1\}^* \\ \{F_2\}^* \end{Bmatrix}$$

$$\text{with } [K_{11}]^* = [K_{22}] \quad (4.10a)$$

$$[K_{12}]^* = [K_{21}]^{*t} = [K_{21}][A] \quad (4.10b)$$

$$[K_{22}]^* = [k_c] + ([A]^t [K_{11}] [A]) \quad (4.10c)$$

$$\{F_1\}^* = \{F_2\} \quad (4.10d)$$

$$\text{and } \{F_2\}^* = [A]^t \{F_1\} \quad (4.10e)$$

gives a set of linear simultaneous matrix equations which may be solved to give the unknown displacement components $\{\delta_2\}$ and the unknown core parameters $\{\alpha\}$.

Comparing the forms of equations 4.10 with those of equations 4.7 it is easy to appreciate the simplicity of including this procedure into any standard finite element program. It is necessary only to modify the overall stiffness matrix and load vector according to the operations described and to proceed with the solution in the normal manner.

In the case of two or more singularities present in the same problem, necessitating the use of several core elements, the partitioning of equation 4.7 is slightly more involved but the procedure is identical. For example, consider the problem of Fig. 4.2, where two crack tips are present each with undetermined parameters contained in vectors $\{\alpha_1\}$ and $\{\alpha_2\}$, the subscripts referring to tips 1 and 2 respectively.

Equation 4.5 becomes,

$$V^* = U_{c_1} + U_{c_2} + U_e - \Omega \quad (4.11)$$

$$\text{or } V^* = \frac{1}{2} \{\alpha_1\}^t [k_{c_1}] \{\alpha_1\} + \frac{1}{2} \{\alpha_2\}^t [k_{c_2}] \{\alpha_2\} - \frac{1}{2} \{\delta\}^t [K] \{\delta\} - \{\delta\}^t \{F\}$$

Partitioning vectors $\{\delta\}$, $\{F\}$ and the matrix $[K]$ for nodes on the first core/element interface, $\{\delta_1\}$, nodes on the second core/element interface $\{\delta_2\}$ and those in the remainder of the mesh, $\{\delta_3\}$,

$$\{\delta\} = \begin{Bmatrix} \{\delta_1\} \\ \{\delta_2\} \\ \{\delta_3\} \end{Bmatrix} [K] = \begin{bmatrix} [K_{11}] & [K_{21}] & [K_{31}] \\ [K_{21}] & [K_{22}] & [K_{32}] \\ [K_{31}] & [K_{32}] & [K_{33}] \end{bmatrix} \text{ and } \{F\} = \begin{Bmatrix} \{F_1\} \\ \{F_2\} \\ \{F_3\} \end{Bmatrix} \quad (4.7)$$

Once again careful mesh numbering around core elements eliminates the need for rearranging the rows of the above matrices.

Proceeding in a similar manner to that used in the example above, and noting that :

$$\{\delta_1\} = [A_1]\{\alpha_1\} \quad \{\delta_2\} = [A_2]\{\alpha_2\}$$

the following modified matrix equation results :

$$\begin{bmatrix} [K_{11}]^* & [K_{12}]^* & [K_{13}]^* \\ [K_{21}]^* & [K_{22}]^* & [K_{23}]^* \\ [K_{31}]^* & [K_{32}]^* & [K_{33}]^* \end{bmatrix} \begin{Bmatrix} \{\delta_3\} \\ \{\alpha_1\} \\ \{\alpha_2\} \end{Bmatrix} = \begin{Bmatrix} \{F_1\}^* \\ \{F_2\}^* \\ \{F_3\}^* \end{Bmatrix} \quad (4.8)$$

where

$$\begin{aligned} [K_{11}]^* &= [K_{33}] \\ [K_{21}]^* &= [K_{12}]^{*t} = [A_1]^t [K_{31}] \\ [K_{22}]^* &= [A_1]^t [K_{11}] [A_1] \\ [K_{31}]^* &= [K_{13}]^{*t} = [A_2]^t [K_{32}] \\ [K_{32}]^* &= [K_{23}]^{*t} = [A_2]^t [K_{21}] [A_1] \\ [K_{33}]^* &= [A_2]^t [K_{22}] [A_2] \\ \{F_1\}^* &= \{F_3\} \\ \{F_2\}^* &= [A_1]^t \{F_1\} \\ \{F_3\}^* &= [A_2]^t \{F_2\} \end{aligned}$$

As in the example given above, the matrix equation 4.8 may be solved to give values for the vectors $\{\delta_j\}$, $\{\alpha_1\}$ and $\{\alpha_2\}$.

This procedure may be generalised for the case of any number of tips present and is discussed in the subsequent chapter on Numerical Procedures.

5. NUMERICAL PROCEDURES FOR VARIOUS CRACK CONFIGURATIONS

5.1 INTRODUCTION

The general theory for augmenting a standard finite element program to include a singular subregion has been discussed in Chapter 4. In this chapter the theory is applied to several crack configurations of both symmetric, Mode I loading and skew-symmetric, Modes I and II loading.

As previously noted, one of the principal advantages of the technique is its easy inclusion in any standard finite element code, necessitating a modification of the overall stiffness matrix and load vector for the particular crack problem being considered. In order, therefore, to retain the flow of the finite element program, the only changes made are, firstly to include a procedure which modifies the overall stiffness matrix and load vector before the load-deflection equations are solved in the usual manner and, secondly, to rearrange the print statement to include the new core parameters. For the four problems to be discussed in this Chapter, procedures are presented in each case which perform the required changes in the system equations. Final complete programs incorporating these procedures are given in Appendix 10.D.

The first example is of a single tip crack under symmetric Mode I loading. Geometry and loading symmetry about the x-axis, Fig. 2.9, require that only a semi-circular core be constructed round the crack tip. In the second example, an array of tips all subjected to opening Mode I loading is considered. This represents a generalisation of the single tip case of the first example. The single tip becomes, therefore, a special case of the 'N-tip, Mode I' (several tip) procedure. Thirdly, the theory is applied to single edge cracks with Modes I and II stress intensity factors, this is generally referred to here as the mixed-mode single tip case. For mixed mode problems there is no symmetry about

the crack plane as in Mode I problems and so a full circular core is constructed around the tip, Fig. 4.1. Finally, as an extension of the mixed mode single tip problem, the case of an internal crack of arbitrary shape is considered. This represents two crack tips each with mixed mode stress intensity factors.

Obviously the next step would be to develop a general procedure capable of handling several crack tips each with general mixed mode singularities. This procedure would then replace all the previous procedures in the sense that each is a special case with Mode I or Mode II equal to zero as appropriate. Indeed this is probably one of the ultimate objectives of the method as applied to two-dimensional problems of plane stress or plane strain but there are problems associated with such a program, not the least of which is crack closure, which will be the subject of further discussion with reference to the mixed mode two tip procedure later in this thesis. Also in the case of a Mode I analysis, using the general mixed mode program, no allowance could be made for symmetry since a full circular core would have to be constructed around each tip. The number of degrees of freedom are thus increased with a corresponding increase in computer storage and time requirements. In summary, therefore, when the problem is symmetrical it is worth using the special program compared with the general one because of the savings.

The problem of crack tip closure was briefly referred to in the preceeding paragraph with reference to mixed-mode fracture. Closure or partial closure can also occur for the Mode I case, e.g. consider loading of a centre cracked plate as shown in Fig. 5.1 with a subsequent crack profile of Fig. 5.2. In an application of the finite element method the equilibrium displacement solution is obtained from a minimisation of the total potential energy of the body. For the normal mesh, inter-element compatibility is maintained by assuming suitable displacement functions

within each element. No allowance is made, however, with regard to the compatibility requirements of free boundaries in close proximity to each other, as in the case of crack surfaces. In such cases, without special precautions, the resulting crack configuration will be of the form shown in Fig. 5.3 for the loading of Fig. 5.1. The minimum energy configuration results in the physically inadmissible situation of the two crack surfaces crossing with a negative value for the Mode I stress intensity factor at crack tip number 2. Having established that surfaces have crossed, in the case of a negative K_{I1} , then remedial action is relatively straightforward, involving constraints applied to the nodes on the crack surface in the y-direction giving $K_{II2} = 0$. Iteration is necessary to determine the extent of the closure. For mixed-mode problems, however, where cracks may be inclined to the global axes, corrective action is rather more difficult. Consider, for example, the inclined crack of Fig. 5.4 with loading such that after a normal solution, ignoring closure, K_{I1} is found to be negative. Again compatibility of displacements has been violated by crossing of the crack surfaces. In such cases it is not sufficient simply to constrain opposite nodes, A and B, Fig. 5.4, to move together, i.e. to replace nodes A and B by a single node C. This is because K_{II} may not be zero and thus nodes A and B will move relative to each other as shown in Fig. 5.5. Special provision is necessary and the scheme developed here, therefore, is to modify the overall system matrices such that on subsequent solution, displacements of crack surface nodes in directions normal and tangential to the crack surface are obtained, Fig. 5.6. Using the normal values of these displacement components for adjacent nodes, an average value of normal displacement can be calculated for nodes A and B; this value can then be applied as a displacement boundary condition to the original problem and the resulting overall system equations resolved giving $K_{I1} = 0$ and $K_{II1} \neq 0$. The approach is described more fully and the computer procedures are presented in the section describing the mixed mode two-tip program.

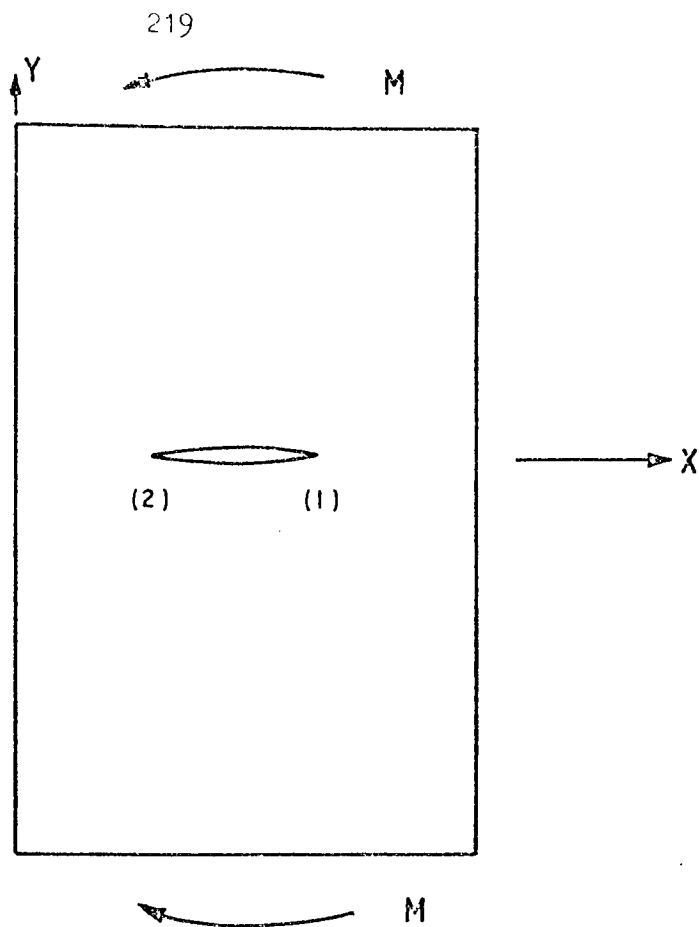


FIG 5.1 CENTRE CRACKED PLATE SUBJECTED TO IN-PLANE BENDING.

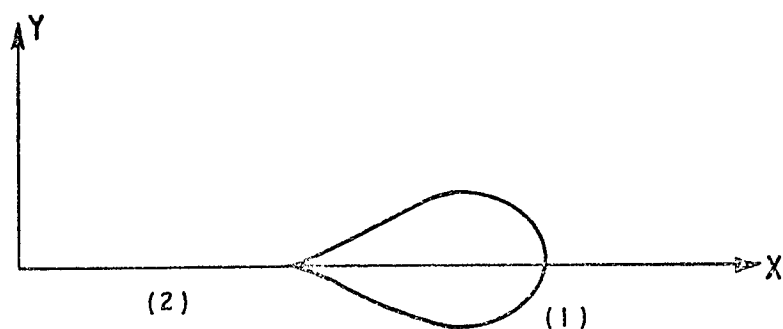


FIG 5.2 CORRECT CRACK PROFILE

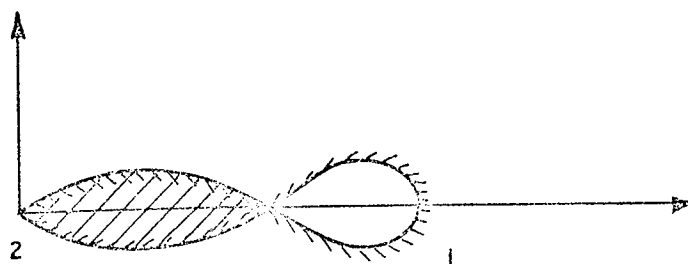


FIG 5.3 INCORRECT CRACK PROFILE WHERE SURFACES HAVE CROSSED

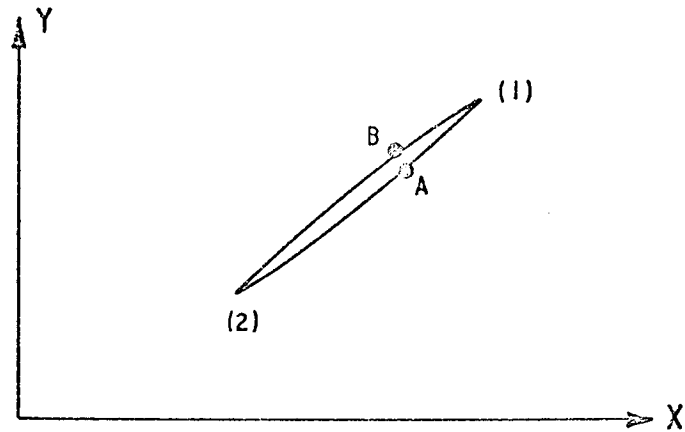


FIG 5.4 INCLINED CRACK SUBJECT TO CLOSURE FORCES.

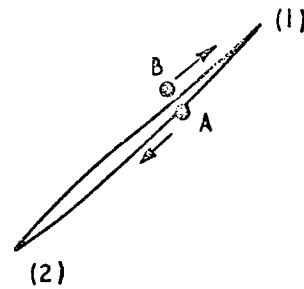


FIG 5.5 SLIDING OF CRACK SURFACES FOR A NON-ZERO, K_{II} .

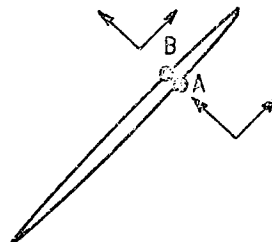


FIG 5.6 NORMAL AND TANGENTIAL DISPLACEMENT COMPONENTS FOR CRACK SURFACE NODES.

In an application of the normal finite element code, the size of the overall stiffness matrix is obtained after the address sequence has been constructed. See Section I of this thesis. For particularly large problems involving many degrees of freedom, the bounds of the overall stiffness matrix may have to be increased to include the singular core or cores present. To this end a separate procedure is included in several of the programs to check whether the bounds as originally calculated are sufficient to include the singular cores. For the particular case of the Mode I single crack tip program this procedure is unnecessary since the array bounds originally calculated are always sufficient to contain the singular core.

In selecting the shape functions for any finite element, due consideration must be given to, and allowance made for, inter-element compatibility and element rigid body displacements. Similarly for the core element all the element rigid body modes must be included, e.g. in the mixed mode problem they are, displacement in the x and y-directions together with a rotation in the xy-plane. Failure to include these terms leads to spurious constraint forces after solution and incorrect values for the stress intensity factors. Also, due to the fact that displacements along the interface Γ , Fig. 2.7, in the core region vary as trigonometric functions of θ whereas the displacements in the triangular elements are polynomial functions of the nodal displacements, the compatibility of displacements between nodes on the interface Γ is not satisfied. The magnitude of the error thus induced is indeterminate but it is worth noting that as the number of elements on Γ is increased, the angular separation $d\theta$ between the nodal points and the difference between the polynomial and trigonometric displacement functions are reduced. The computational experiments described in Chapter 6 will support this argument and also show that adequate accuracy is achieved.

Many of the computational steps involved in modifying the overall stiffness matrix and load vector are similar for each of the cracked configurations considered. The Mode I single tip case will, therefore, be used as an example and each step will be discussed in detail, the remaining cases being summarised only briefly.

5.2 SINGLE TIP SYMMETRIC MODE I CASE

As an example, consider a rectangular plate of dimensions $2b \times 2c$ containing a crack of length $2a$, Fig. 5.7. By symmetry only one quarter of this plate need be considered, with the core element and boundary conditions of Fig. 5.8. As a result of symmetry, displacement of the crack tip in the y -direction is prevented although the tip may move in the x -direction. The near tip displacement field components may be written, using Appendix 10.A, as,

$$\begin{aligned} u_x &= \delta_x + K_I f(r, \theta) \\ u_y &= K_I g(r, \theta) \end{aligned} \quad (5.1)$$

where δ_x is the rigid body displacement of the crack tip in the x -direction.

K_I is the Mode I stress intensity factor,

and

$$\begin{Bmatrix} f(r, \theta) \\ g(r, \theta) \end{Bmatrix} = \frac{1}{4G} \left(\frac{r}{2} \right)^{\frac{1}{2}} \begin{Bmatrix} (2K-1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \\ (2K+1) \sin \frac{\theta}{2} - \sin \frac{3\theta}{2} \end{Bmatrix}$$

Note that derivatives of these displacement components are independent of δ_x and hence it can have no effect on strain, stress or strain energy stored in the core. In order to form the modified stiffness matrix, as in Chapter 4, the potential energy is minimised with respect to K_I , δ_x and the unconstrained nodal displacements, u_j . The unconstrained nodal displacements, u_j , refer to all nodes in the finite element mesh not on the core/finite element interface, Γ , Fig. 5.8.

Referring to Fig. 5.8, where $N1$ nodes surround the core element with the nodal numbering system as shown. This numbering system is particularly convenient for the operations involved in modifying the overall stiffness matrix, $[K]$. As a result of such a numbering convention, the overall stiffness equations are of the form shown in Fig. 5.9 where only nodes on the first ring of elements contribute stiffness coefficients connecting nodes on the core to those in the remainder of the finite element mesh. Partitioning the matrices as described in equation 4.7 results in the subdivision shown in Fig. 5.9.

The modified stiffness matrix, hereafter referred to as $[K^*]$, will have a reassembled form according to equation 4.10, as shown in Fig. 5.10. Having located the generated matrices in their relevant positions in $[K^*]$ it is convenient, for simplicity of notation, to refer to their locations in $[K^*]$ directly, e.g. it is understood that :

$$K_{22}^* [2, 1] = K^* [NG - 2N1 + 2, NG - 2N1 + 1] \quad (5.2)$$

where $N1$ is the number of nodal points on the core and

NG is the number of degrees of freedom of the original problem.

Also it is convenient to relate the submatrices $[K_{21}]$, etc. to their positions in the original stiffness matrix, $[K]$.

e.g. the matrix $[K_{21}]$ has rows $2N1 + 1$ to $6N1$ and columns 1 to $2N1$.

Following the procedure described in section 4.2,

$$\begin{aligned} \{\delta_1\} &= [A] \{\alpha\} \\ \text{where } \{\delta_1\} &= \begin{Bmatrix} \begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix} \\ \begin{Bmatrix} u_2 \\ v_2 \end{Bmatrix} \\ \vdots \\ \begin{Bmatrix} u_{N1} \\ v_{N1} \end{Bmatrix} \end{Bmatrix} \quad \text{and} \quad \{\alpha\} = \begin{Bmatrix} \delta_x \\ K_I \end{Bmatrix} \end{aligned}$$

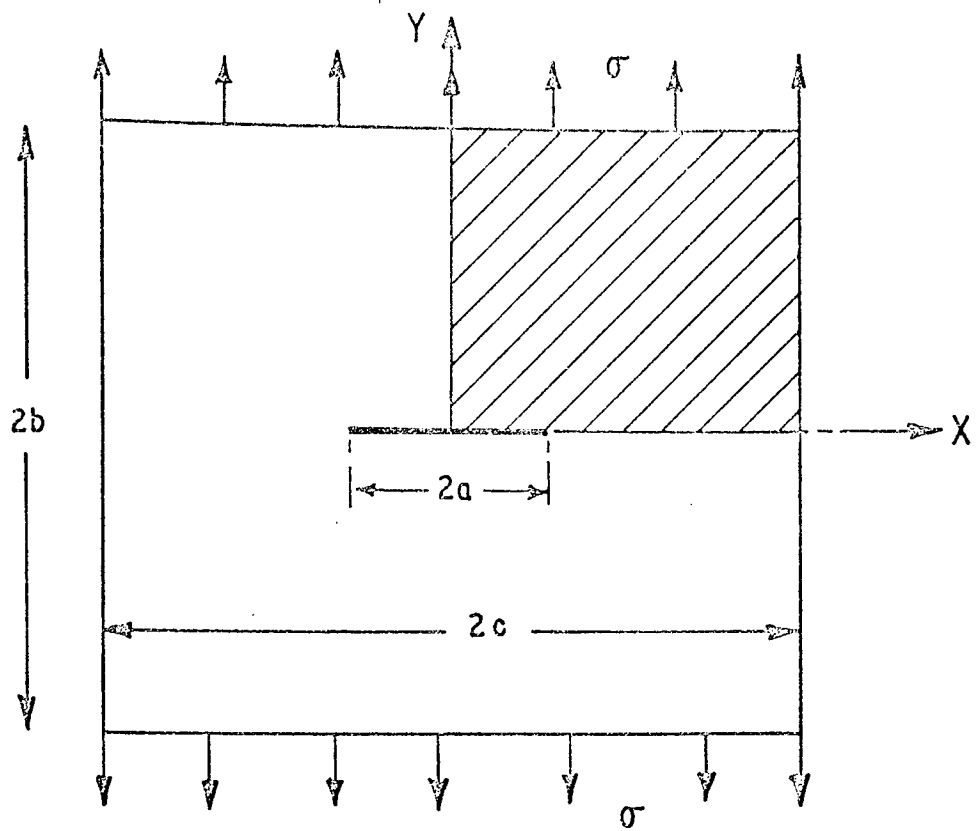


FIG.5.7 CENTRALLY CRACKED RECTANGULAR PLATE.

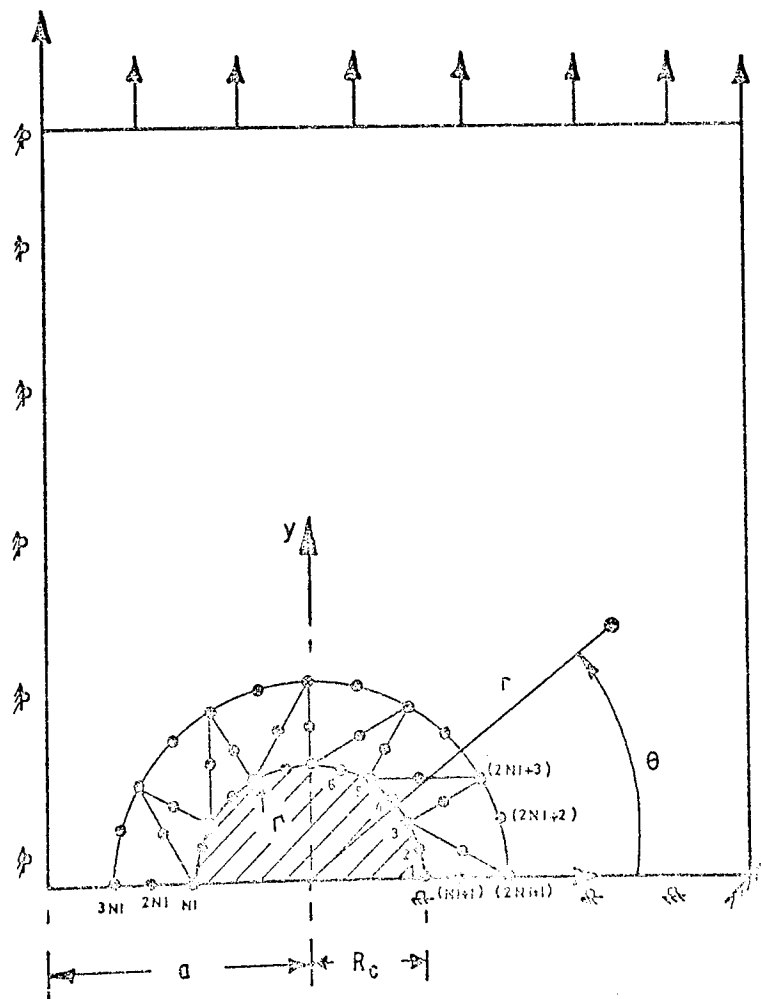


FIG.5.8 QUADRANT OF CENTRALLY CRACKED PLATE WITH CORE ELEMENT.

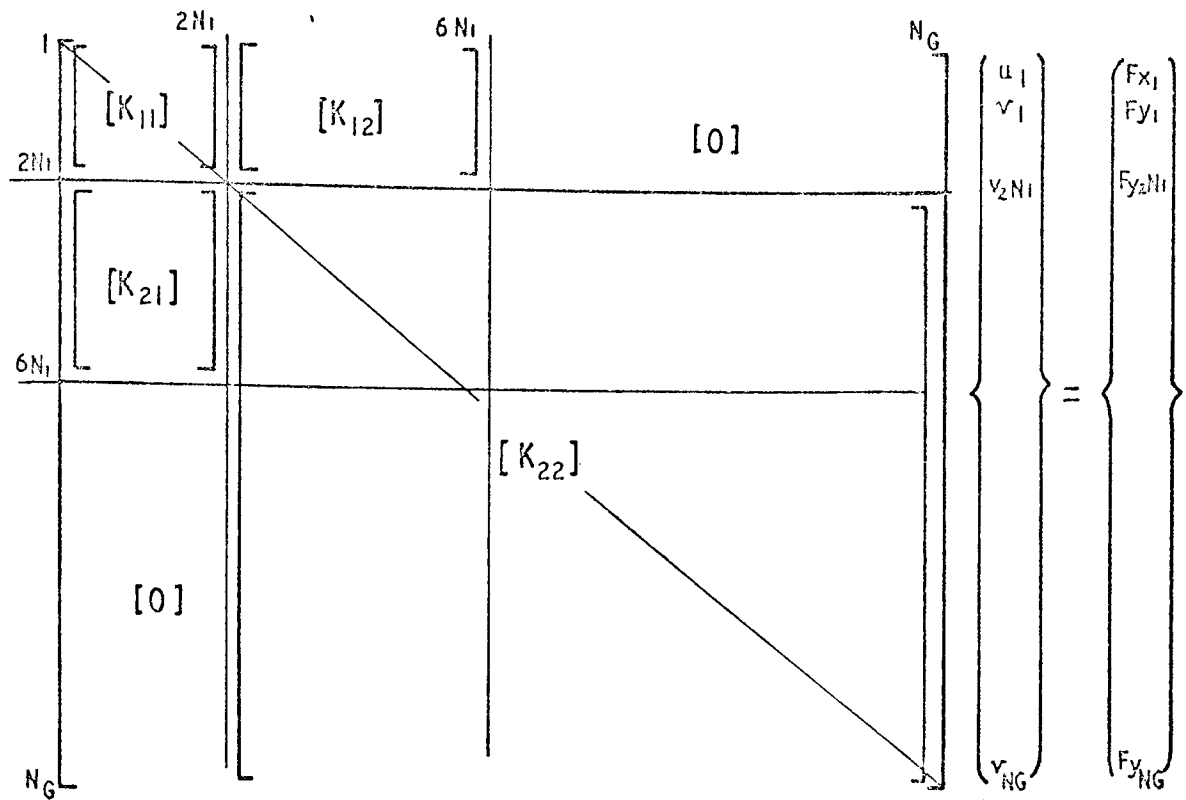


FIG 5.9 PARTITIONING OF STIFFNESS MATRIX, $[K]$.

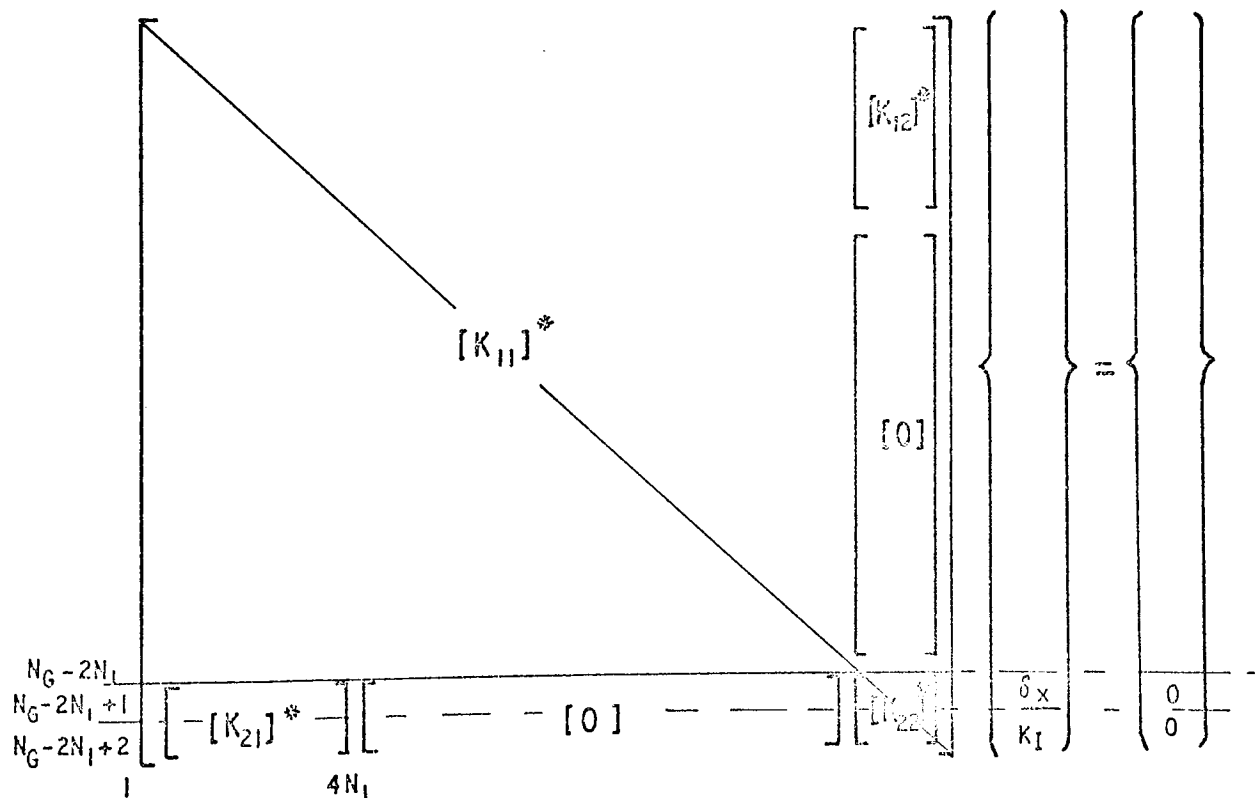


FIG 5.10 CONSTRUCTION OF STIFFNESS MATRIX, $[K^*]$

with

$$[A] = \begin{bmatrix} \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} & \begin{Bmatrix} f(R_c, \theta_1) \\ g(R_c, \theta_1) \end{Bmatrix} \\ \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} & \begin{Bmatrix} f(R_c, \theta_2) \\ g(R_c, \theta_2) \end{Bmatrix} \\ \vdots & \vdots \\ \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} & \begin{Bmatrix} f(R_c, \theta_{N1}) \\ g(R_c, \theta_{N1}) \end{Bmatrix} \end{bmatrix} \quad (5.3)$$

$f(R_c, \theta_i)$ etc. is understood to mean the evaluation of that function with polar coordinates (R_c, θ_i) . θ_i measured from positive x-axis to node number i , Fig. 5.8.

From appendix 10.B, equation 10.11, $[k_c]$ may be constructed as

$$[k_c] = \begin{bmatrix} 0 & 0 \\ 0 & \frac{R_c \pi}{16G} (2K-1) \end{bmatrix}$$

Rewriting the matrix operations of equation 4.10 as summations in a form suitable for programming, equation 4.10C,

$$K_{22}^* [1, 1] = k_c [1, 1] + \sum_{i=1,3,5,\dots}^{2N_1-1} \sum_{j=1,3,5,\dots}^{2N_1-1} K[i, j] \quad (5.4a)$$

$$K_{22}^* [1, 2] = K_{22}^* [2, 1] = k_c [2, 1] + \sum_{i=1,3,5,\dots}^{2N_1-1} \sum_{j=1,2,\dots}^{2N_1} K[i, j] u_j \quad (5.4b)$$

$$K_{22}^* [2, 2] = k_c [2, 2] + \sum_{i=1,2,\dots}^{2N_1} \sum_{j=1,2,\dots}^{2N_1} K[i, j] u_i u_j \quad (5.4c)$$

equation 4.10b,

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

or from symmetry of $[K]$,

$$K_{12}^* [1, j] = \sum_{i=1,3,5,\dots}^{2N_1-1} K [j + 2N_1, i] \quad j = 1, 2, 3, \dots, 4N_1 \quad (5.4d)$$

$$K_{12}^* [2, j] = \sum_{i=1,2,3,\dots}^{2N_1} K [j + 2N_1, i] u_i \quad j = 1, 2, 3, \dots, 4N_1 \quad (5.4e)$$

equation 4.10e, with interface nodes considered free from external loads,

$$\{ F_2 \}^* = \{ 0 \} \quad (5.4f)$$

equation 4.10 a,

$$K_{11}^* [i, j] = K [i + 2N_1, j + 2N_1] \quad i, j = 1, 2, \dots, NG - 2N_1 \quad (5.4g)$$

and finally equation 4.10 d,

$$F_1^* [i] = F [i + 2N_1] \quad i, j = 1, 2, \dots, NG - 2N_1 \quad (5.4h)$$

In the above equations,

$$u_j = \begin{cases} f(R_c, \theta \frac{j+1}{2}) & - j \text{ odd} \\ g(R_c, \theta \frac{j}{2}) & - j \text{ even} \end{cases} \quad (5.4i)$$

The following flowchart and procedure listing contains the steps described in equations 5.4 for the modification of the overall stiffness matrix and load vector. As well as modifying these system equations, the procedure adjusts the coefficients of the address sequence for the new form of $[K^*]$.

DESCRIPTION OF PROCEDURE MODEIA

Temporary storage KT11, KT12, etc. has been allocated for the generation of the coefficients, equation 5.4, prior to their insertion in the modified stiffness matrix $[K^*]$.

Consider, for example, the evaluation of coefficient $K_{22}^*[2, 2]$, the value of which is given by equation 5.4c and is located in the computer store as variable KT11,

$$\begin{aligned} \text{KT11} = & k_c [2, 2] + K[1, 1] u_1 u_1 + K[1, 2] u_1 u_2 + \dots K[1, 2N_1] u_1 u_{2N_1} \\ & + K[2, 1] u_2 u_1 + K[2, 2] u_2 u_2 + \dots \\ & + \\ & \vdots \\ & + K[2N_1, 1] u_{2N_1} u_1 + \dots + K[2N_1, 2N_1] u_{2N_1} u_{2N_1} \end{aligned}$$

with u_i given by equation 5.4i.

This summation is complicated by the fact that only the lower triangle of the overall stiffness matrix $[K]$ has been stored, hence to form KT11 it is expedient to use the symmetry of $[K]$.

As a demonstration of the logic, consider Fig. 5.11 and in particular the evaluation of the sum,

$$\begin{aligned} \sum = & \sum_{j=1}^{2N_1(18)} K[10, j] u_{10} u_j = K[10, 1] u_{10} u_1 + K[10, 2] u_{10} u_2 \\ & + \dots K[10, 18] u_{10} u_{18} \end{aligned}$$

Now, the first non-zero coefficient of row ($i = 10$) is calculated from the address sequence as ;

$$I - \text{ADD}[I] + \text{ADD}[I - 1] + 1 = 5$$

Using symmetry of $[K]$, \sum may be divided into two parts :

$$\sum = \sum_1 + \sum_2$$

where $\sum_1 = K[10, 5] u_{10} u_5 + K[10, 6] u_{10} u_6 + \dots K[10, 10] u_{10} u_{10}$

and $\sum_2 = K[11, 10] u_{10} u_{11} + K[12, 10] u_{10} u_{12} + \dots K[18, 10] u_{10} u_{18}$

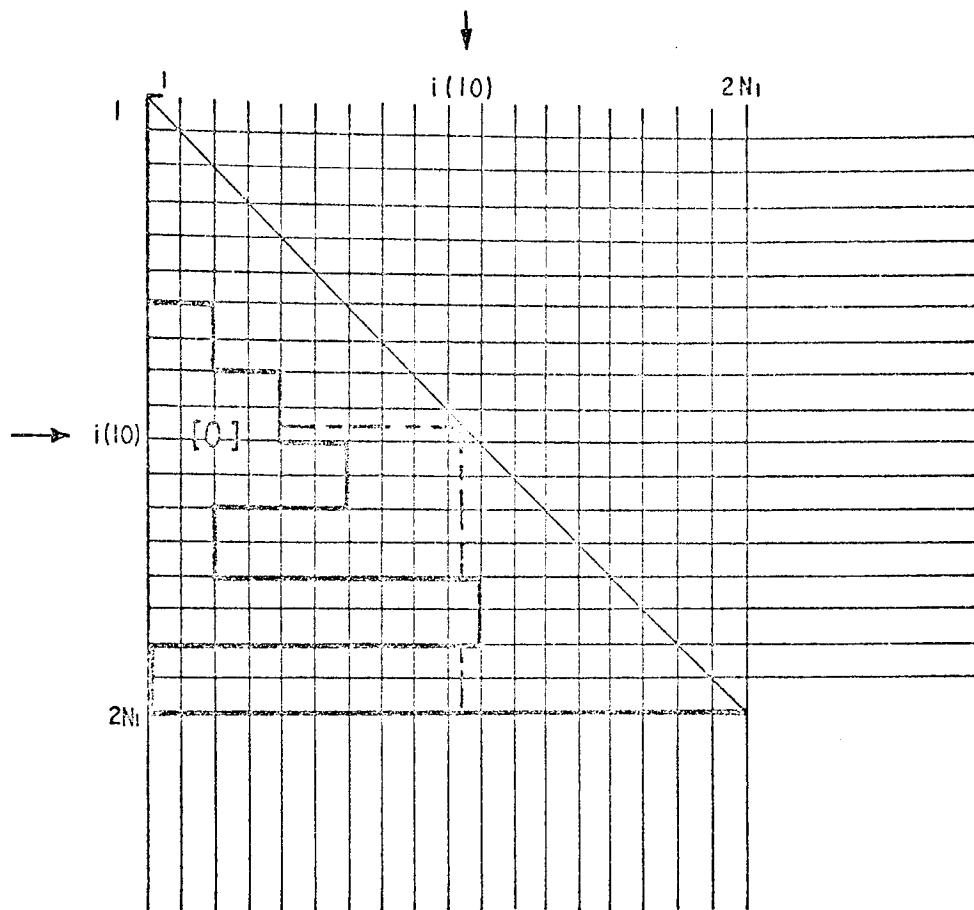


FIG 5.11 SECTION OF THE OVERALL STIFFNESS MATRIX $[K]$.

On checking the first non-zero column numbers for rows 15 and 16, Fig. 5.1, it is found that,

$$K [15, 10] = K [16, 10] = 0.0$$

Therefore,

$$\sum_2 = K[11, 10] u_{10} u_{11} + K[12, 10] u_{10} u_{12} + K[13, 10] u_{10} u_{13} \\ + K[14, 10] u_{10} u_{14} + K[17, 10] u_{10} u_{17} + K[18, 10] u_{10} u_{18}$$

This method for evaluating summations is common to all subsequent programs and its numerical implementation is identical to the symbolic procedure described above.

Coefficients of the overall stiffness matrix stored as a one-dimensional array, are accessed using the formula of Section 1, equation 4.1, i.e.

$$K[i, j] = K[\text{ADD}[i] - i + j]$$

Operations are described, with reference to the flowchart, as :

A. - Elastic coefficients G , μ and K are calculated. Coefficients of the array for the temporary storage of the terms of equation 5.4 are initialised.

B. - Coefficient $KT_{11} = K_{22}^*[2, 2]$ of equation 5.4c is calculated as ;

$$(1) \quad KT_{11} = k_c[2, 2]$$

loop on counter I from rows 1 to $2N1$.

(2) If I odd then $IA = 1$ else $IA = 0$

(3) Values of θ_i and $u_i = \begin{Bmatrix} f(R_c, \theta_i) \\ g(R_c, \theta_i) \end{Bmatrix}$ are calculated depending on whether I is odd or even.

(4) First non-zero column number of row I is found, CI .

(5) Coordinate transformation for u_x if direction of global X and local crack tip x -axis do not coincide.

loop on counter J from column CI to $2N1$.

(6) If J is less than or equal to I then the sum is proceeding along row I as given by \sum_1 and the appropriate coefficient in $[K]$ is $K[I, J]$. If, however, J is greater than I then the sum is proceeding down the column I as given by \sum_2 .

and the position of the first non-zero coefficients must be checked, CJ. If CJ is found to be greater than I, then the coefficient in $[K]$ is zero and counter J increases by one. If, however, the coefficient is not zero, then indices are exchanged and coefficient in $[K]$ is $K[J, I]$.

(7) If J odd then $JA = 1$ else $JA = 0$.

(8) Values of θ_j and $u_j = \begin{cases} f(R_c, \theta_j) \\ g(R_c, \theta_j) \end{cases}$ are calculated depending on whether J is odd or even.

(9) Coordinate transformation similar to step (5) is executed.

(10) Coefficient KT_{11} is calculated, as in equation 5.4c.

(11) Similarly $KT_{12} = K_{22}^* [1, 2]$ is calculated for odd values of I ($IA = 1$) according to equation 5.4b.

End of loops on counters J and I.

C. - Coefficients $KT_{22} = K_{22}^* [1, 1]$, equation 5.4a, is evaluated in an identical manner to that described in steps B(2) to B(11) above.

D. - Array $[KT]$ is declared for the storage of coefficients of equation 5.4d as

$$KT[1, J + 2] = K_{12}^* [2, J] \quad J = 1, 2, 3, \dots, 4N1$$

Again program logic follows that in step B.

E. - Array $[KT]$ is employed for the storage of coefficients of equation 5.4e as

$$KT[2, J] = K_{12}^* [1, J] \quad J = 1, 2, 3, \dots, 4N1$$

Coefficients are formed as in step B.

F. - Equation 5.4g is carried out and the address sequence is modified accordingly. This step may be visualised with reference to Figs. 5.9 and 5.10 where rows and columns 1 to $2N1$ in $[K]$ are eliminated and $[K_{22}]$ replaces them.

loop on counter I from 1 to $NG - 2N1$

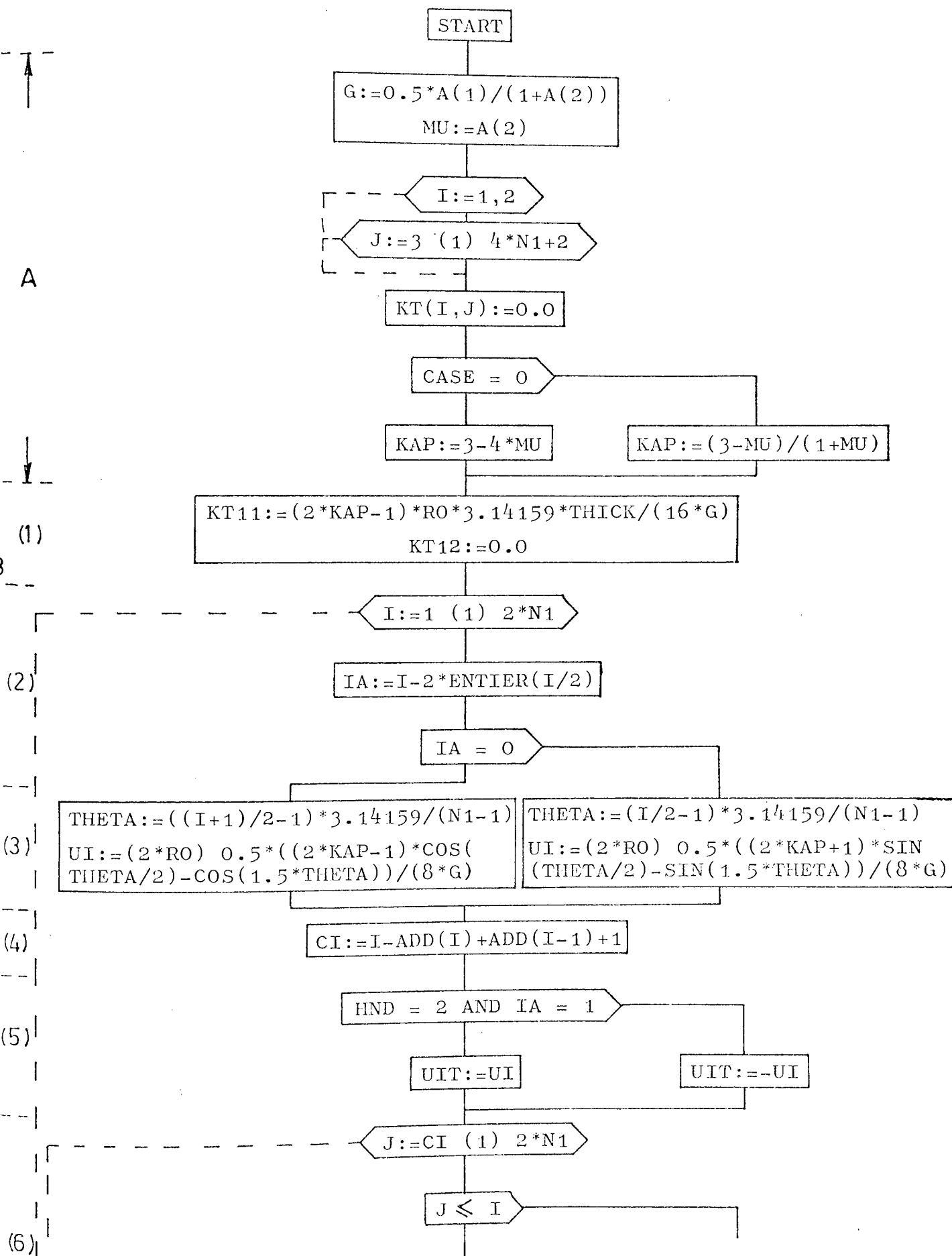
(1) Column number of first non-zero coefficient in row $(I + 2N1)$ is calculated, CI.

- (2) If this column number is less than $2N1$ then first $2N1$ columns are eliminated by making $CI = 2N1 + 1$.
- (3) New coefficient $ADD[I]$ is found by adding the number of columns in row $I + 2N1$ to $ADD[I - 1]$.
- (4) Coefficients of $[K]$ are relocated using the coefficients of the address sequence previously generated in step 3 for row I and coefficients of the original address array for row $(I + 2N1)$

end of loop on counter I .

- G. - Load vector is relocated as described by equation 5.4h.
- H. - Equation 5.4f is implemented for rows $NG - 2N1 + 1$ and $NG - 2N1 + 2$.
- I. - Address coefficients for rows $NG - 2N1 + 1$ and $NG - 2N1 + 2$ of $[K^*]$ are calculated.
- J. - Coefficients of rows $NG - 2N1 + 1$ and $NG - 2N1 + 2$ are initialised.
- K. - Coefficients $KT11$, $KT12$ and $KT22$ are located in their correct positions in $[K^*]$, Fig. 5.10.
- L. - Coefficients of the array KT are located in their correct positions in K^* , Fig. 5.10.

PROCEDURE : MODE1A(N1,RO,NG,NSETF,A,ADD,K,Q,CASE,HND,THICK)



(6)

(7)

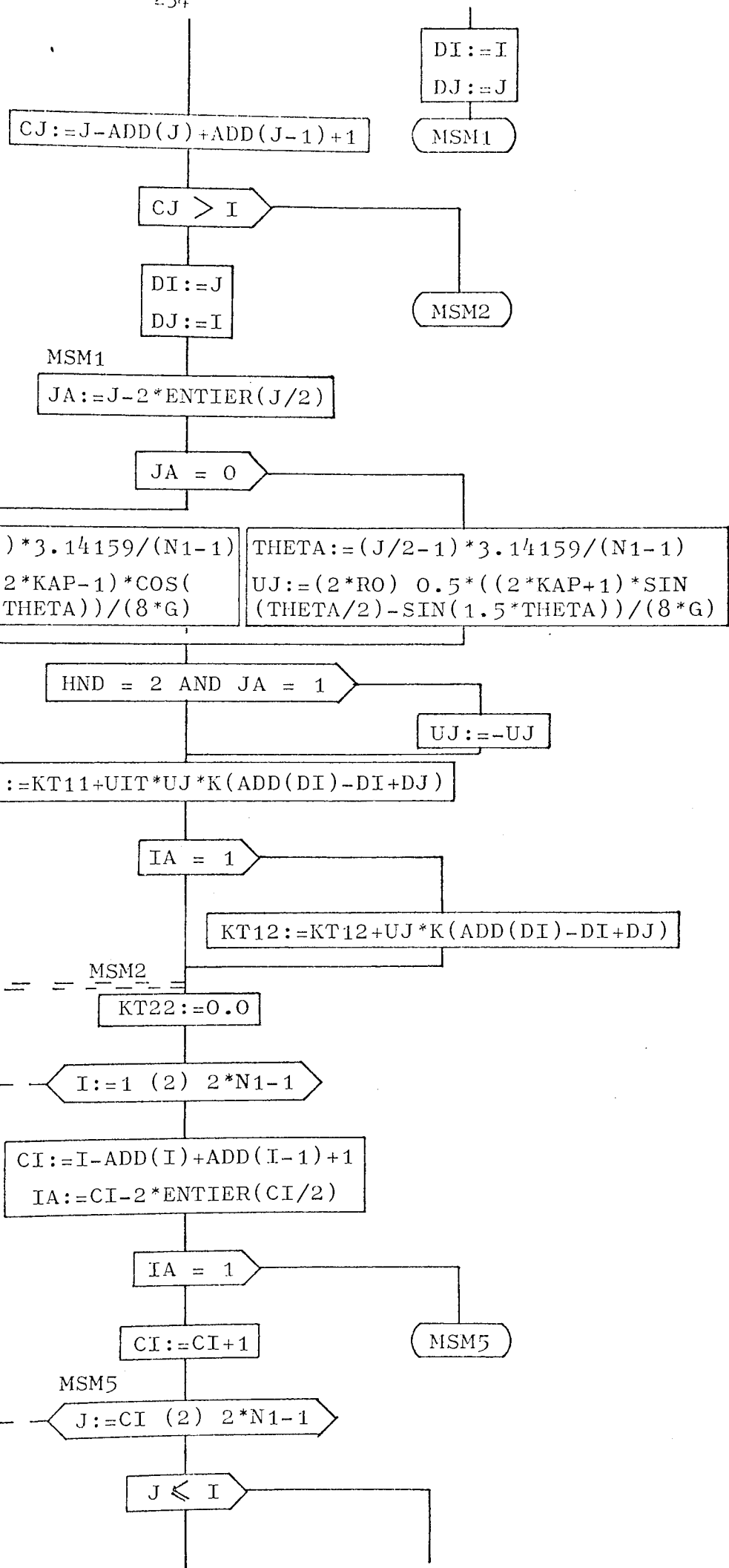
(8)

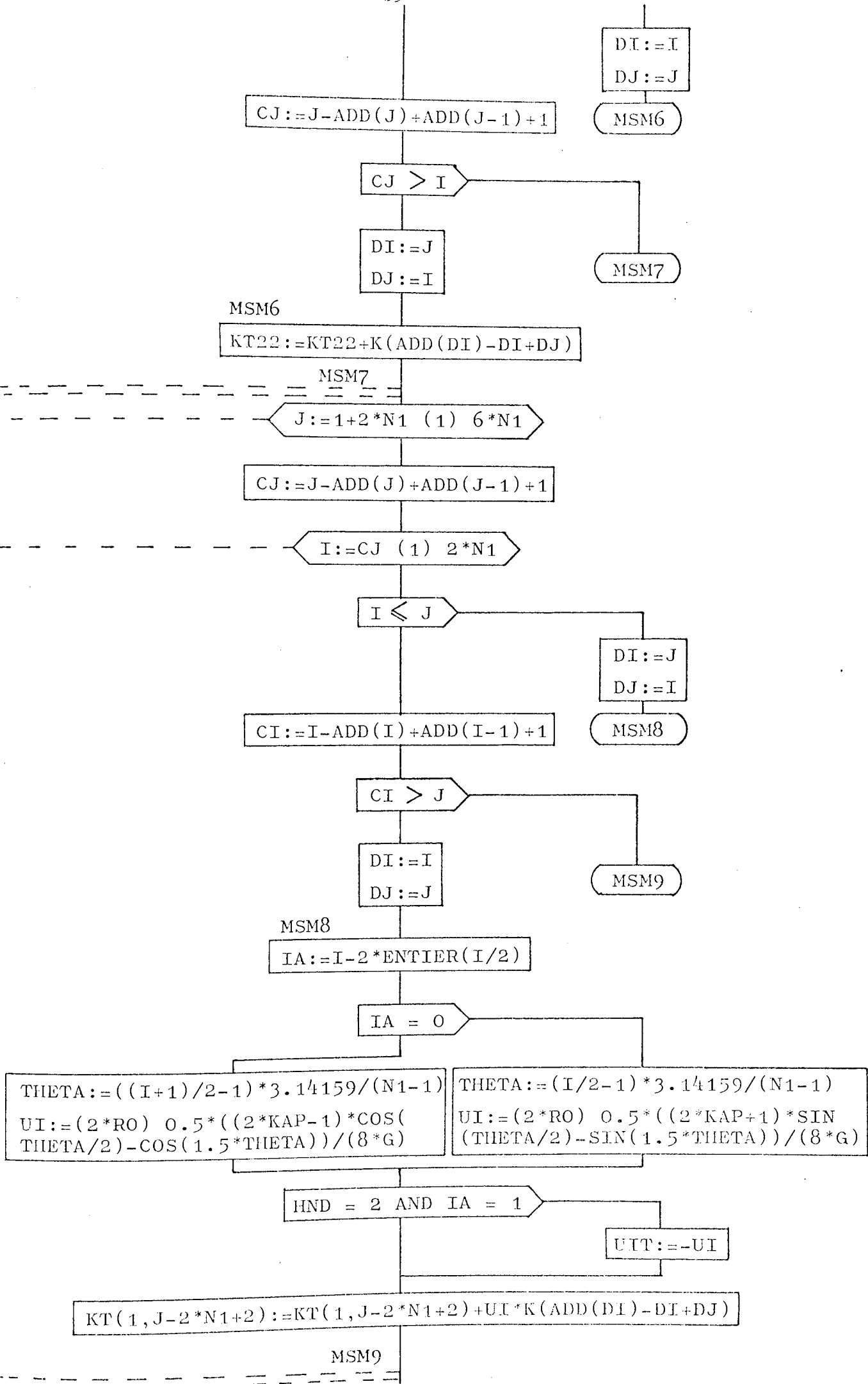
(9)

(10)

(11)

C





E

 $J := 1 + 2 * N1 \quad (1) \quad 6 * N1$
 $CJ := J - ADD(J) + ADD(J-1) + 1$
 $JA := CJ - 2 * ENTIER(CJ/2)$
 $JA = 1$
 $CJ := CJ + 1$

MSM10

MSM10

 $I := CJ \quad (2) \quad 2 * N1 - 1$
 $I \leq J$
 $DI := J$
 $DJ := I$
 $CI := I - ADD(I) + ADD(I-1) + 1$

MSM11

 $CI > J$
 $DI := I$
 $DJ := J$

MSM12

MSM11

 $KT(2, J - 2 * N1 + 2) := KT(2, J - 2 * N1 + 2) + K(ADD(DI) - DI + DJ)$

MSM12

 $I := 1 \quad (1) \quad NG - 2 * N1$
 $CI := I + 2 * N1 - ADD(I + 2 * N1) + ADD(I + 2 * N1 - 1) + 1$
 $CI \leq 2 * N1$
 $CI := 2 * N1 + 1$
 $I = 1$
 $ADD(I) := I + 2 * N1 - CI + ADD(I-1) + 1$
 $ADD(I) := 1$
 $J := CI \quad (1) \quad I + 2 * N1$
 $K(ADD(I) - I + J - 2 * N1) := K(ADD(I + 2 * N1) - I - 2 * N1 + J)$
 $I := 1 \quad (1) \quad NG - 2 * N1$
 $J := 1 \quad (1) \quad NSETF$
 $Q(I, J) := Q(I + 2 * N1, J)$

F

G

I := 1 (1) NSETF

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+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

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

```

PROCEDURE' MODE1A(N1,RO,NG,NSETF,A,ADD,K,O,CASE,THICK,HND);
VALUE' N1,RO,NG,NSETF,CASE,THICK;
INTEGER' N1,NG,NSETF,CASE,HND;
REAL' RO,THICK;
INTEGER' 'ARRAY' ADD;
REAL' 'ARRAY' K,O,A;
BEGIN' 'INTEGER' IA,CI,DI,DJ,CJ,JA;
REAL' UI,UJ,KT11,KT12,KT22,THETA,KAP,G,MU,UIT;
ARRAY' KT[1:2,3:4*N1+2];
MU:=0.5*AL1/(1+A[2]); MU:=AL2;
WRITE('('('('20')'THIS%PROCEDURE%MODIFIES%THE%OVERALL%STIFFNESS%
MATRIX%FOR%ASINGLE%TIP%UNDER%MODE1%LOADING'('20')'NUMBER%OF%MODAL%
DINTS%ON%CORE%=%')'); PRINT(N1,3,0);
WRITE('('('('C')'CORE%RADIUS%=%')'); PRINT(RO,0,3);
WRITE('('('('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[I,J]:=0.0;
IF' CASE=0 'THEN' KAP:=(3-MU)/(1+MU) 'ELSE' KAP:=3-4*MU;
KT11:=(2*KAP-1)*RO*3.14159/(16*G)*THICK; 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);
I:=(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);
I:=(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' UIT:=-UI 'ELSE' UIT:=UI;
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' 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);
J:=(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);
J:=(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+UIT*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' MSMS 'ELSE' CI:=CI+1;
MSMS: 'FOR' J:=CI 'STEP' 2 'UNTIL' 2*N1-1 'DO'
'BEGIN' 'IF' J 'LE' 1 'THEN' 'BEGIN' DI:=J; DJ:=J; 'GOTO' MSM6; 'END';
CJ:=J-ADD[J]+ADD[J-1]+1;
'IF' CJ 'GT' 1 '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:=J; DJ:=J; 'GOTO' MS*8; '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(I/2);
'IF' IA=0 'THEN'
'BEGIN' THETA:=(I/2-1)*3.14159/(N1-1);
UI:=(2*RD)↑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*RD)↑0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(8*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[DI]-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[DI]-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' NSETF 'DO'
Q[I,J]:=Q[I+2*N1,J];
'FOR' I:=1 'STEP' 1 'UNTIL' NSETF '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+2;
'FOR' I:=1 'STEP' 1 'UNTIL' NG-2*N1 'DO'
K[ADD[NG-2*N1+1]-NG+2*N1-1+1]:=K[ADD[NG-2*N1+2]-NG+2*N1-2+1]:=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 MODE1A;

```

↑*****

5.3 SEVERAL CRACK TIPS, SYMMETRIC MODE I CASE

As an example of a multi-tip problem, consider the rectangular plate containing three cracks each of length $2a$, Fig. 5.12. By symmetry only one quadrant of the plate need be analysed with the boundary conditions as shown, Fig. 5.13. The problem reduces, therefore, to the determination of the stress intensity factors K_{I1} , K_{I2} , K_{I3} and the rigid body displacements in the x -direction δ_{x1} , δ_{x2} and δ_{x3} corresponding to each tip, together with the nodal displacements of the unconstrained nodes in the finite element mesh.

The problem may be analysed using an extension of the theory presented in section 5.2, where matrices are partitioned according to the constrained nodes for each singularity. A modified potential energy may be found and minimised with respect to the undetermined parameters as previously described. Subsequent reassembly gives a modified stiffness matrix and load vector which is then solved for the undetermined coefficients. An analogous but somewhat simpler approach, from a programming viewpoint, is to consider the three tip problem as a triple application of the single tip procedure of section 5.2.

Refer to the system equations for the three-point problem, Fig. 5.15, with the nodal numbering sequence as in Fig. 5.14. The matrix may be partitioned and areas related to the tips as shown. Once again the node numbering around cores plays a critical role in determining the final form for the overall stiffness matrix, $[K]$. Array NS contains node numbers at the start of the first ring of nodes surrounding a given tip, Fig. 5.14. From Fig. 5.15 it is a simple matter to eliminate rows and columns corresponding to core nodes and to construct a modified stiffness matrix $[K^*]$, Fig. 5.16. The coefficients $KT11$, $KT12$, etc. of procedure ~~MODEIA~~, section 5.2, may be evaluated in a similar manner and located in their correct positions in $[K^*]$ as in Fig. 5.16; where $(KT11)_1$ refers to coefficient $KT11$ for crack tip number 1.

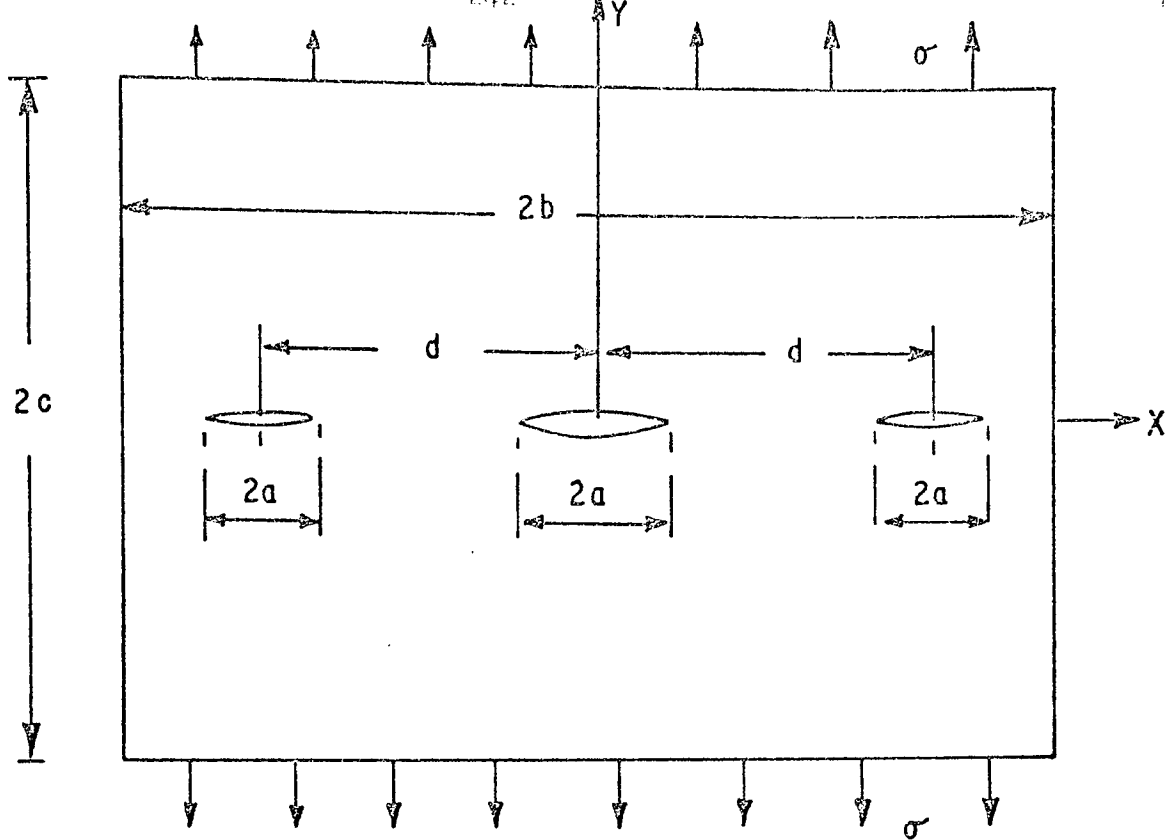


FIG.5.12 RECTANGULAR PLATE CONTAINING THREE CRACKS.

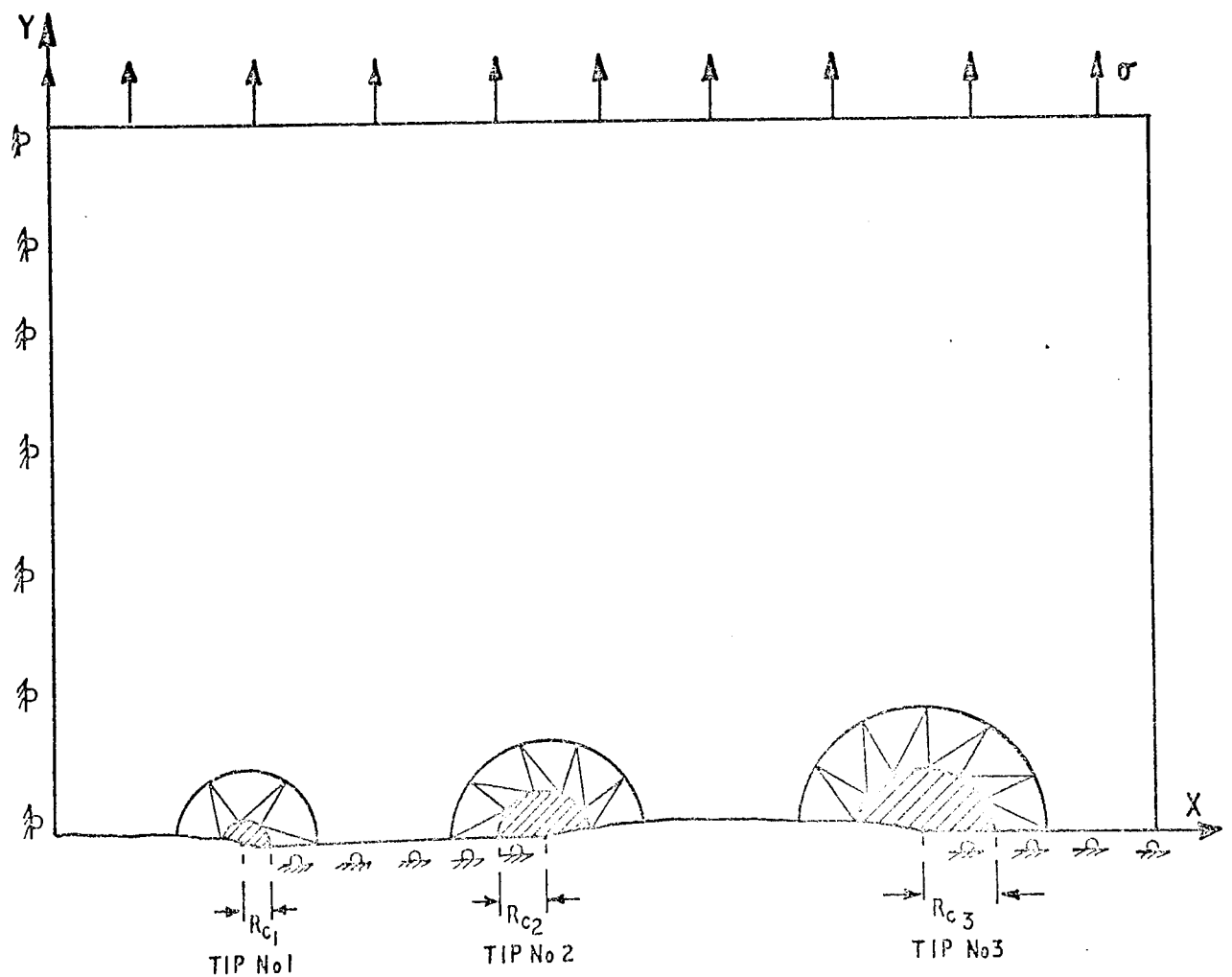


FIG.5.13 QUADRANT OF PLATE WITH CORE ELEMENTS IN POSITION.

NS[1] = 1
 NS[2] = 30
 NS[3] = 70
 NS[4] = 108

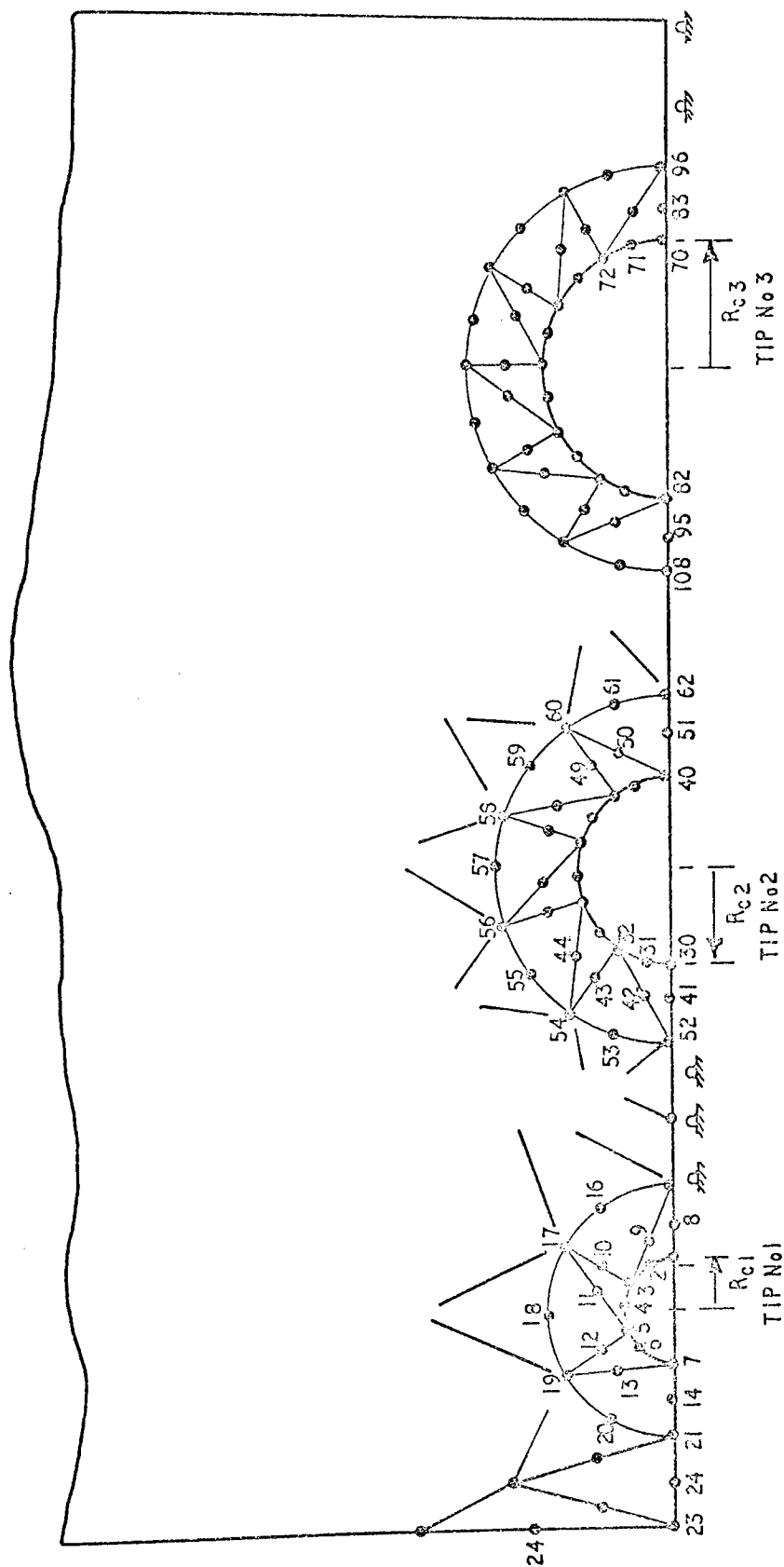


FIG. 5.14 ENLARGEMENT OF TIPS SHOWING CORE NUMBERING.

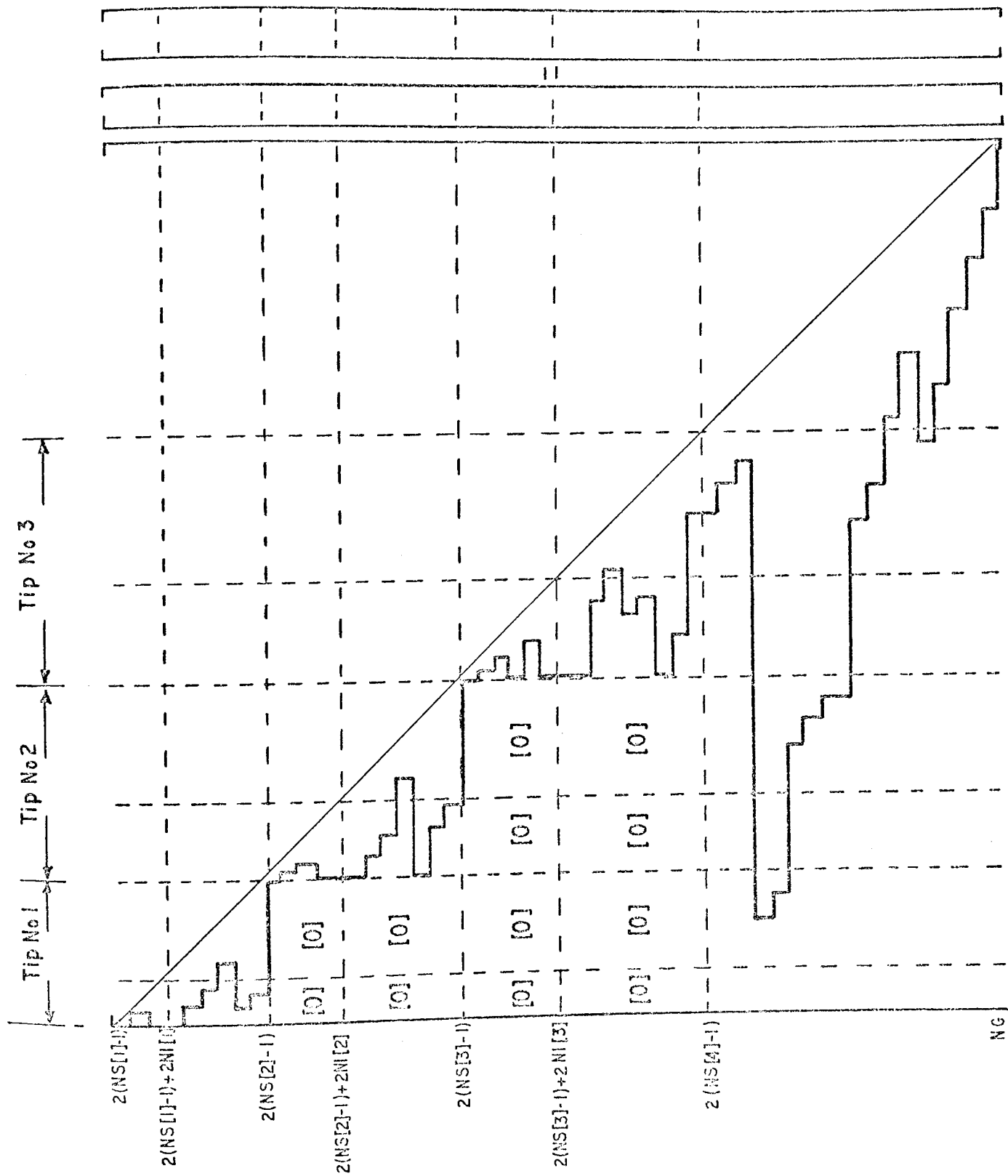


FIG. 5.15 STIFFNESS RELATIONS FOR ORIGINAL PROBLEM.

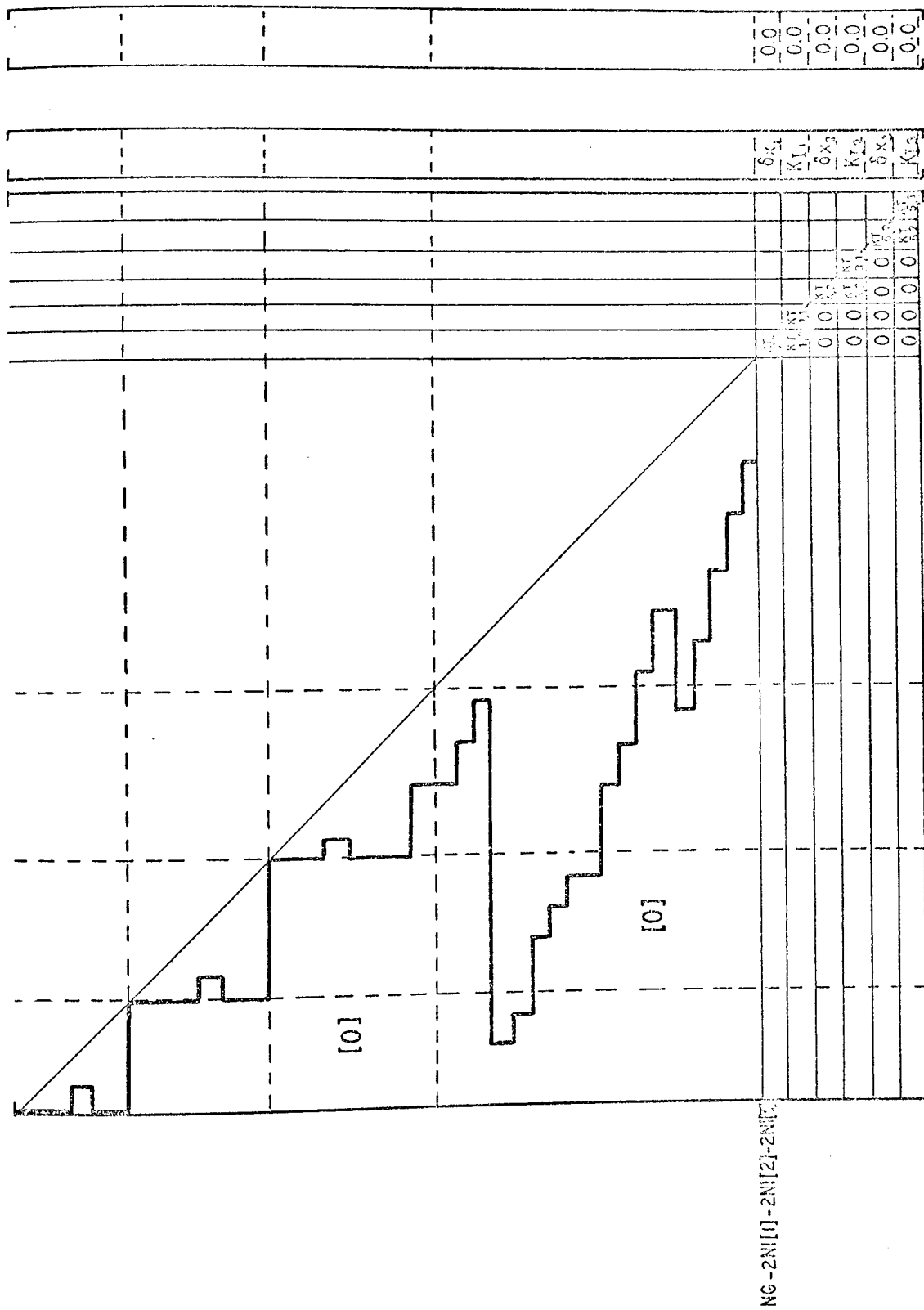


FIG 5.16 MODIFIED STIFFNESS EQUATIONS AFTER INCLUDING THREE CORE ELEMENTS.

Turning now to the flowchart of procedure MODEINTIPS, an array KT is declared for the storage of the relevant coefficients prior to their insertion in $[K^*]$. For the example chosen, with three tips, this array has the form :

		1	2	3	4	5	4 x GRT + 2
Tip 1	{	1	$(KT11)_1$	$(KT12)_1$	$KT[1,1]_1$	$KT[1,2]_1$ $KT[1,4N1[z]]_1$
		2		$(KT22)_1$	$KT[2,1]_1$	$KT[2,2]_2$
Tip 2	{	3	$(KT11)_2$	$(KT12)_2$	$KT[1,1]_2$	$KT[1,2]_2$
		4		$(KT22)_2$	$KT[2,1]_2$	$KT[2,2]_2$
Tip 3	{	5	$(KT11)_3$	$(KT12)_3$	$KT[1,1]_3$	$KT[1,2]_3$
		6		$(KT22)_3$	$KT[2,1]_3$	$KT[2,2]_3$

where GRT is the maximum number of nodes surrounding any given core.

Generation of the relevant coefficients of this array may be summarised, using equations 5.4 as follows;

for crack tip number, z (Here z = 1,2,3)

If $V = 2 (NS[z] - 1)$, Fig. 5.14, then,

$$KT[2z-1,1] = k_c[2,2]_z + \sum_{i=(V+1),(V+2) \dots}^{2N1[z]+V} \sum_{j=(V+1),(V+2) \dots}^{2N1[z]+V} K[i,j] u_i u_j$$

$$KT[2z-1,2] = k_c[2,1]_z + \sum_{i=(V+1),(V+3) \dots}^{2N1[z]+V-1} \sum_{j=(V+1),(V+2) \dots}^{2N1[z]+V} K[i,j] u_j$$

$$KT[2z,2] = k_c[1,1]_z + \sum_{i=(V+1),(V+3) \dots}^{2N1[z]+V-1} \sum_{j=(V+1),(V+3) \dots}^{2N1[z]+V-1} K[i,j]$$

$$KT[2z-1,j] = \sum_{i=(V+1),(V+2) \dots}^{2N1[z]+V} K[j+2N1[z]+V-2,i] u_j$$

$$j = 3,4 \dots 4N1[z]+2$$

$$KT [2z, j] = \sum_{i = (V+1), (V+3) \dots}^{2N1 [z] + V - 1} K [j + 2N1 [z] + V - 2, i]$$

$$j = 3, 4, \dots 4N1 [z] + 2$$

$$u_j = \begin{cases} f(R_{j+1} [z], \theta_{\frac{j+1}{2}}) & j \text{ odd} \\ g(R_{j/2} [z], \theta_{j/2}) & j \text{ even} \end{cases}$$

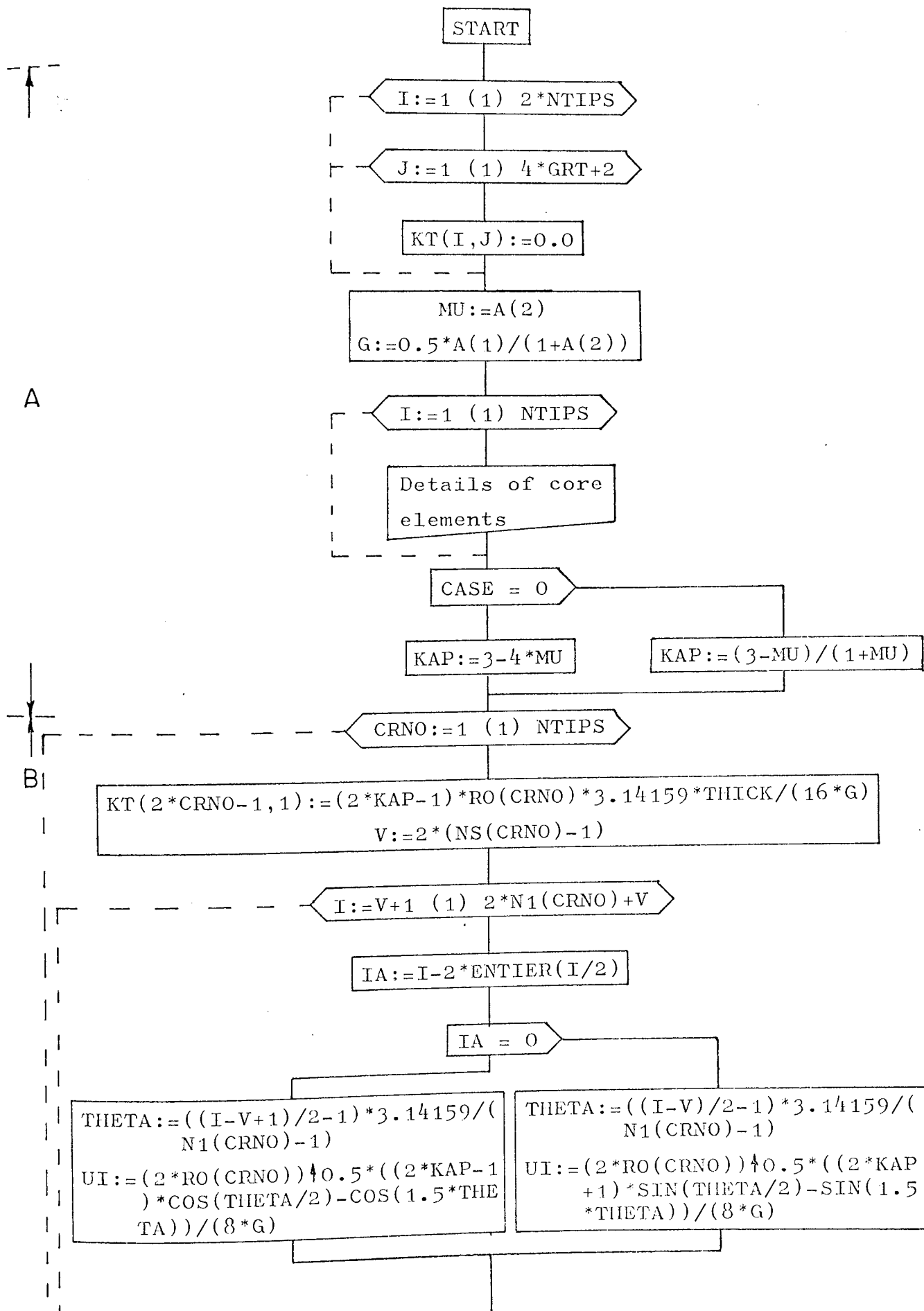
where $R_{j+1} [z]$, are the radii of the core elements.

These coefficients for each tip are evaluated numerically in a similar manner to the preceeding example of section 5.2, for a single tip.

A flowchart and program listing follows for the general 'N-tip' case and the various steps are described, with reference to the flowchart, as :

- A. - Array KT is initialised, elastic constants G, MU are read as data
K is evaluated and the details of the tips are printed.
- B. - Loop is constructed over all the tips present and the coefficients of array KT are evaluated as described above.
- C. - Rows and columns, of rows in the overall stiffness matrix corresponding to core nodes, are eliminated. The address sequence is modified and the coefficients of $[K]$ are relocated in $[K^*]$.
- D. - Rows corresponding to $2(NS[4] - 1) + 1$ to NG, Fig. 5.15, are relocated in $[K^*]$. Once again certain columns are eliminated and the address sequence changed.
- E. - Coefficients of the load vector $\{F\}$ are positioned in $\{F^*\}$, in a similar manner to equation 5.4h.
- F. - Terms in load vector $\{F_2^*\}$, equation 5.4f are set to zero.
- G. - Address coefficients for rows corresponding to
 $NG - 2N1 [1] - 2N1 [2] - 2N1 [3] + 1$ to $NG - 2N1 [1] - 2N1 [2] - 2N1 [3] + 6$,
 Fig. 5.16, are calculated.
- H. - Coefficients of the rows in step G are initialised.
- I. - Coefficients of the array KT are located in their appropriate positions in the rows of step G of the modified stiffness matrix $[K^*]$, see Fig. 5.16.

PROCEDURE : MODE1NTIPS(N1,RO,NG,NSETF,A,ADD,K,Q,CASE,THICK,
NTIPS,HND,GRT,NS)



CI := I - ADD(I) + ADD(I-1) + 1

HND(CRNO) = 2 AND IA = 1

UIT := UI

UIT := -UI

J := CI (1) 2 * N1(CRNO) + V

J ≤ I

CJ := J - ADD(J) + ADD(J-1) + 1

DI := I

DJ := J

MSM1

CJ > I

MSM2

DI := J

DJ := I

MSM1

JA := J - 2 * ENTIER(J/2)

JA = 0

THETA := ((J - V + 1) / 2 - 1) * 3.14159 / (N1(CRNO) - 1)

UJ := (2 * RO(CRNO))^{0.5} * ((2 * KAP - 1) * COS(THETA/2) - COS(1.5 * THETA)) / (8 * G)

THETA := ((J - V) / 2 - 1) * 3.14159 / (N1(CRNO) - 1)

UJ := (2 * RO(CRNO))^{0.5} * ((2 * KAP + 1) * SIN(THETA/2) - SIN(1.5 * THETA)) / (8 * G)

HND(CRNO) = 2 AND JA = 1

UJ := -UJ

KT(2 * CRNO - 1, 1) := KT(2 * CRNO - 1, 1) + UIT * UJ * K(ADD(DI) - DI + DJ)

MSM2

I := V + 1 (2) 2 * N1(CRNO) + V - 1

CI := I - ADD(I) + ADD(I-1) + 1

J := CI (1) 2 * N1(CRNO) + V

J ≤ I

CJ := J - ADD(J) + ADD(J-1) + 1

DI := I

DJ := J

MSM3

CJ > I

DI:=J

DJ:=I

MSM4

MSM3

JA:=J-2*ENTIER(J/2)

JA = 0

THETA:=((J-V+1)/2-1)*3.14159/(
N1(CRNO)-1)

UJ:=(2*RO(CRNO))^{1/2}*0.5*((2*KAP-1)
)*COS(THETA/2)-COS(1.5*
THETA))/(8*G)

THETA:=((J-V)/2-1)*3.14159/(
N1(CRNO)-1)

UJ:=(2*RO(CRNO))^{1/2}*0.5*((2*KAP+1)
)*SIN(THETA/2)-SIN(1.5*THETA
))/(8*G)

HND(CRNO = 2 AND JA = 1

UJ:=-UJ

KT(2*CRNO-1,2):=KT(2*CRNO-1,2)+UJ*K(ADD(DI)-DI+DJ)

MSM4

I:=V+1 (2) 2*N1(CRNO)+V-1

CI:=I-ADD(I)+ADD(I-1)+1
IA:=CI-2*ENTIER(CI/2)

IA = 1

CI:=CI+1

MSM5

MSM5

J:=CI (2) 2*N1(CRNO)+V-1

J ≤ I

CJ:=J-ADD(J)+ADD(J-1)+1

DI:=I
DJ:=J

MSM6

CJ > I

DI:=J

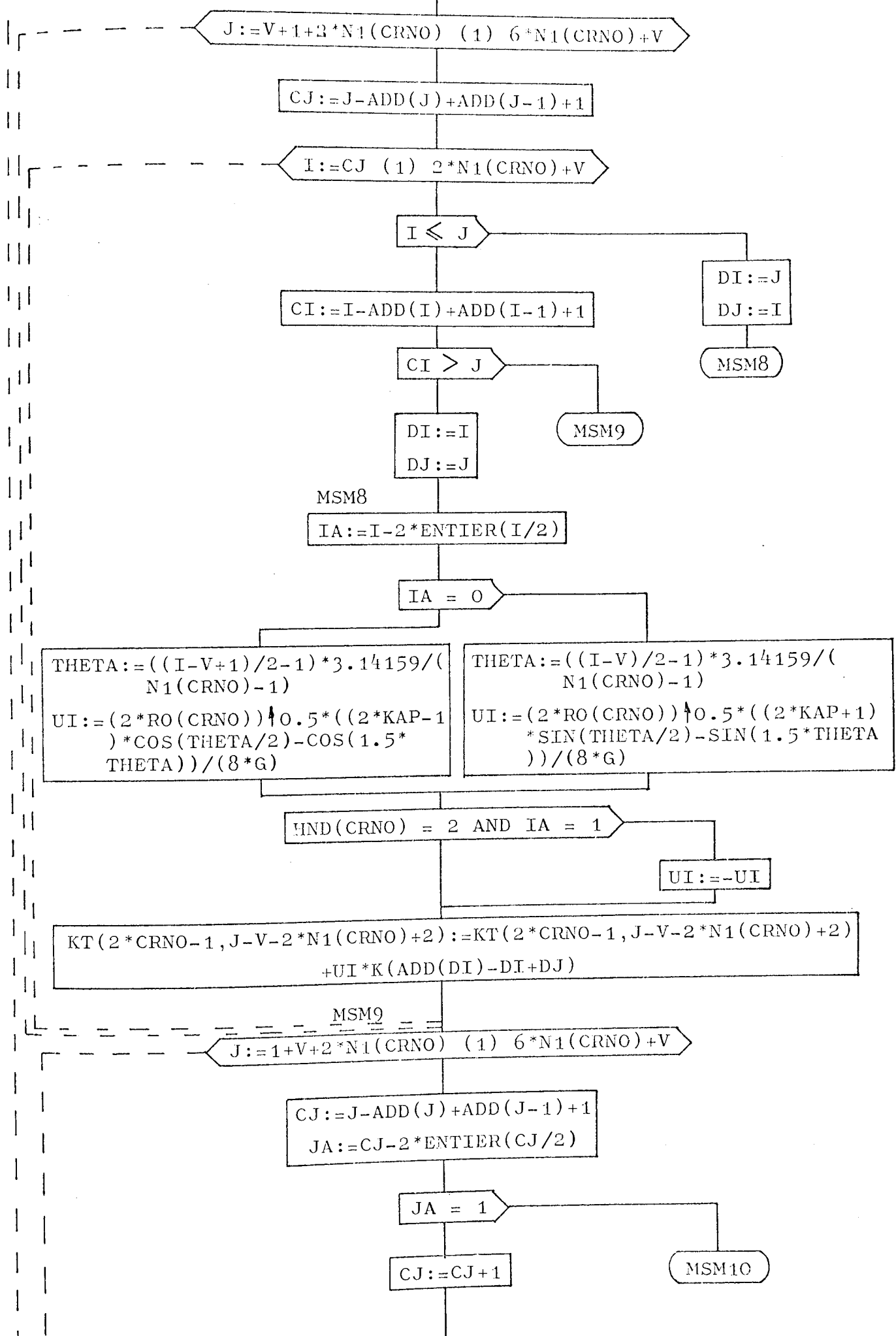
DJ:=I

MSM7

MSM6

KT(2*CRNO,2):=KT(2*CRNO,2)+K(ADD(DI)-DI+DJ)

MSM7



$I := CJ \ (2) \ 2 * N1(CRNO) + V - 1$

$I \leq J$

$CI := I - ADD(I) + ADD(I-1) + 1$

$DI := J$
 $DJ := I$

MSM11

$CI > J$

MSM12

$DI := I$
 $DJ := J$

MSM11

$KT(2 * CRNO, J - V - 2 * N1(CRNO) + 2) := KT(2 * CRNO, J - V - 2 * N1(CRNO) + 2) + K(ADD(DI) - DI + DJ)$

MSM12

$Z := 1 \ (1) \ NTIPS$

$V := 2 * (NS(Z) - 1)$

$Z = 1$

$TEM := 0$

$TEM := TEM + 2 * N1(Z - 1)$

$W := 2 * (NS(Z + 1) - 1)$

$I := 2 * N1(Z) + V + 1 \ (1) \ W$

$CI := I - ADD(I) + ADD(I-1) + 1$

$CI \leq 2 * N1(Z) + V$

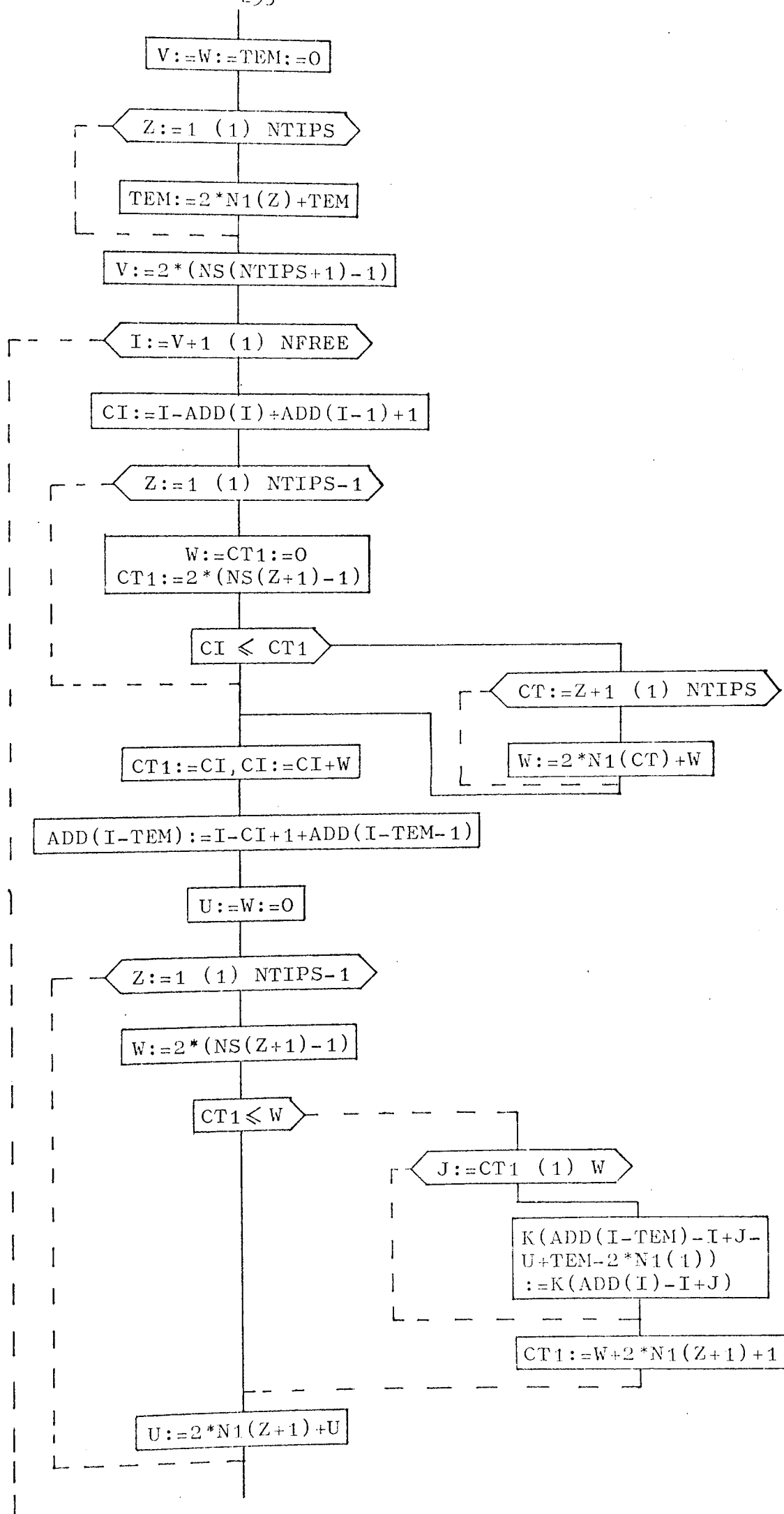
$CI := 2 * N1(Z) + V + 1$

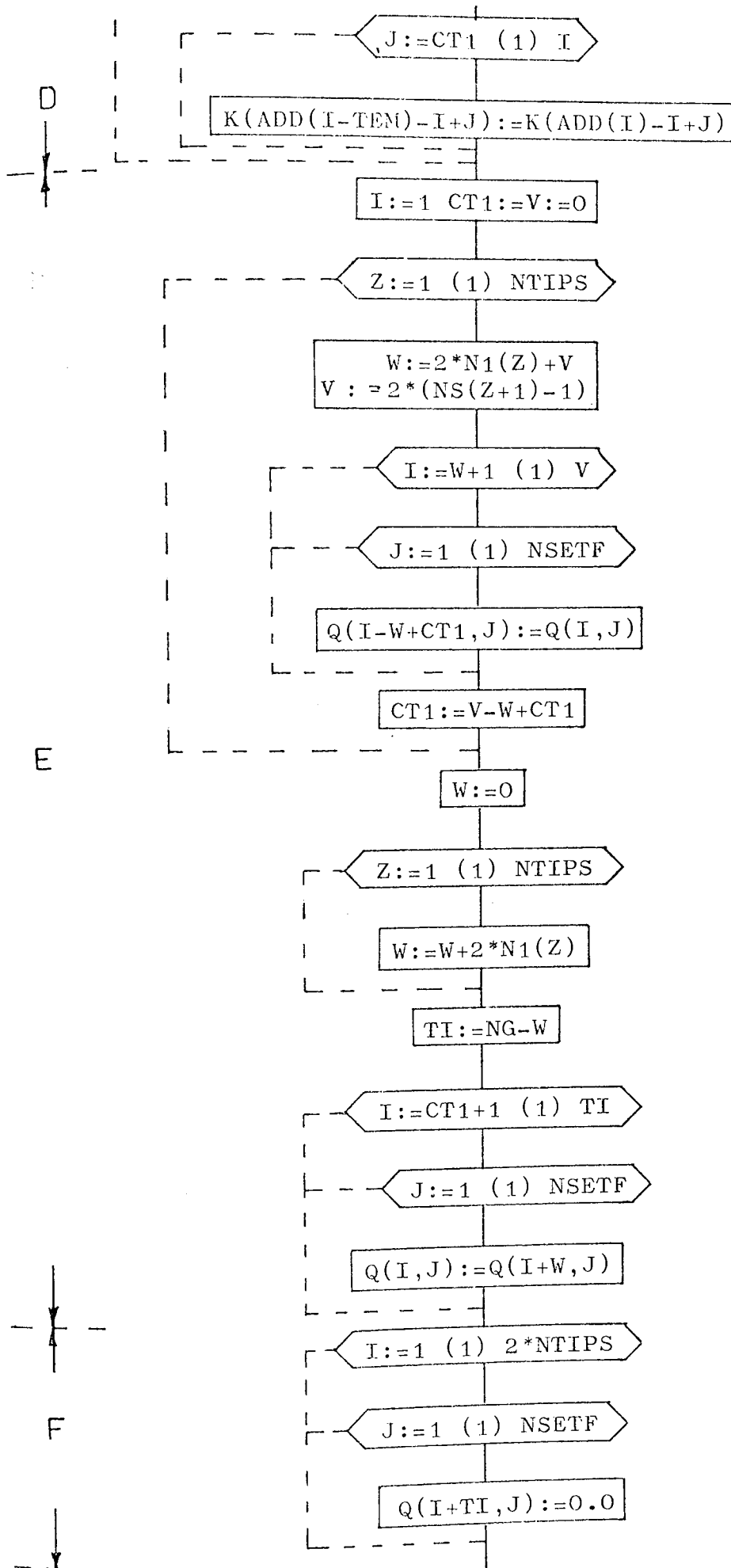
$ADD(I - 2 * N1(Z) - TEM) := I - CI + ADD(I - 2 * N1(Z) - TEM - 1) + 1$

$J := CI \ (1) \ I$

$K(ADD(I - 2 * N1(Z) - TEM) - I + J) := K(ADD(I) - I + J)$

D





CT:=0

I:=1,2

ADD(TI+I):=ADD(TI+I-1)+TI+I

I:=1(1)NTIPS-1

CT:=CT+2*(NS(I+1)-NS(I)-N1(I))

J:=1,2

ADD(TI+2*I+J):=ADD(TI+2*I+J-1)+TI-CT+2*I+J

I:=1 (1) 2*NTIPS

CI:=TI+I-ADD(TI+I)+ADD(TI+I-1)+1

J:=CI (1) TI+I

K(ADD(TI+I)-I-TI+J):=0.0

I:=1 (2) 2*NTIPS-1

K(ADD(TI+I)):=KT(I+1,2)
K(ADD(TI+I+1)):=KT(I,1)
K(ADD(TI+I+1)-1):=KT(I,2)

CT1:=0

I:=1 (1) NTIPS

Z:=2*(NS(I)-1)-CT1
CT1:=CT1+2*N1(I)
W:=4*N1(I)+2
U:=TI+2*I
V:=2*I

J:=3 (1) W

K(ADD(U-1)-U+Z-1+J):=KT(V,J)
K(ADD(U)-U+Z-2+J):=KT(V-1,J)

END

G

H

I

```

'PROCEDURE' MODIFITIPS(N1,RO,AG,NSFTE,A,ADD,R,0,CASE,THICK,NTIPS,
                      RND,GR,ES);
'VALUE' NG,NSFTE,CASE,THICK,NTIPS,GET;
'INTEGER' NG,ASFTE,CASE,NTIPS,GR;
'REAL' THICK;
'INTEGER' 'ARRAY' N1,ADD,RND,NS;
'REAL' 'ARRAY' RO,K,0,A;
'BEGIN' 'COMMENT' THIS PROCEDURE IS DESIGNED TO MODIFY THE
          OVERALL STIFFNESS MATRIX WHERE SEVERAL CRACK TIPS ARE
          PRESENT. CORES MAY HAVE DIFFERENT RADII AND POINTS
          ON THEIR BOUNDARIES.;
'INTEGER' IA,CI,DI,DJ,CJ,JA,CN0,II,7,V,CN0,W,CT1,TE0,0,CT;
'REAL' UI,UJ,THETA,KAP,UIT,W0,G;
'ARRAY' KT[1:2*NTIPS,1:4*GR+2];
WRTTETEXT('('('2C')'THIS PROCEDURE MODIFIES THE OVERALL STIFFNESS%
MATRIX WHERE SEVERAL TIPS ARE PRESENT')');
'FOR' J:=1 'STEP' 1 'UNTIL' 2*NTIPS 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 4*GR+2 'DO' KT[I,J]:=0.0;
MU:=AT[2]; G:=0.5*A[1]/(1+A[2]);
'FOR' I:=1 'STEP' 1 'UNTIL' NTIPS 'DO'
'BEGIN' WRTTETEXT('('('4C')'DETAILS OF EXTIP AND REFZ-Z')');
PRINT(J,2,0);
WRTTETEXT('('('2C5S')'RADIUS OF CORE ELEMENT Z-Z')');
PRINT(N1[I],3,0);
WRTTETEXT('('('C5S')'RADIUS OF CORE ELEMENT Z-Z')'); PRINT(RO[I],0,4);
WRTTETEXT('('('C5S')'DIRECTION OF CRACK AT IP Z-Z')'); PRINT(RND[I],3,0);
'END';
'IF' CASE=0 'THEN' KAP:=(3-MU)/(1+MU) 'ELSE' KAP:=3-4*MU;
'FOR' CN0:=1 'STEP' 1 'UNTIL' NTIPS 'DO'
'BEGIN'
KT[2*CN0-1,1]:=(2*KAP-1)*RO[CN0]*3.14159/(16*G)*THICK;
V:=2*(AS[CN0]-1);
'FOR' I:=1+V 'STEP' 1 'UNTIL' 2*N1[CN0]+V 'DO'
'BEGIN' IA:=1-2*ENTIER(I/2);
'IF' IA=0 'THEN'
'BEGIN' THETA:=((I-V)/2-1)*3.14159/(N1[CN0]-1);
UI:=(2*RO[CN0])^0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(8*G);
'END' 'ELSE'
'BEGIN' THETA:=((I-V+1)/2-1)*3.14159/(N1[CN0]-1);
UI:=(2*RO[CN0])^0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(8*G);
'END';
'IF' I=1 'THEN' CI:=1 'ELSE' CI:=I-ADD[I]+ADD[I-1]+1;
'IF' UND[CN0]=2 'AND' IA=1 'THEN' UIT:=~UI 'ELSE' UIT:=UI;
'FOR' J:=CI 'STEP' 1 'UNTIL' 2*N1[CN0]+V '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[I]+ADD[J-1]+1;
'IF' CJ 'GT' I 'THEN' 'GOTO' MSM2;
DI:=J; DJ:=1;
'END';
MSM1: JA:=J-2*ENTIER(J/2);
'IF' JA=0 'THEN'
'BEGIN' THETA:=((J-V)/2-1)*3.14159/(N1[CN0]-1);
UJ:=(2*RO[CN0])^0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(8*G);
'END' 'ELSE'
'BEGIN' THETA:=((J-V+1)/2-1)*3.14159/(N1[CN0]-1);
UJ:=(2*RO[CN0])^0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(8*G);
'END';

```

```

IF HAD(CRNO)=2 'AND' JA=1 'THEN' DJ:=-DJ;
KT(2*CRNO-1,1):=KT(2*CRNO-1,1)+UJ*K[ADD(DI)-DI+DJ];
MSM2: 'END';
'END';
FOR I:=1+V 'STEP' 2 'UNTIL' 2*N1(CRNO)+V-1 'DO'
'REGIN' 'IF' I=1 'THEN' CI:=1 'ELSE' CI:=(1-ADD(I))+ADD(I-1)+1;
FOR J:=CI 'STEP' 1 'UNTIL' 2*N1(CRNO)+V 'DO'
'REGIN' 'IF' J 'LE' I 'THEN'
'REGIN' DI:=I; DJ:=J;
'GOTO' MSM3;
'END' 'ELSE'
'REGIN' 'IF' J=1 'THEN' CJ:=1 'ELSE' CJ:=J-ADD(J)+ADD(J-1)+1;
'IF' CJ 'GT' I 'THEN' 'GOTO' MSM4;
DI:=J; DJ:=I;
'END';
MSM3: JA:=J-2*ENTIER(J/2);
'IF' JA=0 'THEN'
'REGIN' THETA:=((J-V)/2-1)*3.14159/(N1(CRNO)-1);
UJ:=(2*RO(CRNO))^0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(K*G);
'END' 'ELSE'
'REGIN' THETA:=((J-V)+1)/2-1)*3.14159/(N1(CRNO)-1);
UJ:=(2*RO(CRNO))^0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(K*G);
'END';
IF HAD(CRNO)=2 'AND' JA=1 'THEN' DJ:=-DJ;
KT(2*CRNO-1,2):=KT(2*CRNO-1,2)+UJ*K[ADD(DI)-DI+DJ];
MSM4: 'END';
'END';
FOR J:=V+1 'STEP' 2 'UNTIL' 2*N1(CRNO)+V-1 'DO'
'REGIN' 'IF' I=1 'THEN' CI:=1 'ELSE' CI:=1-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' 2 'UNTIL' 2*N1(CRNO)+V-1 'DO'
'REGIN' 'IF' J 'LE' I 'THEN'
'REGIN' DI:=I; DJ:=J;
'GOTO' MSM6;
'END';
'IF' J=1 'THEN' CJ:=1 'ELSE' CJ:=J-ADD(J)+ADD(J-1)+1;
'IF' CJ 'GT' I 'THEN' 'GOTO' MSM7;
DI:=J; DJ:=I;
MSM6: KT(2*CRNO,2):=KT(2*CRNO,2)+K[ADD(DI)-DI+DJ];
MSM7: 'END';
'END';
FOR J:=V+1+2*N1(CRNO) 'STEP' 1 'UNTIL' 2*N1(CRNO)+V 'DO'
'REGIN' CJ:=J-ADD(J)+ADD(J-1)+1;
FOR I:=CJ 'STEP' 1 'UNTIL' 2*N1(CRNO)+V 'DO'
'REGIN' 'IF' I 'LE' J 'THEN'
'REGIN' DI:=J; DJ:=I;
'GOTO' MSM8;
'END';
CI:=1-ADD(I)+ADD(I-1)+1;
'IF' CI 'GT' J 'THEN' 'GOTO' MSM9;
DI:=I; DJ:=J;
MSM8: IA:=I-2*ENTIER(I/2);
'IF' IA=0 'THEN'
'REGIN' THETA:=((I-V)/2-1)*3.14159/(N1(CRNO)-1);
UJ:=(2*RO(CRNO))^0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(K*G);
'END' 'ELSE'
'REGIN' THETA:=((I-V)+1)/2-1)*3.14159/(N1(CRNO)-1);
UJ:=(2*RO(CRNO))^0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(K*G);
'END';

```

```

IF' HAD(CRNO)=2 'AND' IA=1 'THEN' DI:=-DI;
KT[2*CRNO-1,J-V-2*N1(CRNO)+2]:=IT[2*CRNO-1,J-V-2*N1(CRNO)+2]
+DI*K[ADD(DI)-DI+DJ];
MSM9: 'END';
'END';
FOR' J:=1+V+2*N1(CRNO) 'STEP' 1 'UNTIL' 6*N1(CRNO)+V 'DO'
'BEGIN' CJ:=J-ADD(I)+ADD(I-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(CRNO)+V-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*CRNO,J-V-2*N1(CRNO)+2]:=KT[2*CRNO,J-V-2*N1(CRNO)+2]
+K[ADD(DI)-DI+DJ];
MSM12: 'END';
'END';
'END' OF CRNO;
FOR' Z:=1 'STEP' 1 'UNTIL' NTIPS 'DO'
'BEGIN' V:=2*(NS[Z]-1);
IF' Z=1 'THEN' TEM:=0 'ELSE' TEM:=TEM+2*N1(Z-1);
W:=2*(NS[Z+1]-1);
FOR' I:=2*N1(Z)+V+1 'STEP' 1 'UNTIL' U 'DO'
'BEGIN' CI:=I-ADD(I)+ADD(I-1)+1;
IF' CI 'LE' 2*N1(Z)+V 'THEN' CI:=2*N1(Z)+V+1;
ADD(I-2*N1(Z)-TEM):=I-CI+ADD(I-2*N1(Z)-TEM-1)+1;
FOR' J:=CI 'STEP' 1 'UNTIL' J 'DO'
K[ADD(I-2*N1(Z)-TEM)-I+J]:=K[ADD(I)-I+J];
'END';
'END';
V:=W:=TEM:=0;
FOR' Z:=1 'STEP' 1 'UNTIL' NTIPS 'DO'
'BEGIN' TEM:=2*N1(Z)+TEM; 'END';
V:=2*(NS[NTIPS+1]-1);
FOR' I:=V+1 'STEP' 1 'UNTIL' FREEE 'DO'
'BEGIN' CI:=I-ADD(I)+ADD(I-1)+1;
FOR' Z:=1 'STEP' 1 'UNTIL' NTIPS-1 'DO'
'BEGIN' W:=CI:=0;
CI1:=2*(NS[Z+1]-1);
IF' CI 'LE' CI1 'THEN'
'BEGIN' FOR' CT:=Z+1 'STEP' 1 'UNTIL' NTIPS 'DO' U:=2*N1(CT)+W;
'GOTO' OUT1;
'END';
'END';
OUT1: CI1:=CI; CI:=CI+W;
ADD(I-TEM):=I-CI+1+ADD(I-TEM-1);
U:=W:=0;
FOR' Z:=1 'STEP' 1 'UNTIL' NTIPS-1 'DO'
'BEGIN' W:=2*(NS[Z+1]-1);
IF' CT1 'LE' W 'THEN'
'BEGIN' FOR' J:=CT1 'STEP' 1 'UNTIL' U 'DO'
K[ADD(I-TEM)-I+J-U+TEM-2*N1(1)]:=K[ADD(I)-I+1];
CT1:=W+2*N1(Z+1)+1;
'END';
U:=2*N1(Z+1)+U;
'END';

```

```

'FOR' J:=CT1 'STEP' 1 'UNTIL' 1 'DO'
  K[ADD[I-TF+1]-I+J]:=K[ADD[I]-I+J];
'END';
I:=1; CT1:=V:=0;
'FOR' Z:=1 'STEP' 1 'UNTIL' NTIPS 'DO'
  'BEGIN' W:=2*N1[Z]+V;
  V:=2*(NS[Z+1]-1);
  'FOR' I:=1 'STEP' 1 'UNTIL' V 'DO'
    'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q[I-I+CT1,J]:=Q[I,J];
  CT1:=V-W+CT1;
  'END';
W:=0;
'FOR' Z:=1 'STEP' 1 'UNTIL' NTIPS 'DO' U:=U+2*N1[Z];
TI:=NG-W;
'FOR' I:=CT1+1 'STEP' 1 'UNTIL' TI 'DO'
  'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]:=Q[I+W,J];
  'FOR' I:=1 'STEP' 1 'UNTIL' 2*NTIPS 'DO'
    'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q[I+TI,J]:=0.0;
  CT:=0;
  'FOR' I:=1,2 'DO' ADD[TI+I]:=ADD[TI+I-1]+TI+1;
  'FOR' I:=1 'STEP' 1 'UNTIL' NTIPS-1 'DO'
    'BEGIN' CT:=CT+2*(NS[I+1]-NS[I]-N1[I]);
    'FOR' J:=1,2 'DO' ADD[TI+2*I+J]:=ADD[TI+2*I+J-1]+TI-CT+2*I+J;
    'END';
  'FOR' I:=1 'STEP' 1 'UNTIL' 2*NTIPS 'DO'
    'BEGIN' CI:=TI+I-ADD[TI+I]+ADD[TI+I-1]+1;
    'FOR' J:=CI 'STEP' 1 'UNTIL' TI+I 'DO' Q[ADD[TI+I]-I-TI+J]:=0.0;
    'END';
  'FOR' I:=1 'STEP' 2 'UNTIL' 2*NTIPS-1 'DO'
    'BEGIN' K[ADD[TI+I]]:=KT[I+1,2];
    K[ADD[TI+I+1]]:=KT[I,1];
    K[ADD[TI+I+1]-1]:=KT[I,2];
    'END';
  CT1:=0;
  'FOR' I:=1 'STEP' 1 'UNTIL' NTIPS 'DO'
    'BEGIN' Z:=2*(NS[I]-1)-CT1; CT1:=CT1+2*N1[I];
    V:=4*N1[I]+2; U:=TI+2*I; W:=2*I;
    'FOR' J:=3 'STEP' 1 'UNTIL' W 'DO'
      'BEGIN' K[ADD[U-1]-U+Z-1+J]:=KT[V,J];
      K[ADD[U]-U+Z-2+J]:=KT[V-1,J];
      'END';
    'END';
  'END' OF PROCEDURE MODF1NTIPS;

```

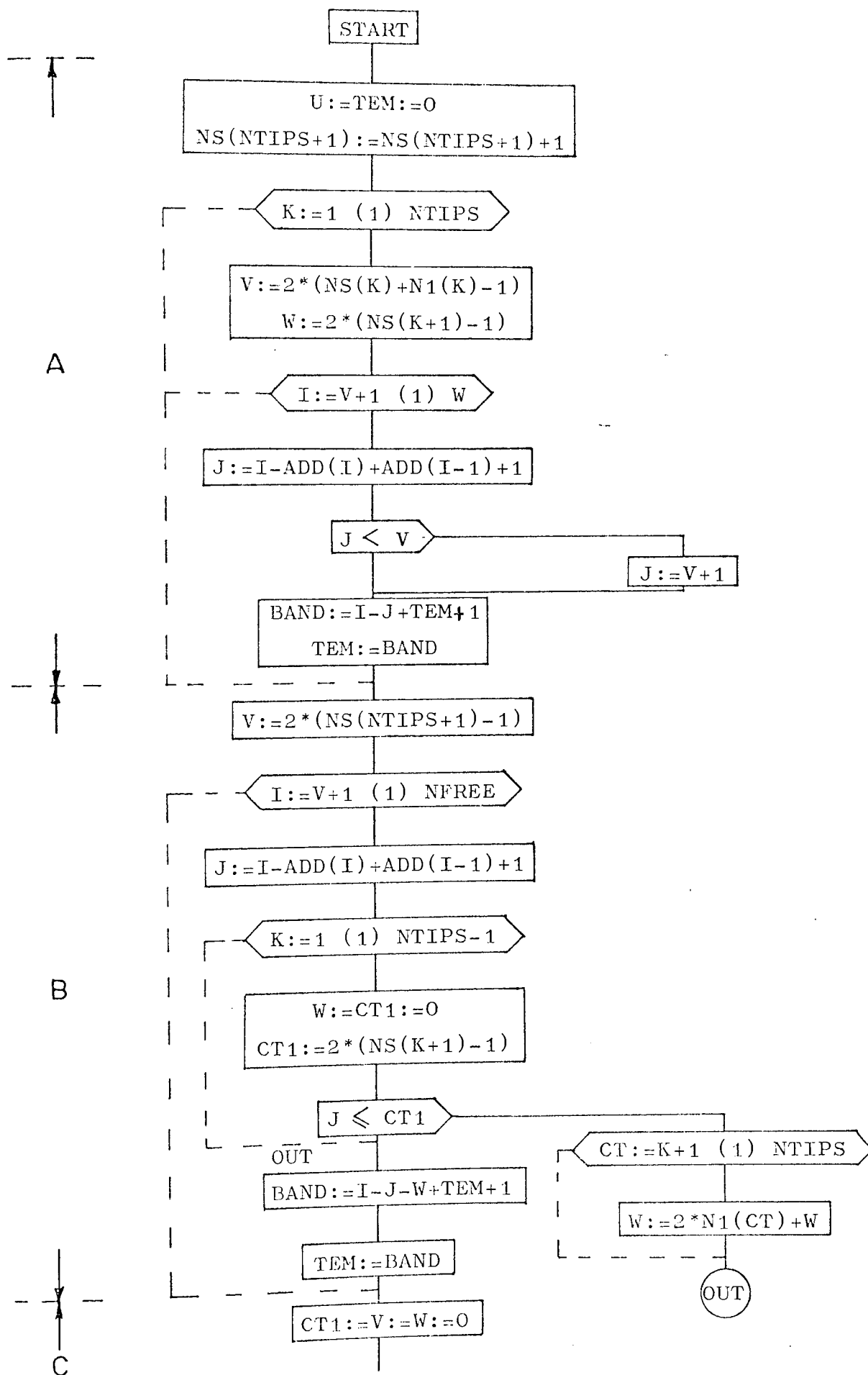
As previously discussed in the introduction, section 5.1, the storage requirements of the modified stiffness matrix may often be greater than those calculated from procedure ADDARRAY for the original stiffness matrix.

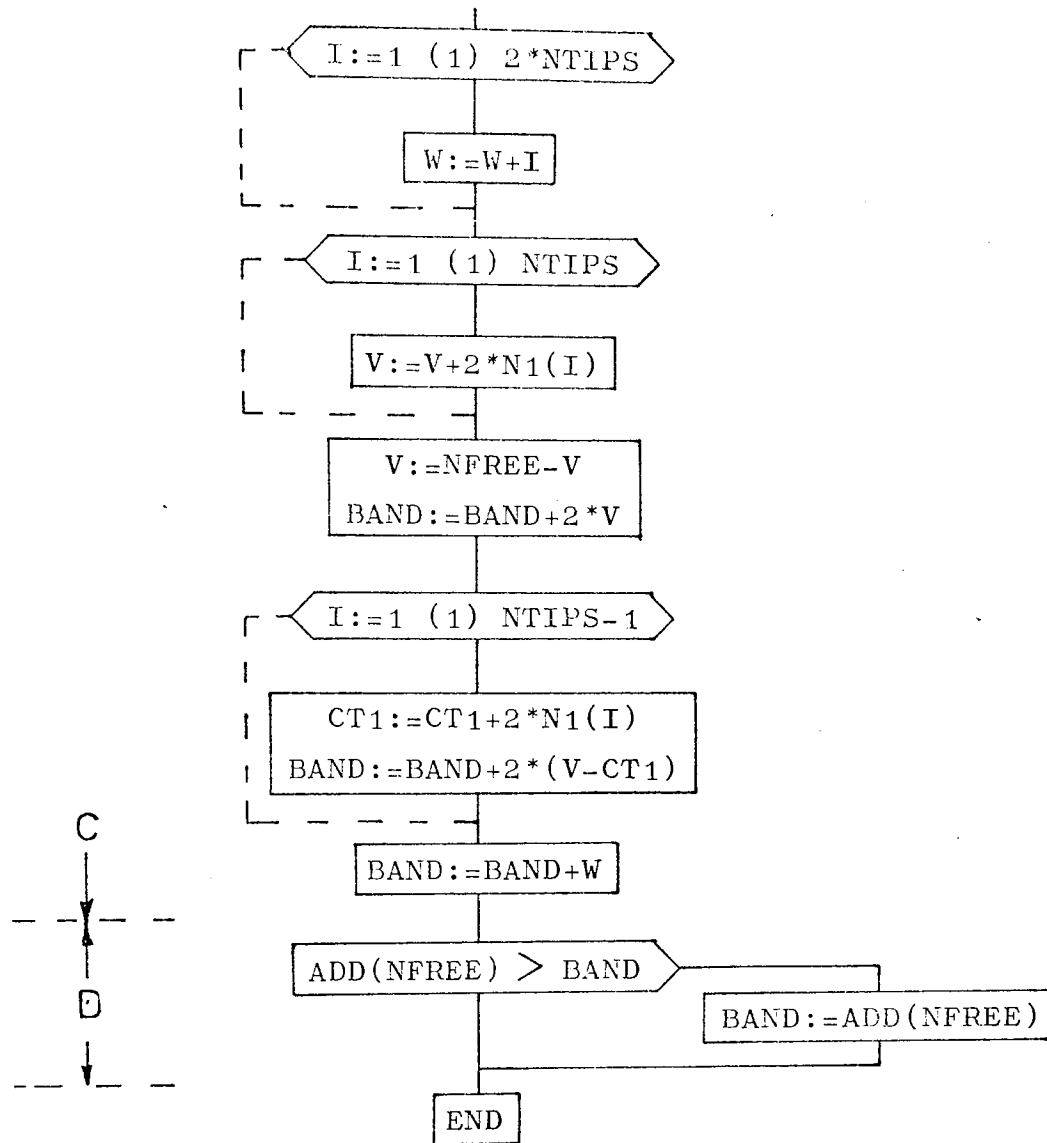
To guard against this eventuality a procedure KARRAYBOUND has been written which is called in the program immediately after procedure ADDARRAY and computes the storage requirements for the modified stiffness matrix $[K^*]$.

A flowchart and program listing follows for this procedure and the various steps are, with reference to the flowchart :

- A. - Using coefficients of ADDARRAY, the number of coefficients present in $[K^*]$ after the elimination of rows corresponding to core nodes are calculated as integer BAND.
- B. - For the remainder of the rows of $[K]$ i.e. rows $2(NS[4]-1)$ to NG, Fig. 5.15, number of coefficients present in $[K^*]$ are calculated.
- C. - The number of coefficients in array KT, i.e. in rows $NG-2N1[1]-2N1[2]-2N1[3]+1$ to $NG-2N1[1]-2N1[2]-2N1[3]+6$, Fig. 5.16, are calculated and this number is added to integer BAND.
- D. - Values found for BAND, i.e. size of the modified stiffness matrix $[K^*]$ is compared with size of $[K]$ as given by ADD[NFREE] and BAND is set to the greatest of these values. Bounds of the array $[K^*]$ will later be declared as $K[1 : BAND]$.

PROCEDURE : KARRAYBOUND(NFREE,N1,BAND,ADD,NTIPS,NS)






```

'PROCEDURE' KARRAYROUND(NFREE,N1,RAND,ADD,NTIPS,NS);
'VALUE' NFREE,NTIPS;
'INTEGER' NFREE,NTIPS,RAND;
'INTEGER' 'ARRAY' N1,ADD,NS;
'BEGIN' 'INTEGER' I,J,V,W,CT,CT1,U,TEM,K;
      U:=0; TEM:=0; NS[NTIPS+1]:=NS[NTIPS+1]+1;
      'FOR' K:=1 'STEP' 1 'UNTIL' NTIPS 'DO'
        'BEGIN' V:=2*(NS[K]+N1(K)-1);
              W:=2*(NS[K+1]-1);
              'FOR' J:=V+1 'STEP' 1 'UNTIL' W 'DO'
                'BEGIN' J:=I-ADD[I]+ADD[I-1]+1;
                      'IF' J<1 'THEN' J:=V+1;
                      BAND:=I-J+TEM+1;
                      TEM:=BAND;
                'END';
              'END';
      V:=2*(NS[NTIPS+1]-1);
      'FOR' I:=V+1 'STEP' 1 'UNTIL' NFREE 'DO'
        'BEGIN' J:=I-ADD[I]+ADD[I-1]+1;
              'FOR' K:=1 'STEP' 1 'UNTIL' NTIPS-1 'DO'
                'BEGIN' W:=CT1:=0;
                      CT1:=2*(NS[K+1]-1);
                      'IF' J<1 'THEN'
                        'BEGIN' 'FOR' CT:=K+1 'STEP' 1 'UNTIL' NTIPS 'DO'
                          W:=2*CT1[CT1]+W;
                        'GOTO' OUT;
                      'END';
                'END';
              OUT: BAND:=I-J-W+TEM+1;
              TEM:=BAND;
        'END';
      CT1:=V:=W:=0;
      'FOR' I:=1 'STEP' 1 'UNTIL' 2*NTIPS 'DO' R:=I+W;
      'FOR' I:=1 'STEP' 1 'UNTIL' NTIPS 'DO' V:=V+2*N1[I];
      V:=NFREE-V; BAND:=BAND+2*V;
      'FOR' I:=1 'STEP' 1 'UNTIL' NTIPS-1 'DO'
        'BEGIN' CT1:=CT1+2*N1[I+1];BAND:=BAND+2*(V-CT1); 'END';
      BAND:=BAND+W;
      'IF' ADD[NFREE] 'GT' BAND 'THEN' BAND:=ADD[NFREE];
      'END' OF PROCEDURE KARRAYROUND;

```

5.4 SINGLE TIP, COMBINED MODES I AND II, CASE

The first example of a fracture problem involving both Mode I and Mode II singularities is the single edge crack of the type shown in Fig. 5.17. In contrast to the Mode I case there is no symmetry about the crack plane and hence a full circular core must be constructed about the crack tip. The near tip displacement field equations of Appendix 10.A, are specified with respect to the crack axes (x,y) , these axes being inclined to the global axes by an angle α , Fig. 5.17.

Considering the rigid body modes of the core element, it is immediately apparent that the core may displace in the x and y directions, hence the two rigid body displacement components δ_x and δ_y must be included in the near tip displacement fields. It is not so apparent, however, that the core element may also rotate as a rigid body about the crack tip and that a suitable rigid body rotation term must be included to allow for this displacement. Referring to Fig. 5.18, where such a rotation, ω^c produces displacements :

$$u_r = 0 \quad \text{and} \quad u_\theta = \omega r \quad (5.5a)$$

in polar coordinates,

$$\begin{aligned} \text{and} \quad u_x &= -\omega r \sin \theta = -\omega y \\ u_y &= \omega r \cos \theta = \omega x \end{aligned} \quad (5.5b)$$

in Cartesian coordinates.

The near tip displacement field may then be written as :

$$\begin{aligned} u_x &= f_1(R_c, \theta) K_I + g_1(R_c, \theta) K_{II} + \delta_x + h_1(R_c, \theta) \omega \\ u_y &= f_2(R_c, \theta) K_I + g_2(R_c, \theta) K_{II} + \delta_y + h_2(R_c, \theta) \omega \end{aligned} \quad (5.6)$$

where

$$\begin{aligned} f_1(R_c, \theta) &= \frac{1}{4G} \sqrt{\frac{R_c}{2}} \left[(2K-1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \right] \\ f_2(R_c, \theta) &= \frac{1}{4G} \sqrt{\frac{R_c}{2}} \left[(2K+1) \sin \frac{\theta}{2} - \sin \frac{3\theta}{2} \right] \end{aligned}$$

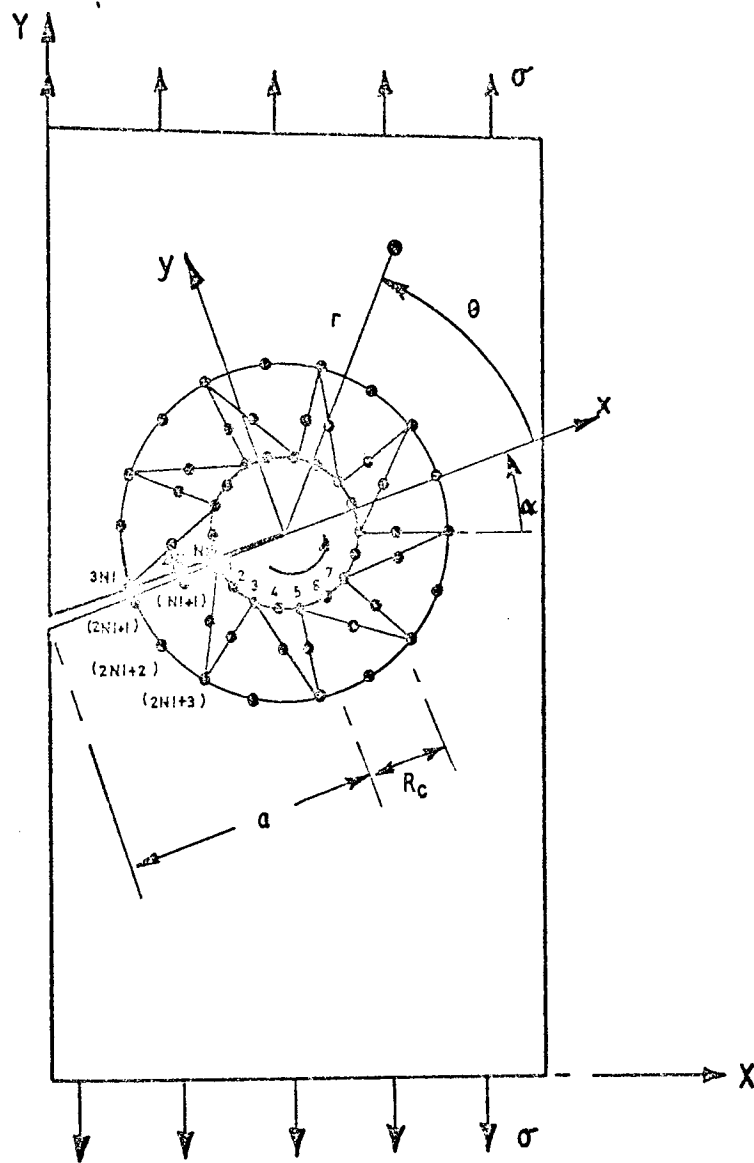


FIG.5.17. INCLINED EDGE CRACK SHOWING FULL CIRCULAR CORE.

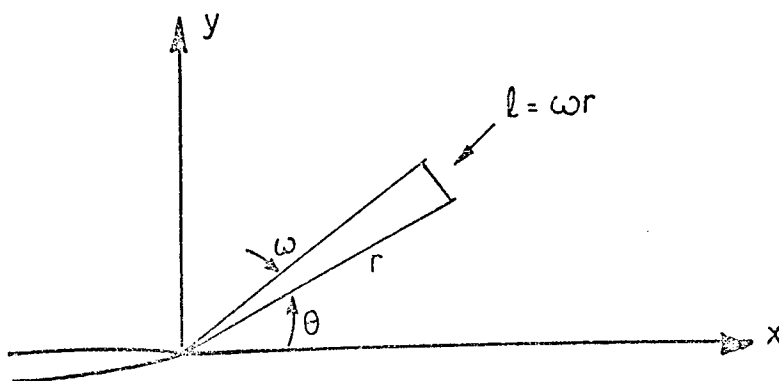


FIG.5.18. RIGID BODY ROTATION OF CORE.

$$g_1(R_c, \theta) = \frac{1}{4G} \sqrt{\frac{R_c}{2}} \left[(2K+3) \sin \frac{\theta}{2} + \sin \frac{3\theta}{2} \right]$$

$$g_2(R_c, \theta) = -\frac{1}{4G} \sqrt{\frac{R_c}{2}} \left[(2K-3) \cos \frac{\theta}{2} + \cos \frac{3\theta}{2} \right]$$

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

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

Derivatives of these displacement components are independent of δ_x , δ_y and ω and hence they can have no effect on strain, stress or strain energy stored in the core. The potential energy is minimised with respect to the five core generalised coordinates, δ_x , δ_y , ω , K_I and K_{II} and the unconstrained nodal displacement components in the remainder of the finite element mesh.

The displacement components of equation 5.6 must be transformed from the local crack tip coordinate system (x,y) to the global X,Y coordinate system, using the equations :

$$\begin{aligned} u_x &= u_x \cos \alpha - u_y \sin \alpha \\ u_y &= u_x \sin \alpha + u_y \cos \alpha \end{aligned} \quad (5.7)$$

With these results, equation 5.6 becomes :

$$u_x = F_1(R_c, \theta) K_I + G_1(R_c, \theta) K_{II} + \delta_x \cos \alpha - \delta_y \sin \alpha + H_1(R_c, \theta) \omega \quad (5.8)$$

$$u_y = F_2(R_c, \theta) K_I + G_2(R_c, \theta) K_{II} + \delta_x \sin \alpha + \delta_y \cos \alpha + H_2(R_c, \theta) \omega$$

Where,

$$F_1(R_c, \theta) = f_1(R_c, \theta) \cos \alpha - f_2(R_c, \theta) \sin \alpha$$

$$F_2(R_c, \theta) = f_1(R_c, \theta) \sin \alpha + f_2(R_c, \theta) \cos \alpha$$

$$G_1(R_c, \theta) = g_1(R_c, \theta) \cos \alpha - g_2(R_c, \theta) \sin \alpha$$

$$G_2(R_c, \theta) = g_1(R_c, \theta) \sin \alpha + g_2(R_c, \theta) \cos \alpha$$

$$H_1(R_c, \theta) = h_1(R_c, \theta) \cos \alpha - h_2(R_c, \theta) \sin \alpha$$

$$H_2(R_c, \theta) = h_1(R_c, \theta) \sin \alpha + h_2(R_c, \theta) \cos \alpha$$

Referring to Fig. 5.17, the core nodes are numbered in an anti-clockwise direction from the lower crack surface to the upper surface as shown. This method of numbering is particularly convenient for partitioning the final stiffness matrix described by equation 4.7. Partitioning is shown in Fig. 5.19, with the bounds of the various sub-arrays as shown. Proceeding in a similar manner as in Chapter 4.2 for the single tip mode I case, the modified stiffness matrix will be referred to as $[K^*]$ and coefficients generated from equations 5.4 will be located directly in their correct positions in $[K^*]$. An array KT with rows 1 to 5 and columns 1 to $4N1 + 5$ is declared in the procedure which follows to store the generated coefficients prior to their insertion in $[K^*]$.

Following the procedure described in Chapter 4.2 :

$$\{\delta_1\} = [A] \{\alpha\} \quad (5.9)$$

where

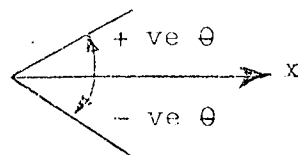
$$\{\delta_1\} = \begin{Bmatrix} \begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix} \\ \begin{Bmatrix} u_2 \\ v_2 \end{Bmatrix} \\ \vdots \\ \begin{Bmatrix} u_{2N1} \\ v_{2N1} \end{Bmatrix} \end{Bmatrix} \quad \{\alpha\} = \begin{Bmatrix} \omega \\ \delta_y \\ \delta_x \\ K_{II} \\ K_I \end{Bmatrix}$$

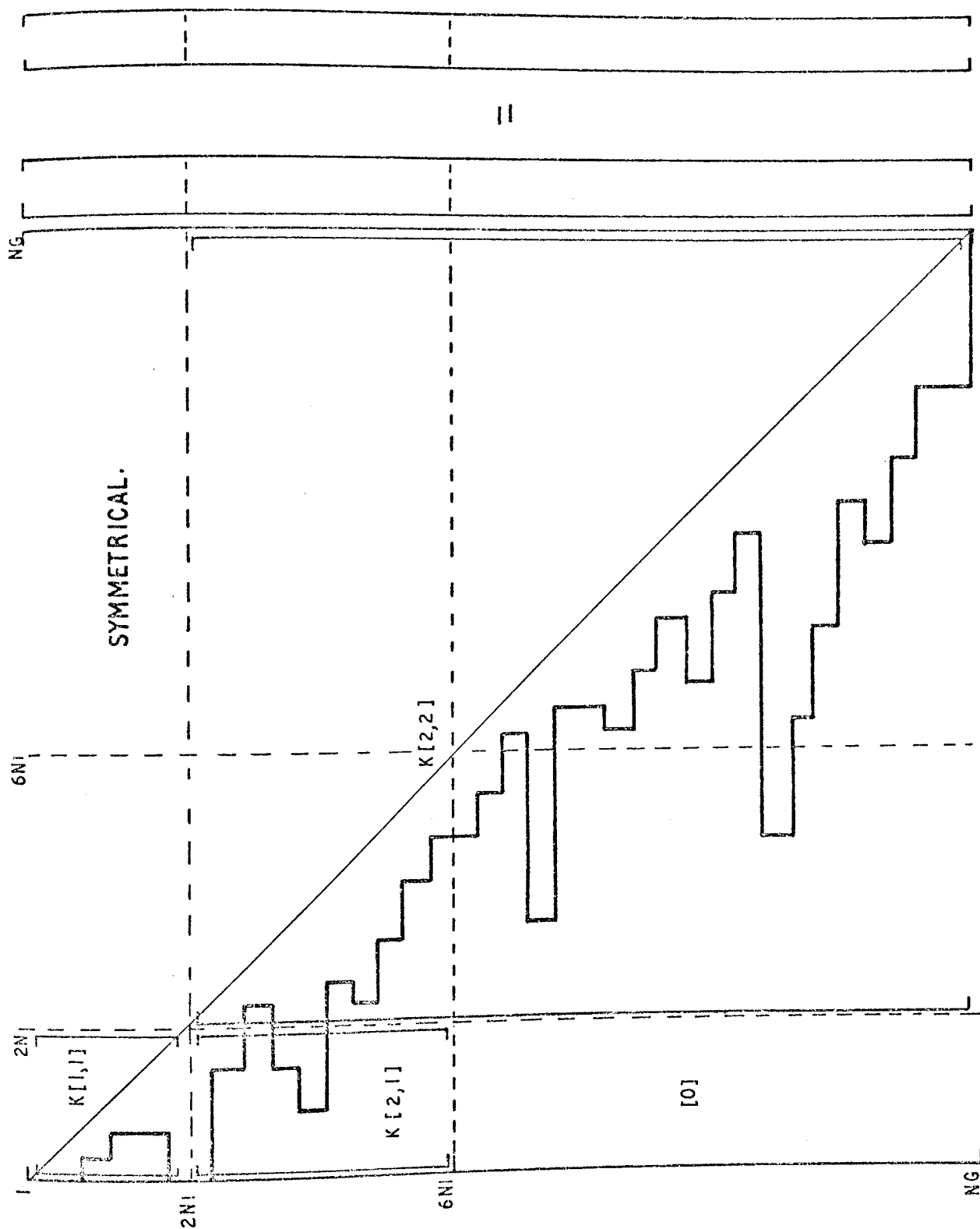
and

$$[A] = \begin{bmatrix} \begin{Bmatrix} H_1(R_c, \theta_1) \\ H_2(R_c, \theta_1) \end{Bmatrix} & \begin{Bmatrix} -\sin \alpha \\ \cos \alpha \end{Bmatrix} & \begin{Bmatrix} \cos \alpha \\ \sin \alpha \end{Bmatrix} & \begin{Bmatrix} G_1(R_c, \theta_1) \\ G_2(R_c, \theta_1) \end{Bmatrix} & \begin{Bmatrix} F_1(R_c, \theta_1) \\ F_2(R_c, \theta_1) \end{Bmatrix} \\ \begin{Bmatrix} H_1(R_c, \theta_2) \\ H_2(R_c, \theta_2) \end{Bmatrix} & \begin{Bmatrix} -\sin \alpha \\ \cos \alpha \end{Bmatrix} & \begin{Bmatrix} \cos \alpha \\ \sin \alpha \end{Bmatrix} & \begin{Bmatrix} G_1(R_c, \theta_2) \\ G_2(R_c, \theta_2) \end{Bmatrix} & \begin{Bmatrix} F_1(R_c, \theta_2) \\ F_2(R_c, \theta_2) \end{Bmatrix} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \begin{Bmatrix} H_1(R_c, \theta_{2N1}) \\ H_2(R_c, \theta_{2N1}) \end{Bmatrix} & \begin{Bmatrix} -\sin \alpha \\ \cos \alpha \end{Bmatrix} & \begin{Bmatrix} \cos \alpha \\ \sin \alpha \end{Bmatrix} & \begin{Bmatrix} G_1(R_c, \theta_{2N1}) \\ G_2(R_c, \theta_{2N1}) \end{Bmatrix} & \begin{Bmatrix} F_1(R_c, \theta_{2N1}) \\ F_2(R_c, \theta_{2N1}) \end{Bmatrix} \end{bmatrix}$$

where θ_i refers to the angle between the x-axis and node number i.

i.e. $\theta_1 = -\pi$, $\theta_{2N1} = \pi$, etc.





PARTITIONING THE ORIGINAL SYSTEM EQUATIONS.

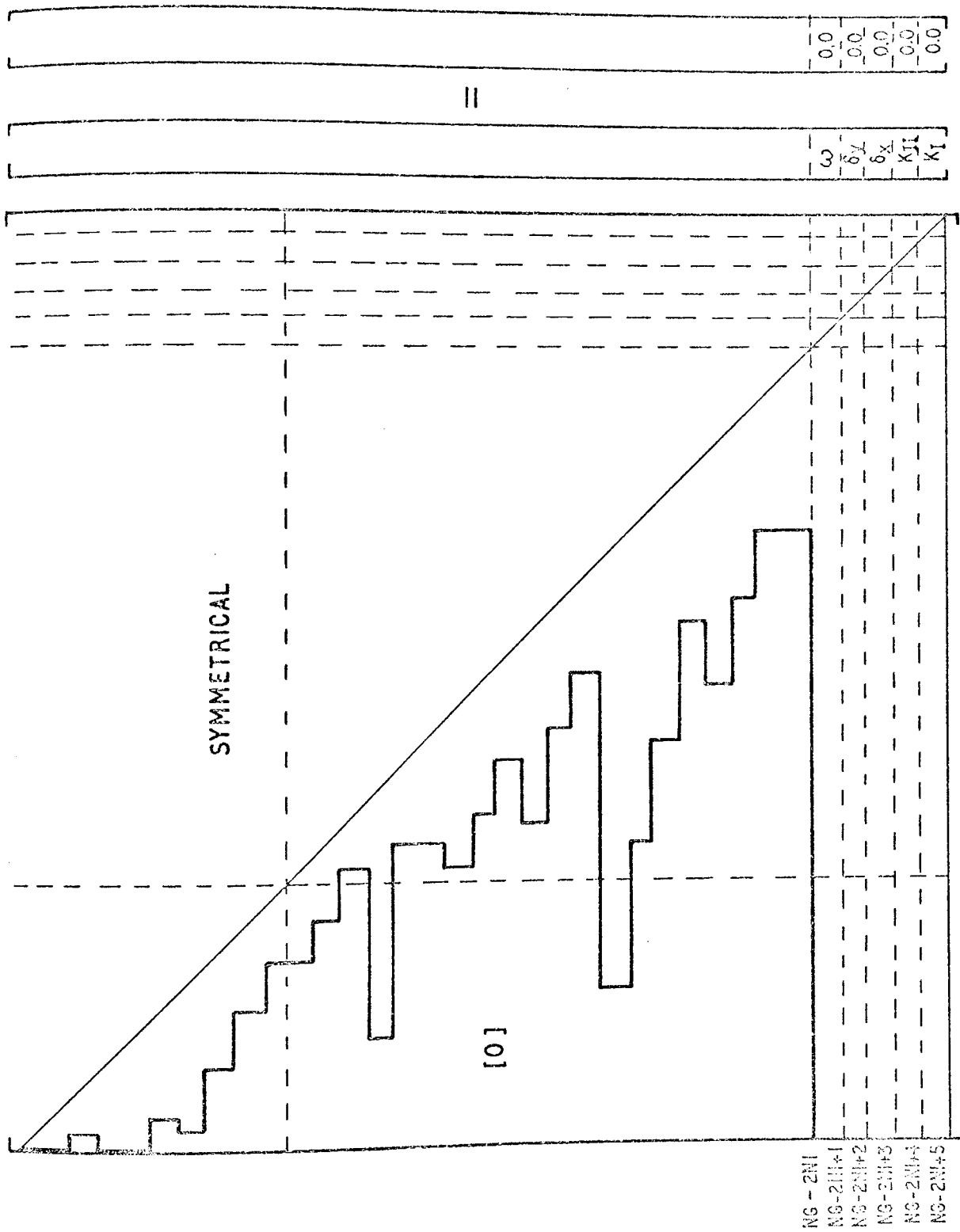


FIG. 5.20 FORMATION OF MODIFIED SYSTEM EQUATIONS.

From Appendix 10.B, the core 'stiffness matrix' may be formed as :

$$[k_c] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{R_c \Lambda}{\epsilon G} (3+2K) & 0 \\ 0 & 0 & 0 & 0 & \frac{R_c \Lambda}{\epsilon G} (2K-1) \end{bmatrix} \quad (5.10)$$

Note, the matrix shown in equation 5.10 is independent of the number of nodes on the core/finite element interface, Γ , and depends only on the number of terms retained in the series expansions, equation 5.6.

Interpreting the matrix products of equations 5.4 in subscript notation, the appropriate coefficients of $[K^*]$ may be generated as follows, with $TI = NG - 2N1$, Fig. 5.20.

$$KT[5,5] = K^*[TI+1, TI+1] = k_c[1,1] + \sum_{i=1}^{2N1} \sum_{j=1}^{2N1} K[i,j] \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_i \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_j$$

$$KT[4,5] = K^*[TI+2, TI+1] = k_c[2,1] + \sum^* \sum^* K[i,j] \begin{vmatrix} -\sin \alpha \\ \cos \alpha \end{vmatrix}_i \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_j$$

$$KT[4,4] = K^*[TI+2, TI+2] = k_c[2,2] + \sum^* \sum^* K[i,j] \begin{vmatrix} -\sin \alpha \\ \cos \alpha \end{vmatrix}_i \begin{vmatrix} -\sin \alpha \\ \cos \alpha \end{vmatrix}_j$$

$$KT[3,5] = K^*[TI+3, TI+1] = k_c[3,1] + \sum^* \sum^* K[i,j] \begin{vmatrix} \cos \alpha \\ \sin \alpha \end{vmatrix}_i \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_j$$

$$KT[3,4] = K^*[TI+3, TI+2] = k_c[3,2] + \sum^* \sum^* K[i,j] \begin{vmatrix} \cos \alpha \\ \sin \alpha \end{vmatrix}_i \begin{vmatrix} -\sin \alpha \\ \cos \alpha \end{vmatrix}_j$$

$$KT[3,3] = K^*[TI+3, TI+3] = k_c[3,3] + \sum^* \sum^* K[i,j] \begin{vmatrix} \cos \alpha \\ \sin \alpha \end{vmatrix}_i \begin{vmatrix} \cos \alpha \\ \sin \alpha \end{vmatrix}_j$$

$$KT[2,5] = K^*[TI+4, TI+1] = k_c[4,1] + \sum^* \sum^* K[i,j] \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_i \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_j$$

$$KT[2,4] = K^*[TI+4, TI+2] = k_c[4,2] + \sum^* \sum^* K[i,j] \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_i \begin{vmatrix} -\sin \alpha \\ \cos \alpha \end{vmatrix}_j$$

$$KT[2,3] = K^*[TI+4, TI+3] = k_c[4,3] + \sum^* \sum^* K[i,j] \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_i \begin{vmatrix} \cos \alpha \\ \sin \alpha \end{vmatrix}_j$$

$$KT[2,2] = K^*[TI+4, TI+4] = k_c[4,4] + \sum^* \sum^* K[i,j] \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_i \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_j$$

(* Sums are carried out from 1 to $2N1$ in each case.)

$$\begin{aligned}
KT[1,5] &= K^*[TI+5, TI+1] = k_c[5,1] + \sum^* \sum^* K[i,j] \begin{matrix} F_1 \\ F_2 \end{matrix} \Big|_i \begin{matrix} H_1 \\ H_2 \end{matrix} \Big|_j \\
KT[1,4] &= K^*[TI+5, TI+2] = k_c[5,2] + \sum^* \sum^* K[i,j] \begin{matrix} F_1 \\ F_2 \end{matrix} \Big|_i \begin{matrix} -\sin\alpha \\ \cos\alpha \end{matrix} \Big|_j \\
KT[1,3] &= K^*[TI+5, TI+3] = k_c[5,3] + \sum^* \sum^* K[i,j] \begin{matrix} F_1 \\ F_2 \end{matrix} \Big|_i \begin{matrix} \cos\alpha \\ \sin\alpha \end{matrix} \Big|_j \\
KT[1,2] &= K^*[TI+5, TI+4] = k_c[5,4] + \sum^* \sum^* K[i,j] \begin{matrix} F_1 \\ F_2 \end{matrix} \Big|_i \begin{matrix} G_1 \\ G_2 \end{matrix} \Big|_j \\
KT[1,1] &= K^*[TI+5, TI+5] = k_c[5,5] + \sum^* \sum^* K[i,j] \begin{matrix} F_1 \\ F_2 \end{matrix} \Big|_i \begin{matrix} F_1 \\ F_2 \end{matrix} \Big|_j
\end{aligned} \tag{5.11}$$

$$\begin{aligned}
KT[5, j+5] &= K^*[TI+1, j] = \sum^* K[j+2N1, i] \begin{matrix} H_1 \\ H_2 \end{matrix} \Big|_i \quad j = 1, 2, 3, \dots, 4N1 \\
KT[4, j+5] &= K^*[TI+2, j] = \sum^* K[j+2N1, i] \begin{matrix} -\sin\alpha \\ \cos\alpha \end{matrix} \Big|_i \quad j = 1, 2, 3, \dots, 4N1 \\
KT[3, j+5] &= K^*[TI+3, j] = \sum^* K[j+2N1, i] \begin{matrix} \cos\alpha \\ \sin\alpha \end{matrix} \Big|_i \quad j = 1, 2, 3, \dots, 4N1 \\
KT[2, j+5] &= K^*[TI+4, j] = \sum^* K[j+2N1, i] \begin{matrix} G_1 \\ G_2 \end{matrix} \Big|_i \quad j = 1, 2, 3, \dots, 4N1 \\
KT[1, j+5] &= K^*[TI+5, j] = \sum^* K[j+2N1, i] \begin{matrix} F_1 \\ F_2 \end{matrix} \Big|_i \quad j = 1, 2, 3, \dots, 4N1
\end{aligned} \tag{5.12}$$

where $\begin{matrix} F_1 \\ F_2 \end{matrix} \Big|_i$ etc. is understood to mean that F_1 is evaluated when i is even and when i is odd F_2 is evaluated.

Nodes on the core interface are considered free from external loads, so that :

$$\left\{ F_2^* \right\} = \left\{ 0 \right\} \tag{5.13}$$

Finally,

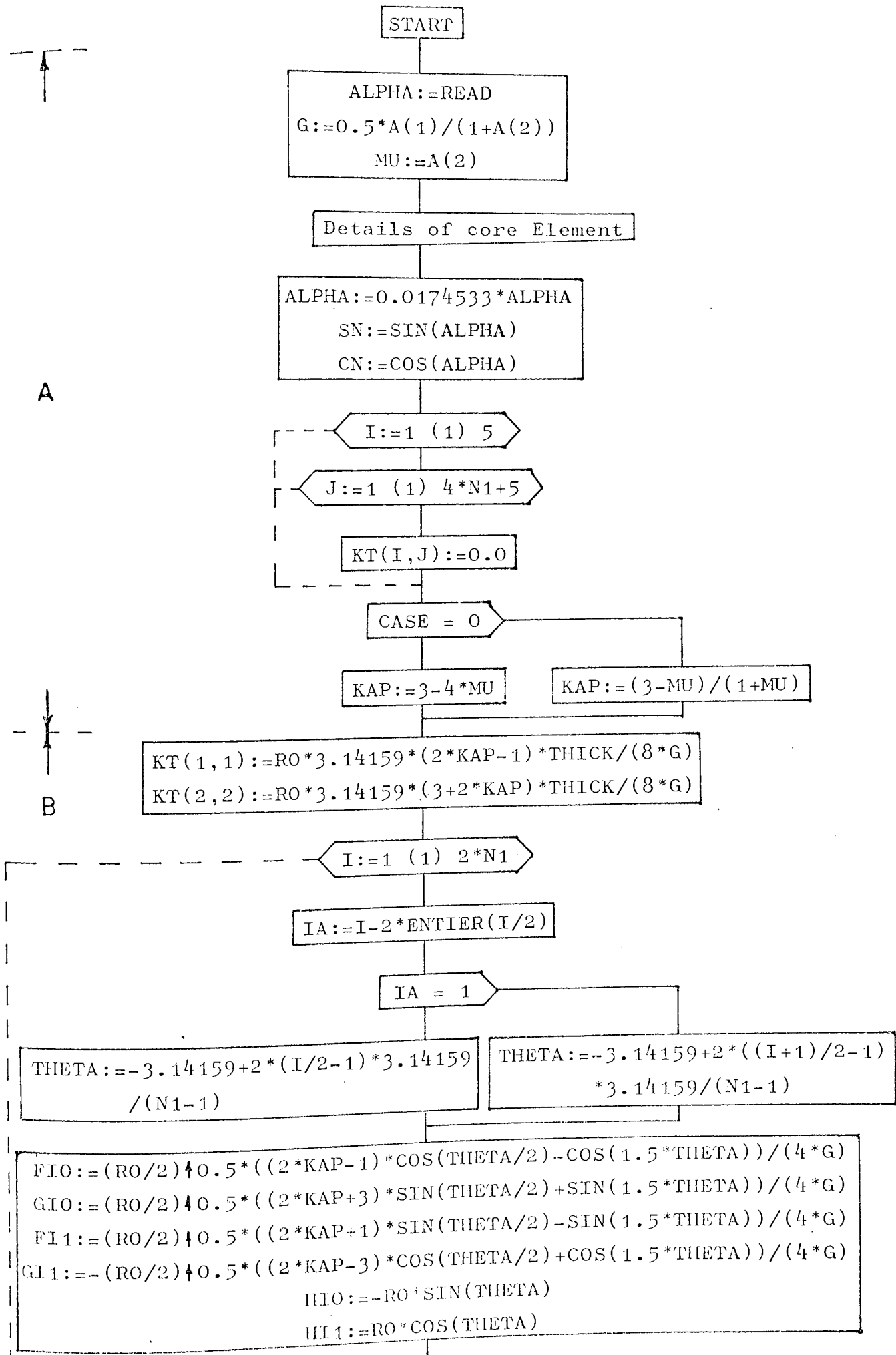
$$K[i, j]^* = K[i + 2N1, j + 2N1] \tag{5.14a}$$

$$F[i]^* = F[i + 2N1] \tag{5.14b}$$

A flowchart and program listing of the procedure written to carry out these operations follows. The steps involved in forming the various summations are identical to those described in section 5.2 and will not be repeated here. The steps of the procedure are summarised, with reference to the flowchart as :

- A. - Tip details are read and printed, elastic constants G , μ and K are calculated and the array KT is initialised.
- B. - Coefficients $KT[1,1]$ to $KT[5,5]$ of the array $[KT]$ are calculated from equations 5.11.
- C. - Coefficients $KT[5,j+5]$ to $KT[1,j+5]$ of the array $[KT]$ are calculated from the equations 5.12.
- D. - Stiffness coefficients are replaced according to equation 5.14a, with appropriate changes in the address sequence.
- E. - Coefficients of the address sequence for rows $NG-2N1+1$ to $NG-2N1+5$ of $[K^*]$ are calculated.
- F. - Load vector moved according to equation 5.14b.
- G. - Coefficients of $\{F_2^*\}$, equation 5.13, are initialised.
- H. - Coefficients of $[K^*]$ are initialised for the rows of step E.
- I. - Coefficients of array KT are positioned in $[K^*]$.

PROCEDURE : MODELIST(N1,RO,NG,NSETF,A,ADD,K,Q,CASE,THICK)



IA = 1

FIO:=FIO*SN+FI1*CN
GIO:=GIO*SN+GI1*CN
HIO:=HIO*SN+HI1*CN
T2:=SN T4:=CN

FIO:=FIO*CN-FI1*SN
GIO:=GIO*CN-GI1*SN
HIO:=HIO*CN-HI1*SN
T2:=CN T4:=-SN

I = 1

CI:=I-ADD(I)+ADD(I-1)+1

CI:=1

J:=CI (1) 2*N1

J ≤ I

J = 1

DI:=I
DJ:=J

MSM1

CJ:=J-ADD(J)+ADD(J-1)+1

CJ:=1

CJ > I

MSM2

DI:=J

DJ:=I

MSM1

JA:=J-2*ENTIER(J/2)

JA = 1

THETA:=-3.14159+2*(J/2-1)*3.14159
/(N1-1)

THETA:=-3.14159+2*((J+1)/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+3)*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-3)*COS(THETA/2)+COS(1.5*THETA))/(4*G)
HJO:=-RO*SIN(THETA)
HJ1:=RO*COS(THETA)

JA = 1

$FJO := FJO * SN + FJ1 * CN$
 $GJO := GJO * SN + GJ1 * CN$
 $HJO := HJO * SN + HJ1 * CN$
 $T1 := CN \quad T3 := SN$

$FJO := FJO * CN - FJ1 * SN$
 $GJO := GJO * CN - GJ1 * SN$
 $HJO := HJO * CN - HJ1 * SN$
 $T1 := -SN \quad T3 := CN$

$TEMP1 := ADD(DI) - DI + DJ$
 $KT(1,1) := KT(1,1) + FIO * FJO * K(TEMP1)$
 $KT(1,2) := KT(1,2) + GIO * FJO * K(TEMP1)$
 $KT(2,2) := KT(2,2) + GIO * GJO * K(TEMP1)$
 $KT(1,4) := KT(1,4) + FIO * T1 * K(TEMP1)$
 $KT(2,4) := KT(2,4) + GIO * T1 * K(TEMP1)$
 $KT(3,4) := KT(3,4) + T2 * T1 * K(TEMP1)$
 $KT(1,3) := KT(1,3) + T3 * FIO * K(TEMP1)$
 $KT(2,3) := KT(2,3) + T3 * GIO * K(TEMP1)$
 $KT(3,3) := KT(3,3) + T3 * T2 * K(TEMP1)$
 $KT(4,4) := KT(4,4) + T4 * T1 * K(TEMP1)$
 $KT(1,5) := KT(1,5) + FJO * HJO * K(TEMP1)$
 $KT(2,5) := KT(2,5) + GJO * HJO * K(TEMP1)$
 $KT(3,5) := KT(3,5) + T3 * HJO * K(TEMP1)$
 $KT(4,5) := KT(4,5) + T1 * HJO * K(TEMP1)$
 $KT(5,5) := KT(5,5) + HJO * HJO * K(TEMP1)$

MSM2

$J := 6 \quad (1) \quad 4 * N1 + 5$

$CJ := J + 2 * N1 - 5 - ADD(J + 2 * N1 - 5) + ADD(J + 2 * N1 - 6) + 1$

$I := CJ \quad (1) \quad 2 * N1$

$IA := I - 2 * ENTIER(I/2)$

$IA = 1$

$THETA := -3.14159 + 2 * (I/2 - 1) * 3.14159$
 $/(N1 - 1)$

$THETA := -3.14159 + 2 * ((I + 1)/2 - 1)$
 $* 3.14159 / (N1 - 1)$

$FIO := (RO/2) \uparrow 0.5 * ((2 * KAP - 1) * COS(THETA/2) - COS(1.5 * THETA)) / (4 * G)$
 $GIO := (RO/2) \uparrow 0.5 * ((2 * KAP + 3) * SIN(THETA/2) + SIN(1.5 * THETA)) / (4 * G)$
 $FI1 := (RO/2) \uparrow 0.5 * ((2 * KAP + 1) * SIN(THETA/2) - SIN(1.5 * THETA)) / (4 * G)$
 $GI1 := -(RO/2) \uparrow 0.5 * ((2 * KAP - 3) * COS(THETA/2) + COS(1.5 * THETA)) / (4 * G)$
 $HIO := -RO * SIN(THETA)$
 $HJ1 := RO * COS(THETA)$

IA = 1

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

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

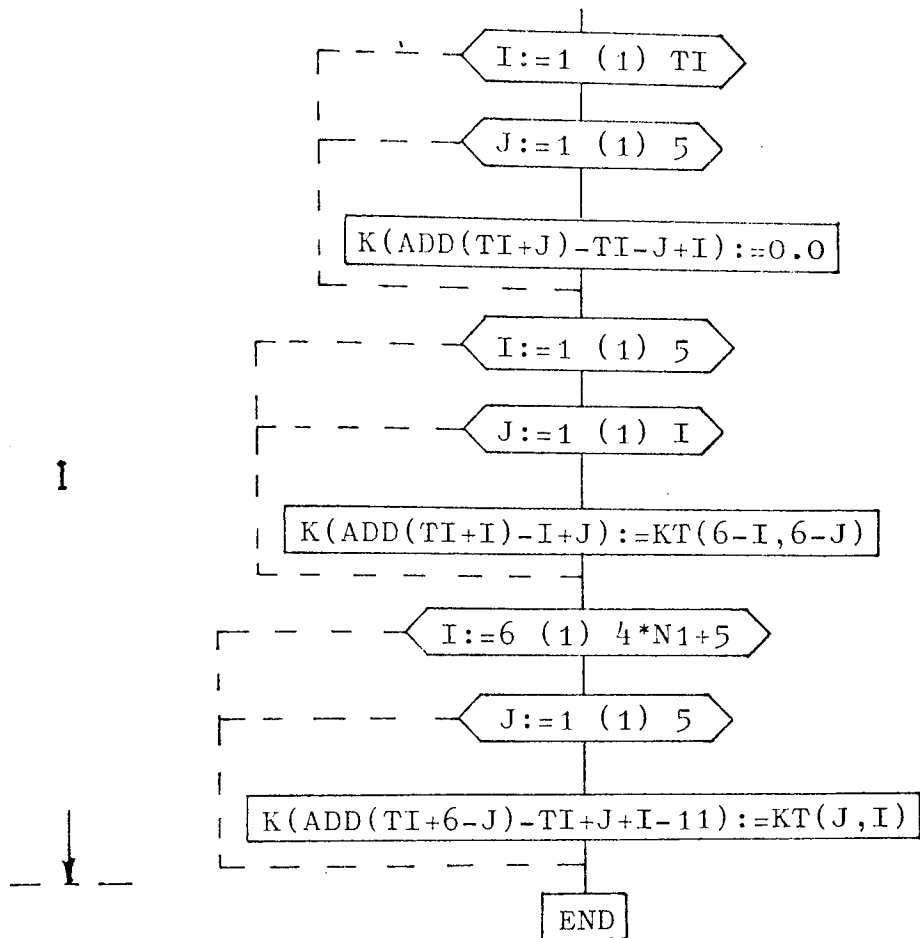
J:=1 (1) NSETF

Q(I,J):=Q(I+2*N1,J)

I:=1 (1) 5

J:=1 (1) NSETF

Q(I+TI,J):=0.0



```

PROCEDURE MODLIST(M1,RO,NG,NSETF,A,ADD,K,0,CASE,THICK);
  VALUE M1,NG,RO,NSETF,CASE,THICK: 'INTEGER' M1,NG,NSETF,CASE;
  REAL RO,THICK: 'INTEGER' 'ARRAY' ADD: 'REAL' 'ARRAY' K,0,A;
  'INTEGER' 'COMMENT' MODIFICATION OF OVERALL STIFFNESS MATRIX FOR THE
    MODE 1 AND II STRESS INTENSITY FACTORS, SINGLE TIP;
  'INTEGER' IA,JI,CJ,DI,DJ,IJ,JA;
  'REAL' FIO,GIO,HIO,GJO,THETA,KAP,ALPHA,F11,G11,HTO,T1,T2,T3,T4,T5,
    SN,CN,FJ1,GJ1,HJ1,HJ2,IO,HIO,G,MU;
  'ARRAY' KII1:5,1:4*M1+51;
  ALPHA:=ATAN2(G:=0.5*A[1]/(1+A[2]),RO:=A[2]);
  WRITETEXT('('('20')'PROCEDURE MODLISTX'GINT('0')'THIS PROCEDUREX
  MODIFIES THE OVERALL STIFFNESS MATRIX FOR MIXED MODE FRACTURE, SINGLE
  CRACK TIP PRESENT)('20')'ANGLE%OF CRACK% TIP% TO THE POSITIVE X-AXIS, DEG
  %')'); PRINT(Alpha,0,4);
  WRITETEXT('('('20')'NUMBER%OF%MODES%ON%CORE%ELEMENT%='')');
  PRINT(M1,0,0);
  WRITETEXT('('('20')'RADIUS%OF%CORE%ELEMENT%='')'); PRINT(RO,0,4);
  WRITETEXT('('('20')'MODULUS%OF%RIGIDITY,CX%='')'); PRINT(G,0,4);
  WRITETEXT('('('20')'POISSON'S RATIO,POX%='')'); PRINT(MU,0,4);
  ALPHA:=0.174533*ALPHA; SN:=SIN(ALPHA); CN:=COS(ALPHA);
  'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'
  'FOR' J:=1 'STEP' 1 'UNTIL' 4*M1+5 'DO' KII1,J1:=0.0;
  'IF' CASE=0 'THEN' KAP:=(3-MU)/(1+MU) 'ELSE' KAP:=3-4*MU;
  KII1,I1:=RO*3.14159*(2*KAP-1)*(8*G)*THICK;
  KII2,J1:=RO*3.14159*(3+2*KAP)*(3*G)*THICK;
  'FOR' I:=1 'STEP' 1 'UNTIL' 2*M1 'DO'
  'BEGIN' IA:=1-2*ENTIER(I/2);
  'IF' IA=1 'THEN' THETA:=-3.14159+2*((I+1)/2-1)*3.14159/(M1-1)
    'ELSE' THETA:=-3.14159+2*(I/2-1)*3.14159/(M1-1);
    FIO:=(RO/2)+0.5*((2*KAP-1)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
    GIO:=(RO/2)+0.5*((2*KAP+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);
    H11:=- (RO/2)+0.5*((2-KAP-3)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
    HIO:=-RO*SIN(THETA); HJO:=RO*COS(THETA);
  'IF' IA=1 'THEN' 'BEGIN' FIO:=FIO*CN-F11*SN;
    GIO:=GIO*CN-G11*SN;
    HIO:=HIO*CN-H11*SN;
    T2:=CN; T1:=-SN;
  'END' 'ELSE'
  'BEGIN' FIO:=FIO*SN+F11*CN;
    GIO:=GIO*SN+G11*CN;
    HIO:=FIO*CN+H11*CN;
    T2:=SN; T1:=CN;
  'END';
  'IF' I=1 'THEN' CJ:=1 'ELSE' CJ:=1-ADD[I]+ADD[I-1]+1;
  'FOR' J:=CJ 'STEP' 1 'UNTIL' 2*M1 'DO'
  'BEGIN' 'IF' J 'LE' I 'THEN'
    'BEGIN' DI:=I; DJ:=J;
    'GOTO' MSN1;
  'END' 'ELSE'
  'BEGIN' 'IF' J=1 'THEN' CJ:=1 'ELSE' CJ:=J-ADD[J]+ADD[J-1]+1;
    'IF' CJ 'GT' I 'THEN' 'GOTO' MSN2;
    DI:=J; DJ:=I;
  'END';
  MSN1: JA:=J-2*ENTIER(J/2);
  'IF' JA=1 'THEN' THETA:=-3.14159+2*((J+1)/2-1)*3.14159/(M1-1)
    'ELSE' THETA:=-3.14159+2*(J/2-1)*3.14159/(M1-1);
    FIO:=(RO/2)+0.5*((2*KAP-1)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
    GIO:=(RO/2)+0.5*((2*KAP+3)*SIN(THETA/2)+SIN(1.5*THETA))/(4*G);
    GJ1:=(RO/2)+0.5*((2*KAP+1)*SIN(THETA/2)+SIN(1.5*THETA))/(4*G);
    HJ1:=(RO/2)+0.5*((2-KAP-3)*COS(THETA/2)+COS(1.5*THETA))/(4*G);

```



```

      G11:=-(R0/2)+0.5*((2*XAP-3)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
      F10:=-R0*SIN(THETA);      HJ1:=R0*COS(THETA);
      IF JA=1 THEN BEGIN
        FJ0:=F10*CN-FJ1*SN;
        GJ0:=G10*CN-G11*SN;
        HJ0:=H10*CN-HJ1*SN;
        T1:=-SN;      T3:=CN;
      END ELSE
      BEGIN
        FJ0:=F10*SN+FJ1*CN;
        GJ0:=G10*SN+G11*CN;
        HJ0:=H10*SN+HJ1*CN;
        T1:=CN;      T3:=SN;
      END;

```

```

      TEMP1:=ADD(IPI)-PI+DJ;
      K1(1,1):=XT(1,1)+F10*FJ0*K(TEMP1);
      K1(1,2):=XT(1,2)+G10*FJ0*K(TEMP1);
      K1(2,1):=XT(2,2)+F10*GJ0*K(TEMP1);
      K1(1,3):=XT(1,4)+F10*T1*K(TEMP1);
      K1(2,3):=XT(2,4)+G10*T1*K(TEMP1);
      K1(3,1):=XT(3,4)+T3*T1*K(TEMP1);
      K1(1,5):=XT(1,3)+T3*F10*K(TEMP1);
      K1(2,5):=XT(2,3)+T3*G10*K(TEMP1);
      K1(3,3):=XT(3,3)+T3*T3*K(TEMP1);
      K1(4,1):=XT(4,4)+T4*T1*K(TEMP1);
      K1(1,6):=XT(1,5)+GJ0*H10*K(TEMP1);
      K1(2,6):=XT(2,5)+GJ0*H10*K(TEMP1);
      K1(3,5):=XT(3,5)+T1*H10*K(TEMP1);
      K1(4,6):=XT(4,5)+T1*H10*K(TEMP1);
      K1(5,5):=XT(5,5)+HJ0*H10*K(TEMP1);
      M2:=END;

```

```

      END;
      FOR I:=1 STEP 1 UNTIL 4*N1+5 DO
      BEGIN
        CJ:=J+2*N1-5-ADD(I+2*-1-5)+ADD(J+2*N1-6)+1;
        FOR J:=C1 STEP 1 UNTIL 2*N1 DO
        BEGIN
          IA:=J-2*NTIER(I/2);
          IF IA=1 THEN THETA:=-3.14159+2*((I+5)/2-1)*3.14159/(N1-1)
            ELSE THETA:=-3.14159+2*(I/2-1)*3.14159/(N1-1);
          F10:=(R0/2)+0.5*((2*XAP-1)*COS(THETA/2)-COS(1.5*THETA))/(4*G);
          F10:=(R0/2)+0.5*((2*XAP+1)*SIN(THETA/2)+SIN(1.5*THETA))/(4*G);
          F11:=(R0/2)+0.5*((2*XAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(4*G);
          G11:=-R0/2+0.5*((2*XAP-3)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
          F10:=-R0*SIN(THETA);      H11:=R0*COS(THETA);
          IF IA=1 THEN BEGIN
            FJ0:=F10*CN-F11*SN;
            GJ0:=G10*CN-G11*SN;
            HJ0:=H10*CN-H11*SN;
            T1:=CN;      T2:=-SN;
          END ELSE
          BEGIN
            FJ0:=F10*SN+F11*CN;
            GJ0:=G10*SN+G11*CN;
            HJ0:=H10*SN+H11*CN;
            T1:=SN;      T2:=CN;
          END;

```

```

      TEMP1:=ADD(J+2*N1-5)-J-2*N1+5+1;
      K1(1,J):=XT(1,J)+K(TEMP1)*F10;
      K1(2,J):=XT(2,J)+K(TEMP1)*G10;
      K1(3,J):=XT(3,J)+T1*K(TEMP1);
      K1(4,J):=XT(4,J)+T2*K(TEMP1);
      K1(5,J):=XT(5,J)+H10*K(TEMP1);
      END;
      END;
      T1:=N3-2*N1;

```

```

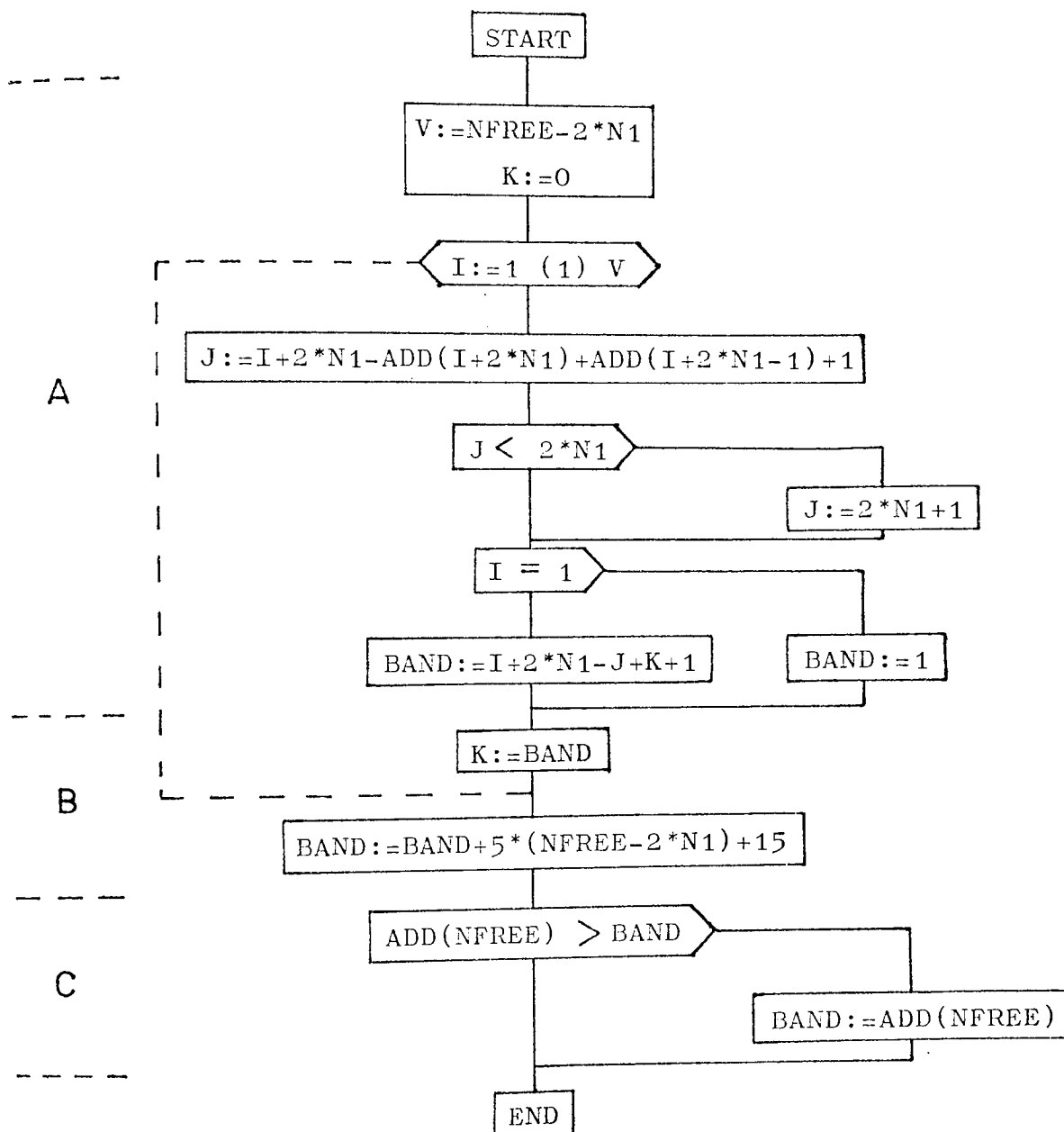
FOR I:=1 'STEP' 1 'UNTIL' TI 'DO'
  BEGIN C1:=I+2*N1-ADD[I+2*N1]+ADD[I+2*N1-1]+1;
  IF C1 'LE' 2*N1 'THEN' C1:=2*N1+1;
  ADD[I]:=I+2*N1-C1+ADD[I-1]+1;
  FOR J:=1 'STEP' 1 'UNTIL' I+2*N1 'DO'
    K[ADD[I]-I+J-2*N1]:=K[ADD[I+2*N1-1-2*N1+J]];
  END;
FOR I:=1 'STEP' 1 'UNTIL' 5 'DO' ADD[TI+1]:=ADD[TI+1-1]+TI+1;
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[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+1-1+1]:=KT[6-I,6-J];
  FOR J:=1 'STEP' 1 'UNTIL' 4*N1+5 'DO'
    FOR I:=1 'STEP' 1 'UNTIL' 5 'DO' K[ADD[TI+6-J]-TI+J+1-11]:=KT[J,I];
END OF PROCEDURE MODEL1ST;

```

In a similar manner to section 5.3 a procedure KARBDMMST, is included to calculate the bounds of $[K^*]$ and to ensure that this value is not greater than bounds originally calculated for $[K]$. A flowchart and program listing follows, the steps involved are discussed with reference to the flowchart as :

- A. - Total number of coefficients present in array $[K]$ are calculated after the elimination of the rows and columns corresponding to core nodes, i.e. rows and columns 1 x 2N1, Fig. 5.19.
- B. - Number of coefficients in rows NG-2N1+1 to NG-2N1+5 of matrix $[K^*]$, Fig. 5.20, are added to the figure found in step A.
- C. - Value found from steps A and B above (integer BAND) is compared with the bounds of $[K]$ (ADD[NFREE]). The subsequent declaration of storage will be $K[1:BAND]$.

PROCEDURE : KARBDMMST(NFREE,N1,BAND,ADD)



```

'PROCEDURE' KARBDMMST(NFREE,N1,RAND,ADD);
'VALUE' NFREE,N1;
'INTEGER' RAND,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' RAND:=1 'ELSE' RAND:=I+2*N1-J+K+1;
K:=RAND;
'END';
RAND:=RAND+5*(NFREE-2*N1)+15;
'IF' ADD[NFREE] 'GT' RAND 'THEN' RAND:=ADD[NFREE];
'END' OF PROCEDURE KARBDMMST;

```

5.5 DOUBLE TIP, COMBINED MODES I AND II CASE.

An example of the type of fracture problem analysed in this section is shown, Fig. 5.21. A plane body contains an arbitrary shaped internal crack, each tip of which may contain both Mode I and Mode II singularities.

For general loading conditions, the crack may close over part or all of its length giving zero values for stress intensity factors. As discussed in the introduction, section 5.1, special provision must be made for this phenomenon and although program changes do not affect the modification of the stiffness matrix for the cores they are included here in subsection 5.5.2. The complete section will then include all the modifications to a standard finite element program necessary for the particular case of an internal crack.

5.5.1 Modification of the Overall System Equations

Referring to Fig. 5.2.1, full circular cores must be constructed around each tip as in Fig. 5.22. Displacement fields for each core contain the stress intensity factors K_I and K_{II} together with the rigid body displacements ω , δ_x and δ_y as in the case of the Mixed Mode Single Tip Problem of section 5.4. The unknowns of the problem are therefore, K_{II} , K_{III} , δ_{xi} , δ_{yi} and ω_i ($i = 1, 2$, referring to each tip) together with the undetermined nodal displacement components in the remaining finite element mesh.

Details of core dimensions and nodal numbering around the tips is shown in Fig. 5.22. Angles α_1 and α_2 are measured in a counter-clockwise direction from the global X-axis to the local crack axes x_1 and x_2 . The nodal numbering system is particularly convenient when partitioning the overall stiffness matrix, Fig. 5.23, areas may be related to tips 1 and 2 as shown.

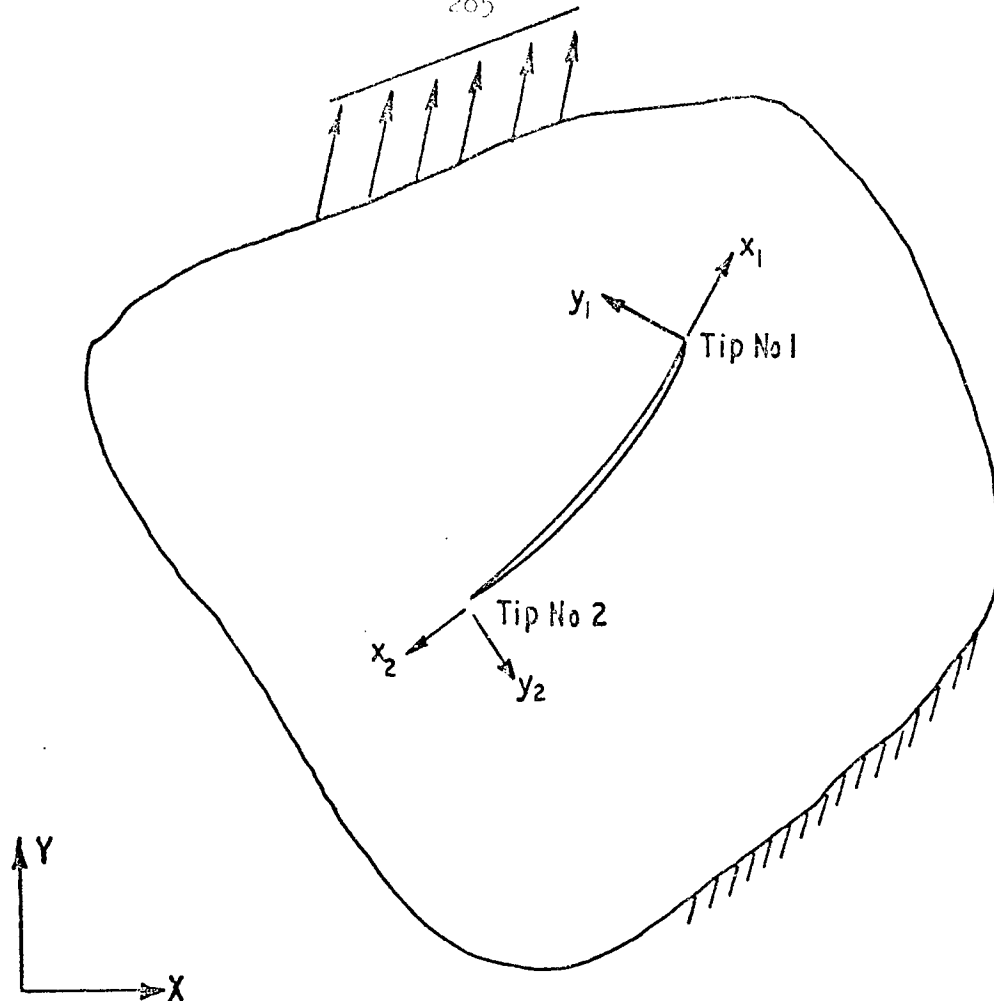


FIG.5.21. INTERNAL CRACK, MODES I AND II SINGULARITIES.

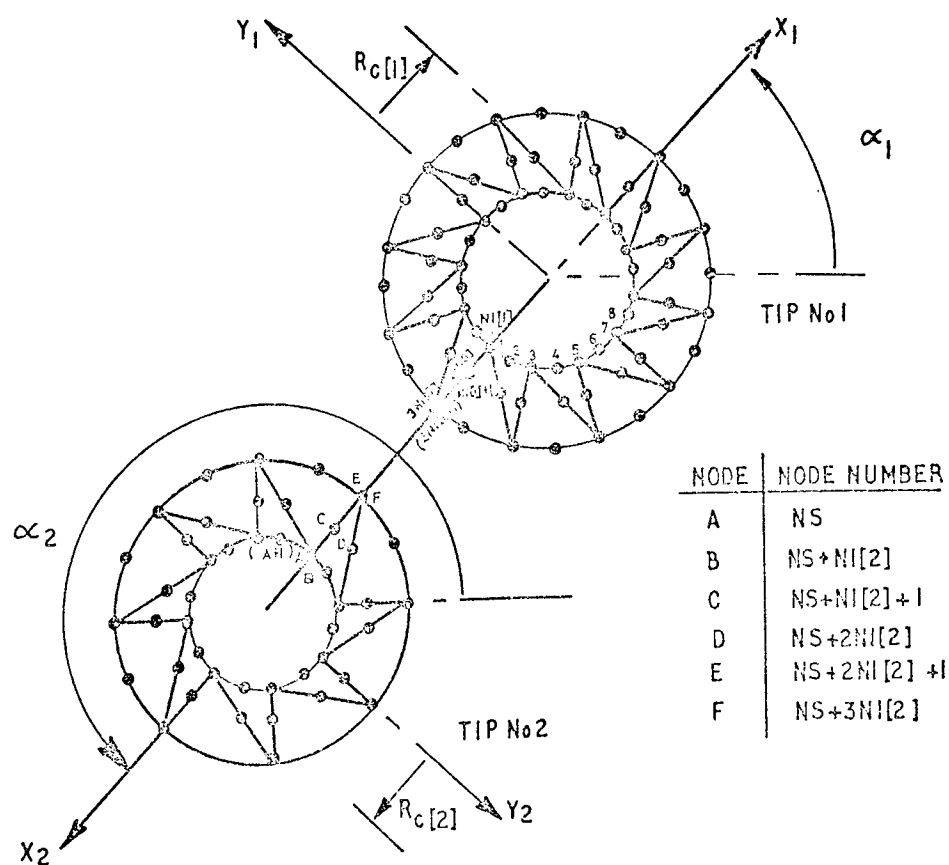


FIG.5.22. NODAL NUMBERING AROUND CORES.

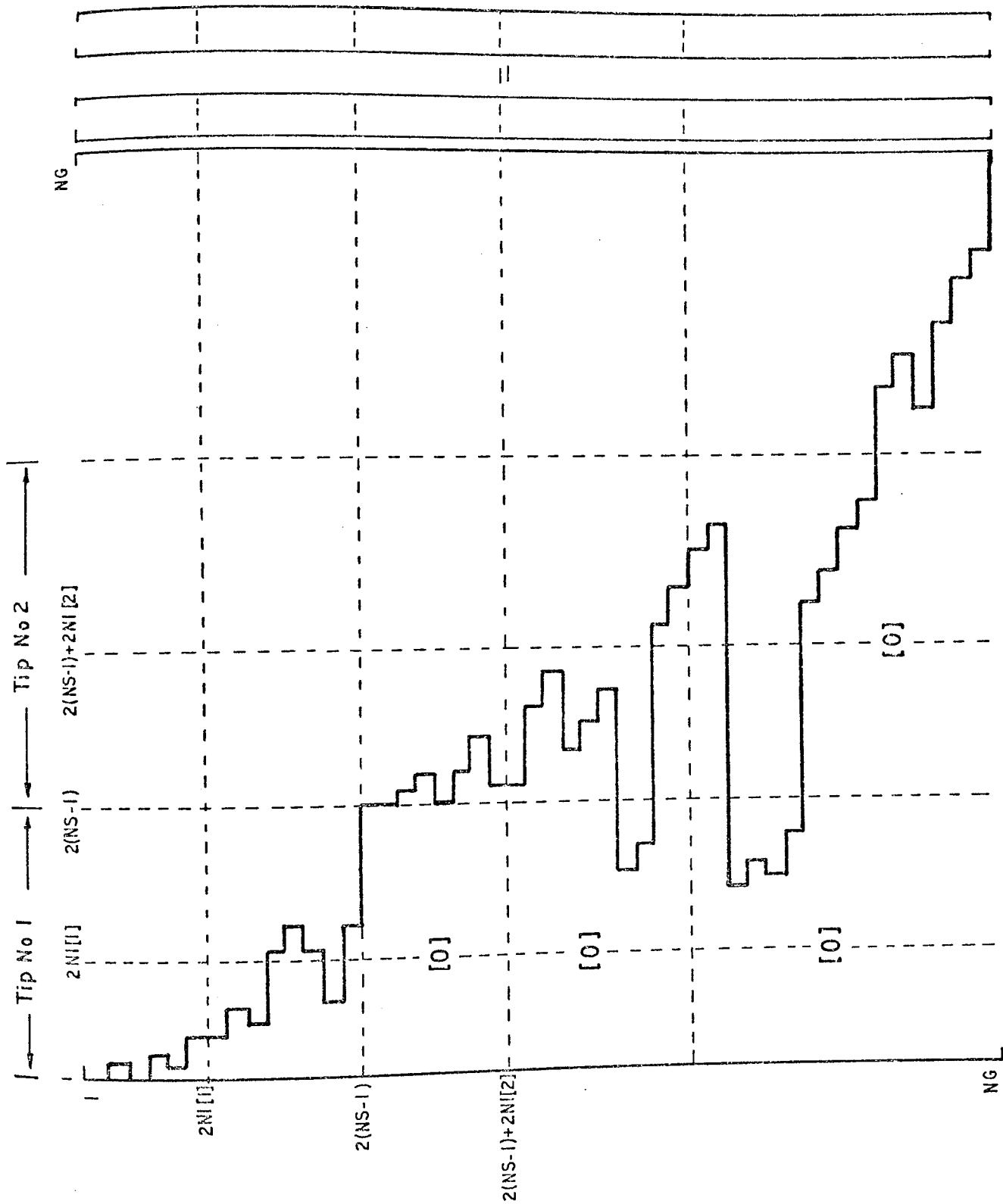


FIG 5.23 PARTITIONING THE OVERALL SYSTEM EQUATIONS .

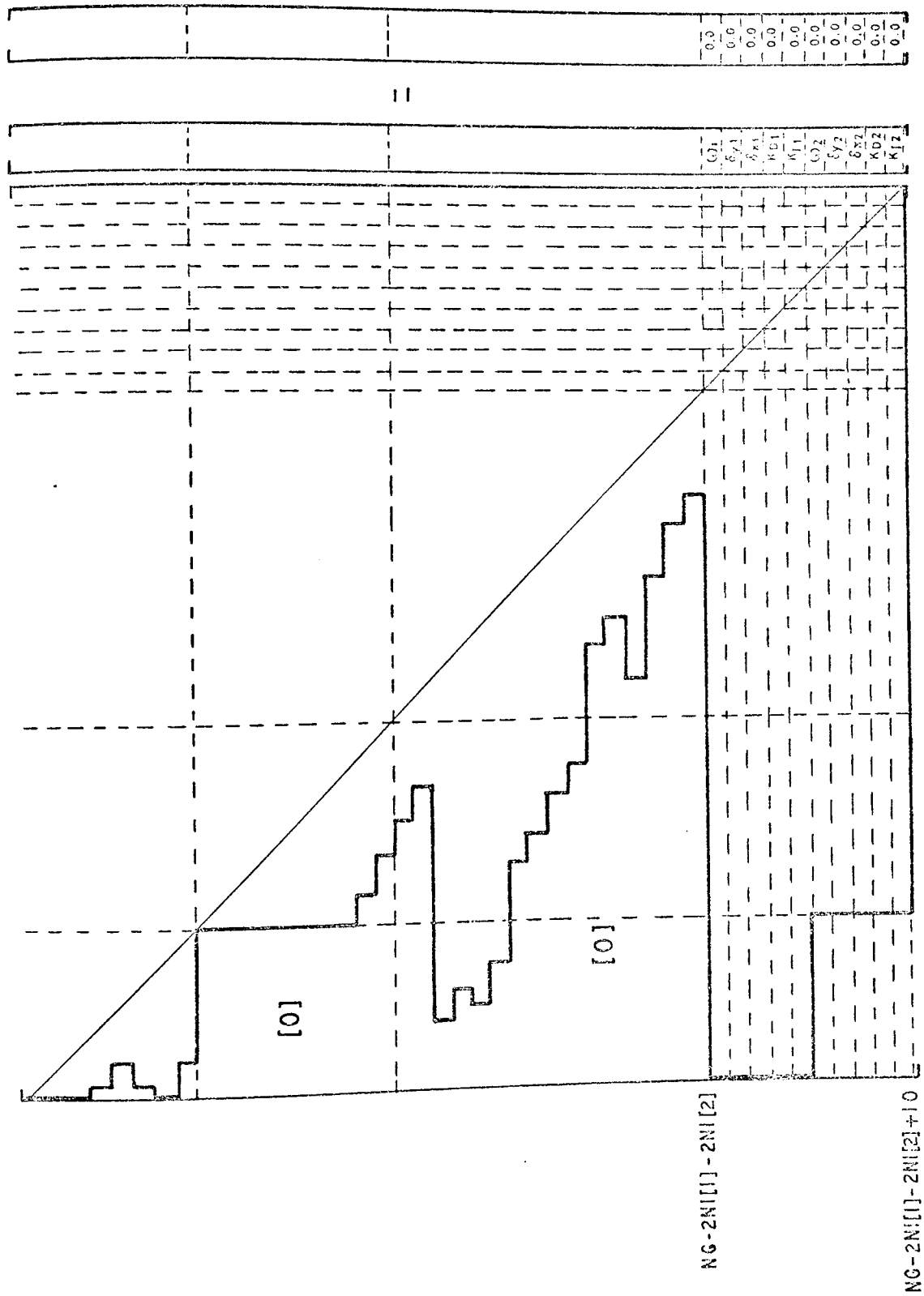


FIG.5.24 FORMATION OF THE MODIFIED SYSTEM EQUATIONS.

The method of obtaining the modified stiffness matrix $[K^*]$ consists essentially of a double application of the Mixed Mode Single Tip Procedure of section 5.4.

The basic steps can be summarised as :

- (a) An array KT is declared for temporary storage of the coefficients of $[K^*]$. These coefficients are generated by applying equations 5.11 and 5.12 to each tip in turn.
- (b) Rows and columns in $[K]$ corresponding to core nodes are eliminated.
- (c) Coefficients of KT are positioned in the modified stiffness matrix $[K^*]$.

If NS is the node number starting the second core, equations 5.11 and 5.12 can be rewritten for the two tip case as :

For $C = 1, 2$ ($C = 1$, tip 1; $C = 2$, tip 2)

When $C = 1$ then $D = F = 0$

and when $C = 2$ then $D = 2(NS-1)$ and $F = 5$.

$$\begin{aligned}
 KT[5+F, 5] &= k_c[1, 1] + \sum_{i=(D+1), (D+2)}^{2N1[C]+D} \sum_{j=(D+1), (D+2)}^{2N1[C]+D} K[i, j] \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_i \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_j \\
 KT[4+F, 5] &= k_c[2, 1] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} -\sin \alpha_c \\ \cos \alpha_c \end{vmatrix}_i \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_j \\
 KT[4+F, 4] &= k_c[2, 2] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} -\sin \alpha_c \\ \cos \alpha_c \end{vmatrix}_i \begin{vmatrix} -\sin \alpha_c \\ \cos \alpha_c \end{vmatrix}_j \\
 KT[3+F, 5] &= k_c[3, 1] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} \cos \alpha_c \\ \sin \alpha_c \end{vmatrix}_i \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_j \\
 KT[3+F, 4] &= k_c[2, 2] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} \cos \alpha_c \\ \sin \alpha_c \end{vmatrix}_i \begin{vmatrix} -\sin \alpha_c \\ \cos \alpha_c \end{vmatrix}_j \\
 KT[3+F, 3] &= k_c[3, 3] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} \cos \alpha_c \\ \sin \alpha_c \end{vmatrix}_i \begin{vmatrix} \cos \alpha_c \\ \sin \alpha_c \end{vmatrix}_j \\
 KT[2+F, 5] &= k_c[4, 1] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_i \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_j \\
 KT[2+F, 4] &= k_c[4, 2] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_i \begin{vmatrix} -\sin \alpha_c \\ \cos \alpha_c \end{vmatrix}_j
 \end{aligned} \tag{5.15}$$

(* Limits are identical in each case).

$$\begin{aligned}
KT[2+F, 3] &= k_c[4, 3] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_i \begin{vmatrix} \cos \alpha_c \\ \sin \alpha_c \end{vmatrix}_j \\
KT[2+F, 2] &= k_c[4, 4] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_i \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_j \\
KT[1+F, 5] &= k_c[5, 1] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} F_1 \\ F_2 \end{vmatrix}_i \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_j \\
KT[1+F, 4] &= k_c[5, 2] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} F_1 \\ F_2 \end{vmatrix}_i \begin{vmatrix} -\sin \alpha_c \\ \cos \alpha_c \end{vmatrix}_j \\
KT[1+F, 3] &= k_c[5, 3] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} F_1 \\ F_2 \end{vmatrix}_i \begin{vmatrix} \cos \alpha_c \\ \sin \alpha_c \end{vmatrix}_j \\
KT[1+F, 2] &= k_c[5, 4] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} F_1 \\ F_2 \end{vmatrix}_i \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_j \\
KT[1+F, 1] &= k_c[5, 5] + \sum_{i=1}^* \sum_{j=1}^* K[i, j] \begin{vmatrix} F_1 \\ F_2 \end{vmatrix}_i \begin{vmatrix} F_1 \\ F_2 \end{vmatrix}_j
\end{aligned}$$

(* Limits are identical in each case).

$$KT[5+F, j+5] = \sum_{i=(D+1)(D+2)}^{2N1[C]+D} K_{j+2N1[C]+D, i} \begin{vmatrix} H_1 \\ H_2 \end{vmatrix}_i \quad j = 1, 2, \dots, 4N1[C]$$

$$KT[4+F, j+5] = \sum_{i=1}^{**} K[j+2N1[C]+D, i] \begin{vmatrix} -\sin \alpha_c \\ \cos \alpha_c \end{vmatrix}_i \quad j = 1, 2, \dots, 4N1[C]$$

$$KT[3+F, j+5] = \sum_{i=1}^{**} K[j+2N1[C]+D, i] \begin{vmatrix} \cos \alpha_c \\ \sin \alpha_c \end{vmatrix}_i \quad j = 1, 2, \dots, 4N1[C] \quad (5.16)$$

$$KT[2+F, j+5] = \sum_{i=1}^{**} K[j+2N1[C]+D, i] \begin{vmatrix} G_1 \\ G_2 \end{vmatrix}_i \quad j = 1, 2, \dots, 4N1[C]$$

$$KT[1+F, j+5] = \sum_{i=1}^{**} K[j+2N1[C]+D, i] \begin{vmatrix} F_1 \\ F_2 \end{vmatrix}_i \quad j = 1, 2, \dots, 4N1[C]$$

where $\begin{vmatrix} F_1 \\ F_2 \end{vmatrix}_i$ is understood to mean that F_1 is evaluated when i is odd and F_2 when i is even.

F_1, F_2 , etc. are defined by equations 5.6 and 5.8.

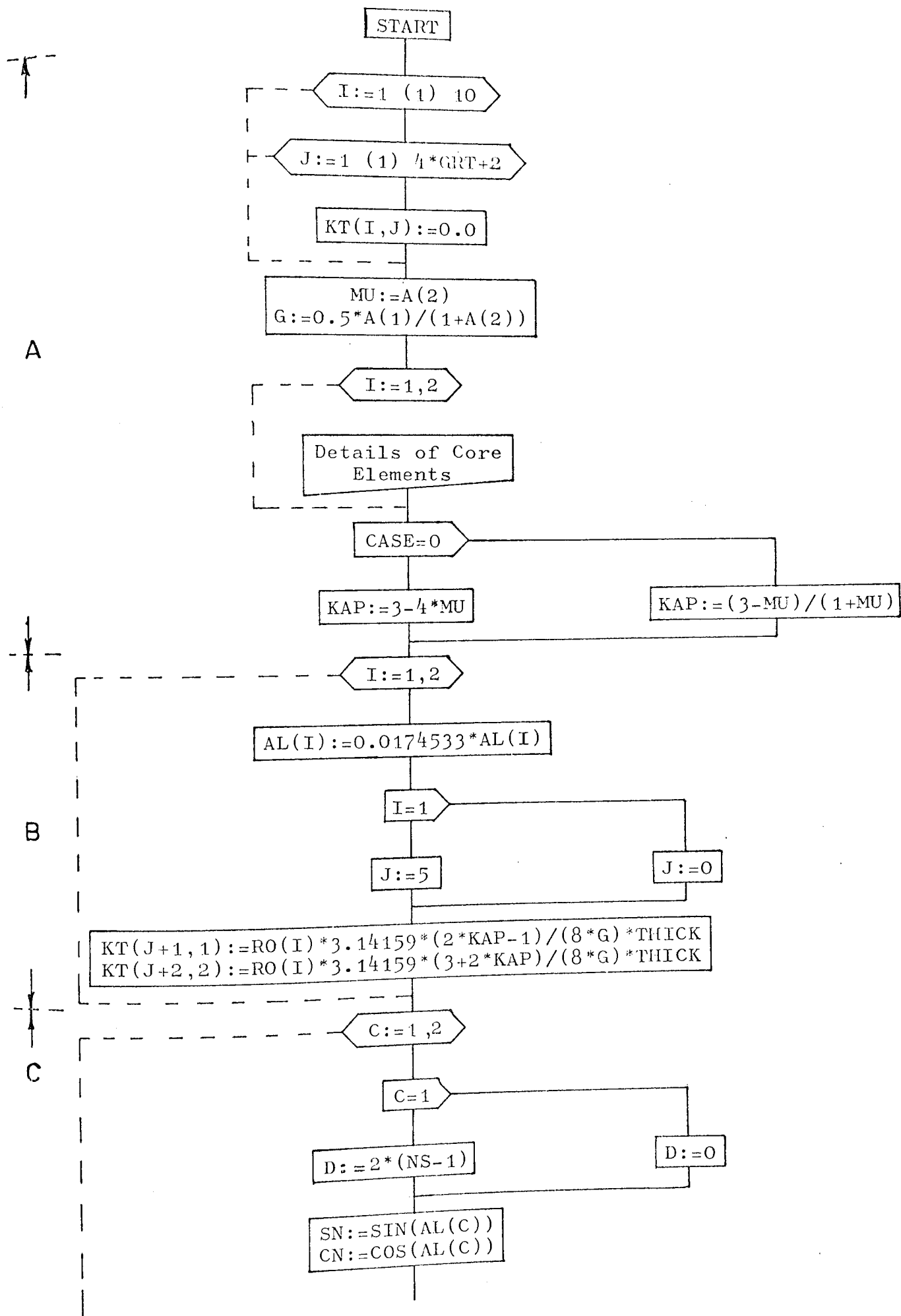
$N1[C]$, $C = 1, 2$, number of core nodes around tips 1 and 2 respectively.

(** Limits are identical in each case.)

290

A flowchart and program listing follows for obtaining the modified stiffness matrix $[K^*]$, Fig. 5.24, and the various steps are summarised with reference to the flowchart as :

- A. - Array KT is initialised and elastic constants G, MU and K are calculated.
- B. - Core stiffness coefficients are located in their correct positions in array KT.
- C. - Equations 5.15 are programmed for each tip.
- D. - Equations 5.16 are programmed for each tip.
- E. - Rows and columns pertaining to nodes surrounding the cores are eliminated, i.e. rows 1 to $2N1[1]$ and $2(NS-1)+1$ to $2(NS-1)+2N1[2]$; columns 1 to $2N1[1]$ and $2(NS-1)+1$ to $2(NS-1)+2N1[2]$, Fig. 5.23.
 - (i) rows and columns 1 to $2N1[1]$
 - (ii) rows and columns $2(NS-1)+1$ to $2(NS-1)+2N1[2]$
- F. - Integer $TI = NG - 2N1[1] - 2N1[2]$, Fig. 5.24. Address coefficients for rows $TI + 1$ to $TI + 10$ are calculated.
- G. - Modified load vector $\{F^*\}$ is formed by eliminating rows 1 to $2N1[1]$ and $2(NS-1)+1$ to $2(NS-1)+2N1[2]$ and initialising $F^*[i, j]$ from $i = TI+1$ to $TI+10$, TI as in step F.
- H. - Coefficients of $[K^*]$ for rows $TI+1$ to $TI+10$ are initialised, TI as in step F.
- I. - Coefficients of array KT are located in modified stiffness matrix $[K^*]$.



C = 1

F := 5

F := 0

I := D + 1 (1) 2 * N1(C) + D

IA := I - 2 * ENTIER(I/2)

IA = 1

THETA := -3.14159 + 2 * ((I - D) / 2 - 1) * 3.14159 / (N1(C) - 1)

THETA := -3.14159 + 2 * ((I - D + 1) / 2 - 1) * 3.14159 / (N1(C) - 1)

FIO := (RO(C) / 2) * 0.5 * ((2 * KAP - 1) * COS(THETA / 2) - COS(1.5 * THETA)) / (4 * G)
GIO := (RO(C) / 2) * 0.5 * ((2 * KAP + 3) * SIN(THETA / 2) + SIN(1.5 * THETA)) / (4 * G)
FI1 := (RO(C) / 2) * 0.5 * ((2 * KAP + 1) * SIN(THETA / 2) - SIN(1.5 * THETA)) / (4 * G)
GI1 := -(RO(C) / 2) * 0.5 * ((2 * KAP - 3) * COS(THETA / 2) + SIN(1.5 * THETA)) / (4 * G)
HIO := -RO(C) * SIN(THETA)
HI1 := RO(C) * COS(THETA)

IA = 1

FIO := FIO * SN + FI1 * CN
GIO := GIO * SN + GI1 * CN
HIO := HIO * SN + HI1 * CN
T2 := SN T4 := CN

FIO := FIO * CN - FI1 * SN
GIO := GIO * CN - GI1 * SN
HIO := HIO * CN - HI1 * SN
T2 := CN T4 := -SN

CI := I - ADD(I) + ADD(I - 1) + 1

J := CI (1) 2 * N1(C) + D

J ≤ I

CJ := J - ADD(J) + ADD(J - 1) + 1

CJ > I

DI := I

DJ := J

MSM1

MSM2

DI := J

DJ := I

MSM1

JA := J - 2 * ENTIER(J/2)

JA = 1

THETA := -3.14159 + 2 * ((J-D)/2 - 1)
* 3.14159 / (N1(C) - 1)

THETA := -3.14159 + 2 * ((J-D+1)/2 - 1)
* 3.14159 / (N1(C) - 1)

FJO := (RO(C)/2) * 0.5 * ((2*KAP-1) * COS(THETA/2) - COS(1.5*THETA)) / (4*G)
GJO := (RO(C)/2) * 0.5 * ((2*KAP+3) * SIN(THETA/2) + SIN(1.5*THETA)) / (4*G)
FJ1 := (RO(C)/2) * 0.5 * ((2*KAP+1) * SIN(THETA/2) - SIN(1.5*THETA)) / (4*G)
GJ1 := -(RO(C)/2) * 0.5 * ((2*KAP-3) * COS(THETA/2) + COS(1.5*THETA)) / (4*G)
HJO := -RO(C) * SIN(THETA)
HJ1 := RO(C) * COS(THETA)

JA = 1

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

TEM1 := ADD(DI) - DI + DJ

KT(1+F, 1) := KT(1+F, 1) + FJO * FJO * K(TEM1)
KT(1+F, 2) := KT(1+F, 2) + GJO * FJO * K(TEM1)
KT(2+F, 2) := KT(2+F, 2) + GJO * GJO * K(TEM1)
KT(1+F, 4) := KT(1+F, 4) + FJO * T4 * K(TEM1)
KT(2+F, 4) := KT(2+F, 4) + GJO * T4 * K(TEM1)
KT(3+F, 4) := KT(3+F, 4) + T3 * T4 * K(TEM1)
KT(1+F, 3) := KT(1+F, 3) + T2 * FJO * K(TEM1)
KT(2+F, 3) := KT(2+F, 3) + T2 * GJO * K(TEM1)
KT(3+F, 3) := KT(3+F, 3) + T2 * T3 * K(TEM1)
KT(4+F, 4) := KT(4+F, 4) + T4 * T1 * K(TEM1)
KT(1+F, 5) := KT(1+F, 5) + FJO * HJO * K(TEM1)
KT(2+F, 5) := KT(2+F, 5) + GJO * HJO * K(TEM1)
KT(3+F, 5) := KT(3+F, 5) + T3 * HJO * K(TEM1)
KT(4+F, 5) := KT(4+F, 5) + T1 * HJO * K(TEM1)
KT(5+F, 5) := KT(5+F, 5) + HJO * HJO * K(TEM1)

MSM2

J := 6 (1) 4 * N1(C) + 5

CI := J + 2 * N1(C) - 5 + D
CJ := CI - ADD(CI) + ADD(CI - 1) + 1

$I := CJ(1) \cdot 2 \cdot N1(C) + D$

$IA := I - 2 \cdot \text{ENTIER}(I/2)$

$IA = 1$

$THETA := -3.14159 + 2 \cdot ((I-D)/2-1) \cdot 3.14159 / (N1(C)-1)$

$THETA := -3.14159 + 2 \cdot ((I-D+1)/2-1) \cdot 3.14159 / (N1(C)-1)$

$FIO := (RO(C)/2) \uparrow 0.5 \cdot ((2 \cdot KAP-1) \cdot \cos(THETA/2) - \cos(1.5 \cdot THETA)) / (4 \cdot G)$
 $GIO := (RO(C)/2) \uparrow 0.5 \cdot ((2 \cdot KAP+3) \cdot \sin(THETA/2) + \sin(1.5 \cdot THETA)) / (4 \cdot G)$
 $FI1 := (RO(C)/2) \uparrow 0.5 \cdot ((2 \cdot KAP+1) \cdot \sin(THETA/2) - \sin(1.5 \cdot THETA)) / (4 \cdot G)$
 $GI1 := -(RO(C)/2) \uparrow 0.5 \cdot ((2 \cdot KAP-3) \cdot \cos(THETA/2) + \cos(1.5 \cdot THETA)) / (4 \cdot G)$
 $HIO := -RO(C) \cdot \sin(THETA)$
 $HI1 := RO(C) \cdot \cos(THETA)$

$IA = 1$

$FIO := FIO \cdot SN + FI1 \cdot CN$
 $GIO := GIO \cdot SN + GI1 \cdot CN$
 $HIO := HIO \cdot SN + HI1 \cdot CN$
 $T1 := SN \quad T2 := CN$

$FIO := FIO \cdot CN - FI1 \cdot SN$
 $GIO := GIO \cdot CN - GI1 \cdot SN$
 $HIO := HIO \cdot CN - HI1 \cdot SN$
 $T1 := CN \quad T2 := -SN$

$TEM1 := \text{ADD}(CI) - CI + I$
 $KT(1+F, J) := KT(1+F, J) + FIO \cdot K(TEM1)$
 $KT(2+F, J) := KT(2+F, J) + GIO \cdot K(TEM1)$
 $KT(3+F, J) := KT(3+F, J) + T1 \cdot K(TEM1)$
 $KT(4+F, J) := KT(4+F, J) + T2 \cdot K(TEM1)$
 $KT(5+F, J) := KT(5+F, J) + HIO \cdot K(TEM1)$

$I := 1(1) \cdot 2 \cdot (NS-1-N1(1))$

$CJ := I + 2 \cdot N1(1)$
 $CI := CJ - \text{ADD}(CJ) + \text{ADD}(CJ-1) + 1$

$CI \leq 2 \cdot N1(1)$

$CI := 2 \cdot N1(1) + 1$

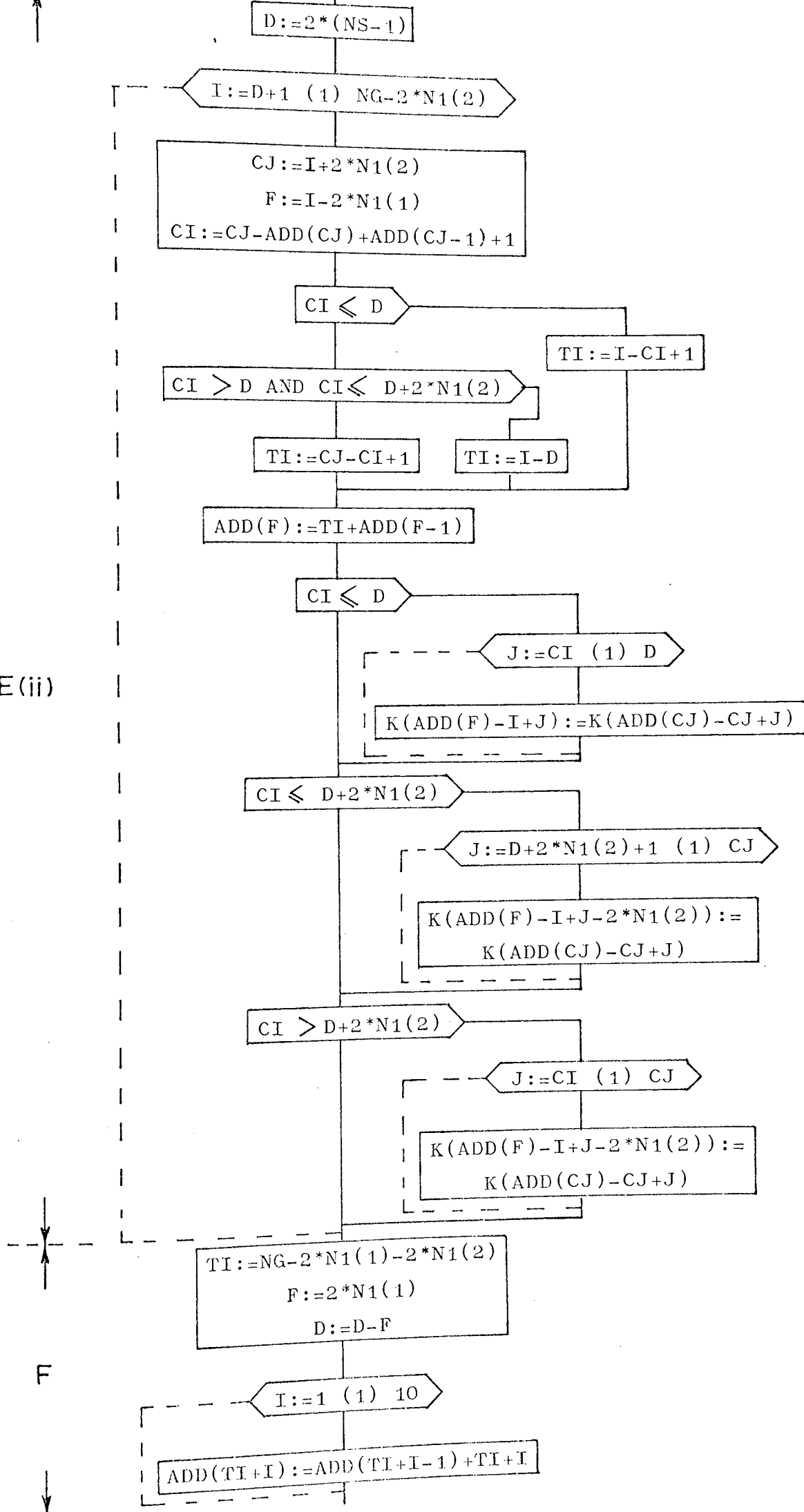
$\text{ADD}(I) := I + 2 \cdot N1(1) - CI + \text{ADD}(I-1) + 1$

$J := CI(1) \cdot I + 2 \cdot N1(1)$

$K(\text{ADD}(I) - CJ + J) := K(\text{ADD}(CJ) - CJ + J)$

E(i)

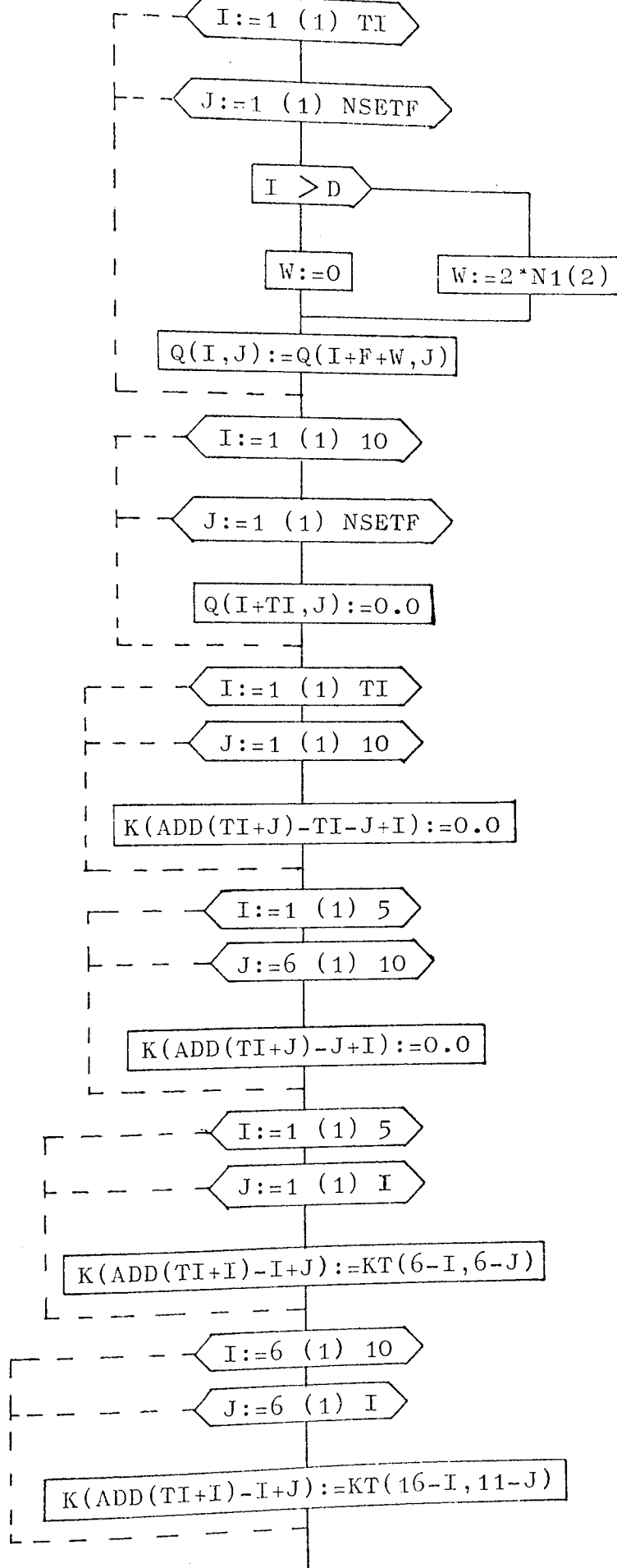
E(ii)

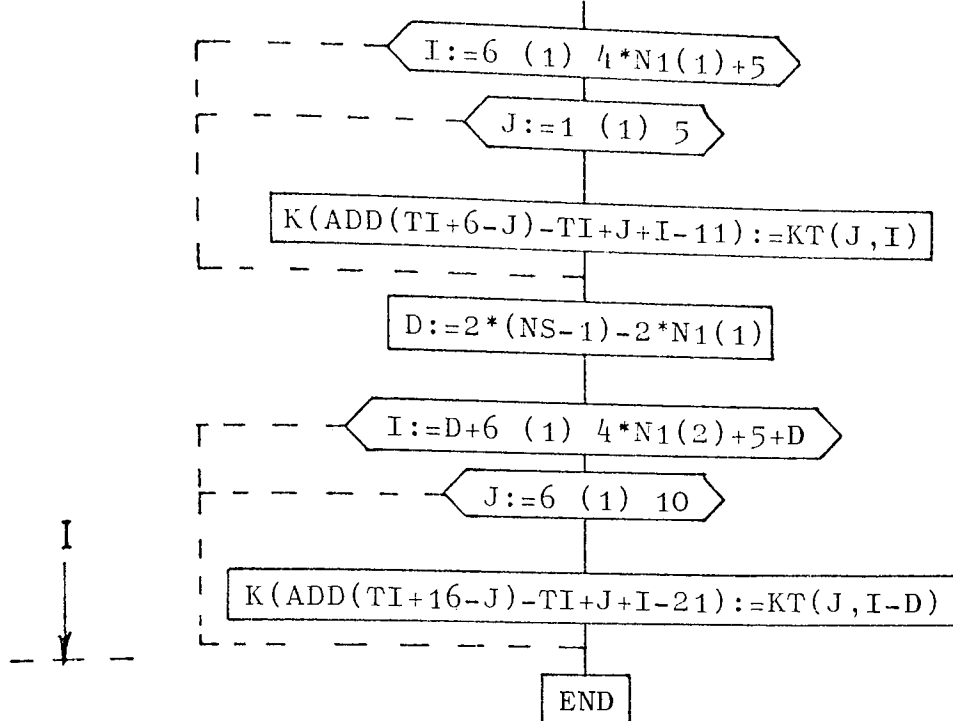


G



H





```

PROCEDURE MM2T(NG,NSETF,A,ADD,K,0,CASE,THICK,N1,RO,GRT,AL,NS);
VALUE NG,NSETF,CASE,THICK; INTEGER NG,NSETF,CASE,GRT,NS;
REAL THICK; INTEGER ARRAY ADD,N1; REAL ARRAY K,0,A,RO,AL;
BEGIN COMMENT MODIFICATION OF OVERALL STIFFNESS MATRIX FOR THE
MIXED MODE CASE,TWO CRACK TIPS;
INTEGER IA,II,C1,D1,DJ,IJ,JA,DUP,TEM1,TEM2,I,J,V,C,F,F;
REAL F10,FJ0,G10,GJ0,THETA,KAP, SN1,CN1,F1,G1,H1,T1,T2,T3,T4,T5,H11,
T11,T21,T31,T41,FJ,GJ,IJ,FJ1,FJ1,GJ1,GJ1,SN,CN,IJ1,H10,HJ0,MU,G;
ARRAY KFI1:10,1:4*GRT+51;
FOR I:=1 STEP 1 UNTIL 10 DO
FOR J:=1 STEP 1 UNTIL 4*GRT+5 DO KFI1(I,J):=0.0;
RO:=AL/2; G:=0.5*AL/11/(1+A[2]);
WRITE TEXT('('('20')'PROCEDURE MM2T BEGINS('('('('THIS PROCEDUREX
MODIFIES THE OVERALL STIFFNESS MATRIX FOR MIXED MODE FRACTURE,TWO CRACK
TIPS PRESENT'))');
FOR I:=1,2 DO
BEGIN
WRITE TEXT('('('30')'DETAILS OF TIP NUMBER-'('('('1'))'); PRINT(I,2,0);
WRITE TEXT('('('35')'ANGLE OF TIP TO X+VE X-AXIS,DEG='('('('1'))');
PRINT(AL(I),0,4);
WRITE TEXT('('('35')'NUMBER OF NODES ON CORE ELEMENT='('('('1'))');
PRINT(N1(I),3,0);
WRITE TEXT('('('35')'RADIUS OF CORE ELEMENT='('('('1'))'); PRINT(RO(I),0,4);
END;
WRITE TEXT('('('30')'MODULUS OF RIGIDITY,G='('('('1'))'); PRINT(G,0,4);
WRITE TEXT('('('30')'POISSON'S RATIO,MU='('('('1'))'); PRINT(MU,0,4);
COMMENT NS IS THE NODE NUMBER STARTING THE SECOND CORE;
IF CASE=0 THEN KAP:=(3-MU)/(1+MU) ELSE KAP:=3-4*MU;
FOR I:=1,2 DO
BEGIN AL(I):=6.0174533*AL(I);
IF I=1 THEN J:=0 ELSE J:=5;
ST(J+1,1):=RO(I)*3.14159*(2*KAP-1)/(A*G)*THICK;
ST(J+2,2):=RO(I)*3.14159*(3+2*KAP)/(A*G)*THICK;
END;
FOR C:=1,2 DO
BEGIN IF C=1 THEN D:=0 ELSE D:=2*(NS-1);
SN:=SIN(AL(C)); CN:=COS(AL(C));
IF C=1 THEN F:=0 ELSE F:=5;
FOR I:=1+D STEP 1 UNTIL 2*N1(C)+D DO
BEGIN IA:=I-2*ENTIER(I/2);
IF IA=1 THEN THETA:=-3.14159+2*((I-D+1)/2-1)*3.14159/(N1(C)-1)
ELSE THETA:=-3.14159+2*((I-D)/2-1)*3.14159/(N1(C)-1);
F10:=(RO(I/2)+0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA)))/(4*G);
G10:=(RO(I/2)+0.5*((2*KAP+3)*SIN(THETA/2)+SIN(1.5*THETA)))/(4*G);
F11:=(RO(I/2)+0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA)))/(4*G);
G11:=-(RO(I/2)+0.5*((2*KAP-3)*COS(THETA/2)+COS(1.5*THETA)))/(4*G);
H10:=-RO(I)*SIN(THETA); H11:=RO(I)*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*SN+F11*CN;
G10:=G10*SN+G11*CN;
H10:=H10*SN+H11*CN;
T2:=SN; T4:=CN;
END;
CI:=I-ADD(I)+ADD(I-1)+1;
FOR J:=C1 STEP 1 UNTIL 2*N1(C)+D DO

```

```

'BEGIN' 'IF' J 'IF' I 'THEN'
  'BEGIN' DI:=I; DJ:=J;
    'GOTO' MSM1;
  'END' 'ELSE'
    'BEGIN' CJ:=J-ADD[CJ]+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-D+1)/2-1)*3.14159/(N1[C]-1);
  'ELSE' THETA:=-3.14159+2*((J-D)/2-1)*3.14159/(N1[C]-1);
  FJ0:=(PO[C]/2)*0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(4*G);
  GJ0:=(PO[C]/2)*0.5*((2*KAP+3)*SIN(THETA/2)+SIN(1.5*THETA))/(4*G);
  FJ1:=(PO[C]/2)*0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(4*G);
  GJ1:=- (PO[C]/2)*0.5*((2*KAP-3)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
  HJ0:=-PO[C]*SIN(THETA); HJ1:=PO[C]*COS(THETA);
  'IF' JA=1 'THEN'
    'BEGIN' FJ0:=FJ0*CN-FJ1*SN;
      GJ0:=GJ0*CN-GJ1*SN;
      HJ0:=HJ0*CN-HJ1*SN;
      T1:=-SN; T3:=CN;
    'END' 'ELSE'
    'BEGIN' FJ0:=FJ0*SN+FJ1*CN;
      GJ0:=GJ0*SN+GJ1*CN;
      HJ0:=HJ0*SN+HJ1*CN;
      T1:=CN; T3:=SN;
    'END';
  TEM1:=ADD[DI]-DI+DJ;
  KT[1+F,1]:=KT[1+F,1]+FJ0*FJ0*K[TEM1];
  KT[1+F,2]:=KT[1+F,2]+GJ0*FJ0*K[TEM1];
  KT[2+F,2]:=KT[2+F,2]+GJ0*GJ0*K[TEM1];
  KT[1+F,4]:=KT[1+F,4]+FJ0*T4*K[TEM1];
  KT[2+F,4]:=KT[2+F,4]+GJ0*T4*K[TEM1];
  KT[3+F,4]:=KT[3+F,4]+T3*T4*K[TEM1];
  KT[1+F,3]:=KT[1+F,3]+T2*FJ0*K[TEM1];
  KT[2+F,3]:=KT[2+F,3]+T2*GJ0*K[TEM1];
  KT[3+F,3]:=KT[3+F,3]+T2*T3*K[TEM1];
  KT[4+F,4]:=KT[4+F,4]+T4*T1*K[TEM1];
  KT[1+F,5]:=KT[1+F,5]+FJ0*HJ0*K[TEM1];
  KT[2+F,5]:=KT[2+F,5]+GJ0*HJ0*K[TEM1];
  KT[3+F,5]:=KT[3+F,5]+T3*HJ0*K[TEM1];
  KT[4+F,5]:=KT[4+F,5]+T1*HJ0*K[TEM1];
  KT[5+F,5]:=KT[5+F,5]+HJ0*HJ0*K[TEM1];
MSM2: 'END';
'END';
'FOR' J:=6 'STEP' 1 'UNTIL' 4*N1[C]+5 'DO'
  'BEGIN' CI:=J+2*N1[C]-5+D;
    CJ:=CI-ADD[CI]+ADD[CI-1]+1;
    'FOR' I:=CJ 'STEP' 1 'UNTIL' 2*N1[C]+D 'DO'
      'BEGIN' IA:=I-2*ENTIER(I/2);
        'IF' IA=1 'THEN'
          THETA:=-3.14159+2*((I-D+1)/2-1)*3.14159/(N1[C]-1);
        'ELSE' THETA:=-3.14159+2*((I-D)/2-1)*3.14159/(N1[C]-1);
        FJ0:=(PO[CI]/2)*0.5*((2*KAP-1)*COS(THETA/2)-COS(1.5*THETA))/(4*G);
        GJ0:=(PO[CI]/2)*0.5*((2*KAP+3)*SIN(THETA/2)+SIN(1.5*THETA))/(4*G);
        FJ1:=(PO[CI]/2)*0.5*((2*KAP+1)*SIN(THETA/2)-SIN(1.5*THETA))/(4*G);
        GJ1:=- (PO[CI]/2)*0.5*((2*KAP-3)*COS(THETA/2)+COS(1.5*THETA))/(4*G);
        HJ0:=-PO[CI]*SIN(THETA); HJ1:=PO[CI]*COS(THETA);
        'IF' IA=1 'THEN'
          'BEGIN' FJ0:=FJ0*CN-FJ1*SN;
            GJ0:=GJ0*CN-GJ1*SN;
            HJ0:=HJ0*CN-HJ1*SN;
          'END'
        'ELSE'
          'BEGIN' FJ0:=FJ0*SN+FJ1*CN;
            GJ0:=GJ0*SN+GJ1*CN;
            HJ0:=HJ0*SN+HJ1*CN;
          'END'
        TEM1:=ADD[CI]-CI+I;
        KT[1+F,1]:=KT[1+F,1]+FJ0*FJ0*K[TEM1];
        KT[1+F,2]:=KT[1+F,2]+GJ0*FJ0*K[TEM1];
        KT[2+F,2]:=KT[2+F,2]+GJ0*GJ0*K[TEM1];
        KT[1+F,4]:=KT[1+F,4]+FJ0*T4*K[TEM1];
        KT[2+F,4]:=KT[2+F,4]+GJ0*T4*K[TEM1];
        KT[3+F,4]:=KT[3+F,4]+T3*T4*K[TEM1];
        KT[1+F,3]:=KT[1+F,3]+T2*FJ0*K[TEM1];
        KT[2+F,3]:=KT[2+F,3]+T2*GJ0*K[TEM1];
        KT[3+F,3]:=KT[3+F,3]+T2*T3*K[TEM1];
        KT[4+F,4]:=KT[4+F,4]+T4*T1*K[TEM1];
        KT[1+F,5]:=KT[1+F,5]+FJ0*HJ0*K[TEM1];
        KT[2+F,5]:=KT[2+F,5]+GJ0*HJ0*K[TEM1];
        KT[3+F,5]:=KT[3+F,5]+T3*HJ0*K[TEM1];
        KT[4+F,5]:=KT[4+F,5]+T1*HJ0*K[TEM1];
        KT[5+F,5]:=KT[5+F,5]+HJ0*HJ0*K[TEM1];
      'END'
    'END'
  'END'

```

```

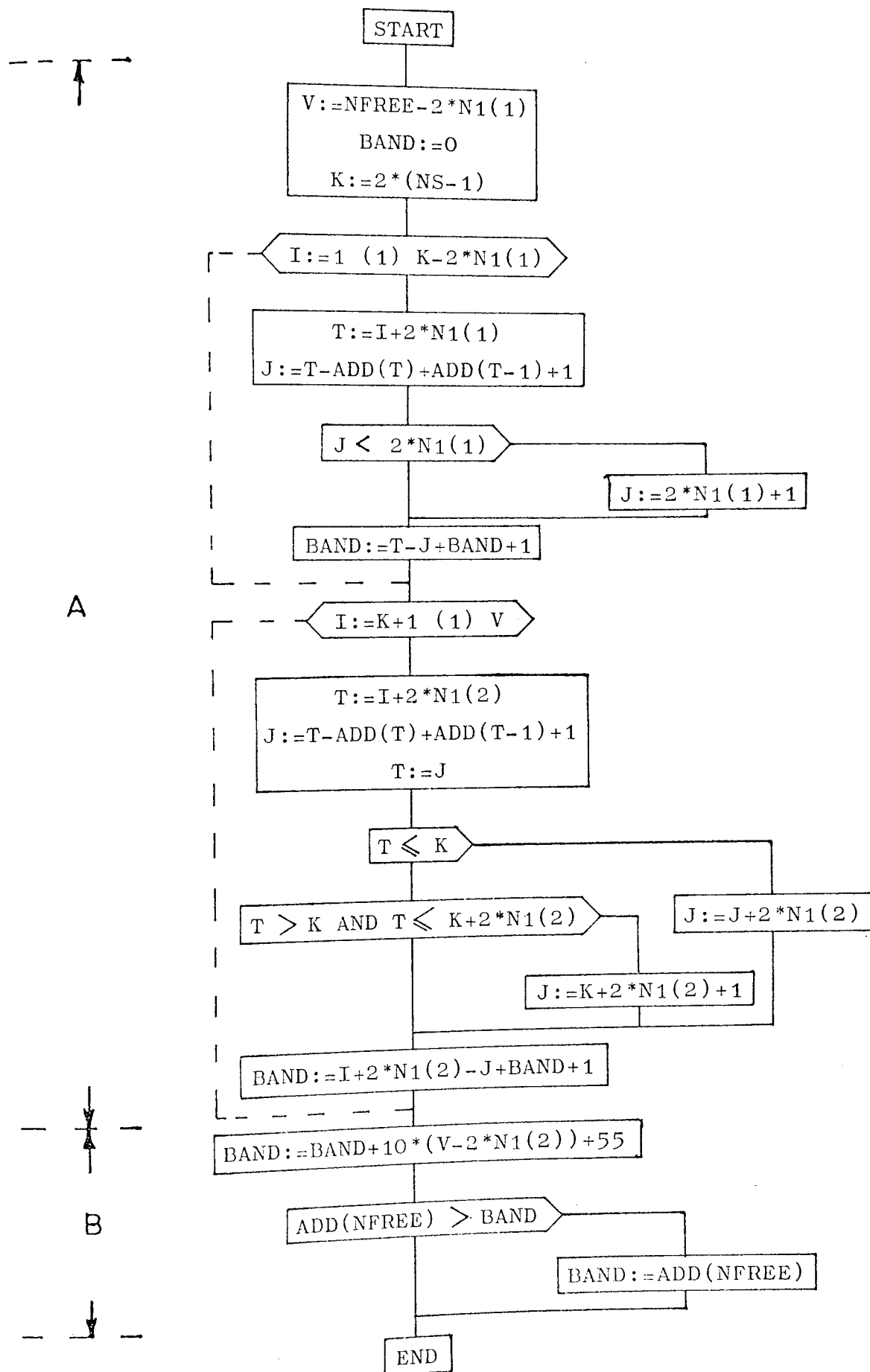
      T1:=CN; T2:=-SN;
    'END' 'ELSE'
  'BEGIN' F10:=F10*SN+F11*CN;
        G10:=G10*SN+G11*CN;
        H10:=H10*SN+H11*CN;
        I1:=SN; I2:=CN;
  'END';
  TEM1:=ADD(CI)-CI+1;
  KT[1+F,J]:=KT[1+F,J]+F10*K[TEM1];
  KT[2+F,J]:=KT[2+F,J]+G10*K[TEM1];
  KT[3+F,J]:=KT[3+F,J]+T1*K[TEM1];
  KT[4+F,J]:=KT[4+F,J]+T2*K[TEM1];
  KT[5+F,J]:=KT[5+F,J]+F10*K[TEM1];
'END';
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' 2*(NS-1-N1[1]) 'DO'
'BEGIN' CJ:=I+2*N1[1];
      CI:=CI-ADD[CJ]+ADD[CJ-1]+1;
      'IF' CI 'LE' 2*N1[1] 'THEN' CI:=2*N1[1]+1;
      ADD[I]:=I+2*N1[1]-CI+ADD[I-1]+1;
      'FOR' J:=CI 'STEP' 1 'UNTIL' I+2*N1[1] 'DO'
      KT[ADD[I]-CJ+J]:=K[ADD[CJ]-CJ+J];
'END';
      F:=2*(NS-1);
'FOR' I:=D+1 'STEP' 1 'UNTIL' NG-2*N1[2] 'DO'
'BEGIN' CJ:=I+2*N1[2]; F:=I-2*N1[1];
      CI:=CJ-ADD[CJ]+ADD[CJ-1]+1;
      'IF' CI 'LE' D 'THEN' TI:=I-CI+1 'ELSE'
      'IF' CI 'GT' D 'AND' CI 'LE' D+2*N1[2] 'THEN' TI:=I-D
      'ELSE' TI:=CJ-CI+1;
      ADD[F]:=TI+ADD[F-1];
      'IF' CI 'LE' D 'THEN'
      'FOR' J:=CI 'STEP' 1 'UNTIL' D 'DO'
      KT[ADD[F]-I+J]:=K[ADD[CJ]-CJ+J];
      'IF' CI 'LE' D+2*N1[2] 'THEN'
      'FOR' J:=D+2*N1[2]+1 'STEP' 1 'UNTIL' CJ 'DO'
      KT[ADD[F]-I+J-2*N1[2]]:=K[ADD[CJ]-CJ+J];
      'IF' CI 'GT' D+2*N1[2] 'THEN'
      'FOR' J:=CI 'STEP' 1 'UNTIL' CJ 'DO'
      KT[ADD[F]-I+J-2*N1[2]]:=K[ADD[CJ]-CJ+J];
'END';
TI:=NG-2*N1[1]-2*N1[2]; F:=2*N1[1]; D:=D-F;
'FOR' I:=1 'STEP' 1 'UNTIL' 10 'DO' ADD[TI+I]:=ADD[TI+I-1]+TI+I;
'FOR' I:=1 'STEP' 1 'UNTIL' TI 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO'
'BEGIN' 'IF' J 'GT' D 'THEN' W:=2*N1[2] 'ELSE' W:=0;
      QET,J1:=QET+P+W,J;
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' 10 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' QET+TI,J:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' TI 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 10 'DO' KT[ADD[TI+J]-TI-J+1]:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 10 'DO' KT[ADD[TI+J]-J+1]:=0.0;
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 1 'DO' K[ADD[TI+I]-I+J]:=KT[6-I,6-J];
'FOR' J:=1 'STEP' 1 'UNTIL' 10 'DO'
'FOR' I:=6 'STEP' 1 'UNTIL' 1 'DO' K[ADD[TI+I]-I+J]:=KT[16-I,11-J];
'FOR' J:=6 'STEP' 1 'UNTIL' 1 'DO' K[ADD[TI+I]-I+J]:=KT[16-I,11-J];
'FOR' I:=6 'STEP' 1 'UNTIL' 4+N1[3]+5 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 5 'DO' K[ADD[TI+6-J]-TI+J+I-11]:=KT[J,I];
D:=2*(NS-1)-2*N1[1];
'FOR' I:=D+6 'STEP' 1 'UNTIL' 4*N1[2]+5+D 'DO'
'FOR' J:=6 'STEP' 1 'UNTIL' 10 'DO'
KT[ADD[TI+16-J]-TI+J+I-21]:=KT[J,I-D];
'END' OF PROCEDURE MM2T;

```

In a similar manner to section 5.3, a procedure MM2TKABND is included which calculates the bounds of $[K^*]$ in advance to ensure that the value is not greater than bounds originally calculated for $[K]$, and so that adequate new storage capacity is allocated. A flowchart and program listing for this procedure follows, and the steps are summarised with reference to the flowchart as :

- A. - Total number of coefficients present in array $[K]$ are calculated in integer BAND, after the elimination of rows and columns corresponding to core nodes, i.e. rows and columns 1 to $2N1[1]$ and $2(NS-1)+1$ to $2(NS-1)+2N1[2]$, Fig. 5.23.
- B. - Value of BAND found from step A is compared with the bounds of $[K]$ (ADD[NFREE]). The subsequent declaration of storage for $[K]$ will be $K[1 : BAND]$.

PROCEDURE : MM2TKABND(NFREE,N1,BAND,ADD,NS)




```

'PROCEDURE' MY2TKABND(NREFE,N1,BAND,ADD,NS);
'VALUE' NREFE: 'INTEGER' BAND,NREFE,NS;
'INTEGER' 'ARRAY' ADD,N1;
'BEGIN' 'INTEGER' I,J,K,V,T;
V:=NREFE-2*N1[1]; BAND:=0; K:=2*(NS-1);
'FOR' I:=1 'STEP' 1 'UNTIL' K-2*N1[1] 'DO'
'BEGIN' T:=I+2*N1[1];
      J:=T-ADD[T]+ADD[T-1]+1;
      'IF' J 'LT' 2*N1[1] 'THEN' J:=2*N1[1]+1;
      BAND:=T-J+BAND+1;
'END';
'FOR' I:=K+1 'STEP' 1 'UNTIL' V 'DO'
'BEGIN' T:=I+2*N1[2];
      J:=T-ADD[T]+ADD[T-1]+1; T:=J;
      'IF' T 'LE' K 'THEN' J:=J+2*N1[2] 'ELSE'
      'IF' T 'GT' K 'AND' T 'LE' K+2*N1[2] 'THEN' J:=K+2*N1[2]+1;
      BAND:=I+2*N1[2]-J+BAND+1;
'END';
BAND:=BAND+10*(V-2*N1[2])+55;
'IF' ADD[NREFE] 'GT' BAND 'THEN' BAND:=ADD[NREFE];
'END' OF MY2TKABND;

```

5.5.2 Crack Closure

The closure phenomenon has been discussed in the introduction, section 5.1, where the basic principles for correction have been outlined. Expanding these ideas in preparation for their numerical implementation, the steps involved are :

- A. - Crack surface nodes are positioned contiguously, Fig. 5.25, on opposite crack surfaces, during idealisation. The displacements of these nodes are expressed in local skew coordinates using procedures ~~SKWEDCON~~ and ~~SKWLOAD~~ of Section I such that displacements normal and tangential to the crack surface are obtained as shown, Fig. 5.25.
- B. - Procedure MM2T is called and modified system equations are obtained, i.e. $[K^*]$, $\{F^*\}$ and $[ADD^*]$. These three matrices are copied on magnetic tape for subsequent recall if required.
- C. - Kinematic constraints are introduced in the usual manner using procedure ~~GEOMBC~~ and the resulting equations are solved.
- D. - Check is made on the values of K_{I1} and K_{I2} obtained from the solution of step C, as follows :
 - (i) If both K_{I1} and K_{I2} are negative then the problem does not have a singular solution and the program terminates.
 - (ii) If either K_{I1} or K_{I2} are negative, indicating that only one tip is singular and that the crack surfaces have crossed, then a procedure ~~COMPVIO~~ is called. This procedure examines the normal displacements of the crack surface nodes, establishes where incompatibility has occurred, calculates an average value for the normal displacement of two contiguous nodes, recalls the system equations from magnetic tape and using procedure ~~GEOMBC~~ applies this average displacement as a prescribed nodal displacement to those nodes. The system equations thus obtained are re-solved and the solution is printed.

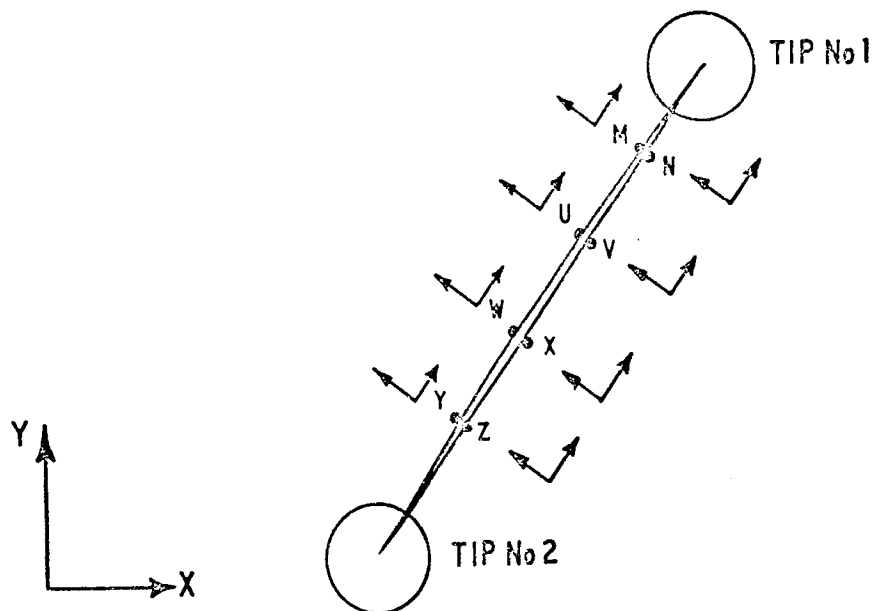


FIG 5.25 ARRANGEMENT OF NODES ON CRACK SURFACE.

- (iii) If both K_{I1} and K_{I2} are positive then no crack closure is involved and the solution is printed in the normal manner.

Fig. 5.26 shows an excerpt from the main program flowchart which illustrates the steps discussed above. A flowchart and program listing follows for procedure ~~CMPVI~~ but before proceeding to a discussion in detail a short note on storage is relevant.

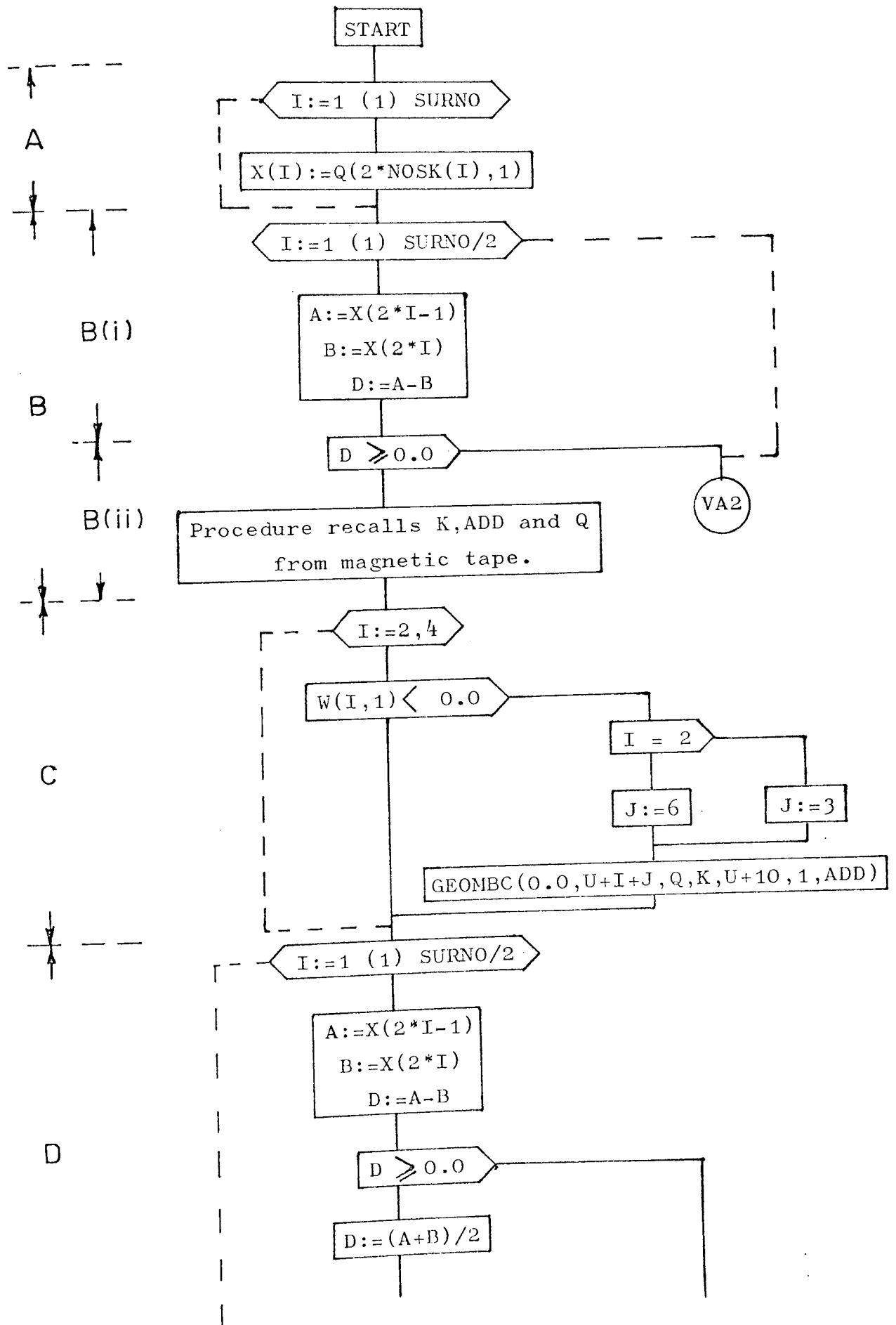
An array ~~N~~SK is declared in the main program which contains the node numbers of the crack surface nodes whose displacements are to be skewed. The node numbers are input to this array in order, upper surface and lower surface for each contiguous pair, e.g. Fig. 5.25, node numbers would be M, N, U, V, W, X, Y, Z. In the procedure an array X, bounds 1 to number of surface nodes, is employed to store the normal displacements of the crack surface nodes in the same order as in ~~N~~SK. Thus in the example of Fig. 5.25 $X[1]$ = Normal displacement of node M and $X[2]$ = Normal displacement of node N, etc.

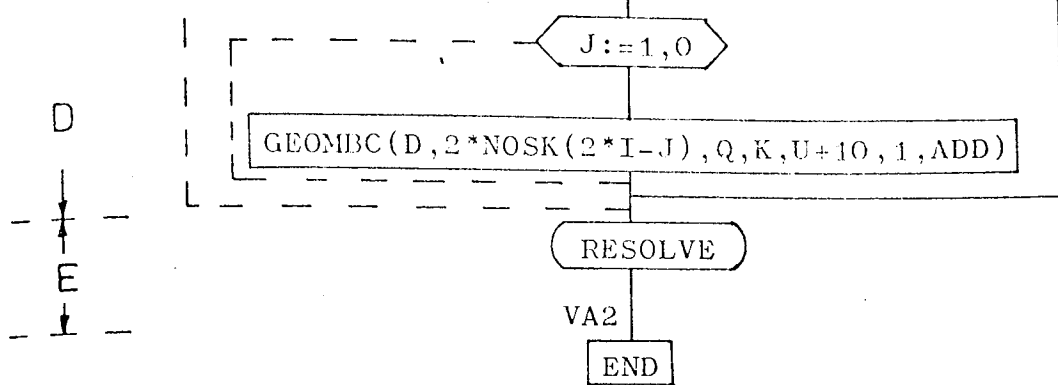
The procedure steps are discussed, with reference to the flowchart, as :

- A. - Coefficients of array X are set to the normal displacement of crack surface nodes.
 - B. - for each pair of surface nodes :
 - (i) difference D is calculated,
 - (ii) if D is greater than zero then compatibility is not violated and the next pair of nodes are checked. If D is always greater than zero then procedure ends, no incompatibility.
- If, however, D is less than zero for any pair of nodes then $[K^*]$, $[ADD^*]$ and $\{F^*\}$ are recalled from magnetic tape in order to prescribe values of nodal displacements to these nodes.

- 381
- C. - $W[2,1] = K_{I1}$, and $W[4,1] = K_{I2}$. If either of these are negative then the procedure ~~GE~~MBC is called and a zero value for K_{I1} or K_{I2} is applied to $[K^*]$ and $\{F^*\}$ as a prescribed displacement.
- D. - for each pair of surface nodes,
- (i) Difference D is calculated.
 - (ii) If D is less than zero indicating incompatibility, then an average value of the normal displacement is calculated, $\left[\frac{A+B}{2} \right]$ and applied as a prescribed displacement to each node using procedure ~~GE~~MBC.
- E. - Label RESOLVE returns the modified stiffness matrix and load vector to the main program, Fig. 5.26, for the application of normal kinematic constraints and the solution of the equations.

PROCEDURE : COMPVIO(Q,K,ADD,U,SURNO,W,NOSK,GEOMBC,GETARRAY,
NVAB1,RESOLVE)






```

'PROCEDURE' COMPVIO(Q,K,ADD,U,SURNO,W,NOSK,GEOMBC,GETARRAY,NVAB1,
    RESOLVE); 'VALUE' U,SURNO; 'INTEGER' U,SURNO,NVAB1;
'INTEGER' 'ARRAY' ADD,NOSK; 'REAL' 'ARRAY' Q,K,W;
'PROCEDURE' GEOMBC,GETARRAY; 'LABEL' RESOLVE;
'REGIN' 'INTEGER' I,J;
'REAL' A,B,D; 'REAL' 'ARRAY' X[1:SURNO];
'FOR' I:=1 'STEP' 1 'UNTIL' SURNO 'DO' X[I]:=Q[2*NOSK[I],1];
'FOR' I:=1 'STEP' 1 'UNTIL' SURNO/2 'DO'
'REGIN' A:=X[2*I-1]; B:=X[2*I]; D:=A-B;
    'IF' D 'LT' 0.0 'THEN' 'GOTO' VA1;
'END'; 'GOTO' VA2;
VA1: NVAB1:=1;
GETARRAY(10,NVAB1,K);
GETARRAY(10,NVAB1,ADD);
GETARRAY(10,NVAB1,Q);
'FOR' I:=2,4 'DO' 'IF' W[I,1] 'LE' 0.0 'THEN'
    GEOMBC(0.0,U+1+('IF' I=2 'THEN' 3 'ELSE' 6),Q,K,U+10,1,ADD);
'FOR' I:=7 'STEP' 1 'UNTIL' SURNO/2 'DO'
'REGIN' A:=X[2*I-1]; B:=X[2*I]; D:=A-B;
    'IF' D 'LT' 0.0 'THEN'
'REGIN' D:=(A+B)/2;
        'FOR' J:=1,0 'DO'
            GEOMBC(0,2*NOSK[2*I-J],Q,K,U+10,1,ADD);
'END';
'END';
'GOTO' RESOLVE;
VA2: 'END' OF PROCEDURE COMPVIO;

```

6. NUMERICAL EXAMPLES

6.1 INTRODUCTION

The various procedures necessary for augmenting a standard finite element program, such as that developed in Section I, to include a core element or elements for the calculation of Mode I or mixed Modes I and II stress intensity factors have been presented in Chapter 5. The complete programs with input data instructions required for the execution of each one are presented in Appendix 10.D. It is the purpose of this chapter to illustrate the application of each of these programs by obtaining numerical solutions for a number of specific crack problems, some of which have been solved previously by other methods and others which have never appeared in the published literature. The problems are chosen to demonstrate the accuracy and flexibility of the technique for arbitrary two-dimensional component shapes.

A series of results are presented to demonstrate the sensitivity of the approach to the various approximations involved. In particular, the influence of the size of the core region and the number of nodes along the core/finite element mesh interface are studied.

The chapter is divided into sections, each of which refer to a particular crack configuration, Mode I single tip, mixed mode single tip, etc.

6.2 SINGLE TIP, SYMMETRIC MODE I

The example chosen to examine the influence of core radius and number of core/finite element mesh interface nodes on the Mode I stress intensity factor K_I is a centrally cracked specimen subjected to tension, Fig. 5.7. The values of K_I obtained were compared with those by Paris and Sih⁽³⁾ from the solution for a periodic array of cracks. For the particular plate geometry with $b/c = 1.4$, $c/a = 0.3$,

$$\overline{k_I} = \frac{k_I}{\sigma\sqrt{a}} = 1.04, \text{ according to Ref. (3).}$$

From symmetry only one quarter of the plate need be considered and it is shown in Fig. 5.8 with a core element around the crack tip. The distribution of nodes and their numbering in the neighbourhood of the core element is shown Fig. 6.1, with three rings of nodes as determined by radii, R_c , R_1 and R_2 . In the first part of the study, radius R_1 was varied, with $R_2 = 0.5$ ins. and $R_c = 0.05$ ins. arbitrarily chosen and fixed in an attempt to examine the influence of the surrounding mesh and hence to determine an optimum ratio R_1/R_c . Results for the range of R_1 considered are shown in Fig. 6.2, giving an optimum value of R_1/R_c in the range 2 - 3.5.

Secondly, the core radius R_c was varied with R_1/R_c kept approximately constant at a value of 2. Results for the range of core sizes considered are shown, Fig. 6.3. From this graph it is seen that the stress intensity factor seems to be very insensitive to the actual value chosen for R_c within the range considered. It may be concluded, therefore, that all the chosen core radii are sufficiently small so that the one term near-tip displacement field is a good approximation to the true solution everywhere in the core region. For the particular case of $R_c/a = 0.01$, the value of K_I is approximately 5% in error and it is thought that this may have been caused by the relatively crude mesh surrounding this small core.

Finally, the effect of the number of nodal points on the core/finite element mesh interface was examined. By increasing the nodes on the interface, the total number of degrees of freedom of the mesh are correspondingly increased. The results of Fig. 6.4 therefore reflect both the convergence of the finite element solution by normal refinement of the mesh and the effect of decrease in displacement incompatibility between nodes on the core/finite element mesh interface with an increasing number of nodes.

For the particular ranges of core radii and interface nodes considered, the values obtained for the stress intensity factor K_I are all within 5% of Paris and Sih's result. It is unjustified to state, therefore, that the program results are within 5% of the correct solution as Paris and Sih's result is also an approximate solution. The results do illustrate that acceptable accuracy may be obtained for a wide range of core radii and number of interface nodes.

The other variable associated with the technique which was not examined was the strain energy of the core region. The one term expansion underestimates the strain energy in the core so that the final value obtained for the stress intensity factor is affected. Exactly how this influences the final result is not clear, but it is felt⁽²¹⁾ that this underestimate gives corresponding overestimates for the stress intensity factors in the numerical calculations.

The results of a finite element displacement formulation with full inter-element compatibility bound the true solution from below with a guaranteed convergence as the mesh is refined. This does not appear to be the case with the embedded singularity, as illustrated in Fig. 6.4. The reason may be that the influences of the several different core variables are inter-related and thus it would be necessary perhaps to change R_c for an increased number of interface nodes to give the optimum solution accuracy. The primary source of error is the incompatibility of displacements between the core and the finite element mesh. This displacement incompatibility will tend to disappear as the number of nodes along the interface approaches infinity, it is expected, therefore, that the result will converge in the limit, to the exact solution. To establish this convergence a great many more computational experiments would have to be undertaken with more attention being given to mesh design in elements adjacent to the core. However, the results obtained here appear to be accurate for most practical engineering purposes as the percentage discrepancies are very small.

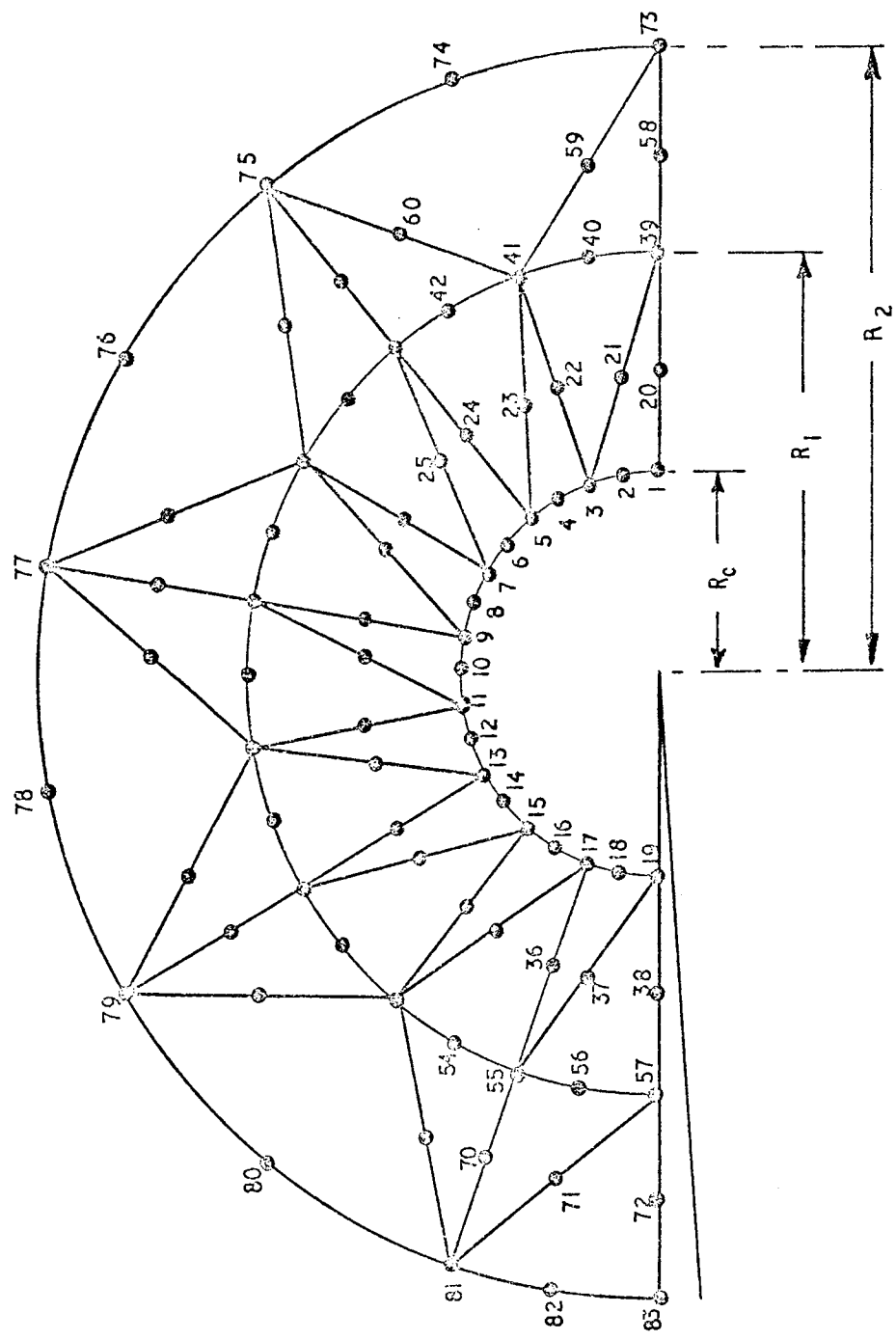


FIG.6.1 DISTRIBUTION OF NODES IN REGION OF CRACK TIP.

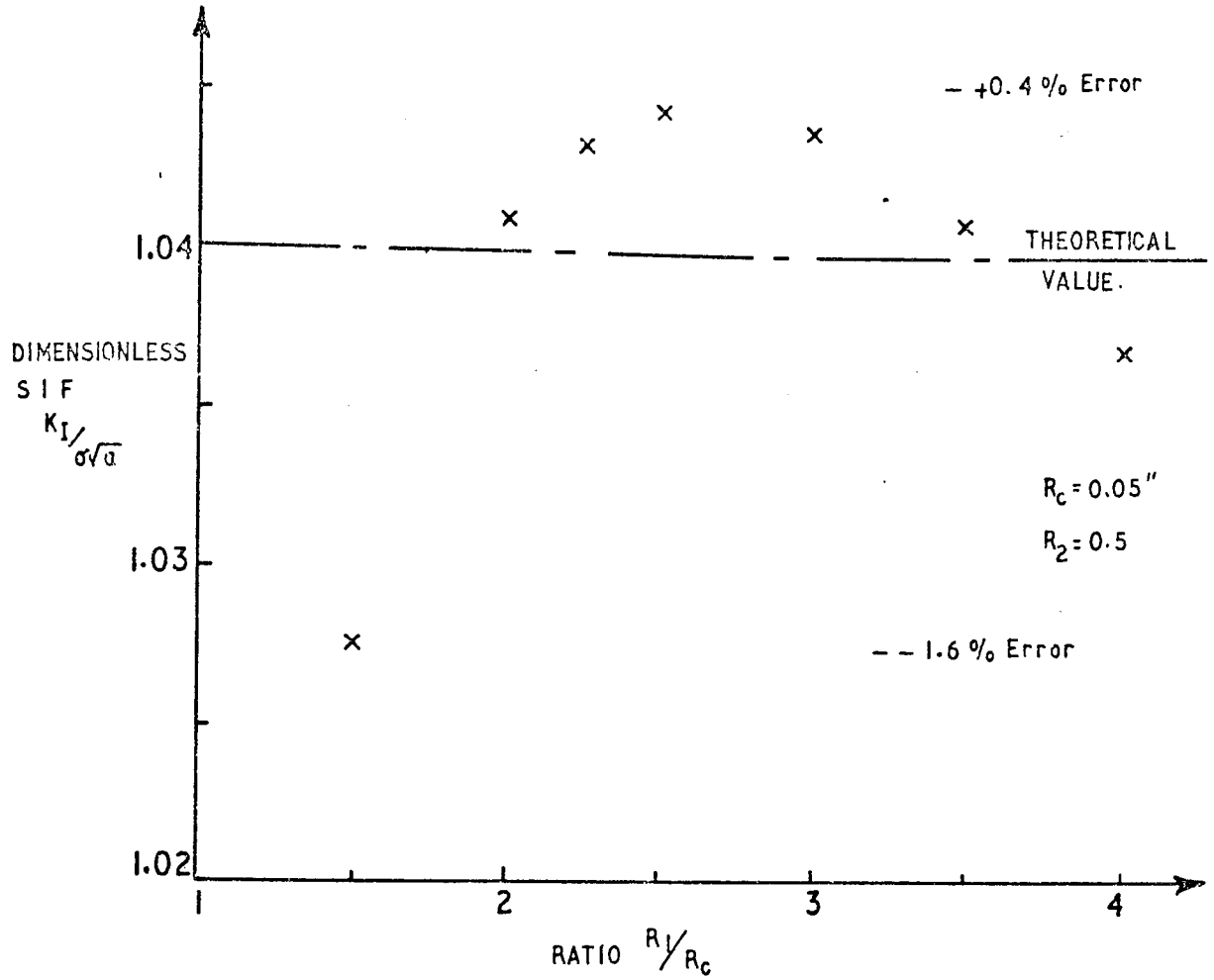


FIG 6.2 GRAPH OF $K_I / \sigma \sqrt{a}$ AGAINST R_1 / R_c .

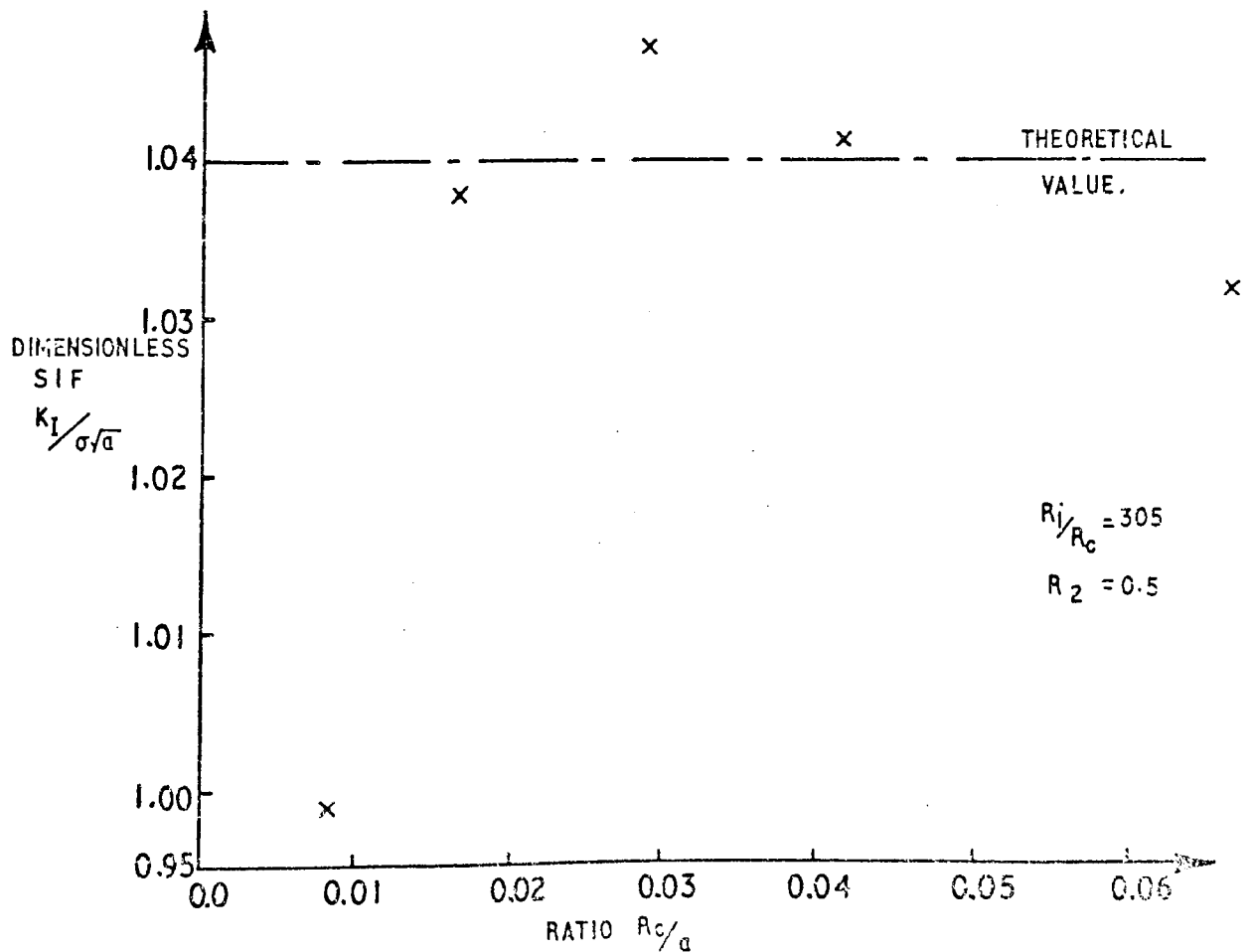


FIG.6.3 GRAPH OF $K_I / \sigma \sqrt{a}$ AGAINST CORE RADIUS (R_c / a)

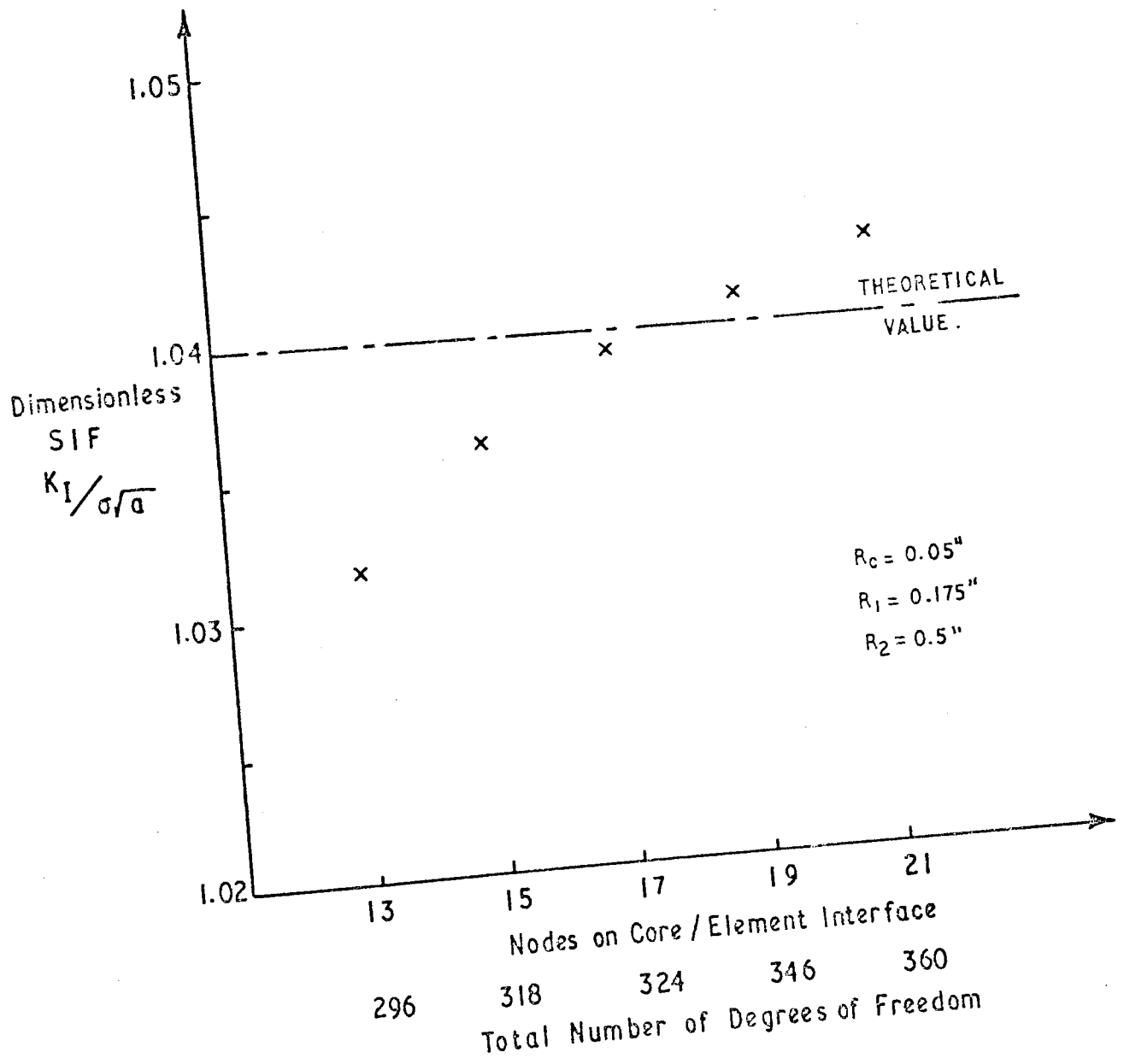


FIG 6.4 GRAPH OF $K_I / \sigma \sqrt{a}$ AGAINST CORE/MESH INTERFACE NODES AND DEGREES OF FREEDOM.

Circular Disc Containing a Central Crack

This example consists of a circular disc of diameter, D , containing a central crack of length $2a$ and subjected to a uniform external tension σ as shown in Fig. 6.5. The same problem was solved by Bowie and Neal⁽⁵⁷⁾ who used a modified mapping-collocation technique. This example provides further confirmation of accuracy and also demonstrates the unique facility of the isoparametric element for modelling curved boundaries.

From symmetry, only one quarter of the disc need be considered, Fig. 6.6, with the boundary conditions as shown. A semi-circular core with $R_c/a = 0.025$, $R_2/R_1 = 2$, $R_1/R_c = 3$, and 19 nodes skirting its boundary, was constructed about the crack tip. The nodal loads corresponding to the uniform external tension were calculated in the usual manner described in Section I.

Table 6.1 shows the results obtained from the program which are compared with those of Bowie and Neal.

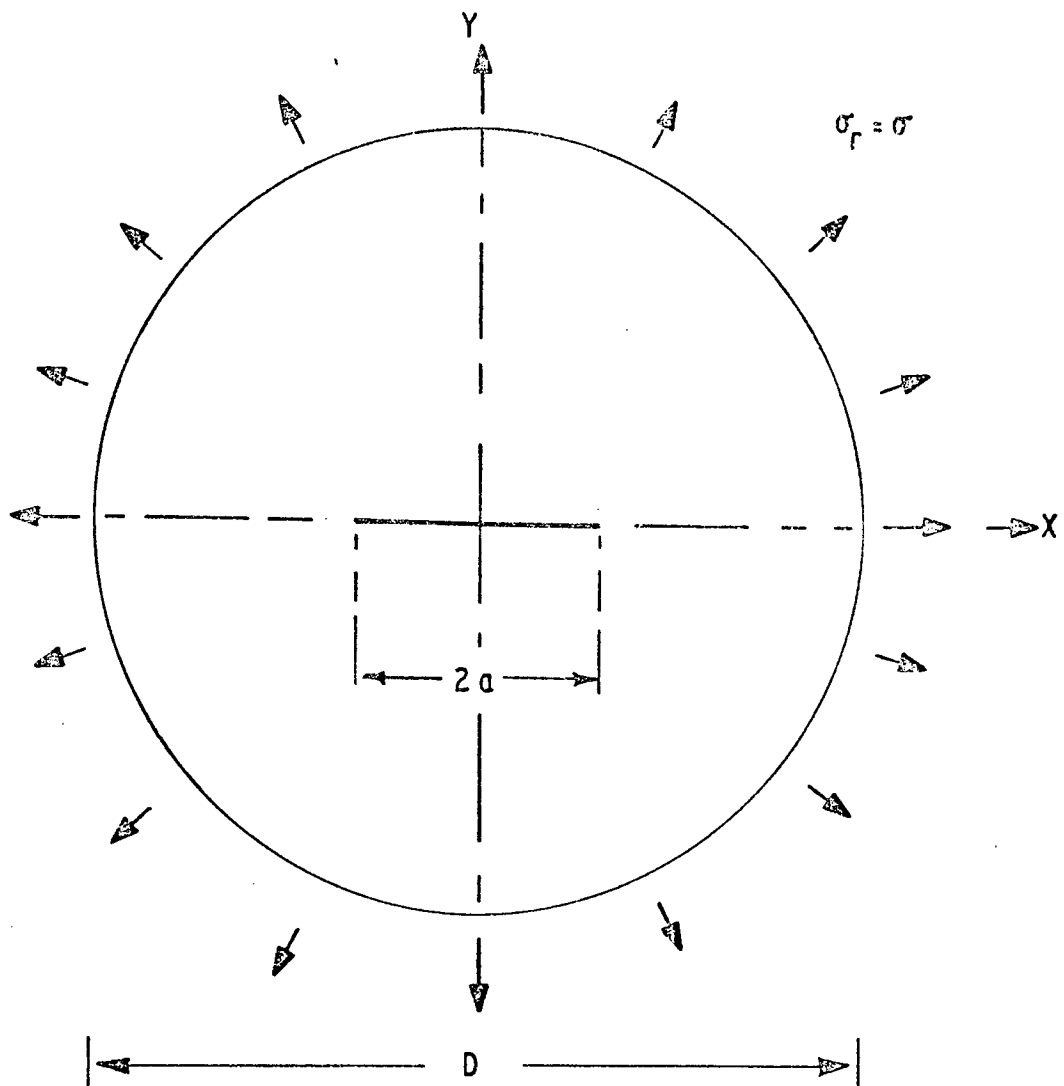


FIG 6.5 CIRCULAR DISC CONTAINING A CENTRAL CRACK.

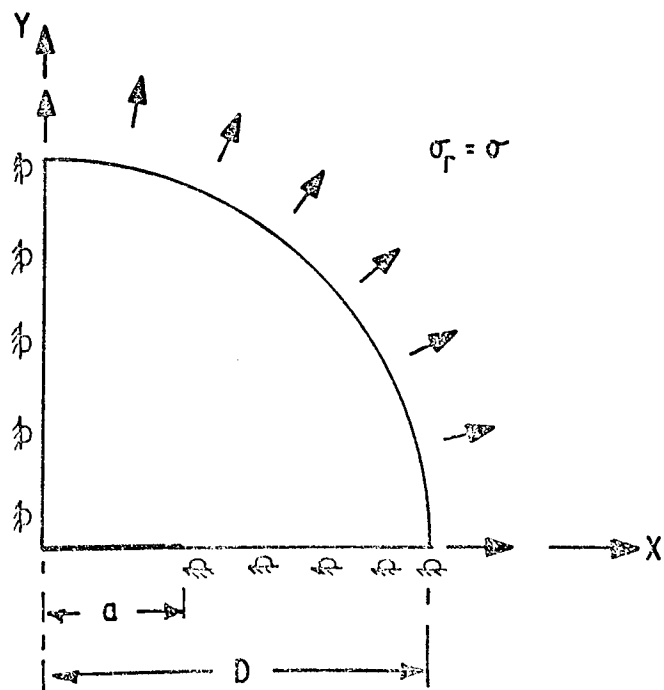


FIG 6.6 QUADRANT OF CIRCULAR DISC WITH BOUNDARY CONDITIONS.

D(ins.)	D/2a	No. of Elements	No. of Nodes	Bowie ⁽⁵⁷⁾ $K_I/\sigma\sqrt{a}$	Computational $K_I/\sigma\sqrt{a}$	Percentage Difference
10	2.5	56	135	1.24	1.241	0.1
10	2.5	64	153	1.24	1.235	0.3
6.0	1.5	63	150	1.74	1.726	<1
5.4	1.35	60	143	1.98	1.971	<0.5
5.0	1.25	61	146	2.24	2.198	1.8
4.8	1.2	58	141	2.43	2.382	2
4.6	1.15	57	138	2.71	2.686	0.8
4.4	1.1	58	144	3.17	3.205	1

Table 6.1 Comparison of K_I Results for Centrally Cracked Circular Disc

Plate Containing Cracks Originating at the Boundary of an Internal Notch

The two cracked configurations analysed are shown in Figs. 6.7 and 6.8 each with two radial cracks emanating from the boundary of the internal circular hole or notch. In each case the overall crack length $2a$ is fixed and the extent of the notch, D , is progressively increased until the cracks are eliminated. The examples are included to assess the performance of the embedded singularity, finite element method in the presence of short crack lengths. These examples are also of interest as part of the wider problem of cracks in the presence of geometric stress concentrations.

The hole or notch has a stress concentrating effect in the absence of cracks and the influence, if any, of their stress gradients on the crack tip stress intensity factors is of considerable interest from a designer's viewpoint. It is, after all, common engineering practice to drill a circular hole to remove crack tips with the object of reducing the concentration of stress at the tips of such cracks.

From symmetry only one quarter of each plate need be analysed, these quadrants are shown together with the boundary conditions in Figs. 6.9 and 6.10. In each case a semi-circular core of radius $R_c = 0.05$ ins. and 19 nodes around its boundary was constructed about the crack tip. This means that the ratio of core radius R_c to crack length $(a - D/2)$ varied as the hole was increased from 0.026 to 0.04.

Results are tabulated for the circular hole and the notch in Tables 6.2 and 6.3 respectively. In Fig. 6.11 the results are compared with existing solutions for the circular hole case as obtained by Cartwright⁽⁵⁸⁾ and Bowie⁽⁵⁹⁾.

The effect of the increased stress concentration of the notch as compared with the circular hole on the stress intensity factors for short crack lengths can be clearly seen from Fig. 6.11. Geometric stress concentrators may therefore have a considerable influence on the growth

characteristics of short cracks and is an area of investigation to which the work described in this thesis can be readily extended.

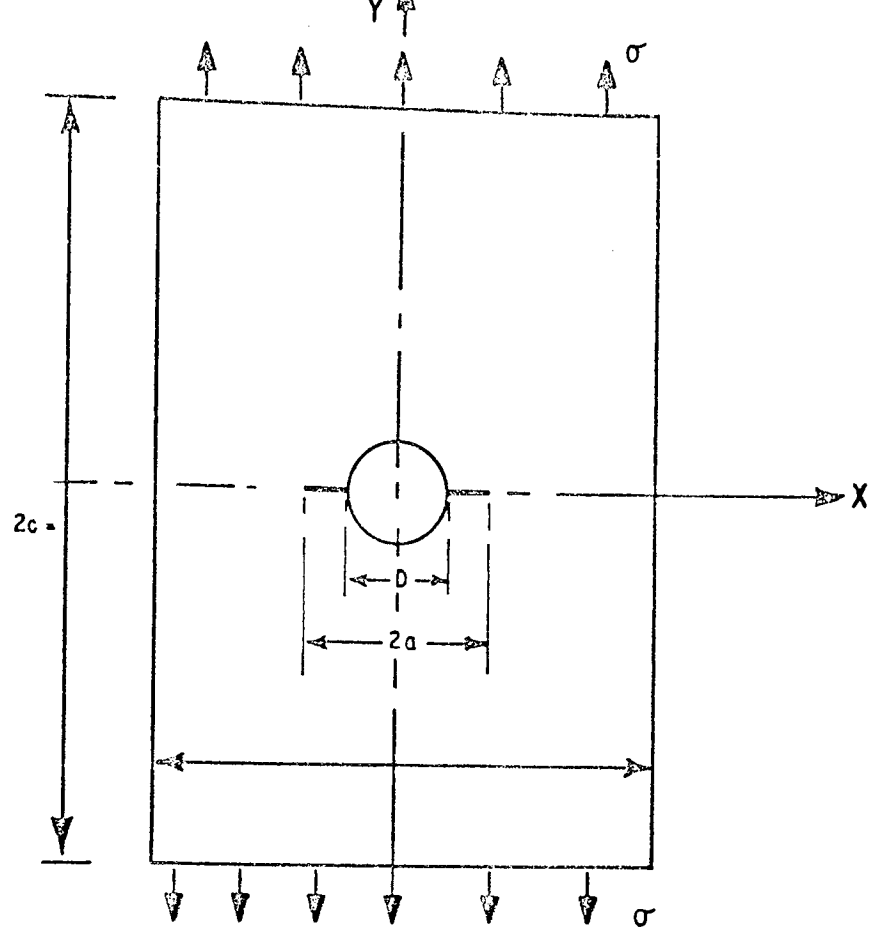


FIG 6.7 RECTANGULAR PLATE CONTAINING A CIRCULAR HOLE AND CRACK.

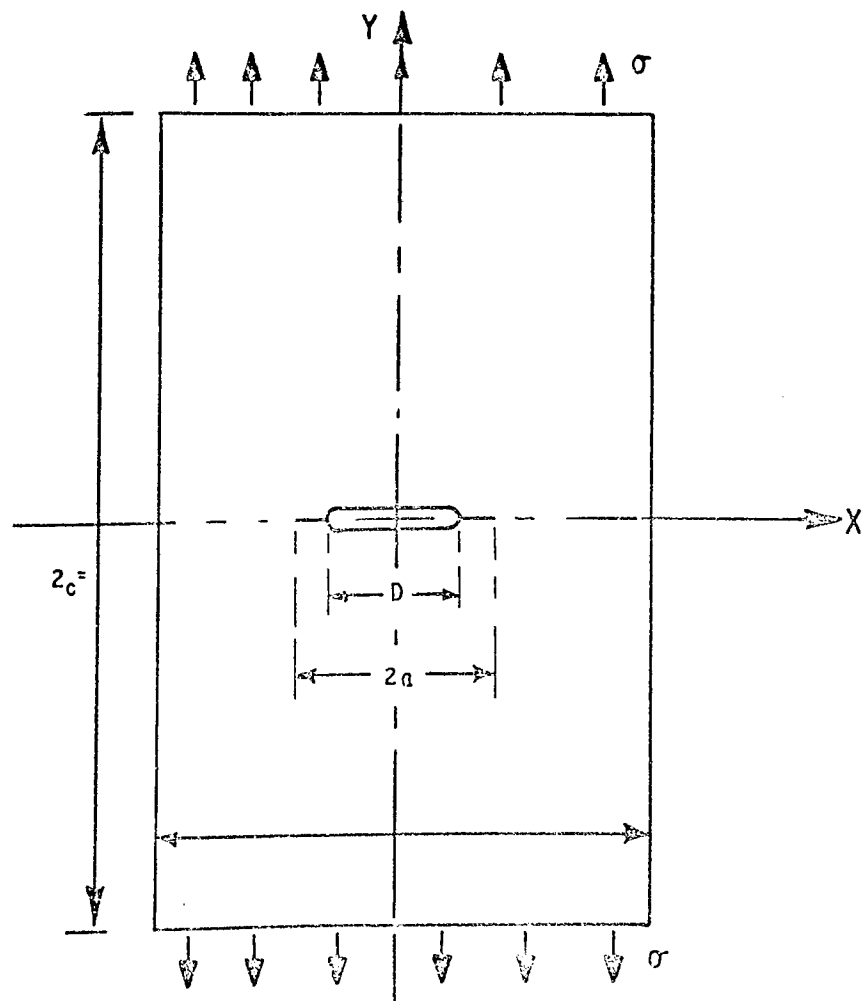


FIG.6.8 RECTANGULAR PLATE CONTAINING A NOTCH AND CRACK .

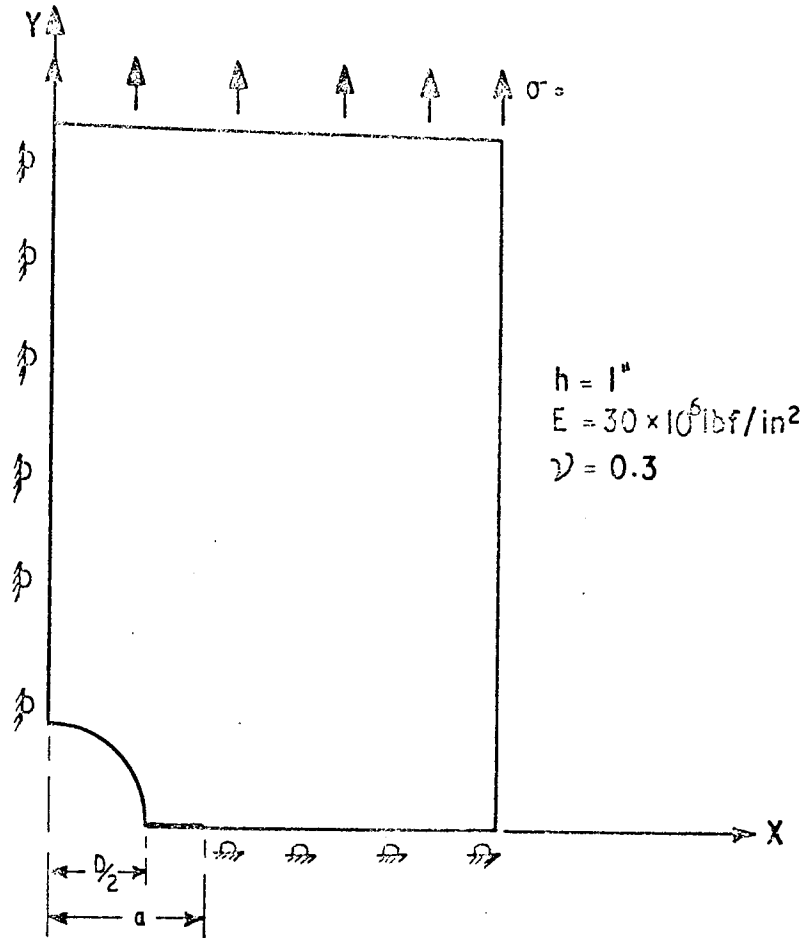


FIG 6.9 QUADRANT OF PLATE CONTAINING CIRCULAR HOLE.

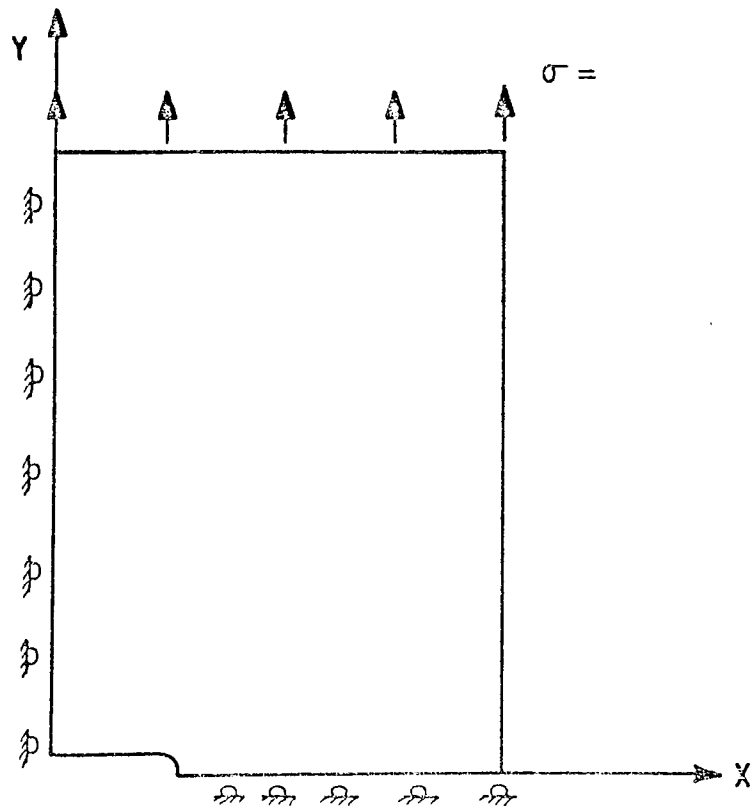


FIG 6.10 QUADRANT OF PLATE CONTAINING NOTCH.

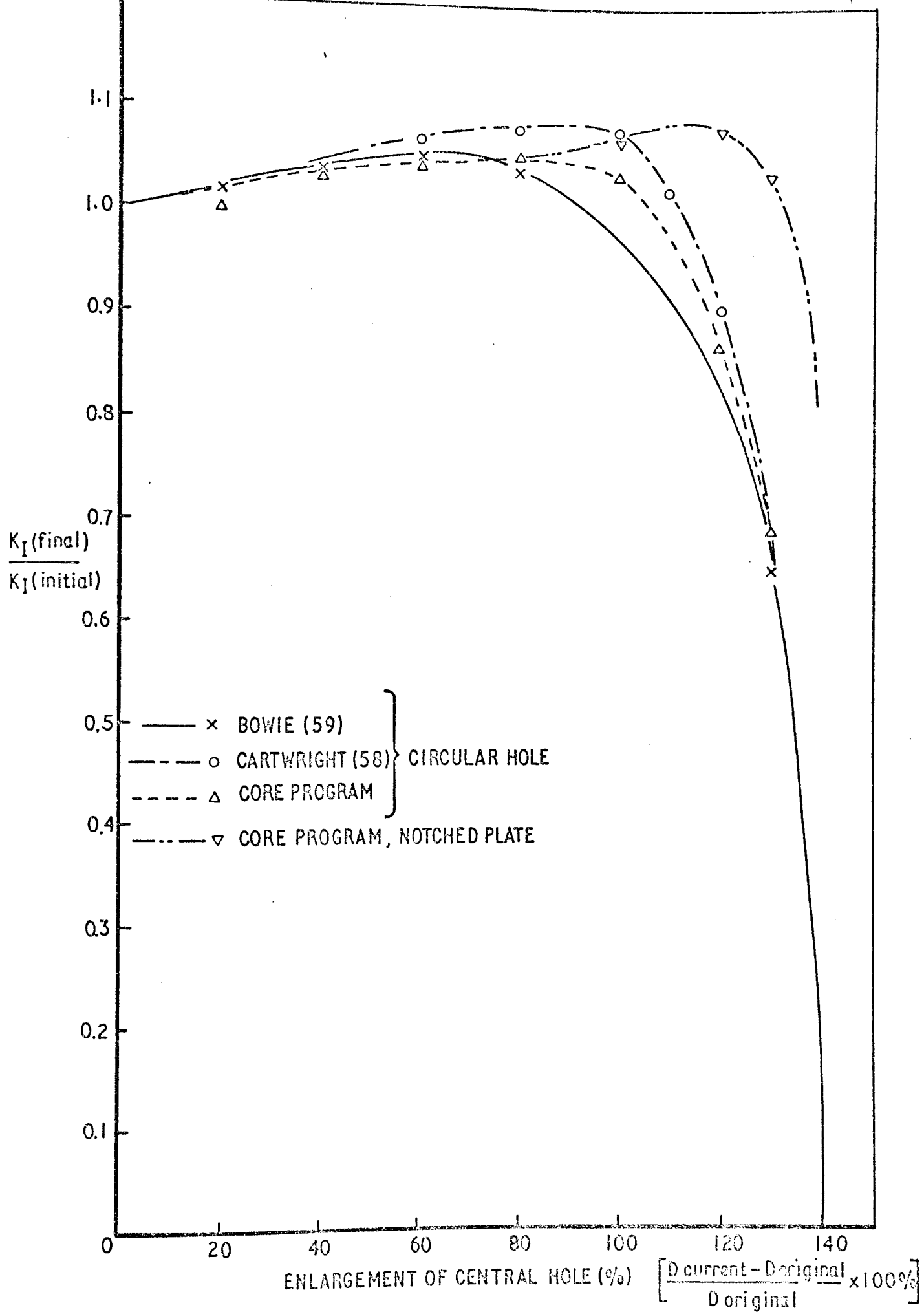


FIG 6.II GRAPH OF $\frac{K_I(\text{initial})}{K_I(\text{final})}$ AGAINST ENLARGEMENT OF CENTRAL HOLE OR NOTCH.

D(ins.)	% enlargement	Bowie ⁽⁵⁹⁾ $\frac{K_{I \text{ final}}}{K_{I \text{ initial}}}$	Cartwright ⁽⁵⁸⁾ $\frac{K_{I \text{ final}}}{K_{I \text{ initial}}}$	Core Program	
				K_I $\text{lb}/\text{in}^2/\sqrt{\text{in}}$	$\frac{K_{I \text{ final}}}{K_{I \text{ initial}}}$
1.0	0	1.0		2.9742	1.0
1.2	20	1.017		3.0208	1.0157
1.4	40	1.037		3.0701	1.0323
1.6	60	1.048	1.07	3.0962	1.0411
1.8	80	1.042	1.08	3.0637	1.0301
2.0	100	0.976	1.075	3.0717	1.0328
2.2	120	0.782	0.9	2.5683	0.8636
2.3	130	0.647	0.675	2.0292	0.6823

Table 6.2 Comparison of Results for Circular Hole

D(ins.)	% enlargement	Core Program	
		K_I $\text{lb}/\text{in}^2/\sqrt{\text{in}}$	$\frac{K_{I \text{ final}}}{K_{I \text{ initial}}}$
1.0	0	2.869	1.0
1.2	20	2.8658	0.999
1.4	40	2.9637	1.033
1.6	60	2.9694	1.035
1.8	80	3.0268	1.055
2.0	100	3.0649	1.068
2.2	120	3.0922	1.078
2.3	130	2.9695	1.034

Table 6.3 Results for Notched Plate.

6.3 PLATE WITH SEVERAL CRACKS IN LINE, SYMMETRIC MODE I

The procedure for several tips was tested for both single and double crack tips, using the data of section 6.2. Identical results were obtained using the multi-tip and single tip procedures, thus establishing the accuracy and range of application of the 'N-tip' program.

Three Collinear Cracks in a Finite Width Plate

The cracked configuration is shown in Fig. 6.12 where a row of three collinear cracks are embedded in a plate subjected to uniaxial tension. Results from the computer program are compared with those of Isida⁽²¹⁾ who solved a similar problem for the case of an infinite plate. The width W , Fig. 6.12, was made approximately twice C and it is considered that this is sufficient to provide a reasonable comparison between the finite and infinite boundary solutions. Limited computer storage capacity places a restriction on the number of degrees of freedom permitted in the idealisation and results are therefore not as accurate as might be hoped for but such loss of accuracy is almost certainly caused by the crude subdivision of the meshes employed.

From symmetry of loads and geometry, only one quarter of the plate need be analysed. The quadrant is shown in Fig. 6.13 with the boundary conditions as specified. Crack tips are labelled as shown. Cores were constructed about each of the tips, each core having different values of interface nodes N_1 and core radii R_c . Input data was prepared for various values of the ratio $2a/d$, with fixed W .

Results of the computer program are shown in Table 6.4 and compared with those of Isida. For the particular case of $2a/d = 0.8$, dimensions of the plate had to be reduced due to excessive storage requirements of the original mesh. It is thought that this change is responsible for the disruption in the pattern of the results for the other $2a/d$ ratios.

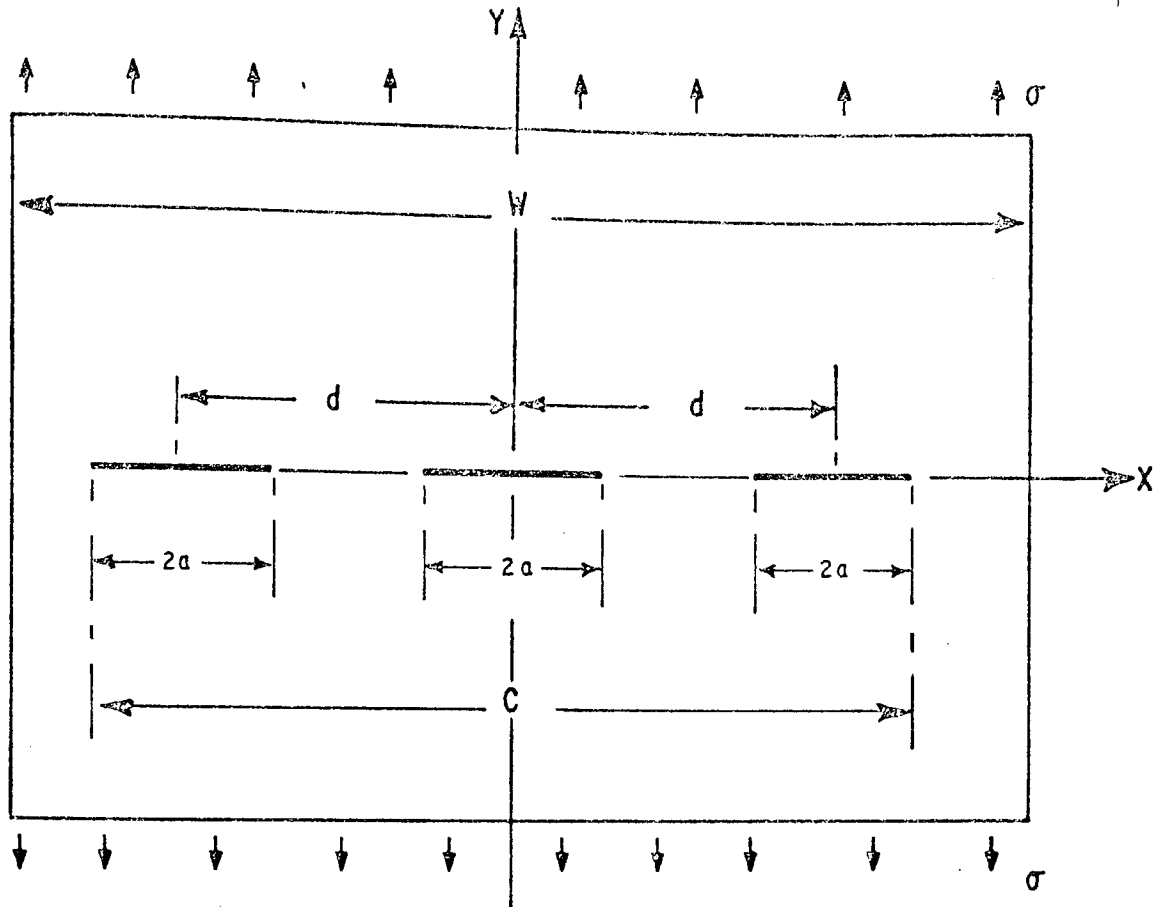


FIG 6.12 RECTANGULAR PLATE CONTAINING THREE COLINEAR CRACKS

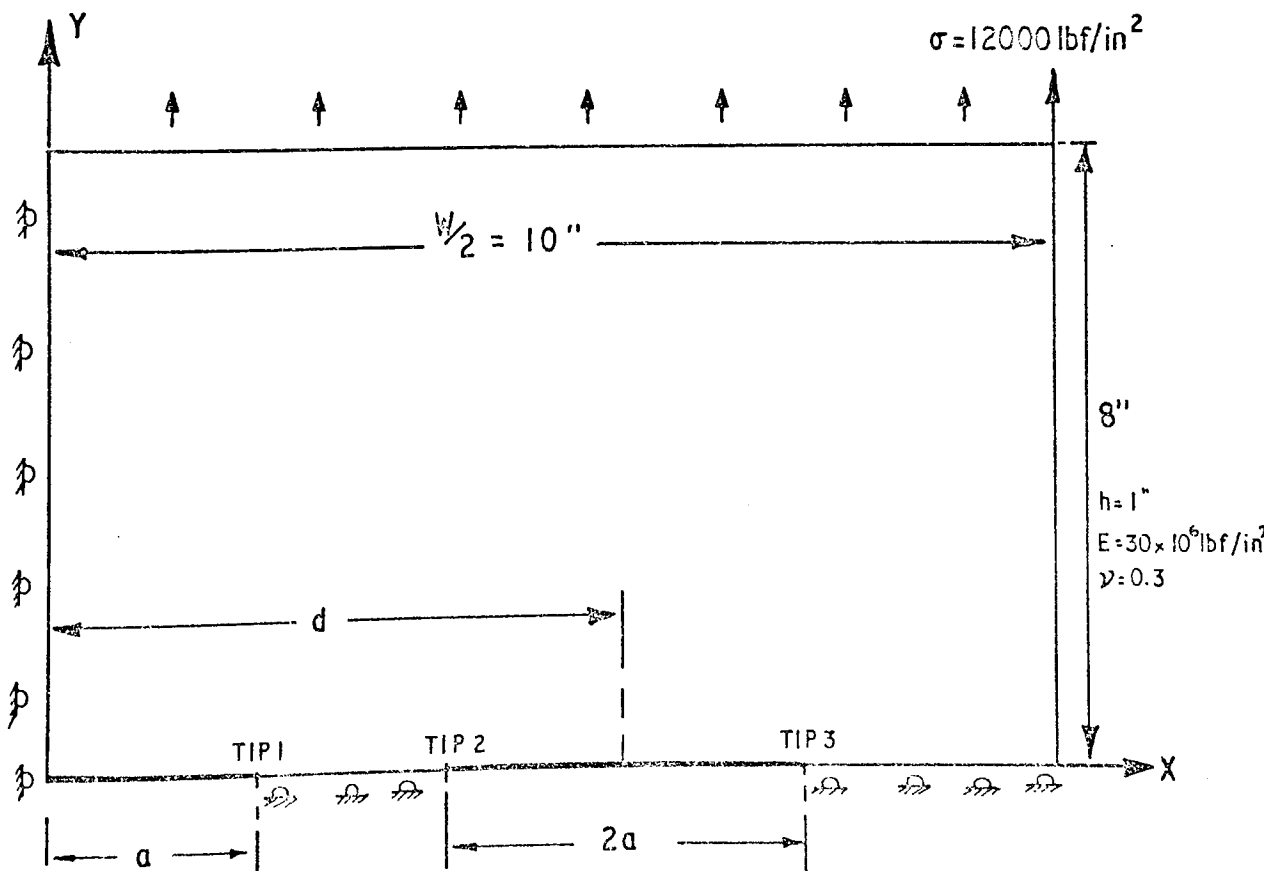


FIG 6.13 QUADRANT OF RECTANGULAR PLATE, THREE CRACK TIPS.

Results of the program are in excellent agreement with those of Isida for tip No. 1, $N1 = 19$, the errors for tips 2 and 3 larger than for tip 1 as a result of the large core radii and the relatively few nodes used on the core/finite element interfaces. The example does, however, illustrate the flexibility of the procedure for general Mode I problems.

Details of tips :

Tip No. 1 $N1_1 = 19$, $RC_1 = 0.05$ $R_{c1}/a = 0.05$
 Tip No. 2 $N1_2 = 15$, $RC_2 = 0.05$ $R_{c2}/a = 0.05$
 Tip No. 3 $N1_3 = 13$, $RC_3 = 0.1$ $R_{c3}/a = 0.01$

$2a/d$	d	No. of Elements	No. of Nodes	$k_I/\sigma \sqrt{a}$ (theory)	$k_I/\sigma \sqrt{a}$ (program)	Percentage Discrepancy
0.5	4.0	129	314	1.0775	(1) 1.024 (2) 0.982 (3) 0.950	(1) 5 (2) 9 (3) 11
0.6	3.333	129	312	1.125	(1) 1.089 (2) 1.064 (3) 0.966	(1) 3 (2) 5 (3) 14
0.7	2.857	131	316	1.19	(1) 1.181 (2) 1.140 (3) 0.996	(1) <1 (2) 4 (3) 16
0.8	2.5	126	303	1.315	(1) 1.3469 (2) 1.3055 (3) 1.0481	(1) 2.4 (2) <1 (3) 20

Table 6.4 Results for Multi-Tip Problem.

6.4 SINGLE TIP, COMBINED MODES I AND II

Curved Crack in a Plate Subjected to Biaxial Tension

The cracked configuration is shown in Fig. 6.14 and consists of an arcular crack of radius R , located symmetrically with respect to a uniform biaxial tension stress field in an infinite plate. The theoretical solution to this problem was originally obtained by Muskhelishvili⁽²⁷⁾ and the results presented here are taken from the Compendium of Stress Intensity Factors by Sih⁽²³⁾. In the finite element idealisation the plate dimensions are large compared with crack length and it is considered therefore that there is a reasonable comparison between the infinite and finite plate results. This example also serves as a further illustration of the unique facility of the isoparametric element for modelling curved element boundaries.

From symmetry of the loading and plate geometry about the x -axis, only one half of the plate need be analysed. The upper half of the plate is shown in Fig. 6.15, with the boundary conditions specified. A full circular core of radius R_c and with $N1$ interface nodes was constructed around the tip, the remainder of the plate being sub-divided in the normal manner with a reasonably fine mesh surrounding the plate boundaries to model the distributed load.

The theoretical solution of Ref. (27) may be written :

$$K_I = \frac{\sigma \sqrt{R}}{(1 + \sin^2 \frac{\alpha}{2})} \left[\frac{\sin \alpha (1 + \cos \alpha)}{2} \right]^{\frac{1}{2}} \quad (6.1)$$

$$K_{II} = \frac{\sigma \sqrt{R}}{(1 + \sin^2 \frac{\alpha}{2})} \left[\frac{\sin \alpha (1 - \cos \alpha)}{2} \right]^{\frac{1}{2}}$$

Table 6.5 shows a comparison of results obtained from the finite element program with those obtained from equation 6.1 for various values of the angle, α . Results show consistent and acceptable accuracy throughout the range of α 's considered, thus providing further confirmation of the suitability of the technique for complex cracked configurations

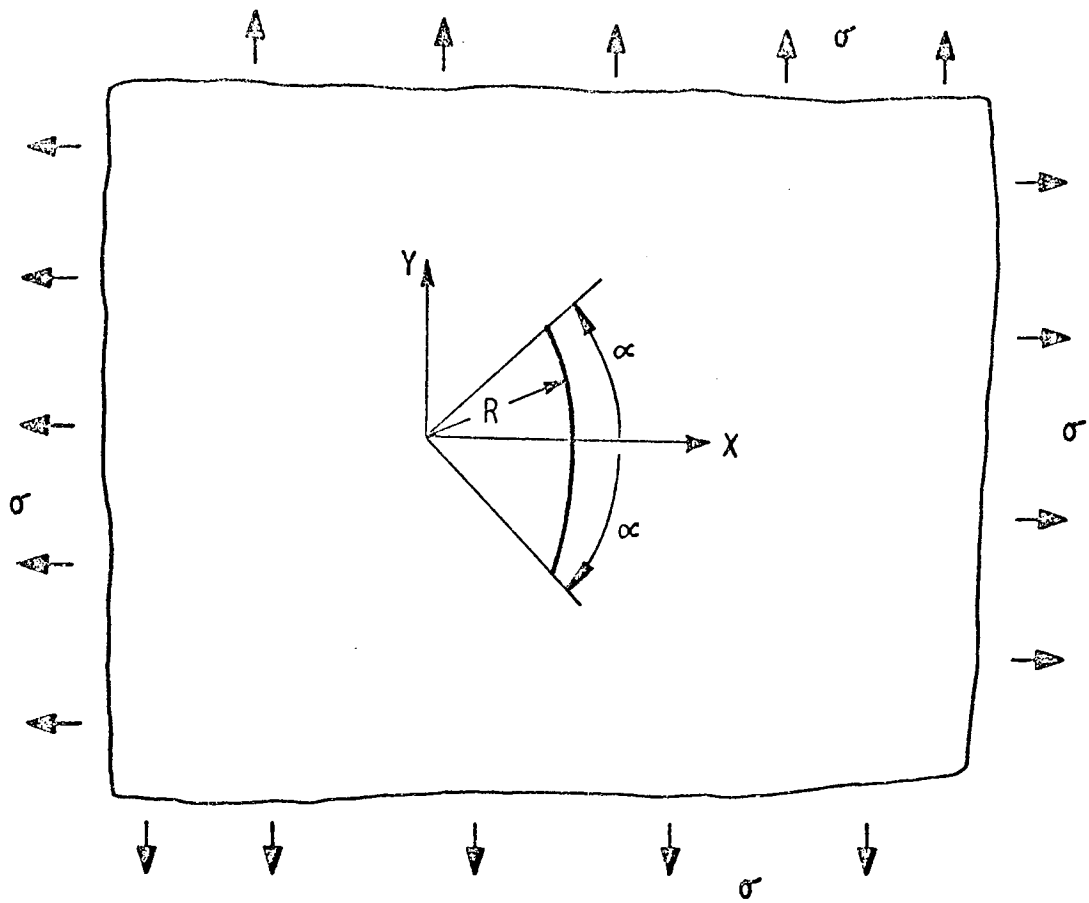


FIG 6.14 CIRCULAR ARC CRACK PLACED SYMMETRICALLY WITH RESPECT TO A UNIFORM BIAXIAL TENSION.

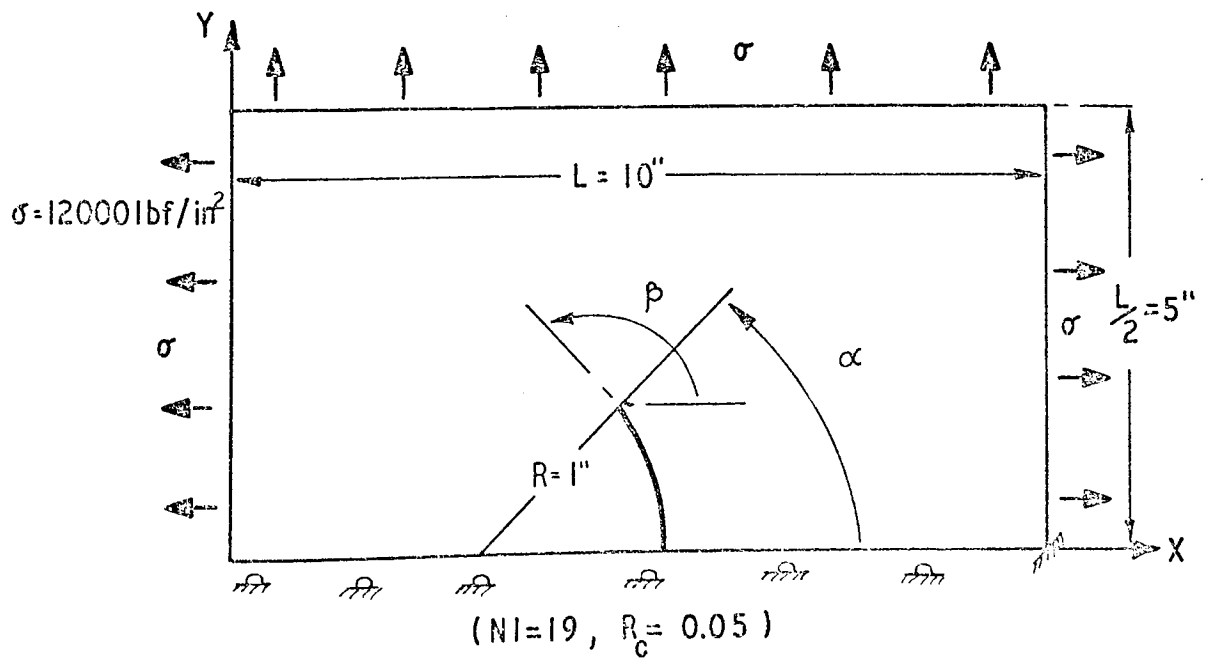


FIG 6.15 ARCULAR CRACK IN A RECTANGULAR PLATE.

α (Deg.)	No. of Elements	No. of Nodes	Theory		Experiment		Percentage Discrepancies	
			$K_I/\sigma\sqrt{R}$	$K_{II}/\sigma\sqrt{R}$	$K_I/\sigma\sqrt{R}$	$K_{II}/\sigma\sqrt{R}$	$K_I/\sigma\sqrt{R}$	$K_{II}/\sigma\sqrt{R}$
30	80	198	0.6401	0.1715	0.5879	0.1663	8	3
45	88	213	0.6776	0.2807	0.6782	0.2727	0	3
60	86	209	0.6447	0.3722	0.6320	0.373	2	0
90	92	223	0.4714	0.4714	0.4585	0.4685	2	0

Table 6.5 Comparison of Results for Circular Crack.

Oblique Edge Crack in a Rectangular Panel

The cracked configuration is shown in Fig. 6.16 and consists of a straight inclined crack of length a , in a rectangular plate subjected to uniaxial tension. The theoretical solution for this finite width case was obtained by Isida⁽²¹⁾.

A full circular core of radius R_c with $N1$ interface nodes was constructed about the crack tip, the remainder of the plate being subdivided in the normal manner. The plate is constrained by fixing point P in both coordinate directions and restraining point C in the Y -direction Fig. 6.16.

Results of the finite element analysis are compared with those of Isida in Table 7.6, Figs. 6.17 and 6.18, for a range of values of λ . For the cases treated, results are both consistent and accurate.

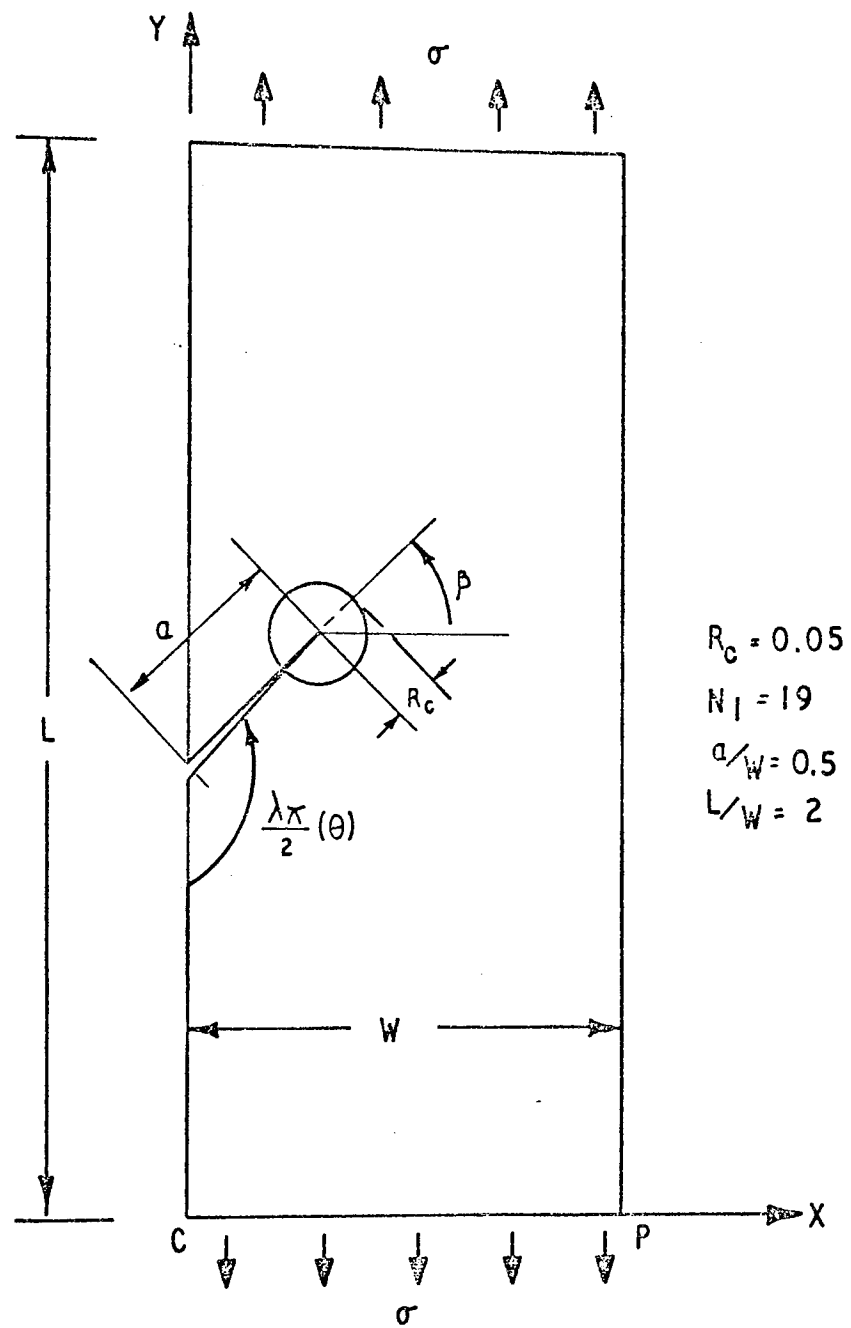


FIG 6.16 OBLIQUE EDGE CRACK IN A RECTANGULAR PLATE.

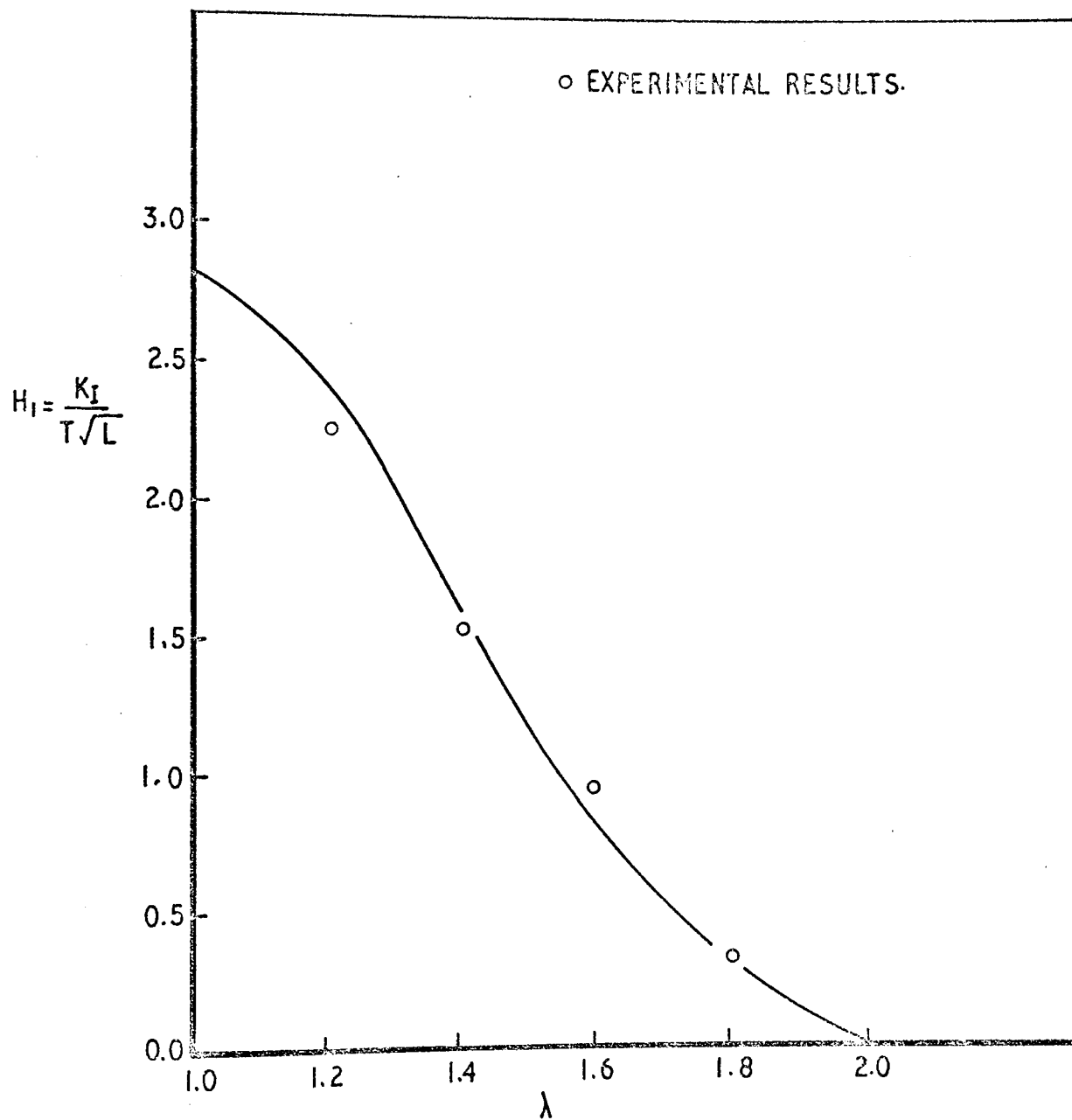


FIG 6.17 GRAPH OF H_I AGAINST λ , OBLIQUE EDGE CRACK.

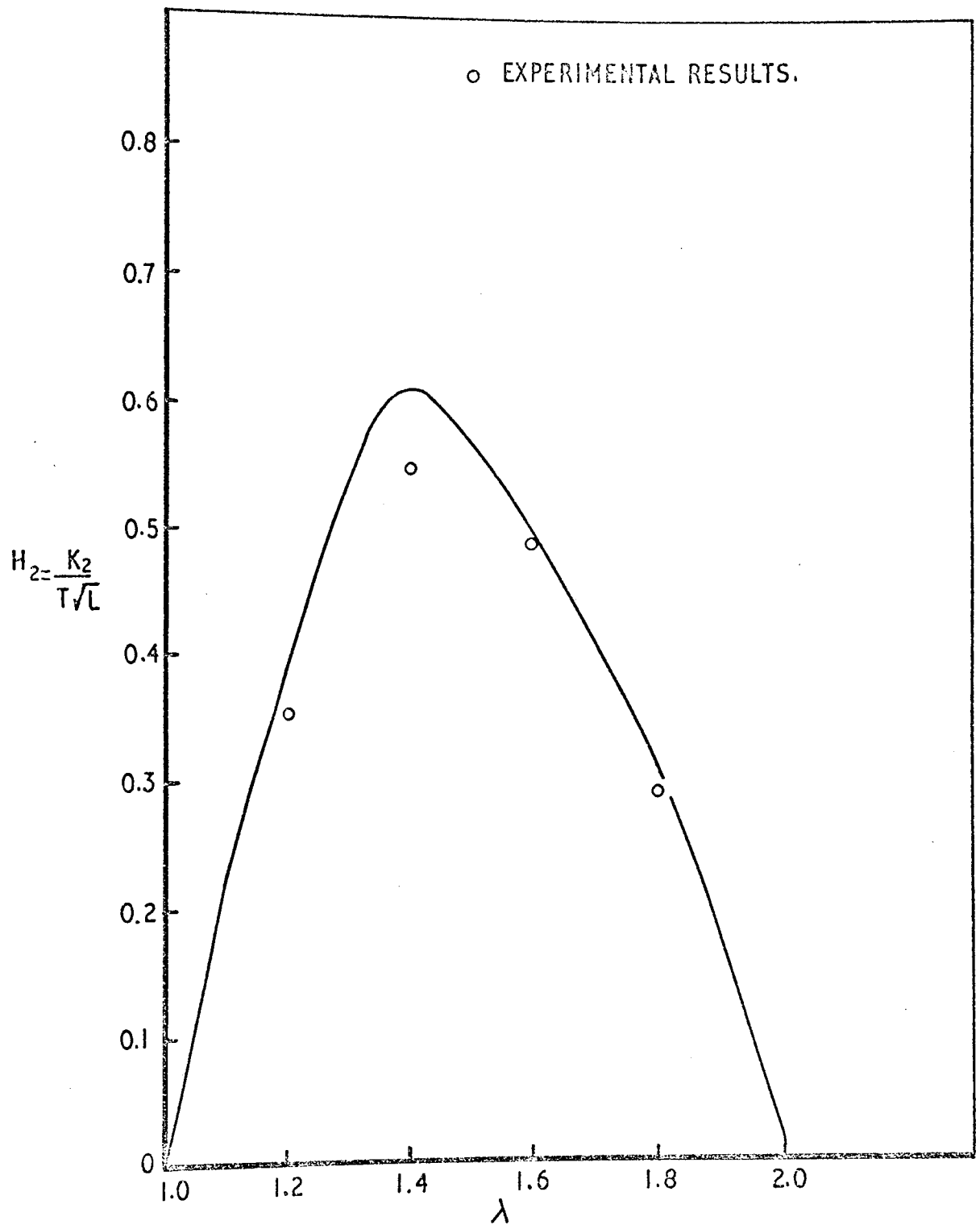


FIG 6.18 GRAPH OF H_2 AGAINST λ , OBLIQUE EDGE CRACK.

λ	Θ (Deg.)	No. of Elements	No. of Nodes	Theory		Experiment	
				$H_I = K_I / \sigma \sqrt{a}$	$H_2 = K_{II} / \sigma \sqrt{a}$	$H_I = K_I / \sigma \sqrt{a}$	$H_2 = K_{II} / \sigma \sqrt{a}$
1.2	108	72	179	2.42	0.366	2.263	0.354
1.4	126	70	175	1.60	0.61	1.534	0.547
1.6	144	72	179	0.86	0.492	0.834	0.485
1.8	162	73	182	0.35	0.308	0.333	0.291

Table 6.6 Comparison of Results for Oblique Edge Crack.

6.5 DOUBLE TIP, COMBINED MODES I AND II

Of the few solutions available in the literature, the one chosen here to demonstrate the application and accuracy of the mixed mode double tip program is shown in Fig. 6.19. A rectangular plate of dimensions $2b \times 2c$ is subjected to a uniaxial tension and contains an inclined straight crack. The theoretical solution for the infinite plate is taken from Rice⁽⁹⁾, where, for tips 1 and 2 :

$$\begin{aligned} K_I &= \sigma \sqrt{a} \sin^2 \alpha \\ K_{II} &= \sigma \sqrt{a} \sin \alpha \cos \alpha \end{aligned} \quad (6.2)$$

In the finite element program, finite width boundaries are employed of such dimensions as to ensure a reasonable comparison of results. In the final part of this work the plate width is reduced to assess the influence of the boundaries on the stress intensity factors.

Dimensions of the plate are shown in Fig. 6.19. In a similar manner to the Mode I single tip examples of section 6.2, a range of cores with varying radii and number of interface nodes were constructed around the crack tips in an effort to assess accuracy and stability of the solution.

Results are presented in tabular form in Table 6.7 and graphically in Figs. 6.20 and 6.21. In each case the value of $\alpha = 45^\circ$ was used, giving, according to equation 6.2,

$$K_I / \sigma \sqrt{a} = K_{II} / \sigma \sqrt{a} = 0.5$$

Fig. 6.20 demonstrates that the actual value of core radius, within the range chosen, has very little influence on the accuracy of the final solution. The figure also gives some indication as to the convergence of the solution with an increase in the number of core/finite element interface nodes, N_1 . In the case of $K_{II} / \sigma \sqrt{a}$ the results exhibit good convergence to the theoretical solution, but for $K_I / \sigma \sqrt{a}$ spurious results are obtained for $N_1 = 13$. A similar situation was observed in the Mode I

study of section 6.2 where it was concluded that although convergence is expected in the limit, $N_1 = \infty$, there is a complicated interaction of the variables N_1 and R_c which influences the core strain energy and the final solution for intermediate values of these variables.

Fig. 6.21 shows a comparison of results obtained from crack tips 1 and 2 for the particular case, $R_c = 0.1$ ins, other core radii show similar trends in their results. In general, results show good agreement, the maximum difference between any two being of the order of 2%.

In Fig. 6.22 values of $K_I/\sigma\sqrt{a}$ and $K_{II}/\sigma\sqrt{a}$ are shown plotted as a function of the plate width $2b$, Fig. 6.19. The graphs show clearly the effect of nearness of plate boundaries on the stress intensity factors and also demonstrate the adequacy of $2b = 15$ inches used in the comparison of the finite and infinite width solutions.

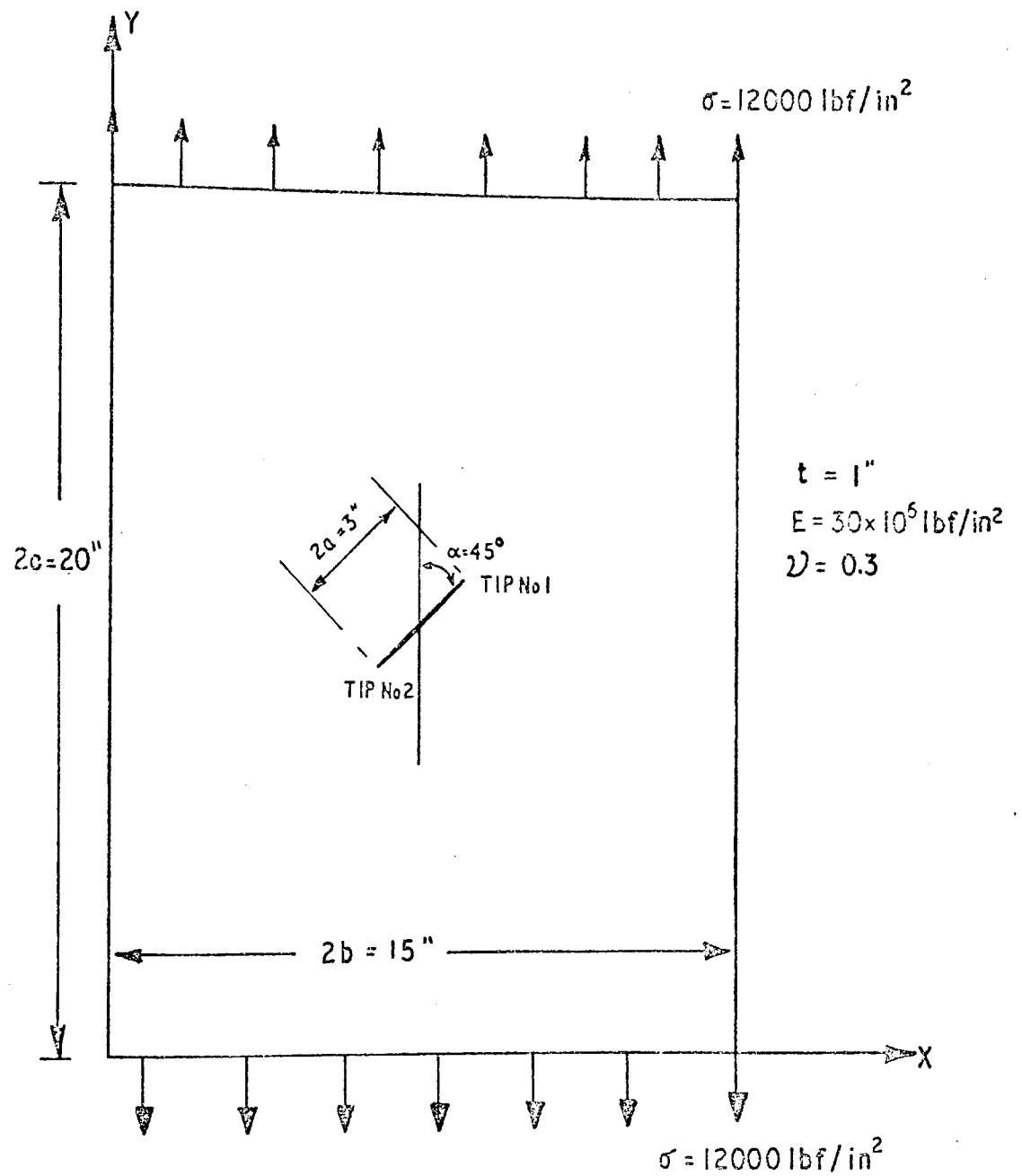


FIG 6.19 STRAIGHT INCLINED CRACK IN A RECTANGULAR PLATE

RC (ins.)	N1 = 19 Nodes (188 Elements, 432 Nodes)						N1 = 17 Nodes (158 elements, 370 Nodes)						N1 = 15 Nodes (144 Elements, 342 Nodes)						N1 = 13 Nodes (142 Elements, 334 Nodes)					
	K _{I1}	K _{II1}	K _{I2}	K _{II2}	K _{I1}	K _{II1}	K _{I2}	K _{II2}	K _{I1}	K _{II1}	K _{I2}	K _{II2}	K _{I1}	K _{II1}	K _{I2}	K _{II2}	K _{I1}	K _{II1}	K _{I2}	K _{II2}	K _{I1}	K _{II1}	K _{I2}	K _{II2}
0.2	0.5016	0.4992	0.503	0.4984	0.4871	0.488	0.4865	0.4866	0.4617	0.4795	0.471	0.48	0.5079	0.4825	0.4931	0.4735								
0.1	0.5036	0.5009	0.5052	0.5001	0.4885	0.4875	0.4881	0.4867	0.4632	0.4806	0.4732	0.4811	0.5099	0.4816	0.494	0.4726								
0.05	0.5	0.5001	0.5015	0.4993	0.4824	0.4851	0.4816	0.4842	0.457	0.479	0.4674	0.4795	0.5004	0.4788	0.4839	0.4697								

Table 6.7 Results for Straight Inclined Crack

(Subscripts 1 and 2 refer to tips 1 and 2, respectively).

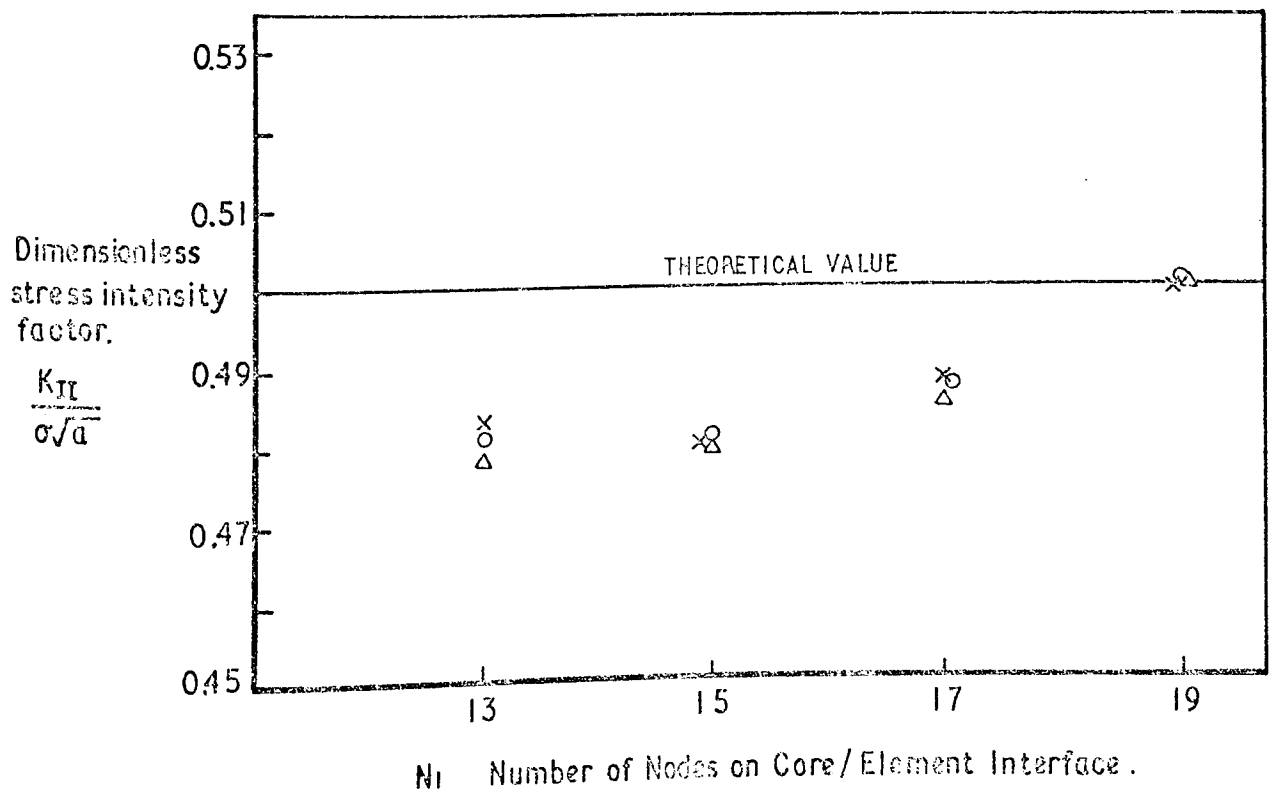
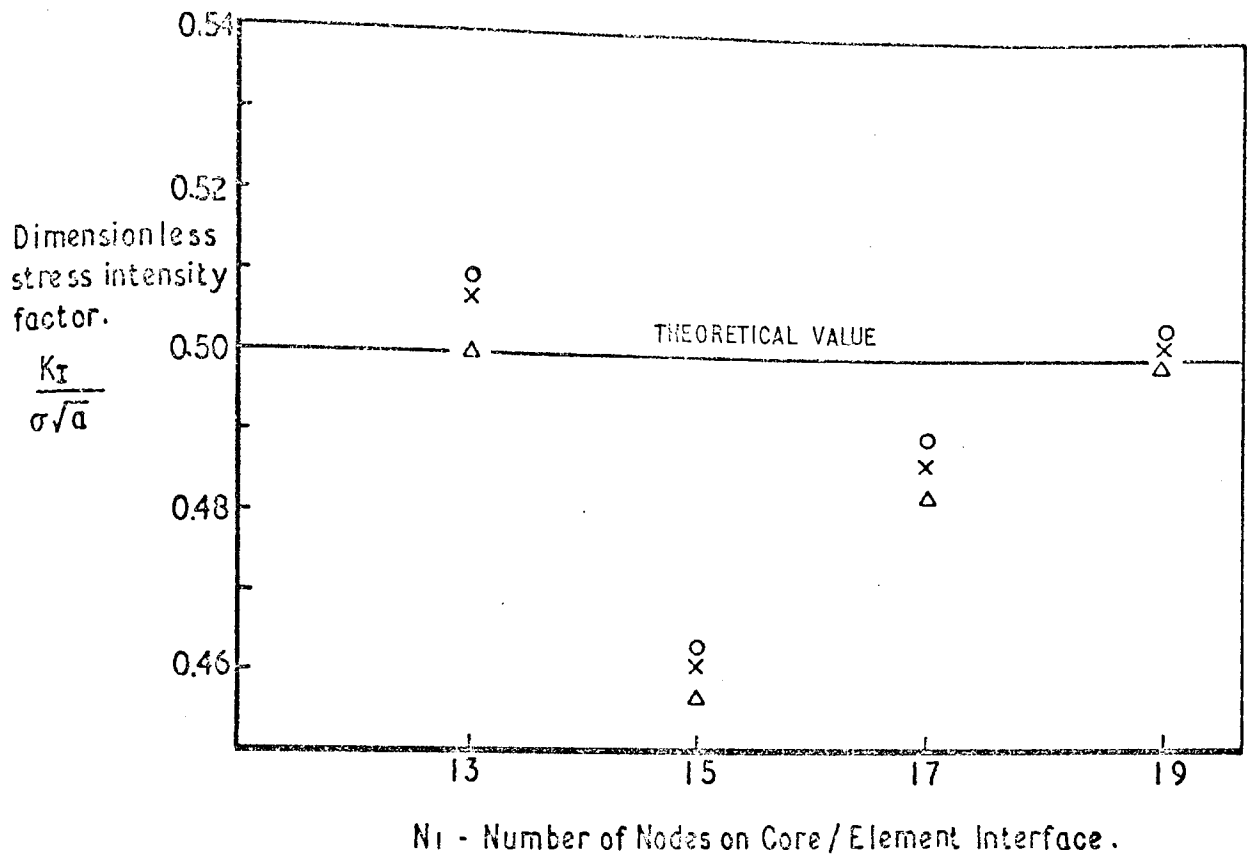
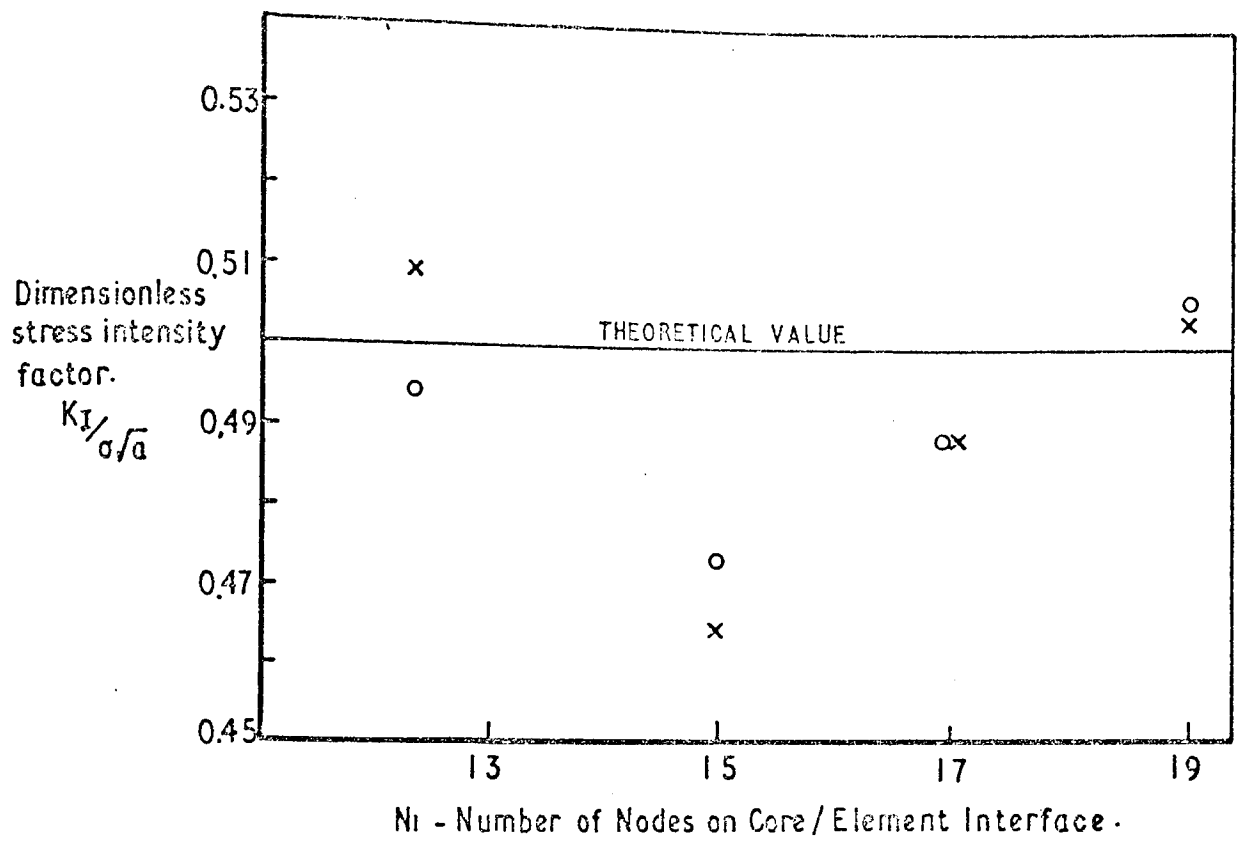


FIG. 6.20 GRAPH OF $\frac{K_I}{\sigma\sqrt{a}}$ AND $\frac{K_{II}}{\sigma\sqrt{a}}$ AGAINST N_I .



x Tip No 1
 o Tip No 2
 $R_c = 0.1 \text{ in}$

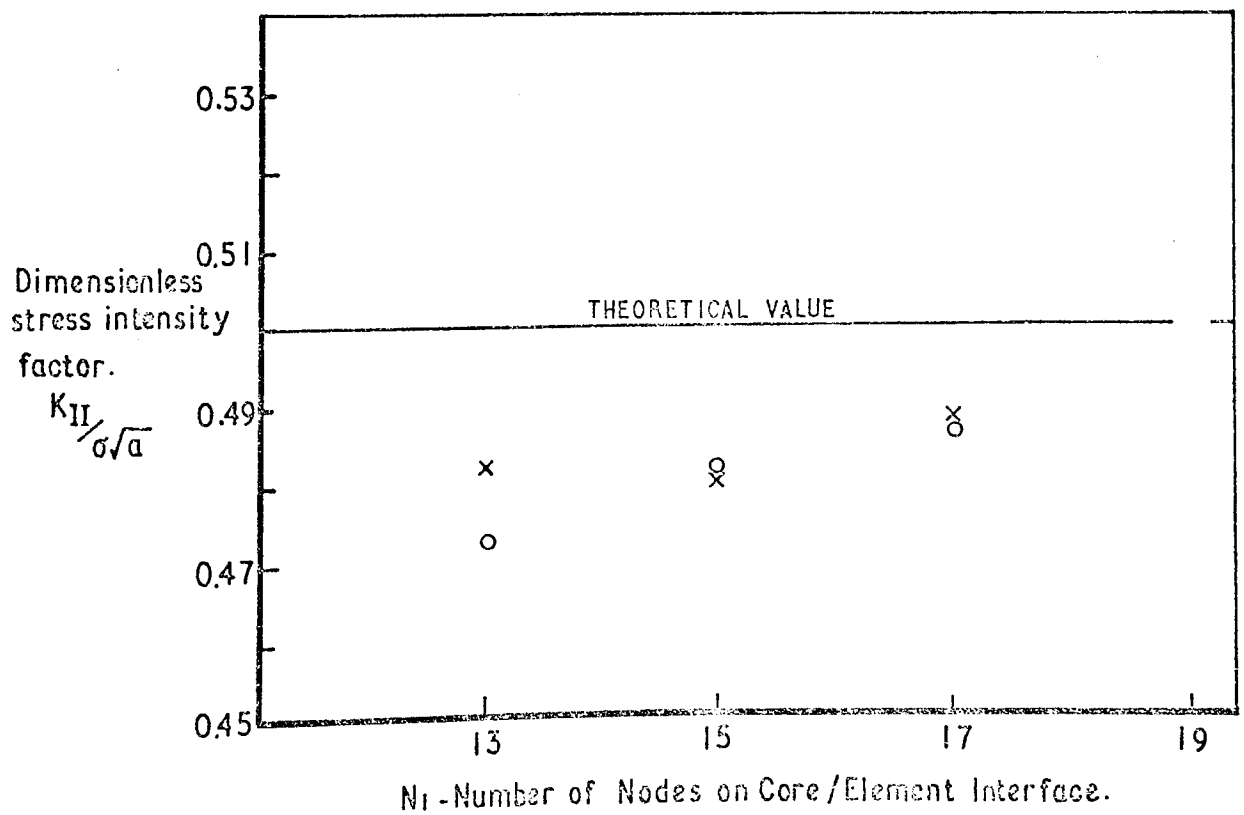


FIG. 6.21 GRAPH OF $K_I/\sigma\sqrt{a}$ AND $K_{II}/\sigma\sqrt{a}$ AGAINST N_I .

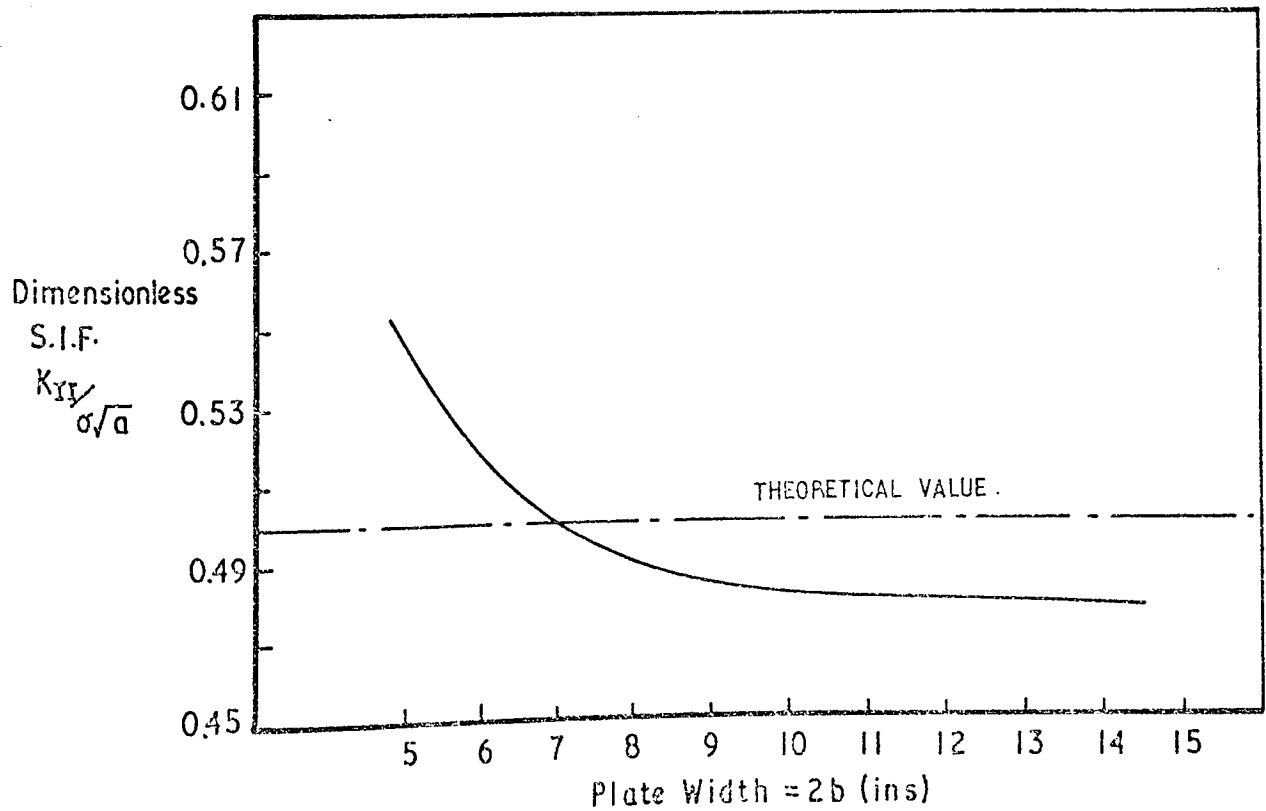
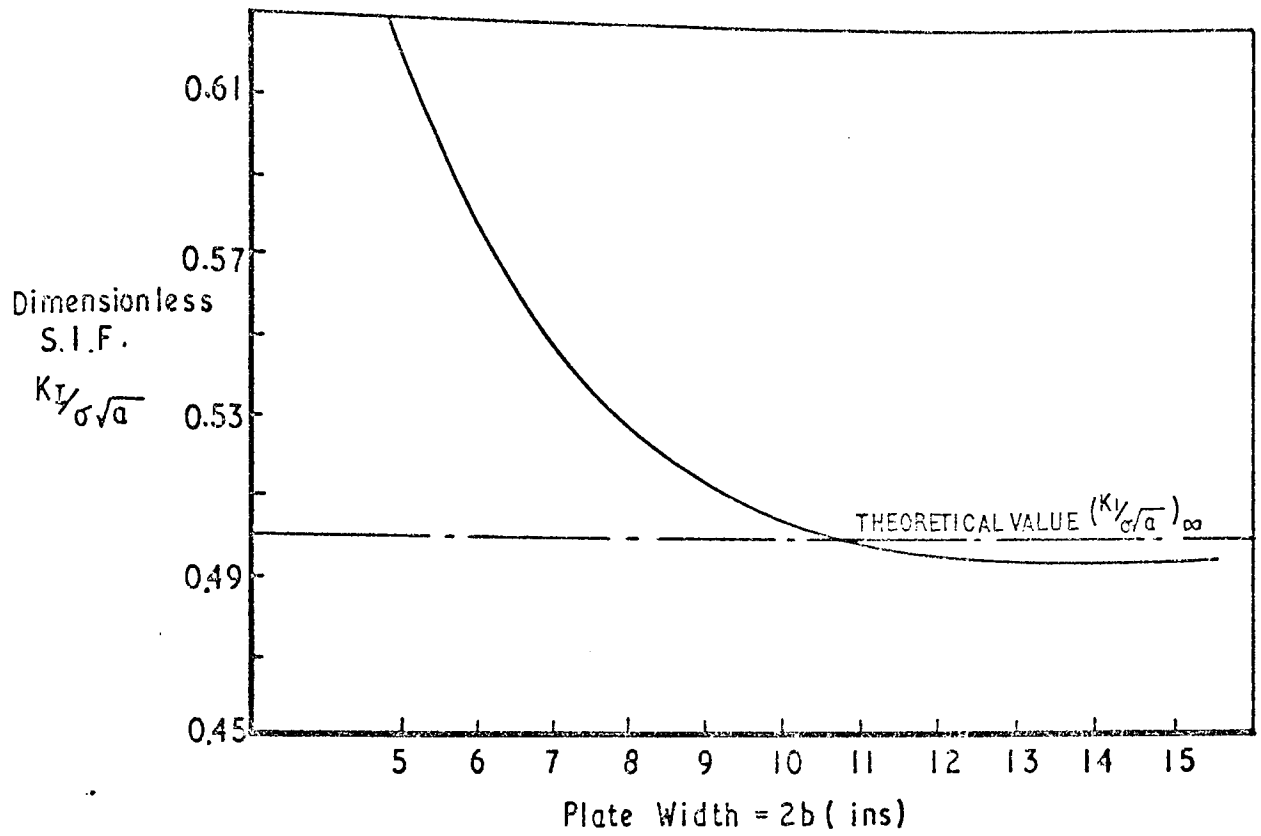


FIG.6.22 GRAPH OF $K_I/\sigma\sqrt{a}$ AND $K_{II}/\sigma\sqrt{a}$ AGAINST PLATE WIDTH, $2b$.

7. DISCUSSION AND SUGGESTIONS FOR FURTHER WORK

The principles of linear elastic fracture mechanics are now well established and the concepts find increasing application in the design of engineering components against brittle fracture. An essential step in applying these concepts to the calculation of critical stress levels or critical defect sizes is the accurate determination of stress intensity factors at complex structural details. Since exact solutions are limited to simple configurations, approximate numerical methods must be utilised. However, such approximate procedures must first be verified on many simple configurations for which comparison with established results may be made before they can be adopted with confidence for the analysis of complex structural details containing flaws. To this end, the technique of Hilton and Hutchinson has been applied here to the calculation of Mode I and Mode II stress intensity factors for a wide variety of cracked component shapes.

The technique of Hilton and Hutchinson is based on a standard finite element code, where a special element whose shape function contains the crack tip singularity and which surrounds the crack tip is embedded in the finite element mesh. The values of the crack tip stress intensity factors are obtained directly as output from the program, together with element stresses and nodal point displacements.

One obvious advantage of such an approach is that it combines the analytical solution, which is accurate near the crack tip but not remote from it, with the finite element solution which is accurate remote from the crack tip but not close to it. Another important advantage, common to any procedure based on the finite element method, is the ease with which complex geometric shapes can be accurately modelled with a suitable mesh subdivision. Also, by using the special crack tip element the need for extremely fine meshes in the crack tip region is avoided and hence the

method is efficient in terms of computer time. The actual amount of computing time will vary from problem to problem depending on the number of nodes present after the subdivision, but as an example the straight inclined crack in a rectangular sheet of Chapter 6.4 with 73 elements and 182 nodes took 12 seconds for compilation and execution on a CDC 7600 computer.

There are several alternative crack tip elements available for the determination of stress intensity factors, but these generally require the calculation of new element stiffness matrices and possibly major alterations to a standard finite element program. In the case of the Hilton and Hutchinson element, however, only minor alterations are required to a finite element program in order to include the crack tip element (core element). Such changes as are necessary are made only to the assembled stiffness matrix and with careful node numbering around the core element, such matrix manipulations are kept to a minimum. In the program described in this thesis, the overall stiffness matrix, $[K]$ has been stored as a one-dimensional array and hence the program steps necessary to modify $[K]$ have been slightly more involved in comparison to a lower triangular or banded storage scheme. However, the matrix operations necessary to modify $[K]$ have been derived for the general case where the full stiffness matrix is stored and hence they can easily be altered for any particular storage scheme.

The primary objective of this thesis was to develop a general program capable of analysing any two-dimensional structure containing any number of cracks. In the attempt to achieve this objective, several problem areas were identified and solved either partially or fully. These areas are discussed in detail below.

The first problem was that of crack surface closure, present in the mixed mode, K_I and K_{II} case and to a lesser extent in the single mode, K_I case. Conceptually, the phenomenon of closure is simple, involving closure of the crack surfaces over part or all of their length due to a

particular combination of applied loads and crack position. The analytical difficulty arises because of displacement incompatibility at crack surface nodes after a normal finite element solution. In a normal application of the finite element procedure no restriction is placed on the movement of crack surface nodes and the minimum potential energy configuration then corresponds to the physically unrealistic situation of two adjacent surfaces crossing. The solution proposed in this thesis is discussed in detail in Chapter 5 and consists essentially of prescribing displacements to crack surface nodes at which compatibility has been violated. This procedure has been tested for the Mode I case of a centre cracked beam in bending and results were compared with the theoretical solution of Savin⁽⁶⁵⁾. Initial results were encouraging but a great deal of research remains to be done, particularly for the mixed mode case where frictional resistance to crack surfaces sliding over each other should be included in the mathematical model of closure. As suitable theoretical solutions are not available which allow for this phenomenon in mixed mode fracture problems, an experimental study must be undertaken.

The second significant problem area concerned the rigid body displacements of the crack tip element. In the Mode I case, where the crack plane is normal to the y-axis, the only rigid body displacement is the displacement of the crack tip core element in the x-direction. For the mixed Mode I and II case, however, there is no symmetry about the crack axis and the crack tip core element may displace in the x and y directions, and also rotate in the x, y- plane. As well as giving erroneous results which could easily have been caused by some unrelated programming error, the omission of any rigid body displacement gives rise to spurious residual forces at the constrained nodes. A special procedure was written to check these residual forces and details of this procedure may be found in Chapter 10.D.3.

Finally, with regard to the mixed mode two-tip program of Chapter 5.5 a solution to the problem of an arcular crack in a uniaxial tensile stress field was attempted. Fig. 7.1 shows the crack position and loading configuration for this problem to which Paris and Sih⁽³⁾ calculated the analytical solution as,

for the crack tip at 0

$$K_I = \frac{\sigma(R \sin \alpha)^{\frac{1}{2}}}{2(1 + \sin^2 \frac{\alpha}{2})} \left\{ \cos \frac{\alpha}{2} + \cos (2\beta + \frac{5}{2}\alpha) \left[\sin^2 \frac{\alpha}{2} \right] - \cos (2\beta + \frac{3}{2}\alpha) \right. \\ \left. \left[\cos^2 \frac{\alpha}{2} - \sin^4 \frac{\alpha}{2} \right] - \sin (2\beta + \frac{3}{2}\alpha) \left[\sin \alpha \sin^2 \frac{\alpha}{2} \right] \right\}$$

$$K_{II} = \frac{\sigma(R \sin \alpha)^{\frac{1}{2}}}{2(1 + \sin^2 \frac{\alpha}{2})} \left\{ \sin \frac{\alpha}{2} + \sin (2\beta + \frac{5}{2}\alpha) \left[\sin^2 \frac{\alpha}{2} \right] + \sin (2\beta + \frac{3}{2}\alpha) \right. \\ \left. \left[\cos^2 \frac{\alpha}{2} - \sin^4 \frac{\alpha}{2} \right] - \cos (2\beta + \frac{3}{2}\alpha) \left[\sin \alpha \sin^2 \frac{\alpha}{2} \right] \right\}$$

Of the several crack positions analysed using the computer program of Chapter 10.D.4, the particular geometry considered in the greatest detail is shown in Fig. 7.2. The results from the program are compared with those of Paris and Sih in Table 7.1. Considerable efforts were made to understand the reason for the discrepancy between these two sets of results, unfortunately without success. In a recent paper by Hussain and Pu⁽⁶⁶⁾ the symmetric problem of Fig. 7.2, was considered, and their results are also shown in Table 7.1. The results of Hussain and Pu seem to support the finite element results, unfortunately Hussain and Pu only considered symmetric configurations, similar to that shown in Fig. 7.2, and hence no general conclusions can be drawn.

Two possible courses of action are suggested in order to resolve this problem :

- (i) to check the results of Refs. (3) and (66) by working from the original paper of Muskhelishvili⁽²⁷⁾,
- or (ii) to carry out an experimental investigation with a suitably cracked plate.

For completeness, the results of all the computational trials carried out are shown in Table 7.2.

There still remain several further refinements to be made to the program before the original objective of a general two-dimensional fracture package is achieved. Nevertheless, the technique of Hilton and Hutchinson has been developed such that a wide variety of cracked structures, not hitherto amenable to solution by this method, can now be analysed. Also, en route to the general program, several important aspects of the method have been identified and for the most part satisfactorily treated. In the light of the experience gained from this work, the following suggestions are made as possible areas for future research :

- (a) To develop a mesh generation scheme for the cracked structure.

One of the most time-consuming and error prone aspects of any finite element analysis lies in the subdivision of the continuum and in the preparation of input data. In order to overcome this problem a special program may be written which automatically subdivides the continuum and prepares input data for the subsequent finite element program. With the special crack tip element described in this thesis, the mesh generation problem is partially solved by the requirements (Chapter 5) placed on node numbering for nodes surrounding the core element. This could well be turned to advantage for multi-tip cracked structures where near crack tip meshes could be automatically generated from a knowledge of core radius (R_c) and number of core/finite element interface nodes (N_1).

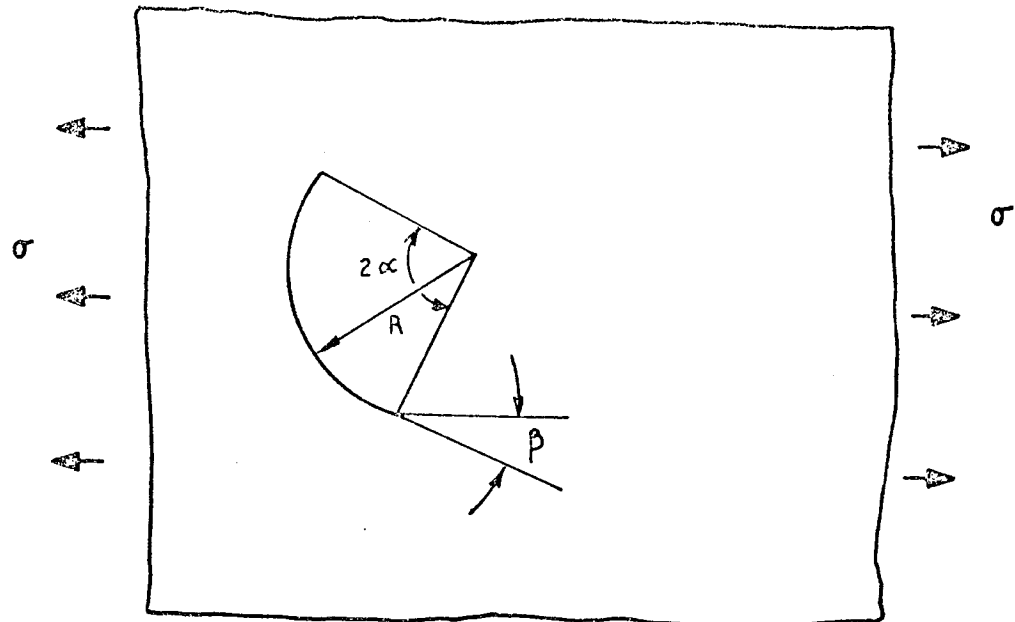


FIG.7.1 CURVED CRACK IN A UNIAXIAL STRESS FIELD.

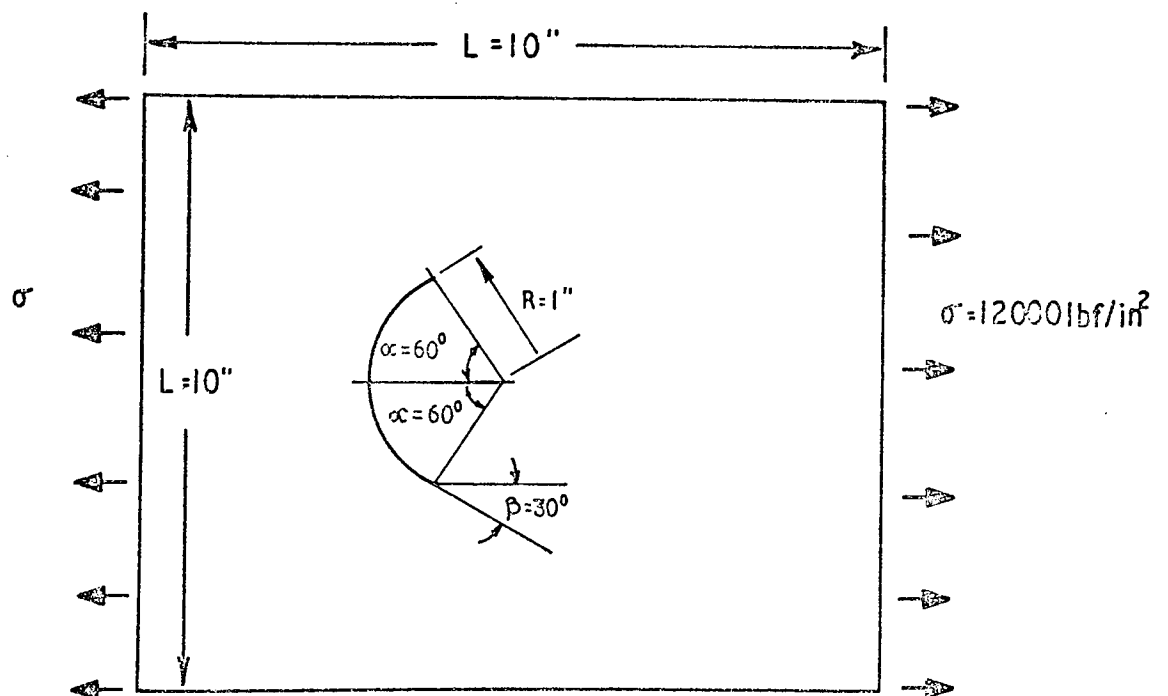


FIG.7.2 CURVED CRACK PLACED SYMMETRICALLY WITH RESPECT TO THE STRESS FIELD.

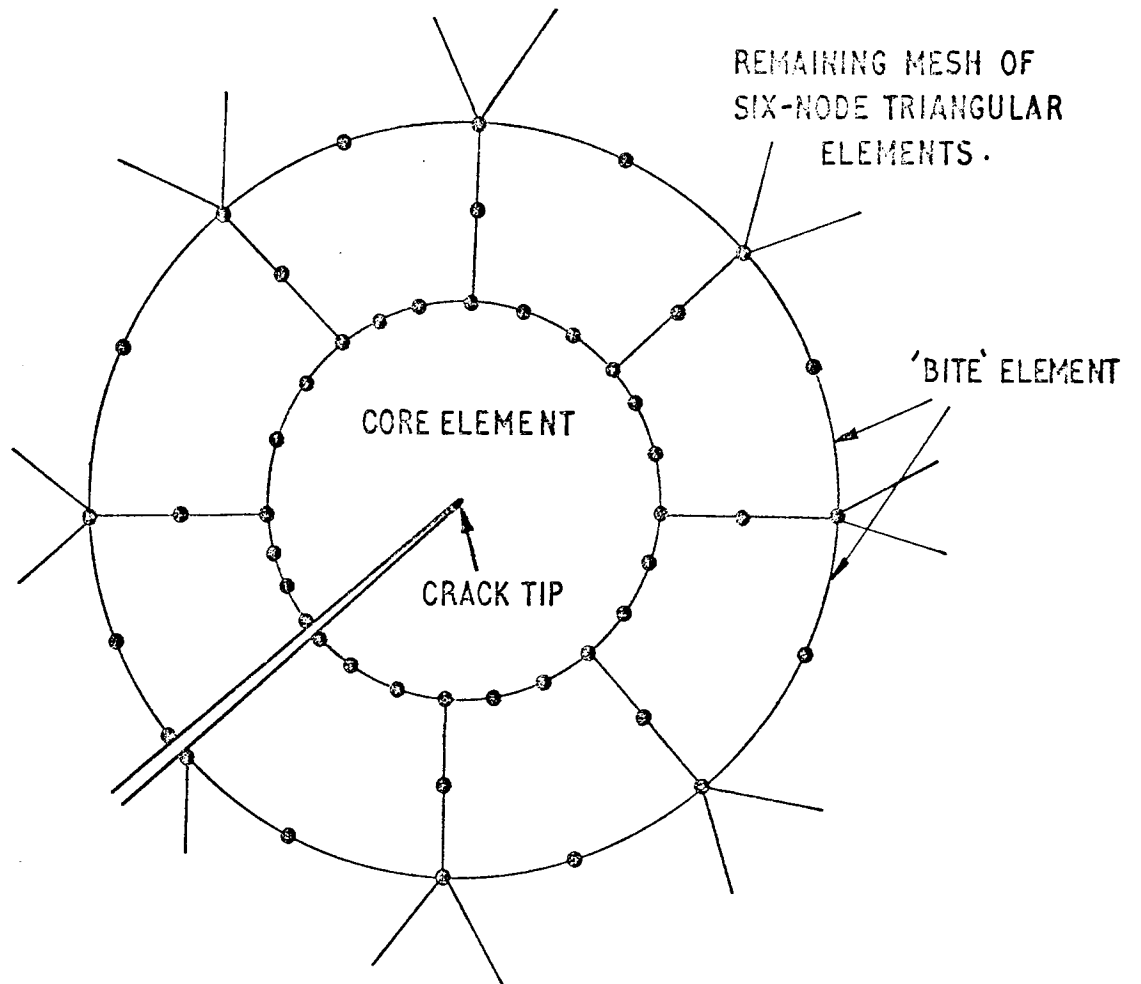


FIG 7.3 PROPOSED MESH SUBDIVISION IN CRACK TIP REGION .

	F.E. Program	Paris and Sih Ref. (3)	Hussain and Pu Ref. (66)
$K_I/\sigma\sqrt{R}$	0.373	0.4231	0.2619
$K_{II}/\sigma\sqrt{R}$	0.6795	0.3373	0.6165

Table 7.1 - Comparison of Results for the Symmetric Crack Problem of Fig. 7.2

$(\alpha = 60^\circ)$ θ	Finite Element Results		Paris and Sih Ref. (3)	
	$K_I/\sigma\sqrt{R}$	$K_{II}/\sigma\sqrt{R}$	$K_I/\sigma\sqrt{R}$	$K_{II}/\sigma\sqrt{R}$
90	0.8445	0.1484	0.4836	-0.1163
60	0.8196	0.3129	0.5843	0.0349
30	0.373	0.6795	0.4231	0.3373
0	0.0	0.5685	0.1612	0.4886
-30	0.0	0.0638	0.0604	0.3373
-60	0.3722	0.2925	0.2216	0.0349
-90	0.8384	0.1439	0.4836	-0.1163

Table 7.2 - Results of All Computational Experiments Carried Out with the Plate Dimensions of Fig. 7.2.

As a complement to the mesh generation programs described above, the input data may be reduced by developing isoparametric elements of the type shown in Fig. 7.3 for surrounding the core element. The value of isoparametric elements in regions of high stress gradients has been recognised previously by Tracey⁽⁴⁸⁾ and elements similar to those shown in Fig. 7.3 have been developed by Bond et al⁽⁶⁷⁾, specifically for problems of stress concentration.

- (b) To prepare a compendium of stress intensity factors for a variety of cracked structures.

Such a compendium could well take a form similar to that of Peterson's book⁽⁶⁸⁾ on Stress Concentration Factors where several standard engineering component shapes are analysed. Also, there are many important practical cracked structures which could be analysed using the embedded singular element described in this thesis. One such area concerns the fracture of gears, where residual stress fields caused by surface hardening as well as stresses induced by the applied loads influence the growth of cracks.

- (c) To retain extra terms in the series expansions for the near crack tip displacement field used for the derivation of the element stiffness properties.

The possible benefits of this are twofold :

- (i) More accurate estimates of stress intensity factors may be obtained using an extended series compared with the shortened series, for a given core element radius, R_c . The results obtained from this thesis do suggest however that core radius is related to solution accuracy and that a detailed study of the influence of the various core variables is required.
- (ii) By using larger core elements, the crack propagation path could be studied, i.e. the crack is extended within the core element. For all the cases considered in this thesis the crack tip was fixed at the centre of a circular core. It should be noted, however, that the core shape is arbitrary and the particular choice of a circular core was based on one

convenient for performing the integrations to find the core strain energy U_c . Using numerical integration, this restriction is lifted and cores of arbitrary shape may be employed with the crack tip placed anywhere within such a core.

- (d) To develop a 'fracture package', or suite of programs capable of analysing any cracked structure.

This has been the general theme of this thesis, where the technique of embedding a core region about a crack tip has been applied to several cracked structures. Before the broad objective of a fracture package could be achieved however, the problem of crack closure must be further clarified. The mathematical model of closure proposed in this thesis is capable of further refinement and effort should be directed towards achieving this.

- (e) To extend the principles of the core element technique to provide a more realistic model of material behaviour.

The limited yielding of material adjacent to a crack tip has been discussed previously, Chapter 2, where the plasticity correction factor was introduced. As solutions for the near crack tip displacement field are available⁽⁸⁾ which include such limited plastic flow then it is possible to employ these in the shape function for the crack tip element. With such elastic/plastic core elements it would be possible to give more realistic guides to structural behaviour and also to study the influence of such material variables as strain hardening exponent or yield stress on fracture toughness values. Hilton and Hutchinson⁽⁵¹⁾ have determined plastic stress intensity factors for some simple cracked structures using a similar approach to that described above.

- (f) To develop the elastic core element for problems of material anisotropy or 'bi-material fracture'.

This is an important area of research, particularly for fibre reinforced composites or laminated materials that fail by delamination. Solutions are already available⁽⁸⁾ for the crack tip displacement field when the material is anisotropic and could be employed in the core element shape functions.

- (g) To combine the core element with the Strain Energy Density Factor (SEDF) concept of Sih⁽²¹⁾ for crack propagation studies.

The SEDF of Sih is a method for predicting both the onset of cracking and the direction of subsequent crack growth and relies on the separate determination of the relevant stress intensity factors. As the method described in this thesis gives values of the stress intensity factors directly as output from the program, it is ideally suited to Sih's SEDF approach. Also it should be possible to link this work with that of suggestion (c.ii) where a crack is allowed to propagate within a larger core element.

- (h) To extend the principles of the core element for the analysis of three dimensional fracture problems.

In such three-dimensional fracture problems, all three modes of crack extension, K_I , K_{II} and K_{III} are present. A prerequisite for the analysis of these problems is a suitable three-dimensional finite element program which could be modified to include the crack tip element. As near crack tip displacement solutions are currently available⁽⁸⁾ for all three modes of crack extension, then it is a simple matter to extend the theory, presented in this thesis for plane problems, for the analysis of certain three-dimensional crack problems.

One possible aspect of this work which would require further research however, is the problem of variation of stress intensity factors along a crack front. Consider, for example, a circumferentially cracked solid circular cylinder subjected to combined tension, bending and torsion loads.

In this case, all three modes of crack extension are operating and each component will vary around the crack tip. Any proposed crack tip element would have to allow for this variation, possibly by employing several tip elements round the crack front.

8. CONCLUSIONS

A method has been developed and applied for the determination of Mode I and Mode II stress intensity factors, K_I and K_{II} , for a variety of cracked structures. The method consists of embedding a circular sub-region, containing the crack tip and within which the analytical solution is used, into the finite element mesh. As a consequence of the experience gained during the writing of computer programs and the results obtained from the computational experiments carried out, the following conclusions have been drawn :-

- (a) The method provides accurate estimates of stress intensity factors for a variety of cracked structures.
- (b) The method is conceptually simple requiring only minor alterations to a standard finite element program.
- (c) There is a considerable scope for further development of the method, both in the areas of two and three dimensional crack problems.

and

- (d) The method can be easily incorporated into a design procedure for the calculation of critical defect sizes or critical stress levels that a cracked structure could sustain.

9. REFERENCES

1. Anderson, W.E., "Some Design-Oriented Views on Brittle Fracture", Gulf Coast Metals Conf., Houston, Texas, Feb., 1969.
2. Bluhm, J.I., "Fracture Mechanics", Soc. Automotive Engrs., Automotive Eng. Congress, Detroit, Mich., Jan., 1963.
3. A.S.T.M. STP-381, "Fracture Toughness Testing and Its Applications".
4. Griffith, A.A., "The Phenomenon of Rupture and Flow in Solids", Phil. Trans. Roy. Soc. (London), Series A, Vol. 221, 1920.
5. Inglis, C.E., "Stresses in a Plate Due to the Presence of Cracks and Sharp Corners", Trans. Inst. Naval Architects, Vol. 55, Pt. 1, 1913.
6. Westergaard, H.M., "Bearing Pressures and Cracks", J. Appl. Mechs., Vol. 6, 1939.
7. Liebowitz, H. (Ed.), "Fracture: An Advanced Treatise", Vol. II, (Mathematical Fundamentals), Academic Press, 1968.
8. Ibid, Article by G.C. Sih and H. Liebowitz, "Mathematical Theory of Brittle Fracture", Chapter 2.
9. Ibid, Article by J.R. Rice, "Mathematical Analysis in the Mechanics of Fracture", Chapter 3.
10. Knott, J.F., "Fundamentals of Fracture Mechanics", Butterworths, 1973.
11. Irwin, G.R., "Fracture Mechanics", Proc. 1st Symp. Naval Structural Mechanics, (Ed. J.N. Goodier and N.J. Hoff), Pergamon Press, 1960.
12. Irwin, G.R., "Fracture", Encyclopaedia of Physics, Vol. 6, Springer, Berlin (1958).
13. Irwin, G.R., Kies, J.A. and Smith, H.L., "Fracture Strengths Relative to Onset and Arrest of Crack Propagation", Proc. Am. Soc. Test. Mat., 58, (1958).
14. A.S.T.M. STP-410 "Plane Strain Crack Toughness Testing of High Strength Metallic Materials", (1967).

15. Irwin, G.R., "Analysis of Stresses and Strains near the End of a Crack Traversing a Plate," Trans. A.S.M.E., J. Appl. Mech., 24, (1957)
16. Irwin, G.R., "Fracture Dynamics", Fracturing of Metals, Am. Soc. Metals, Cleveland, 1948.
17. Orowan, E., "Energy Criteria of Fracture", Welding Research Supplement, Vol. 20, 1955.
18. Irwin, G.R., "Plastic Zone Near a Crack and Fracture Toughness", Proc. 7th Sagamore Ordnance Matls. Conf., Syracuse Univ. Research Inst., 1960.
19. Cartwright, D.J. and Rooke, D.P., "Methods of Determining Stress Intensity Factors", R.A.E. Tech. Report 73031, May. 1973.
20. Cartwright, D.J. and Rooke, D.P., "Stress Intensity Factors: A Review", Univ. of Southampton, Dept. of Mech. Eng., ME/71/39, Sept. 1971.
21. Sih, G.C. (Ed.), "Methods of Analysis and Solutions of Crack Problems", Noordhoff, 1973.
22. Tada, H., Paris, P. and Irwin, G.R., "The Stress Analysis of Cracks Handbook", Del Research Corpn., 1973, Hellertown, Pa.
23. Sih, G.C., "Handbook of Stress-Intensity Factors", 1973, Lehigh Univ. Bethlehem, Pa.
24. Srawley, J.E., Jones, M.H. and Gross, B., "Experimental Determination of the Dependence of Crack Extension Force on Crack Length for a Single-Edge-Notch Tension Specimen", NASA, T.N. D-2396, (1964).
25. Rice, J.R. and Tracey, D.M., "Computational Fracture Mechanics", in 'Numerical and Computer Methods in Structural Mechanics' by Fenves, S.J., Perrone, N., Robinson, A.R. and Schnobrich, W.C., Academic Press, 1973.
26. Williams, M.L., "On the Stress Distribution at the Base of a Stationary Crack", J. Appl. Mech., 24, (1957).

27. Muskhelishvili, N.I., "Some Basic Problems of the Mathematical Theory of Elasticity", Noordhoff, (1953).
28. Gross, B., Srawley, J.E. and Brown, W.F., "Stress Intensity Factors for a Single Notch Tensile Specimen by Boundary Collocation of a Stress Function", NASA T.N. D-2395, (1964).
29. Gross, B. and Srawley, J.E., "Stress Intensity Factors for Single-Edge-Notch Specimens in Bending or Combined Bending and Tension", NASA T.N. D-2603, (1965).
30. Gross, B. and Srawley, J.E., "Stress Intensity Factors for Three Point Bend Specimens by Boundary Collocation", NASA T.N. D-3092 (1965).
31. Kobayashi, A.S., Cherepy, R.B. and Kinsel, W.C., "A Numerical Procedure for Estimating the Stress Intensity Factor of a Crack in a Finite Plate", J. Basic Eng., Vol. 86 (1964).
32. Rowe, G.H., "Matrix Displacement Methods in Fracture Mechanics Analysis of Reactor Vessels", Nuc. Eng. and Design, 20, (1972).
33. Oglesby, J.J. and Lomacky, O., "An Evaluation of Finite Element Methods for the Computation of Elastic Stress Intensity Factors", ASME Paper No. 72-PVP-19, June 1972.
34. Gallagher, R.H., "Survey and Evaluation of the Finite Element Method in Linear Elastic Fracture Mechanics". Presented at the colloquium of IUTAM 'High Speed Computing of Elastic Structures', Liege, (1970).
35. Chan, S.K., Tuba, I.S. and Wilson, W.K., "On the Finite Element Method in Linear Elastic Fracture Mechanics", Eng. Fract. Mech., 2 (1970).
36. Kobayashi, A.S., Maiden, O.E., Simon, B.J. and Iide, S., "Application of the Method of Finite Element Analysis to Two-Dimensional Problems in Fracture Mechanics", Report by Univ. of Washington to Office of Naval Research, Contract No. 477(39), NR.064-478, TR No. 5 (Oct.1968)
37. Watwood, V.B., "The Finite Element Method for the Prediction of Crack Behaviour", Nuc. Eng. and Design, 11, (1969).

38. Anderson, G.P., Ruggles, V.L. and Stibor, G.S., "Use of Finite Element Computer Programs in Fracture Mechanics", *Int. J. Fract. Mech.*, 7, (1971).
39. Miyamoto, H., Shiratori, M. and Miyoshi, T., "Application of the Finite Element Method to the Fracture Mechanics", *J. of Faculty of Eng., Univ. of Tokyo (B)*, Vol. 31, No. 1, 1971.
40. Irwin, G.R. and Kies, J.A., "Critical Energy Rate Analysis of Fracture Strength of Large Welded Structures", *Welding J. Res. Supplement*, 33, (1954).
41. Dixon, J.R. and Pook, L.P., "Stress Intensity Factors Calculated Generally by the Finite Element Method", *Nature*, Vol. 224, (Oct.1969)
42. Rice, J.R., "A Path Independent Integral and the Approximate Analysis of Strain Concentration by Notches and Cracks", *J. Appl. Mech.*, *Trans. A.S.M.E., Ser.E.*, 35, (June 1968).
43. Hayes, D.J., "A Practical Application of Buekner's Formulation for Determining Stress Intensity Factors for Cracked Bodies", *Int. J. Fract. Mech.*, Vol. 8, No. 2, (June 1972).
44. Buekner, H.F., "The Propagation of Cracks and the Energy of Elastic Deformation", *Trans. A.S.M.E.*, 80, (1958).
45. Tong, P. and Pian, T.H.H., "The Convergence of the Finite Element Method in Solving Linear Elastic Problems", *Int. J. Solids and Struct.*, 3, (1967).
46. Tong, P. and Pian, T.H.H., "On the Convergence of the Finite Element Method for Problems with Singularity", *Int. J. Solids. Struct.*, Vol. 9, 1973.
47. Byskov, E., "The Calculation of Stress Intensity Factors Using the Finite Element Method with Cracked Elements", *Int. J. Fract. Mech.*, Vol. 6, No. 2, June 1970.
48. Tracey, D.M., "Finite Elements for the Determination of Crack Tip Elastic Stress Intensity Factors", *Eng. Fract. Mech.*, Vol. 3, No. 3, Oct. 1971.

49. Wilson, W.K., "On Combined Mode Fracture Mechanics", Research Report 69-1E7-FMECH-R1, Westinghouse Research Labs., Pittsburgh, Pennsylvania, (1969).
50. Tracey, D.M., "Finite Elements for Three Dimensional Elastic Crack Analysis", Nuc. Eng. and Design., 26, (1974).
51. Hilton, P.D. and Hutchinson, J.W., "Plastic Stress Intensity Factors for Cracked Plates", Harvard Univ. Rept., SM-34, (May 1969).
52. Wilson, W.K., "Fracture Mechanics Technology for Combined Loading and Low-to-Intermediate Strength Materials", Westinghouse Research Labs. Final Tech. Report under Contract DAAE-07-67-C-4021 (Nov.1968)
53. Leverenz, R.K., "A Finite Element Stress Analysis of a Crack in a Bi-Material Plate", Int. J. Fract. Mech., Vol.8, No. 3, (Sept.1972)
54. Corten, H.T., "Fracture Mechanics of Composites", Fracture : An Advanced Treatise, Vol. VII, Academic Press, 1968.
55. Sih, G.C. and Liebowitz, H., "On the Griffith Energy Criterion for Brittle Fracture", Int. J. Solids and Structures, Vol. 3, No. 1, 1967.
56. Richards, T.H., "On Using the Finite Element Method and Sub-Region Singular Solutions in Fracture Mechanics", Dept. of Mechanical Engineering Report, University of Aston, Nov. 1974.
57. Bowie, C.L. and Neal, D.M. "A Modified Mapping-Collocation Technique For Accurate Calculation of Stress Intensity Factors", Int. J. Fract. Mech., Vol. 6, No. 2, (1970).
58. Cartwright, D.J., "Stress Intensity Factors and Residual Static Strength of Certain Cracked Structural Elements", Ph.D. Thesis, Univ. of Southampton, 1971.
59. Bowie, O.L., "Analysis of an Infinite Plate Containing Radial Cracks Originating at the Boundary of an Internal Circular Hole", J. Math. Phy., Vol. 25, 1956.

60. Erdogan, F. and Sih, G.C., "On the Crack Extension in Plates under Plane Loading and Transverse Shear", J. Basic Eng., Trans. A.S.M.E., Series D, Vol. 85, 1963.
61. Sih, G.C. and Macdonald, B., "What the Designer Must Know About Fracture Mechanics", App. Fract. Mech. to Engg. Problems, Lehigh Univ. Report IFSM 72-73, No. 1972.
62. Shah, R.C., "Fracture Under Combined Modes in 4340 Steel", Fracture Analysis, ASTM STP-560, A.S.T.M., 1974.
63. Timoshenko, S. and Goodier, J.N., "Theory of Elasticity", McGraw Hill, 2nd Ed., 1957.
64. Rice, J.R., "The Mechanics of Crack Tip Deformation and Extension by Fatigue", Symp. on Fatigue Crack Growth, ASTM STP-415, A.S.T.M., 1967.
65. Savin, G.N., "Stress Concentration Around Holes", Pergamon Press, 1961.
66. Hussain, M.A., Pu, S.L. and Underwood, J., "Strain Energy Release Rate for a Crack under 'Combined Mode I and Mode II'", Fracture Analysis, ASTM STP-560, 1974, pp.2-28.
67. Bond, T.J., Swannell, J.H., Henshell, R.D. and Warburton, G.B., "A Comparison of Some Curved Two-Dimensional Finite Elements", J. Strain Analysis, Vol. 8, No. 3, 1973.
68. Peterson, R.E., "Stress Concentration Design Factors", Chapman and Hall, 1953.

10. APPENDICES10.A FIELD EQUATIONS FOR STRAINS, STRESSES AND DISPLACEMENTS NEAR A CRACK TIP

The near crack tip field equations are written for both Polar and Cartesian Coordinate Systems centred at a crack tip, Figs. 3.1 and 3.2. The equations for the Polar Coordinates are taken from equation 3.22 while those for the Cartesian system are obtained by a transformation of this equation.

Polar Coordinates

$$\sigma_r = \frac{1}{2\sqrt{2r}} \left[K_I(3-\cos\theta)\cos\frac{\theta}{2} + K_{II}(3\cos\theta-1)\sin\frac{\theta}{2} \right] + 2\alpha_1(1+\cos 2\theta)$$

$$\sigma_\theta = \frac{1}{2\sqrt{2r}} \left[K_I(1+\cos\theta)\cos\frac{\theta}{2} - K_{II}(3\sin\theta)\cos\frac{\theta}{2} \right] + 4\alpha_1 \sin^2\theta \quad (10.1)$$

$$\tau_{r\theta} = \frac{1}{2\sqrt{2r}} \left[K_I \sin\theta \cos\frac{\theta}{2} + K_{II}(3\cos\theta-1)\cos\frac{\theta}{2} \right] - 2\alpha_1 \sin 2\theta$$

$$u_r = \frac{1}{4G} \left(\frac{r}{2}\right)^{\frac{1}{2}} \left\{ K_I \left[(2K-1)\cos\frac{\theta}{2} - \cos\frac{3\theta}{2} \right] - K_{II} \left[(2K-1)\sin\frac{\theta}{2} - 3\sin\frac{3\theta}{2} \right] \right\} \\ + \alpha_1 \left(\frac{r}{2G}\right) (K-1+2 \cos 2\theta)$$

$$u_\theta = \frac{1}{4G} \left(\frac{r}{2}\right)^{\frac{1}{2}} \left\{ K_I \left[\sin\frac{3\theta}{2} - (1+2K)\sin\frac{\theta}{2} \right] - K_{II} \left[(2K+1)\cos\frac{\theta}{2} - 3\cos\frac{3\theta}{2} \right] \right\} \\ - \alpha_1 \left(\frac{r}{G}\right) \sin 2\theta + \alpha_2 \left(\frac{r}{2G}\right) (K+1) \quad (10.2)$$

$$\epsilon_r = \frac{1}{8\alpha\sqrt{2r}} \left\{ K_I \left[(2K-1)\cos\frac{\theta}{2} - \cos\frac{3\theta}{2} \right] - K_{II} \left[(2K-1)\sin\frac{\theta}{2} - 3\sin\frac{3\theta}{2} \right] \right\} \\ + \alpha_1 \left(\frac{1}{2G}\right) (K-1+2 \cos 2\theta)$$

$$\begin{aligned} \epsilon_{\theta} = \frac{1}{4G\sqrt{2r}} & \left\{ K_I \left[\frac{1}{2} \cos \frac{3\theta}{2} + (K-\frac{3}{2}) \cos \frac{\theta}{2} \right] - K_{II} \left[(K-\frac{3}{2}) \sin \frac{\theta}{2} + \frac{3}{2} \sin \frac{3\theta}{2} \right] \right\} \\ & + \alpha_1 \left(\frac{1}{2G} \right) (K-1-2 \cos 2\theta) \end{aligned} \quad (10.3)$$

$$\gamma_{r\theta} = \frac{1}{4G\sqrt{2r}} \left[K_I \left(\sin \frac{\theta}{2} + \sin \frac{3\theta}{2} \right) + K_{II} \left(\cos \frac{\theta}{2} + 3 \cos \frac{3\theta}{2} \right) \right] - \alpha_1 \left(\frac{2}{G} \right) \sin 2\theta$$

Cartesian Coordinates

$$\sigma_x = \frac{1}{\sqrt{2r}} \left[K_I \cos \frac{\theta}{2} \left(1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) - K_{II} \sin \frac{\theta}{2} \left(2 + \cos \frac{\theta}{2} \cos \frac{3\theta}{2} \right) \right] + 4\alpha_1$$

$$\sigma_y = \frac{1}{\sqrt{2r}} \left[K_I \cos \frac{\theta}{2} \left(1 + \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) + K_{II} \left(\sin \frac{\theta}{2} \cos \frac{\theta}{2} \cos \frac{3\theta}{2} \right) \right] \quad (10.4)$$

$$\begin{aligned} \tau_{xy} = \frac{1}{\sqrt{2r}} & \left[K_I \left(\cos \frac{\theta}{2} \sin \frac{\theta}{2} \cos \frac{3\theta}{2} \right) + K_{II} \cos \frac{\theta}{2} \left(1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) \right] \\ & + 2\alpha_1 (1 - \sin \theta \cos \theta) \end{aligned}$$

$$\begin{aligned} u_x = \frac{1}{4G} \left(\frac{r}{2} \right)^{\frac{1}{2}} & \left\{ K_I \left[(2K-1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \right] + K_{II} \left[(2K+3) \sin \frac{\theta}{2} + \sin \frac{3\theta}{2} \right] \right\} \\ & + \alpha_1 \left(\frac{r}{2G} \right) (K+1) \cos \theta - \alpha_2 \left(\frac{r}{2G} \right) (K+1) \sin \theta \end{aligned}$$

$$\begin{aligned} u_y = \frac{1}{4G} \left(\frac{r}{2} \right)^{\frac{1}{2}} & \left\{ K_I \left[(2K+1) \sin \frac{\theta}{2} - \sin \frac{3\theta}{2} \right] - K_{II} \left[(2K-3) \cos \frac{\theta}{2} + \cos \frac{3\theta}{2} \right] \right\} \\ & + \alpha_1 \left(\frac{r}{2G} \right) (K-3) \sin \theta + \alpha_2 \left(\frac{r}{2G} \right) (K+1) \cos \theta \end{aligned} \quad (10.5)$$

$$\begin{aligned} \epsilon_x = \frac{1}{8G\sqrt{2r}} & \left\{ K_I \left[(2K-3) \cos \frac{\theta}{2} + \cos \frac{5\theta}{2} \right] - K_{II} \left[(2K+1) \sin \frac{\theta}{2} - \sin \frac{5\theta}{2} \right] \right\} \\ & + \alpha_1 \left(\frac{1}{2G} \right) (K+1) \end{aligned}$$

$$\begin{aligned} \epsilon_y = \frac{1}{8G\sqrt{2r}} & \left\{ K_I \left[(2K-1) \cos \frac{\theta}{2} - \cos \frac{5\theta}{2} \right] - K_{II} \left[(2K-5) \sin \frac{\theta}{2} - \sin \frac{5\theta}{2} \right] \right\} \\ & + \alpha_1 \left(\frac{1}{2G} \right) (K-3) \end{aligned} \quad (10.6)$$

$$\gamma_{xy} = \frac{1}{4G\sqrt{2r}} \left[K_I \left(\sin \frac{5\theta}{2} - \sin \frac{\theta}{2} \right) + K_{II} \left(3 \cos \frac{\theta}{2} - \cos \frac{5\theta}{2} \right) \right]$$

10.B STRAIN ENERGY OF CORE REGION

Mode I Strain Energy Formulation

The strain energy stored in a crack tip core is given by the expression :

$$U_c = \int_{\text{Vol. of core}} \frac{1}{2} (\sigma_r \epsilon_r + \sigma_\theta \epsilon_\theta + \tau_{r\theta} \gamma_{r\theta}) d(\text{Volume}) \quad (10.7)$$

For the Mode I case, Fig. 10.1,

$$U_{Ic} = \frac{t}{2} \int_0^\pi \int_0^{R_c} (\sigma_r \epsilon_r + \sigma_\theta \epsilon_\theta + \tau_{r\theta} \gamma_{r\theta}) r dr d\theta \quad (10.8)$$

where t is the plate thickness

and R_c is the core radius.

The stresses are obtained from equations 10.1 by setting $K_{II} = 0$ in the form :

$$\begin{aligned} \sigma_r &= \frac{1}{2\sqrt{2r}} \left[K_I (3 - \cos \theta) \cos \frac{\theta}{2} \right] + 2\alpha_1 (1 + \cos 2\theta) \\ \sigma_\theta &= \frac{1}{2\sqrt{2r}} \left[K_I (1 + \cos \theta) \cos \frac{\theta}{2} \right] + 4\alpha_1 \sin^2 \theta \\ \tau_{r\theta} &= \frac{1}{2\sqrt{2r}} \left[K_I \sin \theta \cos \frac{\theta}{2} \right] - 2\alpha_1 \sin 2\theta \end{aligned} \quad (10.9)$$

Similarly from equation 10.3, with $K_{II} = 0$,

$$\begin{aligned} \epsilon_r &= \frac{K_I}{8G\sqrt{2r}} \left[(2K-1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \right] + \frac{\alpha_1}{2G} (K-1 + 2 \cos 2\theta) \\ \epsilon_\theta &= \frac{K_I}{4G\sqrt{2r}} \left[\frac{1}{2} \cos \frac{3\theta}{2} + (K - \frac{3}{2}) \cos \frac{\theta}{2} \right] + \frac{\alpha_1}{2G} (K - 1 - 2 \cos 2\theta) \\ \gamma_{r\theta} &= \frac{K_I}{4G\sqrt{2r}} \left[2 \sin \theta \cos \frac{\theta}{2} \right] - \frac{2\alpha_1}{G} \sin 2\theta \end{aligned} \quad (10.10)$$

Substituting equations 10.9 and 10.10 into equation 10.8 and performing the necessary integrations gives :

$$U_{Ic} = \frac{K_I^2 R_c \pi}{32G} (2K-1) + \frac{\alpha_1^2 R_c^2 \pi}{2G} (K+1) + \frac{K_I \alpha_1}{12G} \left(\frac{R_c^3}{2} \right)^{\frac{1}{2}} \left(16K - \frac{112}{5} \right) \quad (10.11)$$

Modes I and II Strain Energy Formulation

The strain energy stored in the crack tip core, Fig. 10.2, is given from equation 10.7 by the expression :

$$U_{cII} = \frac{t}{2} \int_0^{2\pi} \int_0^{R_c} (\sigma_r \epsilon_r + \sigma_\theta \epsilon_\theta + \tau_{r\theta} \gamma_{r\theta}) r dr d\theta \quad (10.12)$$

Taking expressions for stresses and strains as given by equations 10.1 and 10.3 respectively, substituting these in equation 10.12 and performing the integrations :

$$U_{cII} = \frac{K_I^2 R_c \pi}{16G} (2K-1) + \frac{\alpha_1^2 R_c^2 \pi}{G} (K+1) + \frac{K_{II}^2 R_c \pi}{16G} (3+2K) - \frac{K_{II} \alpha_1}{6G} \left(\frac{R_c^3}{2} \right)^{\frac{1}{2}} \left(16K + \frac{48}{5} \right) \quad (10.13)$$

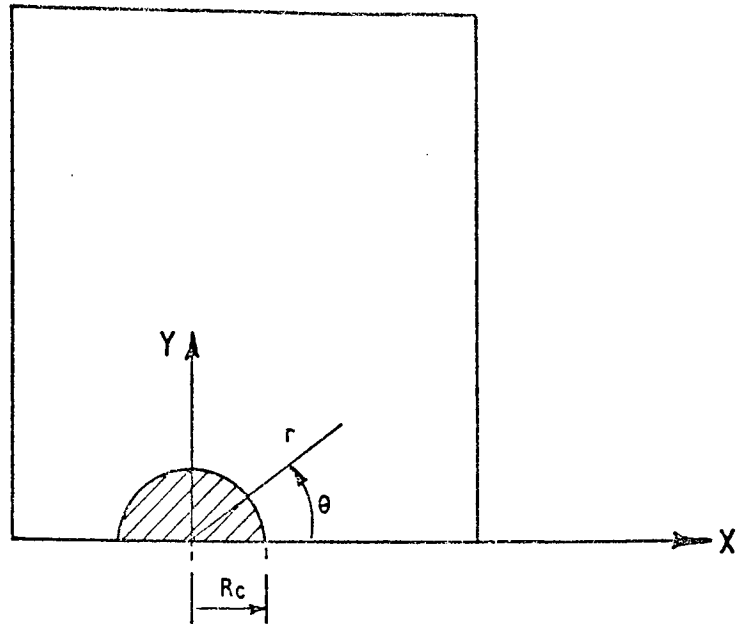


FIG. 10.1 MODE I CORE REGION

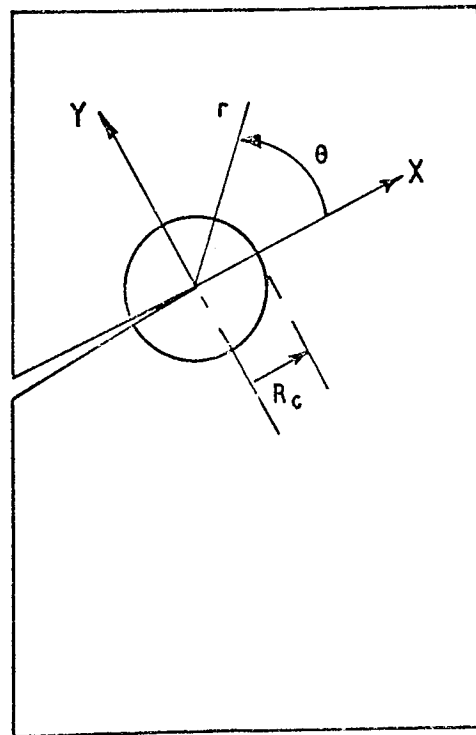


FIG. 10.2 MIXED MODE CORE REGION

10.C PROGRAM FOR PLOTTING DEFINED FINITE ELEMENT MESH

The following program provides a graphical display of the finite element mesh as an aid to identifying errors in data generation. Data required for the execution of the program is identical to that described in Appendix 10.D for the various crack configurations.

```

'BEGIN' 'COMMENT' GRAPHICAL DISPLAY OF INPUT DATA;
'REAL' SCALE,X0,Y0,CHANGE;
'INTEGER' NPLOT,RESTART,NELEMT,NNODE,THICK,NFREE,NSETFS,PRNT,NSETC,
        S,NSETF,1,1,NSPEC,W,NEWLINE,NC1,NC2,NC3;
'INTEGER' 'ARRAY' RCD1,RCD2,RCD3[1:10];
'REAL' 'ARRAY' X,Y[1:7];
NPLOT:=READ;
'FOR' RESTART:=1 'STEP' 1 'UNTIL' NPLOT 'DO'
'BEGIN'
OPENPLOT;
WRITETEXT('('('20')'JOB%NAMEX-----')');
COPYTEXT('('END%OFTITLE')');
NELEMT:=READ; NNODE:=READ; THICK:=READ;
NSETFS:=READ;
WRITETEXT('('('20')'NO%OF%ELEMENTSX-----')');
PRINT(NELEMT,3.0);
WRITETEXT('('('20')'NO%OF%NODESX-----')');
PRINT(NNODE,3.0);
NFREE:=NNODE*2;
NSETC:=READ;
'BEGIN'
'REAL' 'ARRAY' XX,VY[1:NNODE],ULX,VLY[1:NNODE,1:NSETFS];
'INTEGER' 'ARRAY' NODE[1:NELEMT,1:6],KODE[1:NNODE,1:NSETFS];
'FOR' S:=1 'STEP' 1 'UNTIL' NSETC 'DO'
'BEGIN' 'IF' S=1 'THEN'
'BEGIN' NSPEC:=READ;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO' XX(I):=0.00001;
WRITETEXT('('('20')'COORDINATES%OF%SPECIFIED%NODES('204S')'
NODE('5S')'X%COORD('5S')'Y%COORD('5S')'TYPE('7S')'X-DISP('6S')'
Y-DISP('64S')'OR%LOAD('5S')'OR%LOAD('5S')'); NEWLINE(2);
'FOR' I:=1 'STEP' 1 'UNTIL' NSPEC 'DO'
'BEGIN'
J:=READ; XX(J):=READ; VY(J):=READ;
KODE(J,1):=READ; ULX(J,1):=READ; VLY(J,1):=READ;
NEWLINE(1);
SPACE(3); PRINT(J,3.0); SPACE(3); PRINT(XX(J),0.3);
PRINT(VY(J),0.3); SPACE(2); PRINT(KODE(J,1),3.0); SPACE(2);
PRINT(ULX(J,1),0.3); PRINT(VLY(J,1),0.3);
'END';
WRITETEXT('('('40')'ELEMENT%DATA('20')'ELEMENT('18S')'
MODAL%CONNECTIONS')');
'FOR' I:=1 'STEP' 1 'UNTIL' NELEMT 'DO'
'BEGIN' NEWLINE(1); W:=READ; PRINT(W,3.0); SPACE(6);
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN' NODE(I,J):=READ;
PRINT(NODE(I,W,J),3.0); SPACE(2);
'END';
'END';
WRITETEXT('('('20')'MODAL%CONNECTIONS%HAVE%BEEN%READ('30')'

```

```

INTERPOLATIONSTRACE('120')ELEMEN('135')NODE('145')
Y-COORD('175')Y-COORD('1');
'FOR' W:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN' 'IF' XX[NODE[W,3+1]]=0.00001 'THEN'
'BEGIN'
NEWLINE(1); PRINT(W,3,0); SPACE(2);
PRINT(NODE[W,3+1],3,0); SPACE(2);
'IF' I=1 'THEN' J:=1 'ELSE' 'IF' I=2 'THEN' J:=2 'ELSE' J:=0;
XX[NODE[W,3+1]]:=(XX[NODE[W,I]]+XX[NODE[W,1+J]])/2;
PRINT(XX[NODE[W,3+1]],0,3); SPACE(2);
YY[NODE[W,3+1]]:=(YY[NODE[W,I]]+YY[NODE[W,1+J]])/2;
PRINT(YY[NODE[W,3+1]],0,3);
KODE[NODE[W,3+1],1]:=0;
ULX[NODE[W,3+1],1]:=VLY[NODE[W,3+1],1]:=0.0;
'END';
'END';
WRTTEXT('('120')NODALXPOINTXDATA('12045')NODE('155')XCOORD
('155')YCOORD('155')TYPE('175')X-DISP('165')Y-DISP('16485')
ORXLOAD('155')ORLOAD('1');
'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';
'END' 'ELSE'
'BEGIN' NNEWC:=READ;
NSETF:=READ;
'FOR' J:=1 'STEP' 1 'UNTIL' NNEWC 'DO'
'BEGIN' J:=READ;
KODE[J,1]:=READ; ULX[J,1]:=READ; VLY[J,1]:=READ;
'END';
'END';
'BEGIN'
HGPLOT(-5.0,25.0,0.4);
CHANGE:=0.0;
'FOR' W:=1 'STEP' 1 'UNTIL' NNODE 'DO'
'BEGIN'
'IF' XX[W]>CHANGE 'THEN' CHANGE:=XX[W];
'IF' YY[W]>CHANGE 'THEN' CHANGE:=YY[W];

```

```

'END';
SCALE:=3;
WRITE TEXT('('('20')'SCALE')'); PRINT(SCALE,0,4);
'FOR' W:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
X[1]:=X[7]:=XX[NODE[W,1]]*SCALE;
Y[1]:=Y[7]:=YY[NODE[W,1]]*SCALE;
X[2]:=XX[NODE[W,4]]*SCALE;
Y[2]:=YY[NODE[W,4]]*SCALE;
X[3]:=XX[NODE[W,2]]*SCALE;
Y[3]:=YY[NODE[W,2]]*SCALE;
X[4]:=XX[NODE[W,5]]*SCALE;
Y[4]:=YY[NODE[W,5]]*SCALE;
X[5]:=XX[NODE[W,3]]*SCALE;
Y[5]:=YY[NODE[W,3]]*SCALE;
X[6]:=XX[NODE[W,6]]*SCALE;
Y[6]:=YY[NODE[W,6]]*SCALE;

HGPLINE(X,Y,7,1);
'END' OF ELEMENT;

STRARR(BCD1,6,('AXIS')));
STRARR(BCD2,6,('AXIS')));
STRARR(BCD3,23,('STRUCTURAL IDEALIZATION')));
HGPAXIS(0,0,0,0,BCD1,-6,10,0,0,0,0,0,1,0);
HGPAXIS(0,0,0,0,BCD2,6,12,0,90,0,0,0,1,0);
HGPSYMBL(4,10,0.25,BCD3,0,23);
'END' OF GRAPH PLOTTER PROCEDURES;
'END';
'END';
'END';
CLOSEPLOT;
'END';

```

```

XX      LENGTH  1643
EIS USED  38
#AXXX      EC

```


10.D FRACTURE PROGRAM LISTINGS

This Appendix contains program listings and input data instructions for each of the cases considered in Chapter 5.

In most of the following programs the procedures described in Section I are employed in the same form but in a few cases slight modifications have been made to these procedures. The reasons for the modifications are concerned mainly with the simplification of the final fracture program, e.g. for isotropic material only, constant thickness plates. The changed procedures are presented with explanations of the changes where appropriate.

Procedure : CMATRIX (C,CASE,A)

The core element properties have been derived for isotropic materials and so the procedure FECMATRIX of section I may be simplified. A listing of the modified procedure follows, the steps being similar to those of FECMATRIX derived in Section I.

Procedure : ASSEMBLY(NELEMT,K,XX,YY,DETJ,NODE,C,THICK,ADD)

The matrix multiplication $[B]^t[C][B]$ required for formulating element stiffness properties is altered by using the simplified procedure CMATRIX. A listing of procedure ASSEMBLY follows which contains the required modifications.

Procedure : MM2TASEM(NELEMT,K,XX,YY,DETJ,NODE,C,THICK,ADD,NFREE,AUX, SKEWEDCON)

This procedure includes the procedure SKEWEDCON for the application of skewed nodal displacements in a similar manner to the procedure FEASSEMBLY of Section I. A listing of procedure MM2TASEM follows.

```

PROCEDURE CMATRIX(C,CASE,A);
INTEGER CASE; ARRAY A,C;
REGION INTEGER I,J;
COMMENT CALCULATION OF ELASTIC CONSTANTS;
FOR I:=1 STEP 1 UNTIL 3 DO
FOR J:=1 STEP 1 UNTIL 3 DO C[I,J]:=0.0;
CASE:=READ;
FOR I:=1 STEP 1 UNTIL 5 DO A[I]:=READ;
WRITE TEXT('('('204S')'ELASTIC CONSTANTS'('204S')'E1%=')');
PRINT(A[1],0,6); WRITE TEXT('('('3S')'V1%=%')');
PRINT(A[2],0,6); WRITE TEXT('('('203S')'E2%=%')');
PRINT(A[4],0,6); WRITE TEXT('('('3S')'V2%=%')');
PRINT(A[5],0,6); WRITE TEXT('('('203S')'E12%=%')');
PRINT(A[3],0,6);
COMMENT CASE=0 FOR PLANE STRESS AND 1 FOR PLANE STRAIN;
IF CASE=1 THEN GOTO L1;
C[1,1]:=A[1]/(1-(A[4]/A[1])*(A[2]*A[2]));
C[1,2]:=C[2,1]:=A[4]*A[2]/(1-(A[4]/A[1])*(A[2]*A[2]));
C[2,2]:=A[4]/(1-(A[4]/A[1])*(A[2]*A[2]));
C[3,3]:=A[3];
GOTO L2;
L1: C[3,3]:=A[3];
A[3]:=A[1]/((1+A[5])*(1-A[5]-2*(A[4]/A[1])*(A[2]*A[2])));
C[1,1]:=A[3]*(1-A[5]*A[5]);
C[1,2]:=C[2,1]:=A[3]*(A[4]/A[1])*(A[2]*(1+A[5]));
C[2,2]:=(A[4]/A[1])*(1-(A[4]/A[1])*(A[2]*A[2]))+A[3];
L2: END OF PROCEDURE CMATRIX;

```

```

PROCEDURE ASSEMBLY(MELEM,K,XX,YY,DETJ,MODE,C,THICK,ADD);
INTEGER MELEM; REAL DETJ,THICK;
REAL ARRAY K,XX,YY,C; INTEGER ARRAY NODE,ADD;
REGID INTEGER I,J,U,Z,V,SUB2,SUB1,SUB3;
REAL ARRAY R[3,1:12],KE[1:12,1:12],E[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(NEED) DO K[I]:=0.0;
FOR Z:=1 STEP 1 UNTIL MELEM 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 INT PTS IS CONSTRUCTED;
FOR U:=1 STEP 1 UNTIL 3 DO
BEGIN
APX(W[U,2],W[U,3],W[U,4],P,XX,YY,DETJ,MODE,7);
FOR J:=1 STEP 2 UNTIL 11 DO
FOR I:=J STEP 2 UNTIL 11 DO
BEGIN
KE[I,J]:=KE[I,J]+W[U,1]*(R[1,J]*(C[1,1]*R[1,1]+C[1,3]*R[3,1])
+R[3,J]*(C[1,3]*R[1,1]+C[3,3]*R[3,1]))*DETJ*0.5*THICK;
KE[I,I+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]*(R[1,J]*(C[1,2]*R[2,1+1]+C[1,3]*
R[3,I+1])+R[3,J]*(C[2,3]*R[2,I+1]+C[3,3]*R[3,I+1]))*DETJ*0.5*THICK;
END;
FOR J:=2 STEP 2 UNTIL 12 DO
FOR I:=J STEP 2 UNTIL 12 DO
BEGIN
KE[I,J]:=KE[I,J]+W[U,1]*(R[2,J]*(C[2,2]*R[2,1]+C[2,3]*R[3,1])
+R[3,J]*(C[2,3]*R[2,1]+C[3,3]*R[3,1]))*DETJ*0.5*THICK;
IF I=12 THEN GOTO L6 ELSE
KE[I,I+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]*(R[2,J]*(C[1,2]*R[1,I+1]+C[2,3]*
R[3,I+1])+R[3,J]*(C[1,3]*R[1,I+1]+C[3,3]*R[3,I+1]))*DETJ*0.5*THICK;
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;
KIADD(SUB1)-SUB1+SUB2:=KIADD(SUB1)-SUB1+KE[I*2-1,J*2-V];
LABA: IF SUB3 LT SUB2 THEN GOTO LABB;
KIADD(SUB3)-SUB3+SUB2:=KIADD(SUB3)-SUB3+KE[I*2,J*2-V];
LABB: END;
END;
END OF PROCEDURE ASSEMBLY;

```

```

      'PROCEDURE' MN2TASEM(NELEMT,K,XX,YY,DETJ,NODE,C,THICK,ADD,NFREE,AUX,
      SKEWEDCON):
      'INTEGER' NELEMT,NFREE; 'REAL' DETJ,THICK;
      'REAL' 'ARRAY' K,XX,YY,C; 'INTEGER' 'ARRAY' NODE,ADD;
      'PROCEDURE' AUX,SKEWEDCON:
      'BEGIN' 'INTEGER' I,J,U,Z,V,SUB1,SUB2,SUB3;
      'REAL' 'ARRAY' B[1:3,1:12],KE[1:12,1:12],W[1:6,1:4];
      'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO' W[1,1]:=0.433333333;
      W[1,2]:=W[1,3]:=W[2,3]:=W[2,4]:=W[3,2]:=W[3,4]:=0.5;
      W[4,1]:=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' 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' KE[I,J]:=0.0;
      'COMMENT' THE LOOP FOR THE NUMBER OF INT PTS IS CONSTRUCTED;
      'FOR' U:=1 'STEP' 1 'UNTIL' 3 'DO'
      'BEGIN'
      AUX(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,I]*(C[1,1]*B[1,1]+C[1,3]*B[3,1])
      +C[3,1]*(C[1,3]*B[1,1]+C[3,3]*B[3,1]))*DETJ*0.5*THICK;
      KE[J,I+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]*(B[1,I]*(C[1,2]+B[2,1+1]+C[1,3]*
      B[3,I+1])+B[3,I]*(C[2,3]*B[2,1+1]+C[3,3]*B[3,I+1]))*DETJ*0.5*THICK;
      '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]*(B[2,I]*(C[2,2]*B[2,1]+C[2,3]*B[3,1])
      +B[3,I]*(C[2,3]*B[2,1]+C[3,3]*B[3,1]))*DETJ*0.5*THICK;
      'IF' I=12 'THEN' 'GOTO' L6 'ELSE'
      KE[J,I+1]:=KE[I+1,J]:=KE[I+1,J]+W[U,1]*(B[2,I]*(C[1,2]+B[1,I+1]+C[2,3]*
      B[3,I+1])+B[3,I]*(C[1,3]*B[1,I+1]+C[3,3]*B[3,I+1]))*DETJ*0.5*THICK;
      L6: 'END';
      'END';
      'IF' NSKEW.NE.0 'THEN' SKEWEDCON(Z,NODE,ANGSK,NOSK,NSKEW,KE,NSETF);
      '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;
      KIADD(SUB1)-SUB1+SUB2:=KIADD(SUB1)-SUB1+SUB2+KE[I*2-1,J*2-V];
      LABA: 'IF' SUB3.LT.SUB2 'THEN' 'GOTO' LABB;
      KIADD(SUB3)-SUB3+SUB2:=KIADD(SUB3)-SUB3+SUB2+KE[I*2,J*2-V];
      LABB: 'END';
      'END';
      'END' OF PROCEDURE ASSEMBLY;

```

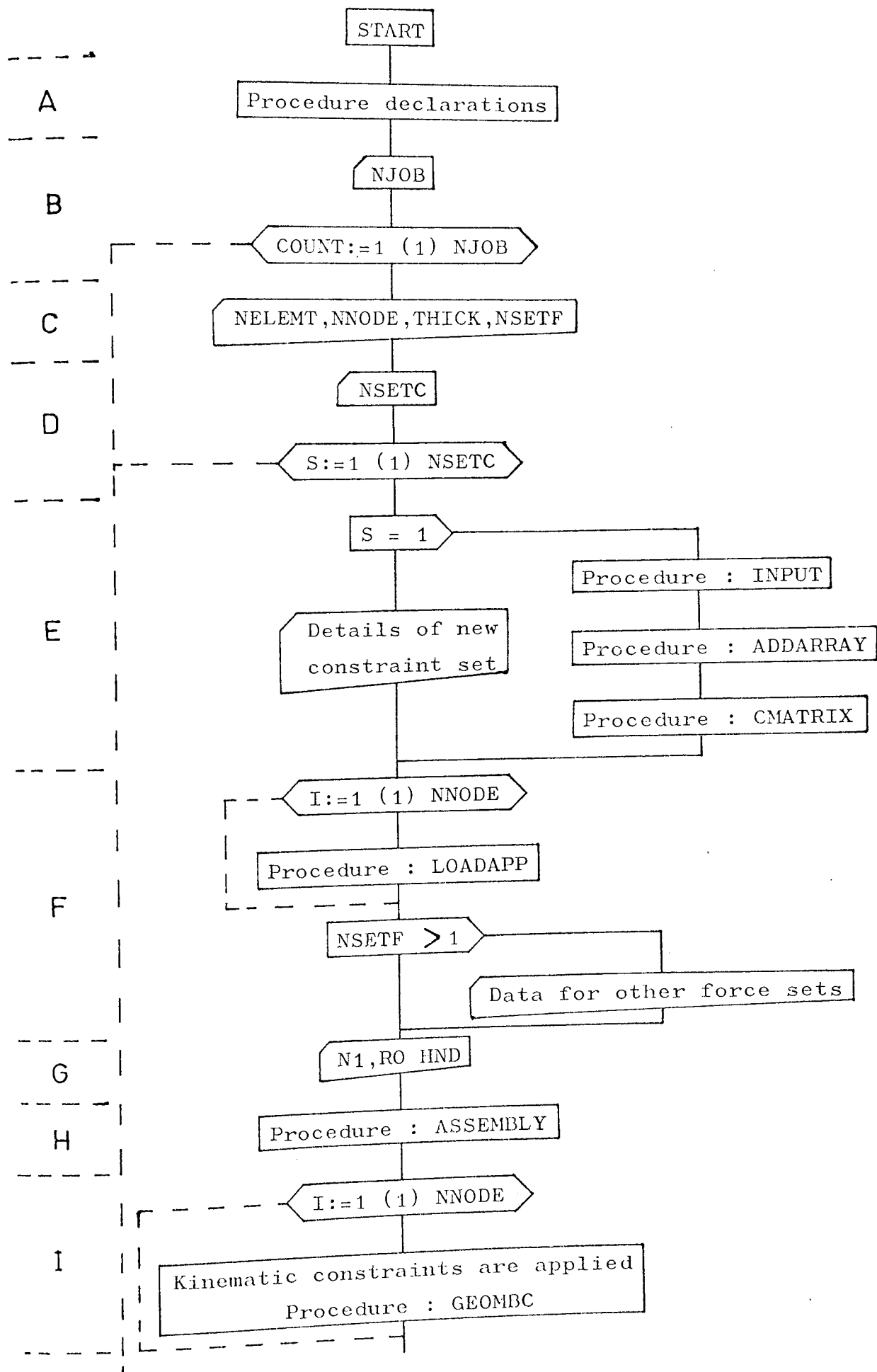
10.D.1 Single Tip, Symmetric Mode I

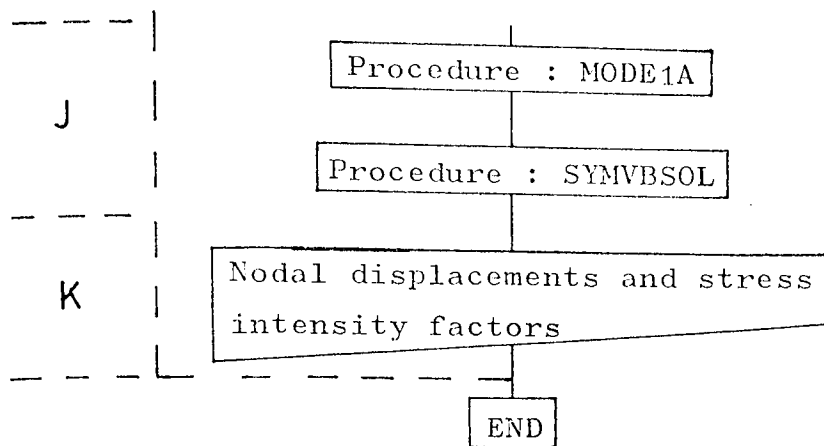
This program determines Mode I stress intensity factors, K_I , for symmetrically loaded single tip crack problems. In a similar manner to the finite element program of Section I, several sets of forces and different sets of constraints may be treated during a single run. In contrast to the finite element program, however, only isotropic materials may be analysed and the plate is of uniform thickness throughout.

A flowchart and computer listing of the program follows, the steps are described with reference to the flowchart, as :

- A. - Procedures used are declared, see Section I and Chapter 5.2.
- B. - Number of jobs (NJOB) are read and a loop is constructed round this number.
- C. - Number of elements (NELEMT), number of nodes (NNODE), thickness of plate (THICK) and number of sets of forces (NSETF) are read and printed.
- D. - Number of sets of constraints (NSETC) are read and a loop is constructed around this number.
- E. - Procedures INPUT, ADDARRAY and CMATRIX are called.
- F. - Load vector is formed using procedure LOADAPP.
- G. - Core Element details are read; number of core/element interface nodes (N1), core radius(RØ) and direction of crack tip (HND).
- H. - Stiffness matrix is assembled, procedure ASSEMBLY.
- I. - Kinematic constraints are applied, procedure GEOMBC.
- J. - Stiffness matrix is modified, procedure MØDEIA and resulting equations are solved, procedure SYMVSOL.
- K. - Stress intensity factor and nodal displacements are printed.

MODE 1 SINGLE TIP PROGRAM FLOWCHART.





LISTING OF :SOLAPS,ARTEM1(1/) PRODUCED ON 10JAN76 AT 04.36.28

OUTPUT BY LISTFILE IS :SOLAPS,CY--AR4' ON 10JAN76 AT 04.37.24

DOCUMENT ARTEM1

BEGIN COMMENT THIS PROGRAM EMPLOYS THE ISOPARAMETRIC
FORMULATION FOR 2-D PROBLEMS OF PLANE STRESS/STRAIN
THE ELEMENT USED IS A SIX-NODE TRIANGLE;

INTEGER NELEMT,NODE,NSETF,NFREE,I,J,V,U,H,K,CASE,NSETC,NHEFC,S,
NJUR,COUNT;
REAL DELTA,DETJ,THICK;
ARRAY C(1:3,1:3),A(1:5);

PROCEDURE AD*(L1,L2,L3,B,X,Y,U,V,Z);

PROCEDURE ADDARRAY(NELEMT,NODE,ADD,NODE);

PROCEDURE GEOMBOCU(X,P,AK,MFC,F,A);

PROCEDURE LOADAPP(A,B,C,D,E,F);

PROCEDURE SYMBSOL(A,L,S,R)DIMENSIONS:(*,*)FAILURE EXIT:(FAIL);

PROCEDURE NODE1A(N1,RD,NG,NSETF,A,ADD,K,Q,CASE,THICK,RND);

PROCEDURE INPUT(ADD,XX,YY,NFREE,NODE,NSETF,KODE,NSPEC,ULX,VLY,
NELEMT,NODE);

PROCEDURE CMATRIX(C,CASE,A);

PROCEDURE ASSEMBLY(NELEMT,K,AX,YY,DETJ,NODE,C,THICK,ADD);

NJUR:=READ;

FOR COUNT:=1 STEP 1 UNTIL NJUR 'DO'

BEGIN

WRITETEXT('('('2C')NJURXNAMEZ-----')');

WRITETEXT('('ENDXOFATITLE')');

NELEMT:=READ; NODE:=READ; THICK:=READ; NSETF:=READ;

WRITETEXT('('('2C')N%DEZELFALNTSZ-----')');

PRINT(NELEMT,5,0);

WRITETEXT('('('2C')N%DEZ*NODESZ-----')');

PRINT(NODE,5,0);

NFREE:=N%NODE*2;

BEGIN INTEGER NSPEC,Z,COMPA;

REAL ARRAY XX,VLY(1:NODE),ULX,VLY(1:NODE,1:NSETF),

O(1:NFREE,1:NSETF);


```

INTEGER 'ARRAY' NNODE(1:NELMT,1:6),KODE(1:NFILE,1:NSETF),ADD(0:NFREE);
NSETC:=READ;
IFOR I:=1 'STEP' 1 'UNTIL' NSETC 'DO'
  IREGI 'IF' S=1 'THEN'
    IREGI '
    INPUT(ADD,XX,YY,NFREE,NODE,NSETF,KODE,NSPEC,ULX,VLY,NELMT,NODE);
    ADDARRAY(NELMT,NODE,ADD,NODE);
    CHATSI(X(C,CASE,A));
  IEND 'ELSE'
    IREGI 'NNEWC:=READ; NSETF:=READ;
    IFOR I:=1 'STEP' 1 'UNTIL' NNEWC 'DO'
      IREGI 'J:=READ;
      KODE(J,1):=READ; ULX(J,1):=READ; VLY(J,1):=READ;
    IEND '
  IEND '
  WRITETEXT('('('('2C')'KODE%DEGREES%OF%FREEDOM---')');
  PRINT(NFREE,3,0);
  NEWLINE(2);
  IFOR I:=1 'STEP' 1 'UNTIL' NFREE 'DO'
    IFOR J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q(I,J):=0.0;
    IFOR I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
      LOADAPP(KODE(I,1),ULX(I,1),VLY(I,1),0,1,1);
    IF NSETF 'GT' 1 'THEN'
      IREGI '
      IFOR I:=1 'STEP' 1 'UNTIL' NSETF 'DO'
        IREGI '
        NSPEC:=READ;
        WRITETEXT('('('('2C')'FORCEXSET---')');
        PRINT(1,3,0);
        WRITETEXT('('('('2C')'NODE('5S')'TYPE('5S')'X-DISP('5S')'Y-DISP
        '('10S')'OR%LOAD('4S')'OR%LOAD')');
        IFOR J:=1 'STEP' 1 'UNTIL' NSPEC 'DO'
          IREGI '
          I:=READ; KODE(K,I):=READ; ULX(K,I):=READ; VLY(K,I):=READ;
          LOADAPP(KODE(K,I),ULX(K,I),VLY(K,I),Q,K,I);
        NEWLINE(2);
        PRINT(K,3,0); SPACE(2);
        PRINT(KODE(K,I),0,4); SPACE(2);
        PRINT(ULX(K,I),0,4);
        PRINT(VLY(K,I),0,4);
      IEND '
    IEND '
  IEND '
  NEWLINE(6);
  IREGI 'INTEGER' N1,NG,HND; 'REAL' RO;
  'REAL' 'ARRAY' K(1:ADD[NFREE]);
  NG:=NFREE;
  N1:=READ; RO:=READ; HND:=READ;
  ASSEMBLY(NELMT,K,XX,YY,DETJ,NODE,C,TRICK,ADD);
  'COMMENT' INTRODUCTION OF KINEMATIC CONSTRAINTS;
  IFOR I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
    IREGI '
    IF KODE(I,1)=0 'THEN' 'GOTO' KC1;
    IF KODE(I,1)=2 'THEN' 'GOTO' KC2;
    GEUBBC(ULX(I,1),2*I-1,Q,K,NFREE,1,ADD);

```

```

      IF J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q[1,J]:=Q[1,1];
      IF NNODE[1,1]=1 'THEN' 'GOTO' K01;
      K02: GEOMEC(VLY[1,1],2*1,0,K,REFE,1,ADD);
      IF J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q[1,J]:=Q[1,1];
      K01: 'END';
      WRITETEXT('('('('C2S')'NDE1%EGIS'))');
      NDE1A(S),R1,NG,NSETF,A,ADD,K,0,CASE,THICK,RND);
      SYMVSOL(K,K,ADD,0,NG-2*1+2,NSETF,FAIL);
      NEWLINE(6);
      IF J:=1 'STEP' 1 'UNTIL' NSETF 'DO'
      BEGIN
        WRITETEXT('('('('40')'NDDAL%DISPLACEMENTS%FOR%FORCE%SET'))');
        PRINT(1,2,0);
        WRITETEXT('('('('30S')'NDE1('5S')'X-DIRECTION('5S')'Y-DIRECTION
        ('5S')'NDE1('5S')'X-DIRECTION('5S')'Y-DIRECTION'))');
        J:=2*(NNODE-1)/2;
        IF 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-1,1],0,8);
          SPACE(2); PRINT(Q[V,1],0,8); SPACE(13);
          PRINT(Z,3,0); SPACE(2); PRINT(Q[V+1,1],0,8);
          SPACE(2); PRINT(Q[V+2,1],0,8);
        END;
        IF NNODE=N1 'GT' U 'THEN'
        BEGIN
          NEWLINE(2); PRINT(NNODE-N1,3,0); SPACE(2);
          V:=2*(NNODE-1);
          PRINT(Q[V-1,1],0,8); SPACE(2); PRINT(Q[V,1],0,8);
        END;
        V:=NG-2*N1;
        WRITETEXT('('('('20')'CRACKTIP%DISPLACEMENT%IN%Y-DIRECTION%=''))');
        PRINT(N[V+1,1],0,10);
        WRITETEXT('('('('20')'MODE1%STRESS%INTENSITY%FACTOR,KIX=%'))');
        PRINT(Q[V+2,1],0,10);
      END;
    END;
  END;
  END;
  END;
  FAIL: 'END';
  END;
  ****

```

```

*****

```

Input Data

The following list of instructions enables a program user to prepare the input data necessary for running the program.

- (A) Number of Jobs to be Analysed.
- (B) for first job,
 - (1) JOB TITLE, followed by END OF TITLE
 - (2) NUMBER OF ELEMENTS
 - (3) NUMBER OF NODES
 - (4) THICKNESS OF PLATE
 - (5) NUMBER OF SETS OF FORCES
 - (6) NUMBER OF SETS OF CONSTRAINTS.
- (C) for first set of constraints,
 - (1) NUMBER OF NODAL POINTS WHOSE COORDINATES ARE SPECIFIED^{(1)*}
 - (2) for the number of points specified in C(1), input
 - (a) NODE NUMBER
 - (b) X-COORDINATE
 - (c) Y-COORDINATE
 - (d) CODE⁽²⁾ (prescribed load or displacement)
 - (e) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN X-DIRECTION
 - (f) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN Y-DIRECTION
 - (3) NODAL CONNECTIONS ARE READ FOR EACH ELEMENT⁽³⁾
 - (4) TYPE OF PROBLEM (0 - plane stress, 1 - plane strain)
 - (5) ELASTIC CONSTANTS (E_x ν_x G_{xy} E_x ν_x)
 - (6) If the number of sets of forces for this set of constraints is greater than 1, then input 6(a)-(e), else go to (7)
- (a) NUMBER OF SPECIFIED FORCES
 - for the number specified in 6(a), input
 - (b) NODE NUMBER
 - (c) CODE⁽²⁾

* See Notes which follow input data instructions.

- (d) VALUE OF PRESCRIBED LOAD IN X-DIRECTION
- (e) VALUE OF PRESCRIBED LOAD IN Y-DIRECTION
- (7) NUMBER OF NODES ON CORE/MESH INTERFACE (N1)
- (8) RADIUS OF CORE (R_c)
- (9) DIRECTION OF CRACK TIP (4)
- (D) for second set of constraints
 - (1) NUMBER OF NEW CONSTRAINTS
 - (2) NUMBER OF SETS OF FORCES WITH THIS SET OF CONSTRAINTS
 - (3) for the number of new constraints, which must include the first set of forces for this set of constraints, input
 - (a) NODE NUMBER
 - (b) KODE⁽²⁾
 - (c) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN X-DIRECTION
 - (d) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN Y-DIRECTION
 - (4) If D(2) is greater than 1, then repeat C(6)(a-e)
 - (5) Repeat C(7-9)
- (E) For subsequent sets of constraints repeat from (D)
- (F) For next job repeat from (B).

NOTES

1. Most elements are straight sided and in these cases it is necessary to specify the coordinates of the vertices only. The program will automatically interpolate to calculate the coordinates of the missing nodes.

If an element side is curved or there is some prescribed load or displacement at a mid-side node then the coordinates of that node must be specified.

2. If $KODE =$

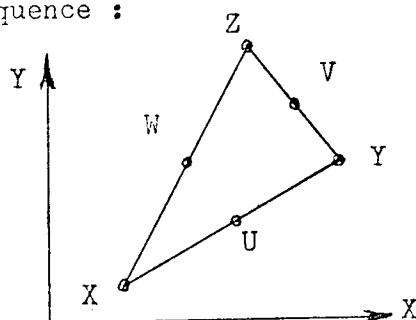
0, this represents prescribed loads in the X and Y directions.

1, this represents a prescribed displacement in the X and load in the Y direction.

2, this represents a prescribed load in X and displacement in the Y-direction

3, this represents prescribed displacements in the X and Y directions.

3. The nodal connections for each element must be specified using the following sequence :



If N is the element number and X, Y, Z, U, V, W are the node numbers then input

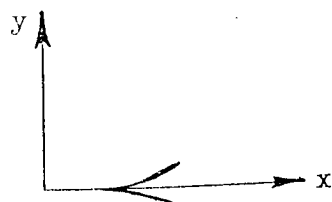
N X Y Z U V W

for each element

4.



DIRECTION = 1



DIRECTION = 2

As an example of a typical data set, consider the problem shown in Fig. 10.D.1 with the boundary conditions as shown. According to the data instructions given previously, the input data for this problem is,

1

EXAMPLE END OF TITLE

24 62 1.0 1 1

26

1 0.05 0.0 0 0 0

2 x_2 y_2 0 0 0

3

4

⋮

7

8 x_8 y_8 2 0 0

⋮

61 x_{61} y_{61} 0 0 0

62 0 0.13 0 0 1000

1 3 1 15 2 8 9

2 3 15 17 9 16 10

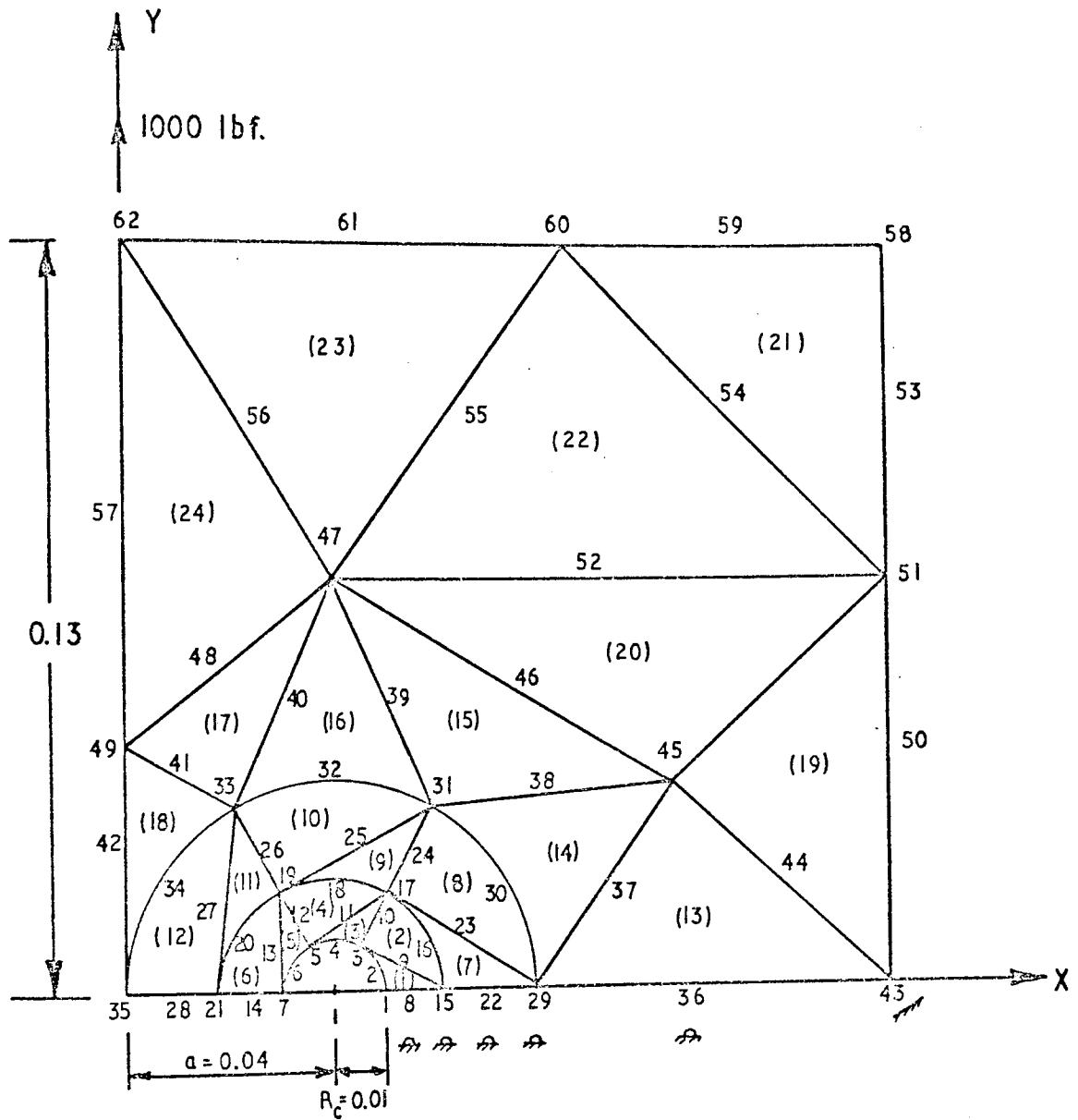
3

⋮

24 49 47 62 48 56 57

1 30 0.1 13.63 30 0.1

7 0.01 1



Thickness of plate = 1

$N1 = 7$

$E = 30$

$\nu = 0.1$

$$G = \frac{E}{2(1+\nu)} = 13.63$$

FIG. 10.D.1 MESH SUBDIVISION FOR SYMMETRIC MODE I PROGRAM.

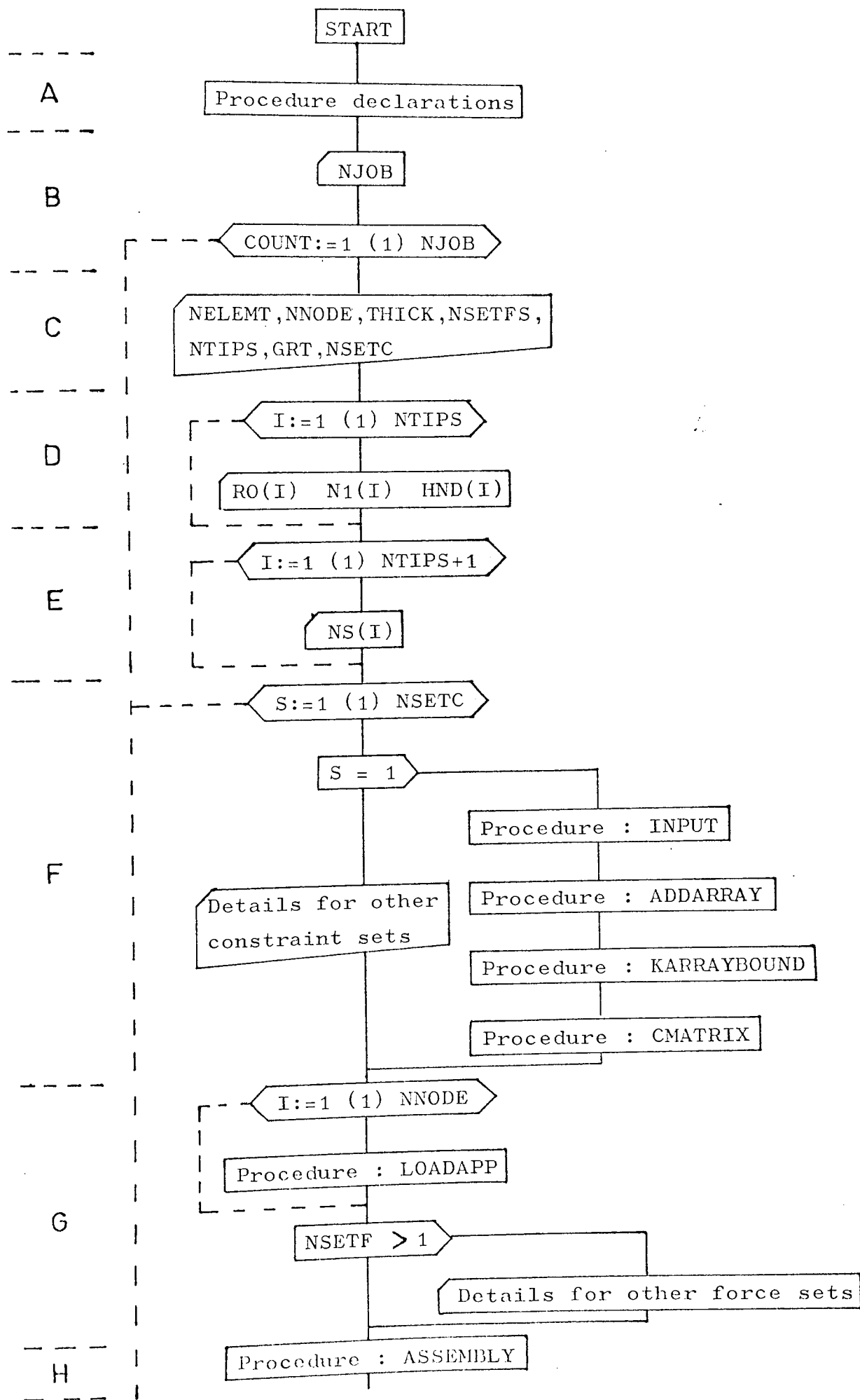
10.D.2 Several Tips, Symmetric Mode I

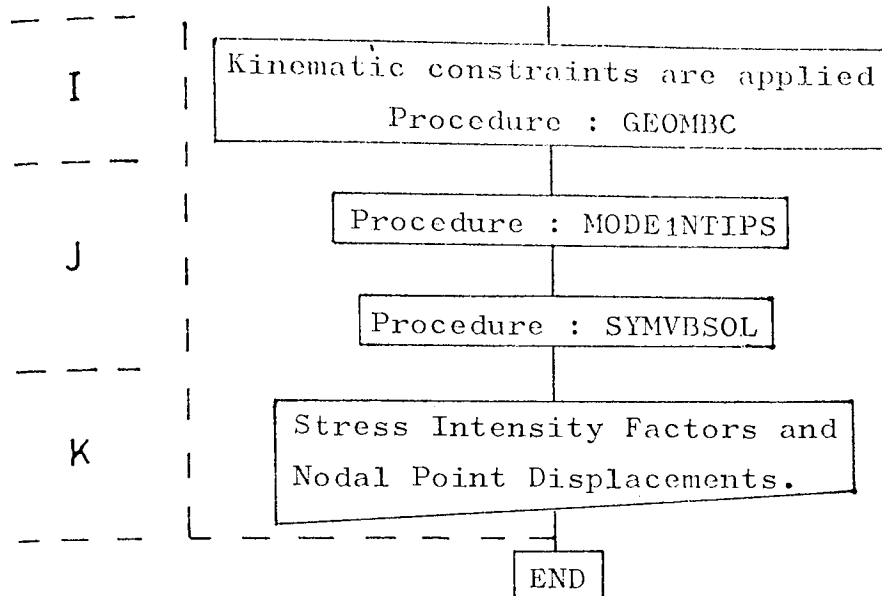
An explanation of the program construction follows. A list of instructions is given to enable a user to prepare the input data necessary for running the program.

A flowchart and computer listing of the program follows. The steps, described with reference to the flowchart, are :

- A. - Procedures are declared, see Section I and Chapter 5.3.
- B. - Number of Jobs are read (NJ/B) and a loop is constructed around this number.
- C. - Numbers of elements ($NELEMT$), nodes ($NNODE$), sets of forces ($NSETFS$), tips ($NTIPS$) and sets of constraints ($NSETC$) are read, together with thickness of the plate ($THICK$) and greatest number of interface nodes on any core (GRT).
- D. - Core element details for each tip are read; $R[I]$, $N1[I]$ and $HND[I]$ are radius, number of interface nodes and tip direction, respectively, of core number I.
- E. - Node numbers of those nodes which start the numbering of each core are contained in array NS. (This point is described more fully with reference to an example).
- F. - Procedures INPUT, ADDARRAY, KARRAYBOUND and CMATRIX are called.
- G. - Load vector is constructed, procedure LOADAPP.
- H. - Stiffness matrix is assembled, procedure ASSEMBLY.
- I. - Kinematic constraints are applied, procedure ~~GEOMBC~~.
- J. - Stiffness equations are modified, procedure ~~MODEINTIPS~~, and resulting equations are solved, procedure SYMBVSOL.
- K. - Stress intensity factors and nodal point displacements are printed.

MODE 1, MULTI-TIP PROGRAM FLOWCHART.





```

PROGRAM (AXXX)
INPUT 'O=CRO
OUTPUT 'O=LPO
EXTENDED DATA
EXTENDED
BEGIN COMMENT THIS PROGRAM EMPLOYS THE ISOPARAMETRIC
FORMULATION FOR 2-D PROBLEMS OF PLATE STRESS/STRAIN
THE ELEMENT USED IS A SIX-NODE TRIANGLE;
INTEGER NELEMT,NNODE,NSETF,NFREE,I,J,V,U,W,CASE,NSETC,NMENC,S,
NJOB,COUNT,NTIPS,GRT,C1,C2,C3,C4,C5,C6,C7,NSETFS;
REAL DELTA,DETJ,THICK;
ARRAY C[1:3,1:3],A[1:5];

PROCEDURE AUX(L1,L2,L3,R,X,Y,U,N,Z);

PROCEDURE ADDARRAY(NELEMT,NNODE,ADD,NODE);

PROCEDURE GEOMBC(U,N,R,AK,NEG,F,A);

PROCEDURE LOADAPP(A,R,C,D,E,F);

PROCEDURE SYMBVSOL(A,L,S,R)DIMENSIONS:(N,R)FAILURE EXIT:(FAIL);

PROCEDURE NODE1NTIPS(N1,R0,M6,NSETF,A,ADD,K,Q,CASE,THICK,NTIPS,
HND,GRT,NS);

PROCEDURE KARRAYBOUND(NFREE,N1,BAND,ADD,NTIPS,NS);

PROCEDURE CMATRIX(C,CASE,A);

PROCEDURE INPUT(ADD,XX,YY,NFREE,NNODE,NSETF,KODE,MSPEC,NSETFS,
ULX,VLY,NELEMT,NODE);

PROCEDURE ASSEMBLY(NELEMT,K,XX,YY,DETJ,NODE,C,THICK,ADD,NFREE,AUX);

NJOB:=READ;
FOR COUNT:=1 'STEP' 1 'UNTIL' NJOB 'DO'
BEGIN
WRITETEXT('('('2C')'JORBNAME%-----')');
COPYTEXT('('END%OF%TITLE')');
NELEMT:=READ;
NNODE:=READ;

```

```

THICK:=READ;
WRITETEXT('('('20')'NO%OF%ELEMENTS%----')');
PRINT(NLEMT,3,0);
WRITETEXT('('('20')'NO%OF%NODES%-----')');
PRINT(NNODE,5,0);
NREF:=NNODE*2;
NSETF:=READ;
NTIPS:=READ; GRT:=READ;
'BEGIN'
'INTEGER' NG,NSPEC,7,COMP, RAND;
'REAL' 'ARRAY' XX,VY[1:NNODE],ULX,VLY[1:NNODE,1:NSETF],
      Q[1:NREF,1:NSETF],PO[1:NTIPS];

'INTEGER' 'ARRAY' NODE[1:NLEMT,1:6],KODE[1:NNODE,1:NSETF],ADDID:NREF,
N1,END[1:NTIPS],NS[1:NTIPS+1];
NSETC:=READ; NG:=NREF;
'FOR' I:=1 'STEP' 1 'UNTIL' NTIPS 'DO'
'REGIN' RO[I]:=READ; R1[I]:=READ; END[I]:=READ; 'END';
'FOR' I:=1 'STEP' 1 'UNTIL' NTIPS+1 'DO' NS[I]:=READ;
'FOR' S:=1 'STEP' 1 'UNTIL' NSETC 'DO'
'REGIN'
'IF' S=1 'THEN' 'BEGIN'
NSETF:=READ;
INPUT(ADD,XX,VY,NREF,NNODE,NSETF,KODE,NSPEC,NSETF,ULX,VLY,
      NLEMT,NODE);
ADDARRAY(NLEMT,NNODE,ADD,NODE);
KARRAYBOUND(NREF,N1,RAND,ADD,NTIPS,NS);
CMATRIX(C,CASE,4);
'END' 'ELSE'
'REGIN' NNEWC:=READ; NSETF:=READ;
'FOR' I:=1 'STEP' 1 'UNTIL' NNEWC 'DO'
'REGIN' J:=READ;
      KODE[J,1]:=READ; ULX[J,1]:=READ; VLY[J,1]:=READ;

'END';
'END';
WRITETEXT('('('20')'NO%OF%DEGREES%OF%FREEDOM----')');
PRINT(NREF,3,0);
NEWLINE(2);
'FOR' I:=1 'STEP' 1 'UNTIL' NREF 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]:=0,0;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
      LOADAPP(KODE[I,1],ULX[I,1],VLY[I,1],0,1,1);
'IF' NSETF 'GT' 1 'THEN'
'REGIN'
'FOR' I:=2 'STEP' 1 'UNTIL' NSETF 'DO'
'REGIN'
NSPEC:=READ;
WRITETEXT('('('20')'FORCE%SET----')');
PRINT(I,3,0);
WRITETEXT('('('20')'MODE('5S')'TYPE('5S')'X-DISP('5S')'Y-DISP
('C10S')'ORZLOAD('4S')'ORZLOAD')');
'FOR' J:=1 'STEP' 1 'UNTIL' NSPEC 'DO'
'REGIN'
      K:=READ; KODE[K,I]:=READ; ULX[K,I]:=READ; VLY[K,I]:=READ;
      LOADAPP(KODE[K,I],ULX[K,I],VLY[K,I],0,K,1);
NEWLINE(2);
PRINT(K,3,0); SPACE(2);
PRINT(KODE[K,I],0,4); SPACE(2);
PRINT(ULX[K,I],0,4);

```

```

PRINT(VLY[K,1],0,4);
'END';
'END';
'END';
NEWLINE(6);
'BEGIN' 'REAL' 'ARRAY' X[1:RAND];
ASSEMBLY(NLENT,X,XY,VY,DET,I,NODE,C,THICK,ADD,NREF,AUX);
'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;
GEOMBO(ULX[I,1],2*I-1,0,K,NREF,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: GEOMBO(VLY[I,1],2*I,0,K,NREF,1,ADD);
'FOR' J:=2 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]:=Q[I,1];
KC1: 'END';
WRITETEXT('('('C2S')'NODE1NTIPS%BEGINS')');
MODE1NTIPS(N1,RO,NG,NSETF,A,ADD,K,0,CASE,THICK,NTIPS,HVD,GRT,MS);
C5:=0; 'FOR' I:=1 'STEP' 1 'UNTIL' NTIPS 'DO' C5:=C5+N1[I];
WRITETEXT('('('C2C')'SYMB%SQL%BEGINS')');
SYMB%SQL(K,K,ADD,0,NG-2*C5+2*NTIPS,NSETF,FAIL);
NEWLINE(6);
'FOR' I:=1 'STEP' 1 'UNTIL' NSETF 'DO'
'BEGIN'
WRITETEXT('('('C4C')'MODAL%DISPLACEMENTS%FOR%FORC%SET')');
PRINT(I,2,0);
WRITETEXT('('('C3CS')'NODE'('5S')'X-DIRECTION'('8S')'Y-DIRECTION'('18S')'MODE'('5S')'X-DIRECTION'('8S')'Y-DIRECTION')');
C6:=2*((NNODE-C5)/2);
'FOR' C7:=2 'STEP' 2 'UNTIL' C6 'DO'
'BEGIN' NEWLINE(2);
'FOR' J:=C7-1,C7 'DO'
'BEGIN' V:=2*J; C3:=2;
U:=NSI[2]-N1[I]-1; C1:=N1[I]; C5:=C5-N1[NTIPS];
'IF' J 'GE' NS[NTIPS]-C5 'THEN' C2:=C5+N1[NTIPS]
'ELSE' 'IF' J 'LE' U 'THEN' C2:=C1
'ELSE' 'BEGIN' C3:=C3+1; U:=0;
'FOR' C4:=1 'STEP' 1 'UNTIL' C3-1 'DO' U:=U+N1[C4];
U:=NS[C3]-U-1;
C1:=C1+N1[C3-1]; C2:=C1;
'END';
C5:=C5+N1[NTIPS];
PRINT(J+C2,3,0); SPACE(2); PRINT(Q[V-1,1],0,8); SPACE(2);
PRINT(Q[V,1],0,8); SPACE(13);
'END';
'END';
'IF' NNODE-C5 'GT' C6 'THEN'
'BEGIN' NEWLINE(2); PRINT(NNODE-C5,3,0); SPACE(2);
V:=2*(NNODE-C5);
PRINT(Q[V-1,1],0,8); SPACE(2); PRINT(Q[V,1],0,8);
'END';
'FOR' J:=1 'STEP' 1 'UNTIL' NTIPS 'DO'
'BEGIN' V:=2*J-2;
WRITETEXT('('('C2C')'CRACK%NUMBER-----')');
PRINT(J,3,0);
WRITETEXT('('('C2C')'CRACK%TIP%DISPLACEMENT%IN%Y-DIRECTION=')');
PRINT(Q[NG-2*C5+1+V,1],0,10);
WRITETEXT('('('C2C')'HOPE1%STRESS%INTENSITY%FACTOR,KI=')');

```

PRINT(Q[NG-2*CS+2+V,1],0,10);

'END';

'END';

'END';

'END';

'END';

FAIL: 'END';

'END';

↑*****

Input Data

An example of the mesh sub-division is shown in Fig. 6.13 for the three tip problem of Chapter 6.3. The node numbering around the core elements is also shown.

One important point which must be noted at the idealisation stage is that no two nodes on the first ring of elements surrounding adjacent cores should be connected across a common element, e.g. with reference to Fig. 10.D.2 nodes 19 and 50 must not be connected across a common element. The reason for this restriction is that the elimination of rows and columns necessary when forming the modified stiffness matrix within the procedure ~~MODE~~INTIPS is considerably simplified.

- (A) NUMBER OF JOBS TO BE ANALYSED
- (B) for first job,
 - (1) JOB TITLE followed by END OF TITLE
 - (2) NUMBER OF ELEMENTS
 - (3) NUMBER OF NODES
 - (4) THICKNESS OF PLATE
 - (5) NUMBER OF SETS OF FORCES^{(1)*}
 - (6) NUMBER OF CRACK TIPS
 - (7) GREATEST NUMBER OF NODES SURROUNDING ANY CORE⁽²⁾
 - (8) NUMBER OF SETS OF CONSTRAINTS
 - (9) for each core, in order 1, 2, etc.
 - (a) CORE RADIUS
 - (b) NUMBER OF CORE/INTERFACE NODES
 - (c) DIRECTION OF CRACK TIP⁽³⁾
 - (10) for each core, in order 1, 2, etc.
 - (a) NODE NUMBER STARTING FIRST RING OF ELEMENTS⁽⁴⁾
 - (11) NUMBER OF LAST NODE ENDING THE FIRST RING OF ELEMENTS SURROUNDING THE FINAL CORE⁽⁵⁾

* See Notes at end of data instructions.

(C) for first set of constraints

- (1) NUMBER OF SETS OF FORCES IN THIS SET OF CONSTRAINTS
- (2) NUMBER OF NODAL POINTS WHOSE COORDINATES ARE SPECIFIED⁽⁶⁾
- (3) for the number of points specified in (C) (2).
 - (a) NODE NUMBER
 - (b) X-COORDINATE
 - (c) Y-COORDINATE
 - (d) KODE⁽⁷⁾ (prescribed load or displacement)
 - (e) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN X-DIRECTION
 - (f) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN Y-DIRECTION
- (4) NODAL CONNECTIONS ARE READ FOR EACH ELEMENT⁽⁸⁾
- (5) TYPE OF PROBLEM (0 - plane stress, 1 - plane strain)
- (6) ELASTIC CONSTANTS (E, ν , G, E, ν)
- (7) If the number of sets of forces for this set of constraints is greater than 1 then input 7(a-e) else go to (D).
 - (a) NUMBER OF SPECIFIED FORCES for the number specified in 7(a), input
 - (b) NODE NUMBER
 - (c) KODE⁽⁷⁾
 - (d) VALUE OF PRESCRIBED LOAD IN X-DIRECTION
 - (e) VALUE OF PRESCRIBED LOAD IN Y-DIRECTION

(D) for second set 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, which must include the first set of forces for this set of constraints,
 - (a) NODE NUMBER
 - (b) KODE⁽⁷⁾
 - (c) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN X-DIRECTION
 - (d) VALUE OF PRESCRIBED LOAD OR DISPLACEMENT IN Y-DIRECTION
- (4) If (D)(2) is greater than 1, then repeat (C)(7)(a-e).

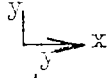
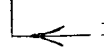
(E) For subsequent sets of constraints, repeat from (D).

(F) For next job repeat from (B).

Notes

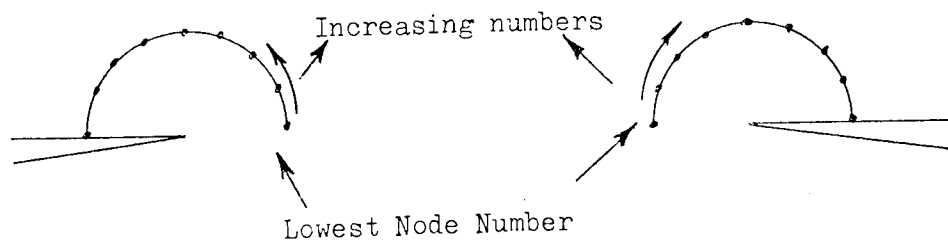
(1) This must be the greatest number for any of the sets of constraints as it fixes array bounds.

(2) The greatest value of N1 on any core/finite element interface for all the cores present, this number fixes array bounds.

(3) If crack tip faces to the right, i.e.  then direction = 1
else if crack tip faces to the left, i.e.  then direction = 2.

(4) With reference to the example of Fig. 10.D.2, the nodes starting each core are, 1 28 61

The numbering convention is,



(5) With reference to the example of Fig. 10.D.2, this value = 99.

(6) See Note 1, Section 10.D.1.

(7) See Note 2, Section 10.D.1.

(8) See Note 3, Section 10.D.1.

For the example shown in Fig. 10.D.2, the input data is :

1

3TIPS END OF TITLE

No. of Elements, No. of Nodes, 0.01, 1, 3, 13, 1

0.001 9 1

0.002 11 2

0.003 13 1

1 28 61

99

No. of nodes whose coordinates are specified

1	x_1	y_1	0	0	0	
2	1	1	1	1	1	
1	1	1	1	1	1	
1	1	1	1	1	1	
1	1	1	1	1	1	
1	1	1	1	1	1	
1	3	1	19	2	10	11
2	3	19	21	11	20	12

1

E V G E V

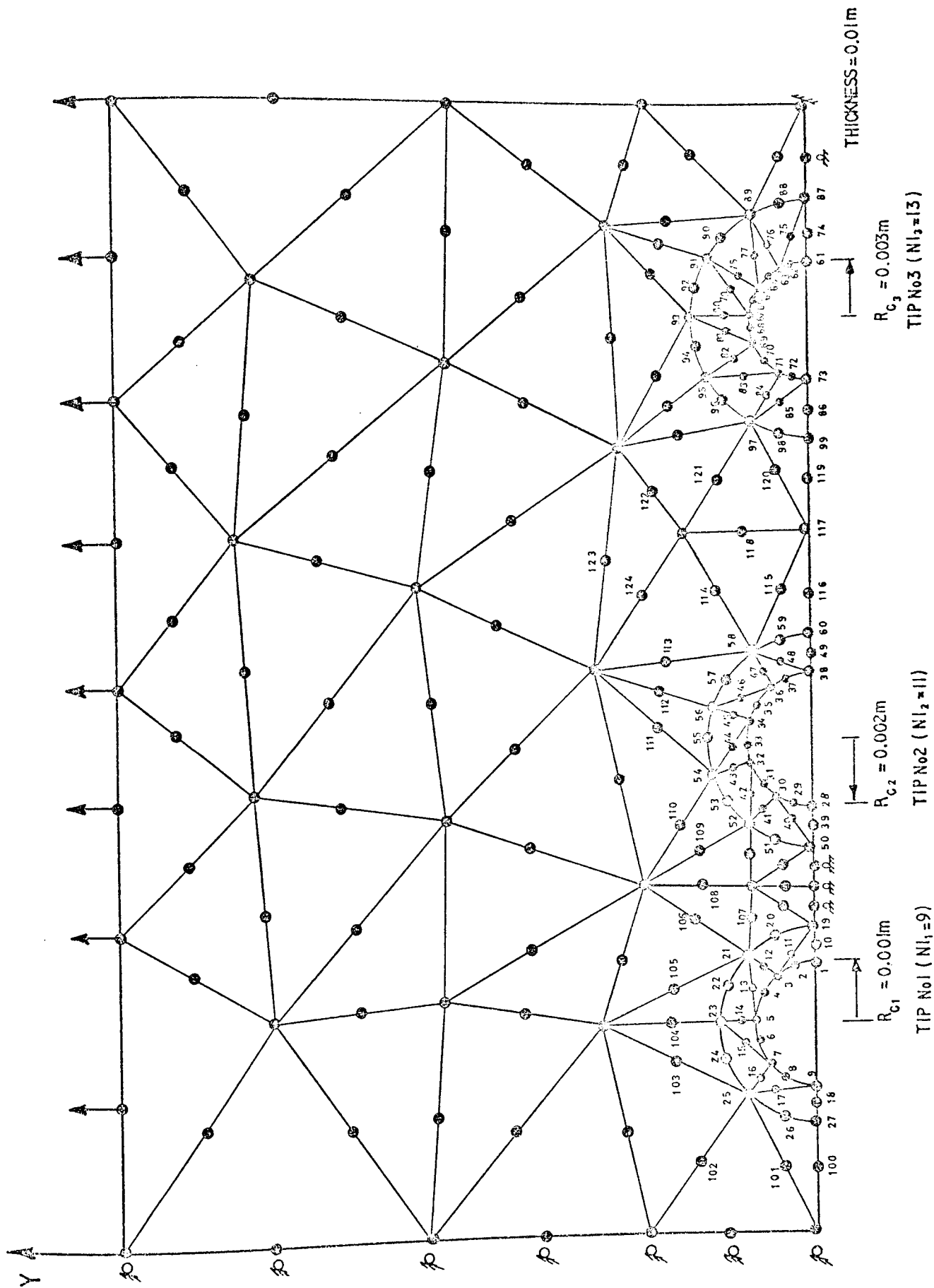


FIG.10.D.2. NODE NUMBERING FOR MULTI-TIP PROGRAM.

10.D.3 Single Tip, Combined Modes I and II

The program described in this section is used for the calculation of Mode I and II stress intensity factors, K_I and K_{II} , for single edge cracked specimens.

This program contains a procedure which checks the accuracy of the solution obtained and also the stability of the overall stiffness matrix. Consider the equilibrium equations :

$$[K]^* \{\delta\}^* = \{F\}^*$$

Procedure SYMVB SOL gives the solution vector $\{\delta\}^*$ using Choleski triangular factorisation. A residual force vector can be found by resubstituting this vector $\{\delta\}^*$ in the above equation and performing the matrix multiplications. The residual force vector $\{F_R\}^*$ so obtained should equal the original force vector $\{F\}^*$. By comparing coefficients in $\{F\}^*$ and $\{F_R\}^*$ any spurious constraints may be identified and the accuracy of the solution checked. A listing of the procedure which performs the matrix multiplications follows the main program listing.

A flowchart and program listing for the main program follows and the steps are described, with reference to the flowchart as :

- A. - Procedures are declared, Section I and Chapter 5.4.
- B. - Number of jobs are read and a loop is constructed round this number.
- C. - Number of elements (NELEMT), number of Nodes (NNODE), thickness of plate (THICK), number of sets of forces (NSETFS), output option (PRNT) and the number of sets of constraints (NSETC) are read.
- D. - Loop round the number of sets of constraints is formed.
- E. - Input data is read, procedure INPUT; address sequence is formed, procedure ADDARRAY; bounds of the stiffness matrix are calculated, procedure KARBDMST; the elasticity matrix is formed, procedure CMATRIX and core parameters are read.
- F. - Load vector is formed, procedure LOADAPP.

- 403
- G. - Stiffness matrix is assembled, procedure ASSEMBLY and modified for the core element, procedure MODEIIST.
 - H. - Kinematic constraints are applied, procedure GEOMBC and the equations are solved, procedure SYMVBSØL.
 - I. - Stress intensity factors and nodal displacements are printed.
 - J. - Residual force vector is calculated and output, using procedure RESIDUAL.

Input Data

An example of the mesh numbering in the core region is shown in Fig.

10.D.3.

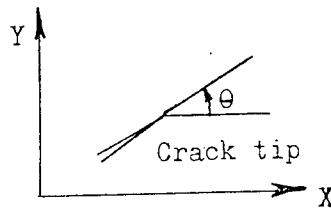
- (A) NUMBER OF JOBS TO BE ANALYSED
- (B) for first job,
 - (1) JOB TITLE followed by END OF TITLE
 - (2) NUMBER OF ELEMENTS
 - (3) NUMBER OF NODES
 - (4) THICKNESS OF PLATE
 - (5) NUMBER OF SETS OF FORCES^{(1)*}
 - (6) OUTPUT OPTION⁽²⁾
 - (7) NUMBER OF SETS OF CONSTRAINTS.
- (C) for first set of constraints,
 - (1) NUMBER OF SETS OF FORCES FOR THIS SET OF CONSTRAINTS.
 - (2) NUMBER OF NODAL POINTS WHOSE COORDINATES ARE SPECIFIED⁽³⁾
 - (3) for the number of points specified in (C)(2),
 - (a) NODE NUMBER
 - (b) X-COORDINATE
 - (c) Y-COORDINATE
 - (d) KØDE⁽⁴⁾(specified load or displacement)
 - (e) VALUE OF PRESCRIBED DISPLACEMENT OR LOAD IN X-DIRECTION
 - (f) VALUE OF PRESCRIBED DISPLACEMENT OR LOAD IN Y-DIRECTION

* See Notes at end of data instructions.

- (4) NODAL CONNECTIONS ARE READ FOR EACH ELEMENT⁽⁵⁾
- (5) TYPE OF PROBLEM (0 - plane stress, 1 - plane strain)
- (6) ELASTIC CONSTANTS (E, ν , G, E, ν)
- (7) If the number of sets of forces for this set of constraints is greater than 1, then input 7(a-e), else go to (8).
 - (a) NUMBER OF SPECIFIED FORCES
 - for the number specified in 7(a) input
 - (b) NODE NUMBER
 - (c) KODE⁽⁴⁾
 - (d) VALUE OF PRESCRIBED LOAD IN X-DIRECTION
 - (e) VALUE OF PRESCRIBED LOAD IN Y-DIRECTION
- (8) NUMBER OF NODES ON CORE/MESH INTERFACE
- (9) RADIUS OF CORE ELEMENT
- (10) ANGLE OF CRACK TIP FROM + ve x-axis⁽⁶⁾
- (D) for the second set of constraints,
 - (1) NUMBER OF NEW CONSTRAINTS
 - (2) NUMBER OF SETS OF FORCES IN THIS SET OF CONSTRAINTS
 - (3) for the number of new constraints, which must include the first set of forces for this set of constraints,
 - (a) NODE NUMBER
 - (b) KODE⁽⁴⁾
 - (c) VALUE OF PRESCRIBED LOAD IN X-DIRECTION,
 - (d) VALUE OF PRESCRIBED LOAD IN Y-DIRECTION
 - (4) If (D)(2) is greater than 1, then repeat (c)(7a-e)
 - (5) Repeat (c)(10)
- (E) for subsequent sets of constraints, repeat from (D)
- (F) for next job, repeat from (B)

Notes

- (1) This must be the greatest number for any of the sets of constraints as it fixes array bounds.
- (2) Procedures similar to those of Section I may be included to calculate element or nodal stresses, as these procedures have been excluded any integer number may be input here.
- (3) See Note 1, Section 10.D.1.
- (4) See Note 2, Section 10.D.1.
- (5) See Note 3, Section 10.D.1
- (6) Angle θ (+ ve) is measured in a clockwise direction from the positive X-axis as shown,



For the example shown in Fig. 10.D.3, the input data is :

1

EDGE CRACK END OF TITLE

No. of elements 113 Thickness 1 1 1 1

No. of Nodes whose coordinates are specified

1 x_1 y_1 0 0 0

2 x_2 y_2 0 0 0

| | | | |

| | | | |

| | | | |

113 x_{113} y_{113} 0 0 P_{y113}

1 39 41 1 40 21 20

2 1 41 3 21 22 2

| | | | | |

| | | | | |

| | | | | |

1

E ν G E ν

19 R_c

α°

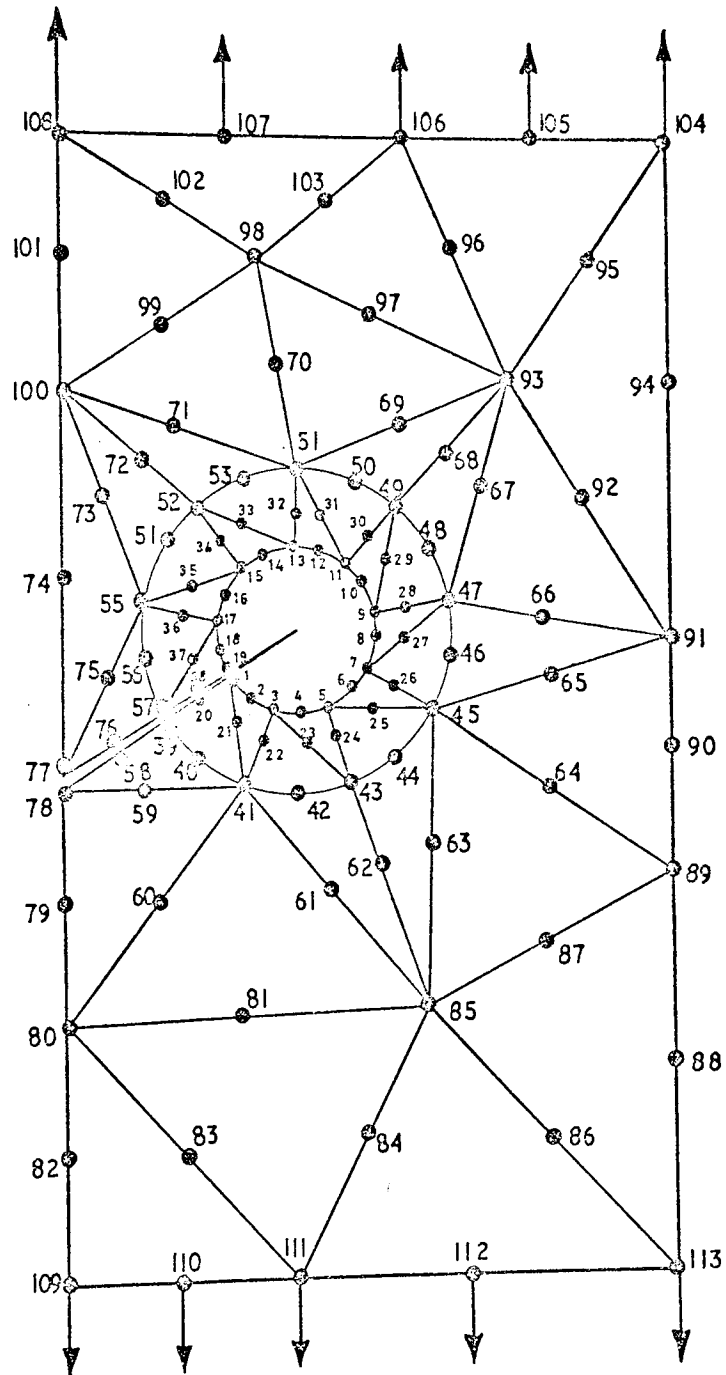
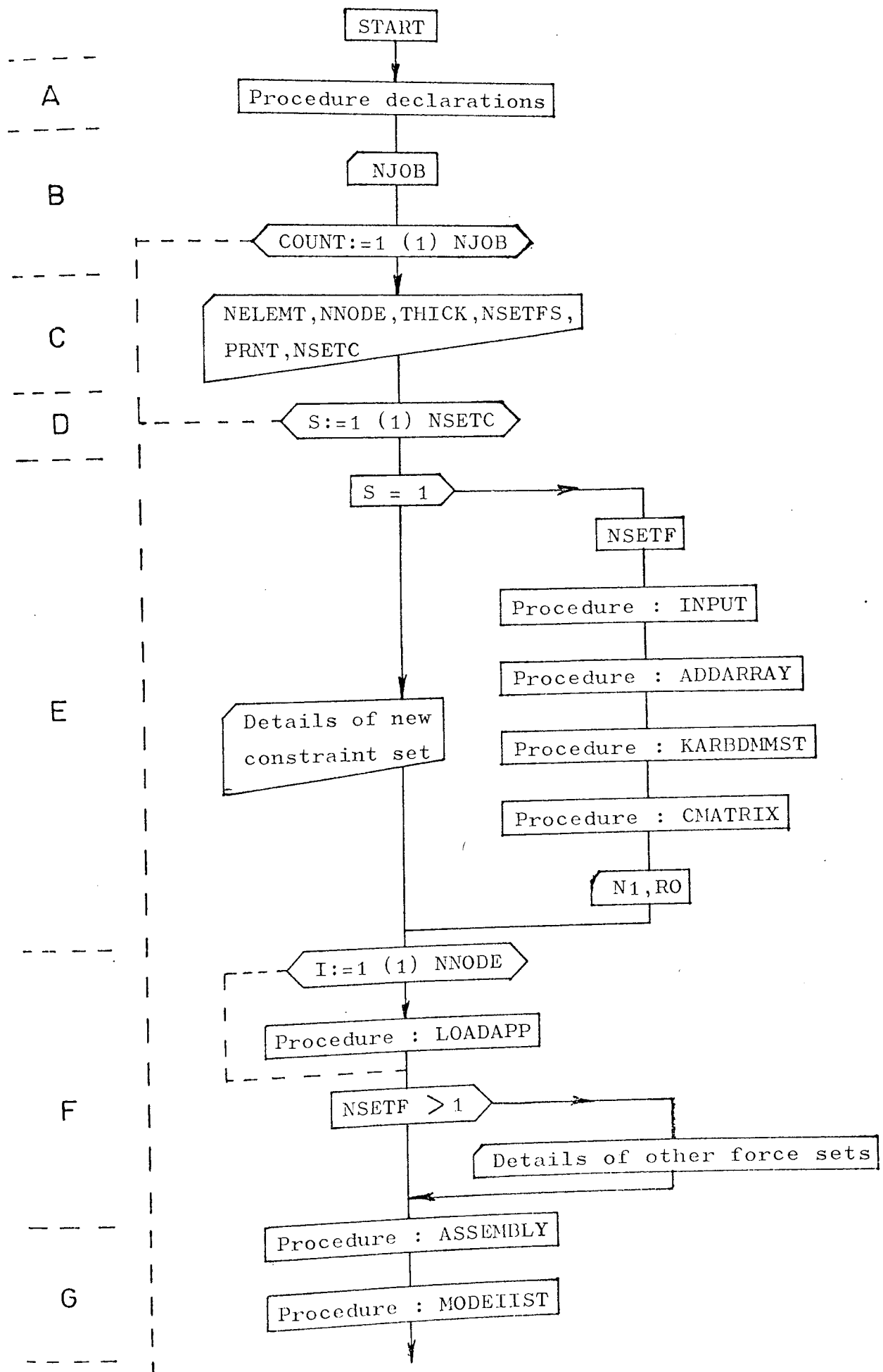
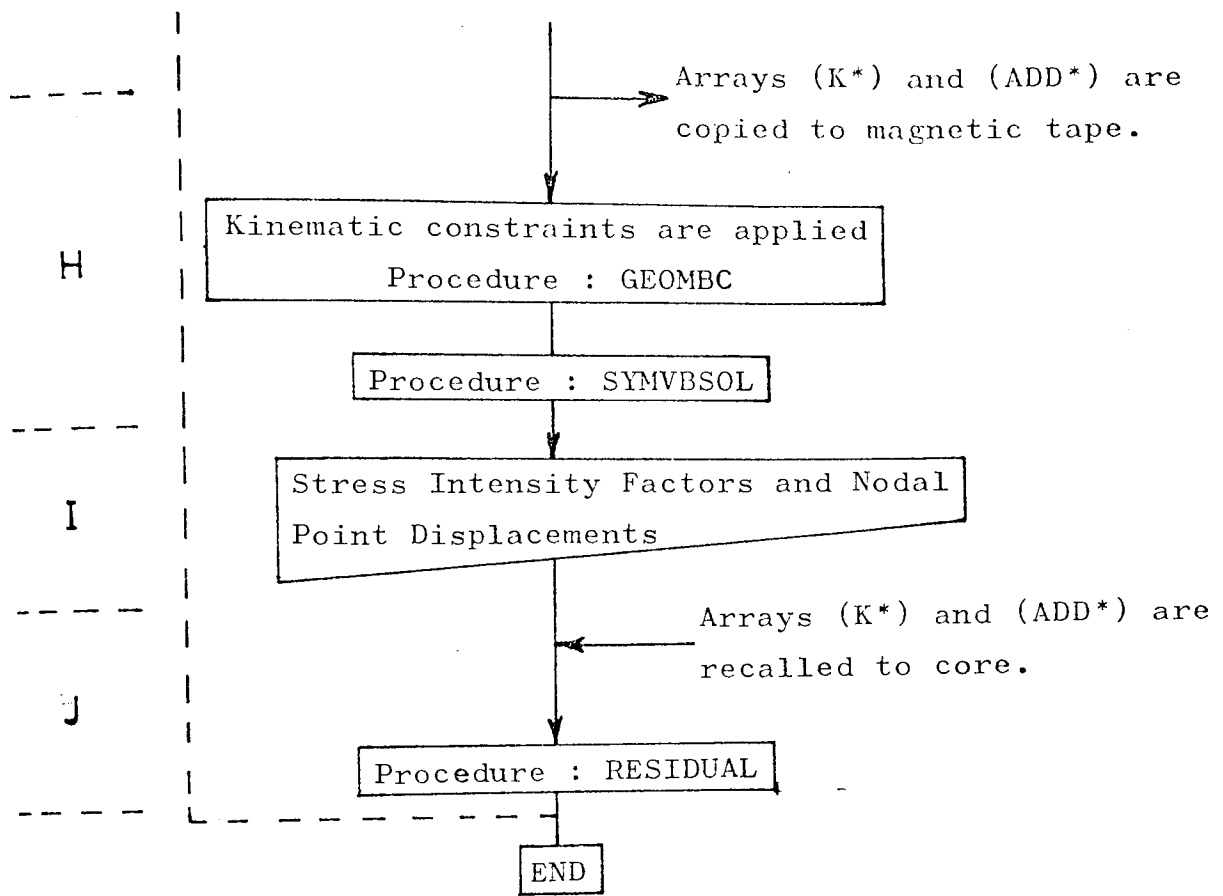


FIG.10.D.3 ENLARGEMENT OF CORE SHOWING NODAL NUMBERING.

MIXED MODE SINGLE TIP PROGRAM FLOWCHART.





#LISTING OF :EMP0024.SHM1THES(17) PRODUCED ON 23JUL75 AT 16.52.00

#OUTPUT BY LISTFILE IN :EMP0024.LISTS ON 28JUL75 AT 18.56.54

DOCUMENT SHM1THES

```
PROGRAM 'CAXXX'
INPUT '0=LP0
OUTPUT '0=LP0
EXTENDED DATA
EXTENDED
LEGIT 'COMMENT' THIS PROGRAM EMPLOYS THE ISOPARAMETRIC
FORMULATION FOR 2-D PROBLEMS OF PLANE STRESS/STRAIN
THE ELEMENT USED IS A SIX-NODE TRIANGLE;
INTEGER' NELEMT,NNODE,NSETF,NREFE,I,J,K,V,N,U,CASE,ISSETC,NNEHC,S,
NJOB,COUNT,PROT,NVAB1;
REAL' DELTA,DETJ,THICK;
ARRAY' C(1:3,1:3),A(1:5);
PROCEDURE' CREATESTORE(N,S,T,G,L);
VALUE' N,L,G; 'INTEGER' N,L,G; 'STRING' S,T; 'EXTERNAL';
PROCEDURE' ISTORE(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' AUX(L1,L2,L3,R,X,Y,U,D,Z);

PROCEDURE' GEOMBC(U,N,R,AK,NEQ,F,A);

PROCEDURE' LOADAPP(A,B,C,D,E,F);

PROCEDURE' SYMVB SOL(A,I,S,R) DIMENSIONS: (N,R) FAILURE EXIT: (FAIL);

PROCEDURE' ADDARRAY(NELEMT,NNODE,ADD,NODE);

PROCEDURE' MODELIST(N1,R0,NG,NSETF,A,ADD,K,Q,CASE,THICK);

PROCEDURE' KARBDMMST(NREFE,N1,BAND,ADD);

PROCEDURE' RESIDUAL(K,Q,ADD,NREFE,NSETF);

PROCEDURE' CMATRIX(C,CASE,A);

PROCEDURE' INPUT(ADD,XX,YY,NREFE,NNODE,NSETF,KODE,NSPEC,NSETF,
ULX,VLY,NELEMT,KODE);
```

```
'PROCEDURE' ASSEMBLY(NFELEM,K,XX,YY,DETJ,CODE,C,THICK,ADD);
```

```
NJOB:=READ;
'FOR' COUNT:=1 'STEP' 1 'UNTIL' NJOB 'DO'
'BEGIN'
  WRTTEXT('('('20')'JOBXNANFY-----')');
  COPYTEXT('('END%OFXTITLE')');
  NFELEM:=READ; NNODE:=READ; THICK:=READ;
  WRTTEXT('('('20')'NO%OF%ELEMENTS%----')');
  PRINT(NFELEM,3,0);
  WRTTEXT('('('20')'NO%OF%NODES%-----')');
  PRINT(NNODE,3,0);
  NREF:=NNODE*2; NSETF:=READ; NNT:=READ;
  'BEGIN' 'INTEGER' BAND,NSPEC,7,COMP,NSETF,N1,NG;
  'REAL' R0;
  'REAL' 'ARRAY' XX,YY[1:NNODE],ULX,VLY[1:NNODE,1:NSETF],
    Q[1:NREF,1:NSETF];
  'INTEGER' 'ARRAY' NODE[1:NFELEM,1:6],KODE[1:NNODE,1:NSETF],ADD[0:NREF];
  NSETC:=READ;
  'FOR' S:=1 'STEP' 1 'UNTIL' NSETC 'DO'
  'BEGIN'
    'IF' S=1 'THEN' 'BEGIN'
      NSETF:=READ;
      INPUT(ADD,XX,YY,NREF,NNODE,NSETF,KODE,NSPEC,NSETF,ULX,VLY,
        NFELEM,NODE);
      ADDARRAY(NFELEM,NNODE,ADD,NODE);
      N1:=NREF; N1:=READ; R0:=READ;
      KAPDMST(NREF,N1,BAND,ADD);
      CMATRIX(C,CASE,A);
      WRTTEXT('('('20')'NO%OF%DEGREES%OF%FREEDOM----')');
      PRINT(NREF,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' NREF 'DO'
    'FOR' J:=1 'STEP' 1 'UNTIL' NSETF 'DO' Q[I,J]:=0.0;
    'FOR' I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
      LOADAPP(KODE[I,1],ULX[I,1],VLY[I,1],0,I,1);
    'IF' NSETF 'GT' 1 'THEN'
      'BEGIN'
        'FOR' I:=2 'STEP' 1 'UNTIL' NSETF 'DO'
        'BEGIN'
          NSPEC:=READ;
          WRTTEXT('('('20')'FORCEXSET-----')');
          PRINT(1,3,0);
          WRTTEXT('('('20')'NODE('5S')'TYPE('8S')'X-DISP('7S')'Y-DISP
            ('21S')'OR%LOAD('6S')'OR%LOAD')');
          'FOR' J:=1 'STEP' 1 'UNTIL' NSPEC 'DO'
          'BEGIN'
            K:=READ; KODE[K,I]:=READ; ULX[K,I]:=READ; VLY[K,I]:=READ;
            LOADAPP(KODE[K,I],ULX[K,I],VLY[K,I],Q,K,I);
          'END';
        'END';
      'END';
  'END';
```

```

NEWLINE(2);
PRINT(K,3,0); SPACE(2);
PRINT(ODECK,11,3,0); SPACE(2);
PRINT(OLX(K,11,0,4);
PRINT(VLY(K,11,0,4);
'END';
'END';
'END';
NEWLINE(6);
'BEGIN' 'INTEGER' SUB1,SUB2,SUB3,SURSCP;
'DEAL' 'ARRAY' K[1:RAND];
ASSEMBLY(XELEM,K,XX,YY,DETJ,MODE,C,THICK,ADD);
NEWLINE(6);
MODELIST(N1,R0,NG,NSFTE,A,ADD,K,C,CASE,THICK);
CREATESTORE(10,('ED'),'('KATLEOT')',1,40000);
NVAR1:=1;
PUTARRAY(10,NVAR1,K);
PUTARRAY(10,NVAR1,ADD);
'COMMENT' INTRODUCTION OF KINEMATIC CONSTRAINTS;
'FOR' I:=1 'STEP' 1 'UNTIL' NNODE-N1 'DO'
'BEGIN' Z:=I+N1;
'IF' ODEI(Z,1)=0 'THEN' 'GOTO' KC1;
'IF' ODEI(Z,1)=2 'THEN' 'GOTO' KC2;
GEOMBO(OLX[Z,1],2*I-1,0,K,NG-2*N1+5,1,ADD);
'FOR' J:=2 'STEP' 1 'UNTIL' NSFTE 'DO' Q[I,J]:=Q[I,1];
'IF' ODEI(Z,1)=1 'THEN' 'GOTO' KC1;
KC2: GEOMBO(VLY[Z,1],2*I,0,K,NG-2*N1+5,1,ADD);
'FOR' J:=2 'STEP' 1 'UNTIL' NSFTE 'DO' Q[I,J]:=Q[I,1];
KC1: 'END';
WRITETEXT('('('2C')'SYNVP SOLX%REGINS')');
SYNVP SOL(X,K,ADD,0,NG-2*N1+5,NSFTE,FALL);
NEWLINE(6);
'FOR' I:=1 'STEP' 1 'UNTIL' NSFTE 'DO'
'BEGIN' WRITETEXT('('('4C')'LOCAL%DISPLACEMENTS%FOR%FORCE%SET')');
PRINT(I,2,0);
WRITETEXT('('('3CS')'NODE'('5S')'X-DIRECTION'('8S')'Y-DIRECTION'('18S')'NODE'('5S')'X-DIRECTION'('8S')'Y-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-1,11,0,8);SPACE(2);
PRINT(Q[V,11,0,8);SPACE(1);PRINT(Z,3,0);SPACE(2);
PRINT(Q[V+1,11,0,8);SPACE(2);PRINT(Q[V+2,11,0,8);
'END';
'IF' NNODE-N1 'GT' U 'THEN'
'BEGIN' NEWLINE(2); PRINT(NNODE-N1,3,0); SPACE(2);
V:=2*(NNODE-N1);
PRINT(Q[V-1,11,0,8); SPACE(2); PRINT(Q[V,11,0,8);
'END';
V:=NG-2*N1;
WRITETEXT('('('4C')'CRACK%TIP%DISPLACEMENT%IN% X-DIRECTION='));
PRINT(Q[V+3,11,0,10);
WRITETEXT('('('4C')'CRACK%TIP%DISPLACEMENT%IN% Y-DIRECTION='));
PRINT(Q[V+2,11,0,10);
WRITETEXT('('('4C')'MODE1%STRESS%INTENSITY%FACTOR%K1='));
PRINT(Q[V+5,11,0,10);
WRITETEXT('('('4C')'MODE2%STRESS%INTENSITY%FACTOR%K2='));
PRINT(Q[V+4,11,0,10);
WRITETEXT('('('4C')'RIGID%BODY%ROTATION%U='));

```

```
PRINT(OLV+1,I1,0,10);
'END';
NVAR1:=1;
GETARRAY(10,NVAR1,K);
GETARRAY(10,NVAR1,ADD);
RESIDUAL(K,Q,ADD,NG-2*N1+5,NSETF);
'END';
'END';
'END';
FAIL: 'END';
'END';
****

↑*****
```

```

PROCEDURE RESIDUAL(K,O,ADD,NREF,NSETF);
VALUE NREF,NSETF;
INTEGER NREF,NSETF;
INTEGER ARRAY ADD;
REAL ARRAY K,O;
BEGIN INTEGER I,J,Z,DI,DJ,C1,CJ;
ARRAY F1:NREF,1:NSETF;
FOR I:=1 STEP 1 UNTIL NREF DO
FOR J:=1 STEP 1 UNTIL NSETF DO F1[J]:=0.0;
FOR Z:=1 STEP 1 UNTIL NSETF DO
FOR I:=1 STEP 1 UNTIL NREF DO
BEGIN C1:=I-ADD[I]+ADD[I-1]+1;
FOR J:=1 STEP 1 UNTIL NSETF DO
BEGIN IF J<LE I THEN
BEGIN DI:=I; DJ:=J;
GOTO LAB1;
END ELSE
BEGIN C1:=J-ADD[J]+ADD[J-1]+1;
IF J<LE I THEN GOTO LAB2;
DI:=J; DJ:=I;
END;
LAB1: F1[I,Z]:=F1[I,Z]+O[J,Z]*K[ADD[I]-DI+DJ];
LAB2: END;
END;
WRITE(EXT(1)('('40')'RESIDUAL FORCES')));
FOR J:=1 STEP 1 UNTIL NSETF DO
BEGIN NEWLINE(4);
FOR I:=1 STEP 1 UNTIL NREF/2 DO
BEGIN PRINT(F1[2+J-1,I],,5); SPACE(2);
PRINT(F1[2+J,I],0,5);
NEWLINE(2);
END;
END;
END OF PROCEDURE RESIDUAL;

```


10.D.4 Double Tip, Combined Modes I and II

This program is used for the determination of Mode I and II stress intensity factors, K_I and K_{II} for problems of a flat plate containing an internal crack. The program is different to the others contained in this Appendix in that provision is made for crack surface closure over part or all of its length. The program flowchart is discussed and a list of input data instructions are presented to enable a user to run the program.

A flowchart and computer listing of the program follows, the steps are discussed with reference to the flowchart, as:

- A. - Procedures are declared, see Section I and Chapter 5.5
- B. - Number of jobs are read (NJ~~OB~~) and a loop is constructed around this number.
- C. - Number of elements (NELEMT), nodes (NN~~ODE~~), sets of forces (NSETFS), skewed nodes (NSKEW), crack surface nodes (SURN~~O~~), sets of constraints (NSETC) are read together with the thickness of the plate (THICK).
- D. - Loop on number of sets of constraints is constructed, mesh details are read (INPUT), address sequence is constructed (ADDARRAY), bounds of the stiffness matrix are calculated (MM2TKAEND) and the elasticity matrix is formed (CMATRIX). Also, tip details are read, number of interface nodes, N1, core radius, R~~O~~, and inclination of crack tip, AL.
- E. - Load vector is formed (LOADAPP).
- F. - Stiffness matrix is assembled, coefficients are modified for skewed displacements (MM2TASEM).
- G. - Load vector is modified for skewed nodes (SKEWLOAD).
- H. - Modified stiffness matrix formed (MM2T), kinematic constraints are applied (GE~~CM~~BBC) and equations are solved (SYMBVS~~O~~L).

- I. - $W[2,1] = K_{I1}$ and $W[4,1] = K_{I2}$, subscripts referring to tips 1 and 2 respectively.
- J. - If both K_{I1} and K_{I2} are less than zero, then problem is non-singular program terminates.
- K. - If either K_{I1} or K_{I2} are less than zero, then crack surface nodes are constrained appropriately (COMPVIO) and equations are resolved.
- L. - Stress intensity factors and nodal point displacements are printed.

Input Data

An example of an internal mixed mode crack problem is shown in Fig. 10.D.4, with the node numbering around cores included. Note that crack surface nodes on opposite faces are contiguous.

- (A) NUMBER OF JOBS TO BE ANALYSED
- (B) for first job,
 - (1) JOB TITLE, followed by END OF TITLE
 - (2) NUMBER OF ELEMENTS
 - (3) NUMBER OF NODES
 - (4) THICKNESS OF PLATE
 - (5) NUMBER OF SETS OF FORCES^{(1)*}
 - (6) NUMBER OF NODES WHERE SKEWED DISPLACEMENTS ARE REQUIRED⁽²⁾
 - (7) NUMBER OF CRACK SURFACE NODES WHERE SKEWED DISPLACEMENTS ARE APPLIED⁽³⁾
 - (8) NUMBER OF SETS OF CONSTRAINTS.
- (C) for first set of constraints,
 - (1) NUMBER OF SETS OF FORCES FOR THIS SET OF CONSTRAINTS
 - (2) NUMBER OF NODES WHOSE COORDINATES ARE SPECIFIED⁽⁴⁾
 - (3) for the number specified in (C)(2),
 - (a) NODE NUMBER
 - (b) X-COORDINATE
 - (c) Y-COORDINATE
 - (d) CODE⁽⁵⁾

* See Notes at end of Input Data.

- (e) VALUE OF PRESCRIBED DISPLACEMENT OR LOAD IN X-DIRECTION
- (f) VALUE OF PRESCRIBED DISPLACEMENT OR LOAD IN Y-DIRECTION
- (4) ELEMENT NUMBER AND NODAL CONNECTIONS ARE READ⁽⁶⁾
- (5) NODE NUMBER OF FIRST NODE STARTING SECOND CORE⁽⁷⁾
- (6) for tips 1 and 2, input
 - (a) NUMBER OF NODES ON CORE/ELEMENT INTERFACE(N1)
 - (b) RADIUS OF CORE ELEMENT (RØ)
 - (c) ANGLE OF CRACK TIP TO +VE AXIS (DEGREES)
- (7) TYPE OF PROBLEM (0 - plane stress, 1 - plane strain)
- (8) ELASTIC CONSTANTS (E, ν , G, E, ν)
- (9) If the number of sets of forces for this set of constraints is greater than 1, then input 9(a-e) else go to (10).
 - (a) NUMBER OF SPECIFIED FORCES
 - for the number specified in 7(a) input
 - (b) NODE NUMBER
 - (c) KODE⁽⁵⁾
 - (d) VALUE OF PRESCRIBED LOAD IN X-DIRECTION
 - (e) VALUE OF PRESCRIBED LOAD IN Y-DIRECTION
- (10) for the number of skewed nodes specified in (B)(6)⁽⁸⁾, input
 - (a) NODE NUMBER
 - (b) ANGLE OF SKEW (DEGREES) TO + VE X-AXIS
- (D) for second set of constraints, input
 - (1) NUMBER OF NEW CONSTRAINTS
 - (2) NUMBER OF SETS OF FORCES IN THIS SET OF CONSTRAINTS
 - (3) for the number of new constrained nodes, which must include the nodal forces for this set of constraints, input
 - (a) NODE NUMBER
 - (b) KODE⁽⁵⁾
 - (c) VALUE OF PRESCRIBED LOAD IN X-DIRECTION
 - (d) VALUE OF PRESCRIBED LOAD IN Y-DIRECTION
 - (4) If (D)(2) is greater than 1, then repeat (C)(9a-e)

(5) Repeat (C)(10)

(E) For subsequent sets of constraints repeat from (D).

(F) For next job, repeat from (B).

Notes

- (1) This is the maximum number for any set of constraints as it sets array bounds.
- (2) This number includes crack surface nodes and any others which may be skewed.
- (3) The extent of displacement incompatibility, if present, is not known before solution of the stiffness equations. It is advisable, therefore, to include all the crack surface nodes, see following example.
- (4) See Note 1, Section 10.D.1.
- (5) See Note 2, Section 10.D.1.
- (6) See Note 3, Section 10.D.1.
- (7) The core elements or crack tips are numbered 1 and 2, see following example, with the numbering as shown in Fig. 10.D.4, the node number starting the second core, numbering in a counter-clockwise direction is 60.
- (8) Starting with the crack surface nodes,
Consider the example of Fig. 10.D.5, where curved crack surfaces have three sets of nodes numbered as shown. The upper crack surface is arbitrarily chosen as that containing nodes 19, 17, and 54. Displacements are skewed such that the new normal displacement to the crack surface was the old global y-displacement of that node. The order for input data is, upper, lower, etc. for each set, e.g.

19	α_1
81	α_1
17	α_2
60	α_2
54	α_3
35	α_3

Finally for skewed nodes remote from the crack surface; node numbers and angles of skew are input in the usual manner, Section I.

For the example shown in Fig. 10.D.4, the input data takes the form,

1

INCLINEDCRACK END OF TITLE

No. of Elements 235 Plate Thickness 1 14 1 1

No. of Nodes whose coords. are specified.

1	x_1	y_1	0	0	0
2					

1	35	37	1	36	19	18
2						
3						

60

17 $R_c, \text{tip1}$ 45

17 $R_c, \text{tip2}$ 225

1

E	ν	G	E	ν
77	45	93	45	
94	45	110	45	
114	45	115	45	
112	45	113	45	
111	45	52	45	
51	45	35	45	
34	45	18	45	

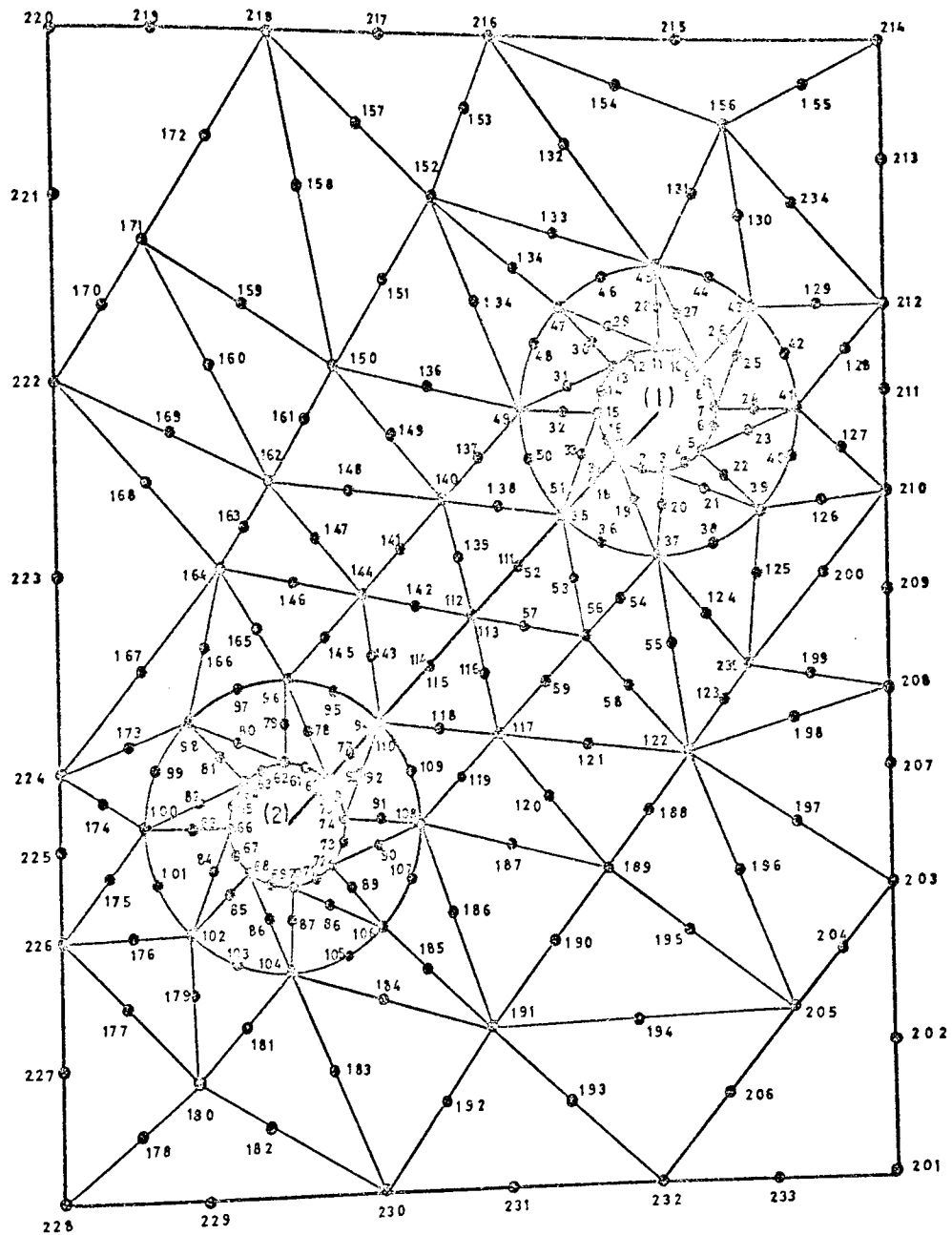


FIG 10.D.4 RECTANGULAR PLATE CONTAINING A STRAIGHT INCLINED CRACK.

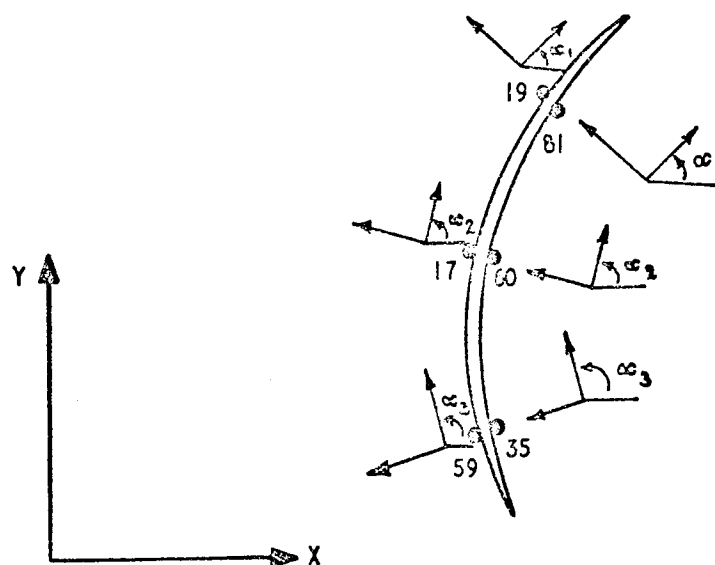
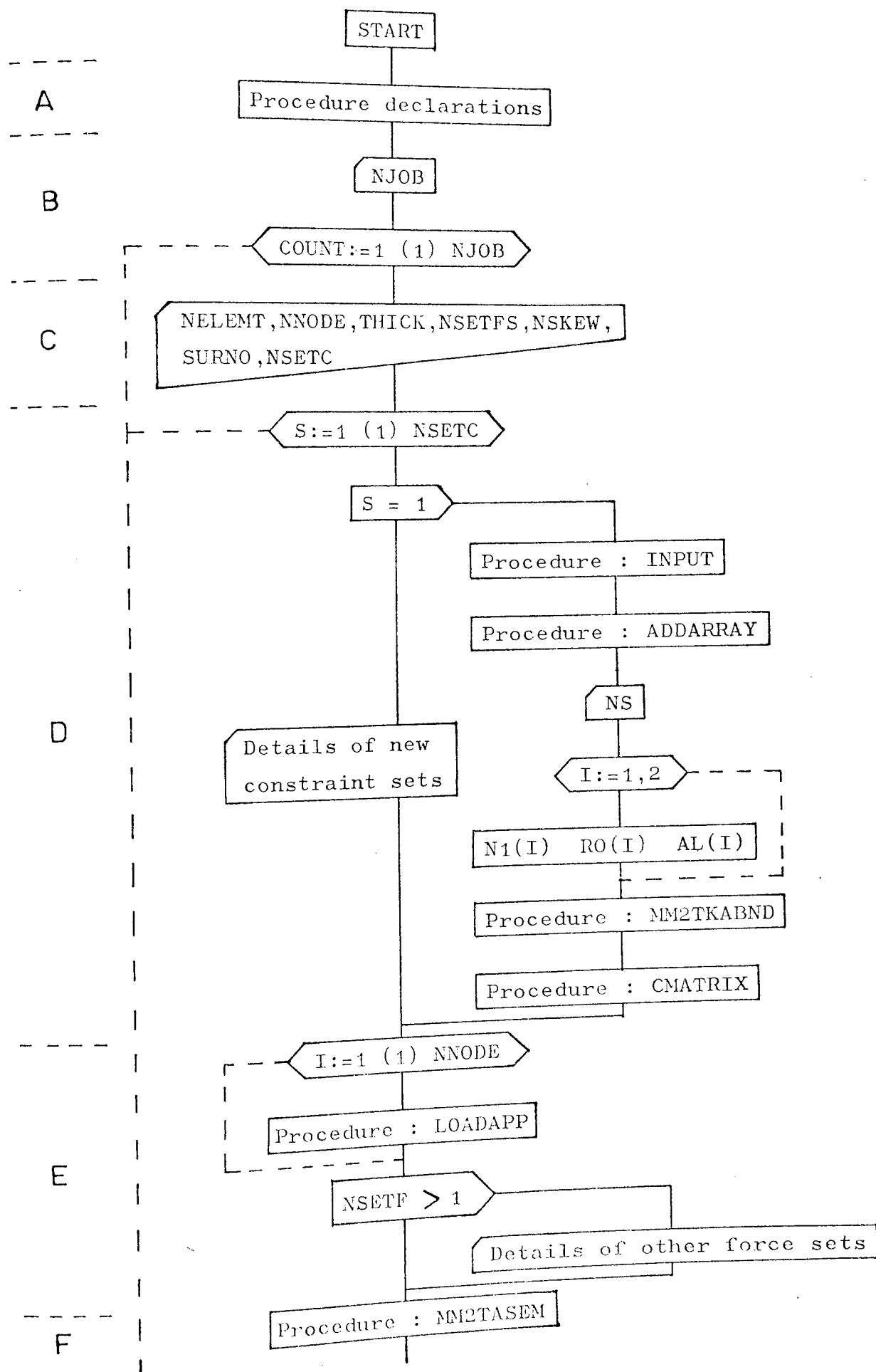
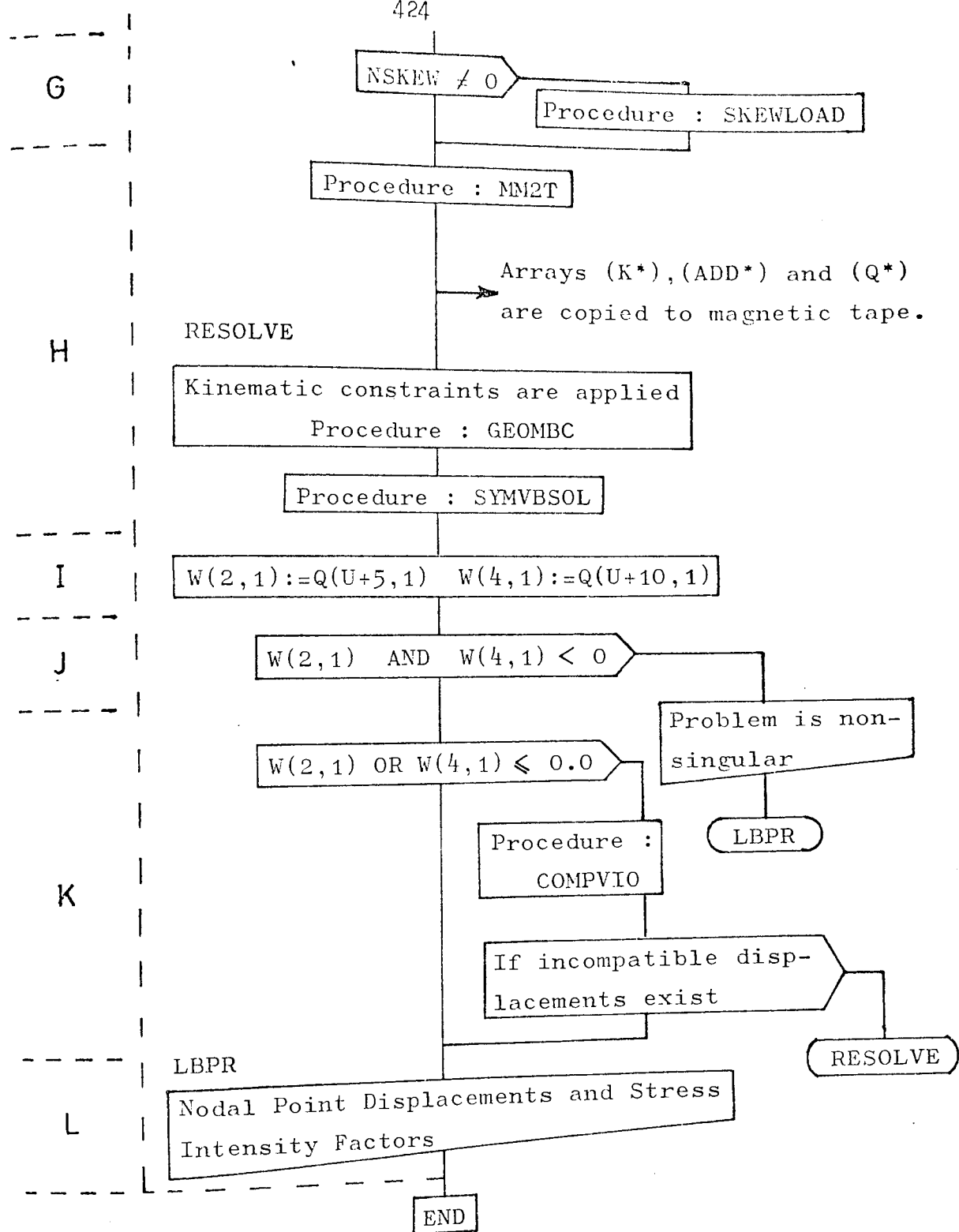


FIG. 10. D.5. CURVED CRACK WITH 3 SETS OF CONTIGUOUS NODES.

MIXED MODE TWO TIP PROGRAM FLOWCHART.





```

*****
#LISTING OF :FMP0824.MM2THES1(1/)      PRODUCED ON 22JUL75 AT 14.18.46
#OUTPUT BY LISTFILE IN :FMP0824.LISTS# ON 23JUL75 AT 18.55.46
DOCUMENT      MM2THES1

```

```

PROGRAM' (AXXX)
INPUT'D=CR0
OUTPUT'D=LP0
EXTENDED DATA'
EXTENDED'
BEGIN' 'COMMENT' THIS PROGRAM EMPLOYS THE ISOPARAMETRIC
                  FORMULATION FOR 2-D PROBLEMS OF PLANE STRESS/STRAIN
                  THE ELEMENT USED IS A SIX-NODE TRIANGLE:
INTEGER' NELEMT, NNODE, NSITES, NFREE, I, J, K, V, W, U, CASE, NSETF, NNEWC, S,
NIOB, COUNT, PR, T, NVAB1, NSKEW, SURNO, 01, 02;
REAL' DELTA, DETJ, THICK;
ARRAY' C(1:3, 1:3), A(1:5);
PROCEDURE' CREATSTORE(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, S: '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' AUX(L1, L2, L3, B, X, Y, U, P, Z);

PROCEDURE' LOADAPP(A, B, C, D, E, F);

PROCEDURE' SYMBVSOL(A, L, S, B) DIMENSIONS: (N, R) FAILURE EXIT: (FAIL);

PROCEDURE' ADDARRAY(NELEMT, NNODE, ADD, NODE);

PROCEDURE' MM2T(NG, NSETF, A, ADD, K, Q, CASE, THICK, N1, R0, GRT, AL, NS);

PROCEDURE' MM2TKABND(NFREE, N1, BAND, ADD, NS);

PROCEDURE' RESIDUAL(K, Q, ADD, NFREE, NSETF);

PROCEDURE' SKEWEDCON(Z, NODE, ANGSK, NOSK, NSKEW, KF, NSETF);

PROCEDURE' SKEWLOAD(NSETF, Q, NOSK, ANGSK, NSKEW, OP);

PROCEDURE' COMPVIO(Q, K, ADD, U, SURNO, W, NOSK, GEOMPC, GETARRAY, NVAB1,
RESOLVE);

```

```
PROCEDURE CMATRIX(C,CASE,A);
```

```
PROCEDURE INPUT(ADD,XX,YY,NFREE,NODE,NSETF,KODE,NSPEC,NSSETS,
                ULX,VLY,NELEMT,NODE);
```

```
PROCEDURE M2TASEM(NELEMT,X,XX,YY,DETJ,NODE,C,THICK,ADD,NFREE,AUX,
                SKEWFCOD);
```

```
NJOB:=READ;
```

```
FOR COUNT:=1 STEP 1 UNTIL NJOB 'DO'
```

```
  BEGIN
```

```
    WRTTEXT('('20')JOB#NAME-----');

```

```
    COPYTEXT('END%OF%TITLE');

```

```
    NELEMT:=READ;
```

```
    NODE:=READ;
```

```
    THICK:=READ;
```

```
    WRTTEXT('('20')NO%OF%ELEMENTS%----');

```

```
    PRINT(NELEMT,3,0);

```

```
    WRTTEXT('('20')NO%OF%NODES%-----');

```

```
    PRINT(NODE,3,0);

```

```
    NFREE:=NNODE*2;
```

```
    NSETF:=READ;
```

```
    NSKEW:=READ; SURNO:=READ;
```

```
  BEGIN
```

```
    INTEGER RAND,NSPEC,2,COMPA,NSETF,NS,CRT,NG;
```

```
    REAL 'ARRAY' XX,YY[1:NODE],ULX,VLY[1:NODE,1:NSETF],
              0[1:NFREE,1:NSETF],ANGSK[1:NSKEW+1],AI,RO[1:2];
```

```
    INTEGER 'ARRAY' NODE[1:NELEMT,1:6],KODE[1:NODE,1:NSETF],ADD[0:NFREE],
```

```
    NSKEW[1:NSKEW+1],N1[1:2];
```

```
    NSKEW:=READ; NG:=NFREE;
```

```
    FOR S:=1 STEP 1 UNTIL NSETF 'DO'
```

```
      BEGIN
```

```
        IF S=1 THEN BEGIN
```

```
          NSETF:=READ;
```

```
          INPUT(ADD,XX,YY,NFREE,NODE,NSETF,KODE,NSPEC,NSSETS,ULX,VLY,NELEMT,
                NODE);
```

```
          ADDARRAY(NELEMT,NODE,ADD,NODE);
```

```
          NS:=READ;
```

```
          FOR I:=1,2 'DO'
```

```
            BEGIN N1[I]:=READ; RO[I]:=READ; AI[I]:=READ; END;
```

```
            IF AI[I] GE N1[2] THEN CRT:=N1[1] ELSE CRT:=N1[2];
```

```
          M2TKABND(NFREE,N1,RAND,ADD,NS);
```

```
          CMATRIX(C,CASE,A);
```

```
          WRTTEXT('('20')NO%OF%DEGREES%OF%FREEDOM----');

```

```
          PRINT(NFREE,3,0);
```

```
          NMLINE(2);
```

```
        END ELSE
```

```
          BEGIN NFWC:=READ;
```

```
            NSETF:=READ;
```

```
            FOR I:=1 STEP 1 UNTIL NFWC 'DO'
```

```
              BEGIN J:=READ;
```

```
                KODE[J,1]:=READ; HX[J,1]:=READ; VLY[J,1]:=READ;
```

```
              END;
```

```
            END;
```

```
            FOR I:=1 STEP 1 UNTIL NFREE 'DO'
```

```

FOR I:=1 'STEP' 1 'UNTIL' NSSETF 'DO' Q[1,1]=0.0;
FOR I:=1 'STEP' 1 'UNTIL' NNODE 'DO'
  LOADAPP(KODE[1,1],ULX[1,1],VLX[1,1],0,1,1);
IF NSSETF 'GT' 1 'THEN'
  BEGIN
    FOR I:=2 'STEP' 1 'UNTIL' NSSETF 'DO'
      BEGIN
        NSPEC=READ;
        WRITE TEXT('('('20')'FORCZSET----')');
        PRINT(1,3,0);
        WRITE TEXT('('('20')'NODE('53')'TYPE('65')'--DISP('75')'Y-DISP
          ('(215')'ORZLOAD('65')'ORXLOAD'))');
        FOR J:=1 'STEP' 1 'UNTIL' NSPEC 'DO'
          BEGIN
            K:=READ; KODE[K,1]=READ; ULX[K,1]=READ; VLX[K,1]=READ;
            LOADAPP(KODE[K,1],ULX[K,1],VLX[K,1],0,K,1);
            NLINE(2);
            PRINT(K,3,0); SPACE(2);
            PRINT(KODE[K,1],3,0); SPACE(2);
            PRINT(ULX[K,1],0,4);
            PRINT(VLX[K,1],0,4);
          END;
        END;
      END;
    FOR I:=1 'STEP' 1 'UNTIL' NSKEW 'DO'
      BEGIN
        NOSK[1]=READ;
        ANGSK[1]=READ;
      END;
    NLINE(6);
    BEGIN
      INTEGER SUB1,SUB2,SUB3,SUBSCP;
      REAL AARRAY(4); R[4]; RAND1,R[1:6,1:4];
      MPTASEM(ELEMENT,K,XX,YY,DETJ,NODE,C,THICK,ADD,HEREF,ADD,SKEWEDCON);
      IF NSKEW 'NE' 0 'THEN' SKEWLOAD(NSSETF,Q,NOSK,ANGSK,NSKEW,1);
      FOR I:=1 'STEP' 1 'UNTIL' NSKEW 'DO'
        IF NOSK[1] 'LE' (NS-1) 'THEN' NOSK[1]=NOSK[1]-N1[1];
        ELSE NOSK[1]=NOSK[1]-N1[1]-N1[2];
      NLINE(6);
      MPT(NSC,NSSETF,A,ADD,K,Q,CASE,THICK,N1,R0,GRT,AL,NS);
      CREATE STORE(10,('ED'),'('KAEILEST'),1,40000);
      NVAB1=1;
      PUTARRAY(10,NVAB1,K);
      PUTARRAY(10,NVAB1,ADD);
      PUTARRAY(10,NVAB1,0);
      U=ENG-2*(N1[1]+N1[2]);
      COMMENT INTRODUCTION OF KINEMATIC CONSTRAINTS;
      RESOLVE: FOR I:=1 'STEP' 1 'UNTIL' NNODE-N1[1]-N1[2] 'DO'
        BEGIN
          IF I 'LE' NS-N1[1]-1 'THEN' Z:=I+N1[1] 'ELSE'
            Z:=I+N1[1]+N1[2];
          IF KODE[2,1]=0 'THEN' GOTO XC1;
          IF KODE[7,1]=2 'THEN' GOTO XC2;
          GOMPRC(ULX[Z,1],2*I-1,0,K,U*10,1,ADD);
          FOR J:=2 'STEP' 1 'UNTIL' NSSETF 'DO' Q[1,J]=Q[1,1];
          IF KODE[2,1]=1 'THEN' GOTO XC1;
          KC2: GOMPRC(VLX[Z,1],2*I,0,K,U*10,1,ADD);
          FOR J:=2 'STEP' 1 'UNTIL' NSSETF 'DO' Q[1,J]=Q[1,1];
          XC1: END;
          WRITE TEXT('('('20')'SYNVPSOLV%REFGINS')));
          SYMBOLE(K,K,ADD,0,U*10,NSSETF,FAIL);
          W[2,1]=Q[1]+5,1; W[4,1]=Q[1]+10,1;
          IF W[2,1] 'LT' 0.0 'AND' W[4,1] 'LE' 0.0 'THEN'
            IF W[2,1] 'LT' 0.0 'AND' W[4,1] 'LE' 0.0 'THEN'
              BEGIN
                WRITE TEXT('('('20')'PROBLEM%IS%NOT%SINGULAR')));
                GOTO LBPR;
              END;
            IF W[2,1] 'LE' 0.0 'OR' W[4,1] 'LE' 0.0 'THEN'

```

