ON THE MECHANICAL BEHAVIOUR OF HUMAN TOOTH STRUCTURES:

An Application of The Finite Element Method of Stress Analysis.

Thesis presented in Two Volumes

for the Degree of Doctor of Philosophy

by

K.W.J. Wright, B. Tech.

VOLUME TWO

DEPARTMENT OF MECHANICAL ENGINEERING

BRUNEL UNIVERSITY

MARCH 1975

PAGE NUMBERING AS ORIGINAL

LIST OF SYMBOLS

The following list defines the principal symbols used in this thesis. Other symbols are defined in context. Rectangular and square matrices are indicated by square brackets [ ], and column vectors by braces { }.

X,Y,Z: $\bar{X},\bar{Y},\bar{Z}$: X',Y',Z':	Right-hand Cartesian coordinate axis systems.
E_z	Young's Modulus of elasticity in Z coordinate direction.
μ_{zx}	Poisson's ratio. Ratio of strain (passive) induced in X direction and the stress induced strain (active) in the Z direction.
G_{xy}	Modulus of rigidity in the XY plane.
σ_{xx}	Normal stress component in X coordinate direction.
ε_{yy}	Normal strain component in Y coordinate direction.
τ_{xy}	Shear stress component in the XY plane.
γ_{yz}	Shear strain component in the YZ plane.
u,v,w	Nodal displacement components in the directions of the corresponding Cartesian coordinate directions.
α_n	Generalized coordinate.
$\{f\}$	Column vector of displacement components.
[A]	Matrix of element nodal point coordinates.
$\{q\}$	Column vector of element nodal point displacements.
N_n	Interpolation function.
δ,η,ξ	Local element coordinate directions.

$\{F\}$	Column vector of element nodal point forces which are equivalent statically to the boundary stresses and internal distributed loading acting on the element.
$[k]$	Element stiffness matrix.
$\{F_d\}$	Column vector of element nodal point forces required to balance any internal distributed loading acting on the element.
$\{F_t\}$	Column vector of element nodal point forces required to balance any initial internal strains in the element.
$[B]$	Matrix relating element nodal point displacements and the internal element strains.
$[D]$	Material elasticity matrix.
$\{R\}$	Column vector of structural nodal point forces.
$[K]$	Structural stiffness matrix.
$\{d\}$	Column vector of structural nodal point displacements.
$[MBK]$	Modified arrangement of the structural stiffness matrix.
$[J]$	Jacobian transformation matrix.

CONTENTS

CHAPTER THIRTEEN

INTRODUCTION TO VOLUME TWO

DENTAL STRUCTURAL STRESS ANALYSIS

13 INTRODUCTION TO VOLUME TWO

DENTAL STRUCTURAL STRESS ANALYSIS

There are many different methods of structural stress analysis. The most commonly employed techniques for general engineering structures have already been discussed in Chapter Five. However, as explained in that chapter, some of the techniques available are either unsuitable for, or are incapable of handling the complexities which the dental structures present.

Because it is difficult, if not impossible, to obtain solutions to many 'real' engineering problems involving either complex material behaviour or boundary conditions, engineers have resorted to various numerical methods of analysis in order to obtain approximate but nevertheless 'acceptable' engineering solutions. Numerical solutions generally yield only approximate values of the unknown quantities at a limited or finite number of positions. The process of selecting the finite number of positions in the body or structure has been called discretization.

Structures, such as the simple frame shown in FIG. 13.1, are already discretized in that the structure consists physically of three separate members. By employing the displacement method of structural analysis the primary unknowns, i.e. the displacements of the frame

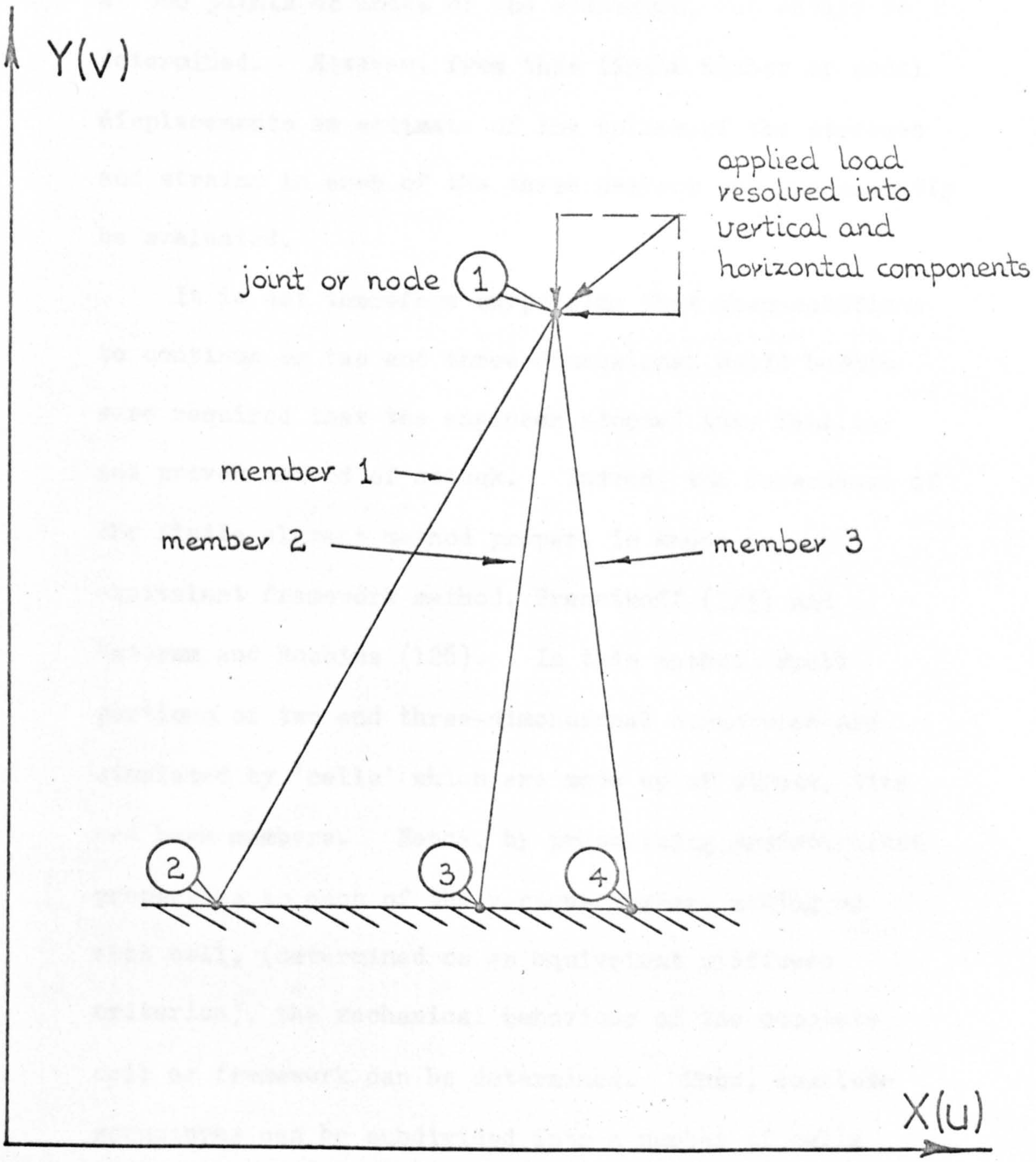


FIG. 13.1 Simple pin-jointed plane frame.
(Structure 'naturally' divided into finite elements.)

at the joints or nodes of the structure, can easily be determined. However, from this finite number of nodal displacements an estimate of the values of the stresses and strains in each of the three members can subsequently be evaluated.

It is not therefore surprising that when solutions to continua or two and three-dimensional solid bodies were required that the engineer adopted this familiar and proven method of attack. Indeed, the forerunner of the finite element method proper, is known as the equivalent framework method, Hrennikoff (125) and Yettram and Robbins (126). In this method, small portions of two and three-dimensional structures are simulated by 'cells' which are made up of struts, ties and beam members. Hence, by prescribing predetermined properties to each of the various members making up each cell, (determined on an equivalent stiffness criterion), the mechanical behaviour of the complete cell or framework can be determined. Thus, complete structures can be subdivided into a number of cells whose individual mechanical characteristics are determined ab initio.

The logical extension of the equivalent framework method was to divide structures up into cells or elements which are physically similar to the structure itself. Thus, in the finite element method, plate structures

are divided up and simulated by cells that are in fact small pieces of plate, and three-dimensional bodies by three-dimensional solid elements. Turner, Clough, Martin and Topp (127), are generally acknowledged to have been the first to have adopted this approach.

The development of the finite element method proper, was only made possible by the advent of the electronic digital computer. Obviously, the simple three member frame previously discussed can easily be solved by hand as only a small number of unknowns are involved. However, for structures consisting of a large number of members or cells, and with each cell or finite element having more complex mechanical behaviour patterns than the simple tension/compression members, numerical solutions are beyond the scope of 'hand' calculation. Consequently, as digital computers have increased in size (storage capacity), and speed, so the finite elements employed and the size of the structures investigated have become more sophisticated and complex. Simultaneously, the finite element approach which was initially conceived as a purely physical method of solving structural stress analysis problems, has been found to be equivalent to the more mathematically rigorous variational methods. In fact, the method has been developed to handle many other variational type field problems such as those due to fluid and heat flow phenomena.

The finite element method is treated in this thesis exclusively in terms of stress analysis. However, because the fundamentals of the method are very well documented, see for instance the works of Desai and Abel (128) and Zienkiewicz (129), only an outline of the important aspects are reiterated here. Nevertheless, a full and more detailed treatment is given for the derivation of the stiffness matrices of finite elements incorporating orthotropic material behaviour, as these, to the author's knowledge, are not yet available in the literature. These derivations appear in the later chapters which deal with the development of the computer programs employed for carrying out the dental structural analyses discussed in Volume One. Three types of finite elements were employed. These were axisymmetric, two-dimensional and three-dimensional elements. However, because of the small size of the computer installation available, each of the element types was incorporated into a separate analysis program.

The programs were developed so as to suit Brunel University's I.C.L. 1903A series computer installation. However, during the course of the project the installation changed considerably and so the programs are perhaps not now so compatible with the machine's capability. Also, the computer staff were initially more conversant with ALGOL and so the programs were written in this language.

CHAPTER FOURTEEN

THE FINITE ELEMENT METHOD

14. THE FINITE ELEMENT METHOD

14.1 GENERAL DESCRIPTION OF THE METHOD

The finite element method of structural stress analysis utilizing the displacement approach, can best be described by a sequential process which consists of a series of seven basic steps. The pattern of these steps is shown by the flow diagram in FIG. 14.1.

14.2 DISCRETIZATION

This step involves the subdivision of the structure to be analysed into a grid or meshwork of finite elements. The type of the elements chosen to represent the structure obviously depends upon the form of the structure being analysed. For example, structures which possess rotational symmetry are usually represented by axisymmetric finite elements, see FIG. 7.13. Other types of finite elements employed include two-dimensional planestress and planestrain elements, three-dimensional elements and elements that have been developed for special applications such as thick plate and shell structures.

Sometimes it is possible to represent fully three-dimensional structures by only a planestrain system of finite elements, see for example Zienkiewicz (129) page 63. This legitimate simplification is very desirable purely from the economic viewpoint. However,

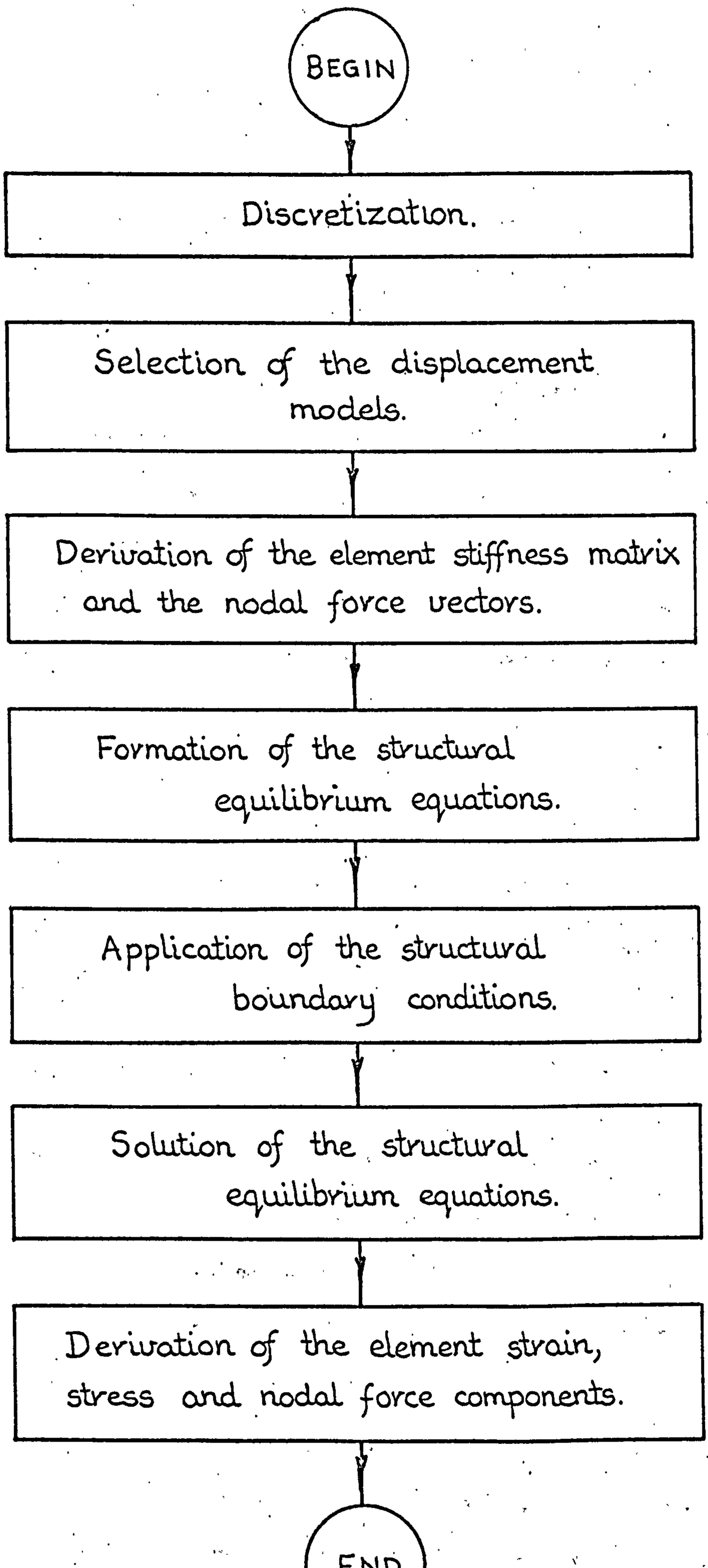


FIG. 14.1 Flow diagram of the finite element analysis method utilizing the displacement approach.

this will become more apparent later on when the cost of using fully three-dimensional finite element models is discussed.

The engineer must consider carefully the type, size and arrangement of the finite elements he is to employ to model the structure,in order that an effective structural representation is achieved. However, he must also consider at this juncture step two in the flow diagram of FIG. 14.1, and give some thought as to the type of the displacement models which will be used for the particular finite elements selected. This will ultimately affect the fineness of the subdivision and in particular the representation of the critical areas in the structure which are of special interest, e.g. areas of high stress concentration.

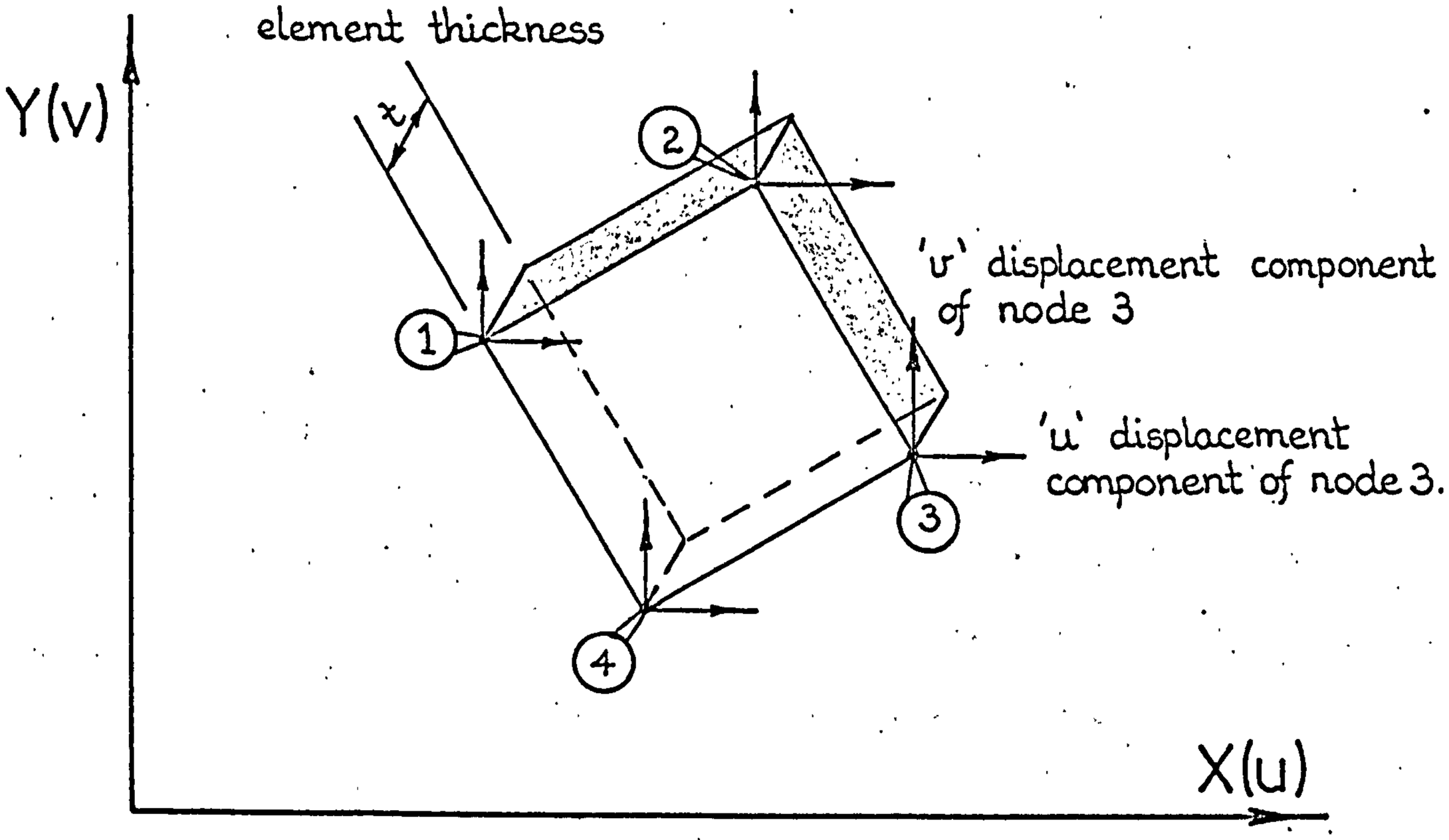
Structural discretization demands skill, competence, experience and judgement on the part of the analyst. In some cases,the extent to which the structure should be modelled may not be clearly defined, e.g. to what extent should the supporting alveolar bone be included in an analysis which is to determine the force distribution around the root of a tooth? Of course, only the significant portion of the alveolar bone need be considered and discretized. Indeed, this must be so,again purely from the aspects of practical limitations and economics.

It is obvious therefore from the foregoing, that the initial step of deciding upon the form of the discretization to be employed is a very important one. Indeed, it is one which will govern the validity of the solutions obtained from the ensuing analyses.

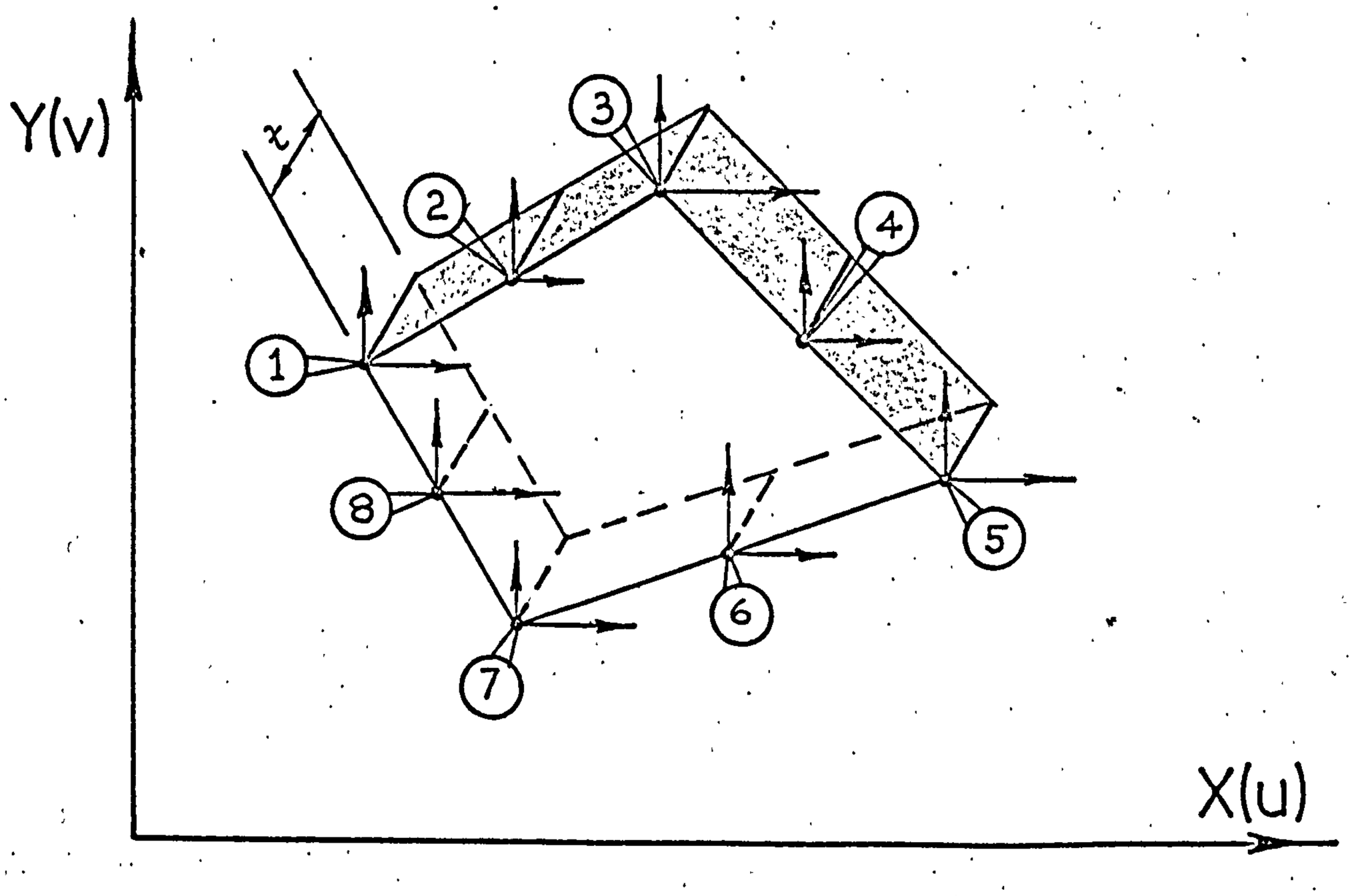
14.3 SELECTING THE DISPLACEMENT MODELS

As mentioned earlier in Chapter Thirteen, the numerical methods employed to solve general structural stress analysis problems provide approximate values of the desired unknown quantities only at a discrete number of points. Using the displacement approach of the finite element method, it is the displacements of the structure at these discrete points that are considered as being the primary unknown quantities. Although it is not essential, the discrete points or nodes in the structure are generally selected to occur along the boundaries of the individual elements forming the structural subdivision. Consequently, it is only through this system of boundary nodes that the elements are assumed to be interconnected.

The number of nodes which each element is prescribed and the number of degrees of freedom prescribed to each node, is quite arbitrary. FIG. 14.2 shows two rectangular plane stress finite elements. Here, element 1 has four nodes, one at each corner, with each node prescribed two separate degrees of freedom. That is,



a) Element One.



b) Element Two.

FIG. 14.2 Two planestress finite elements having 4 and 8 nodes respectively. Each node has 2 degrees of freedom.

one degree of freedom in the X coordinate direction, represented by a 'u' displacement component, and one in the Y coordinate direction and represented by a 'v' displacement component. On the other hand, element 2 has been prescribed eight nodes, one at each corner as before and also one node at the midpoint of each side. It must be pointed out that for this two-dimensional case, the elements have a specific thickness and so each node is not simply a point but is a line or a line node.

To enable the mechanical behaviour of each element to be determined, that is, the element's influence or stiffness coefficients, it is necessary to define the variation of the displacement components within the element in terms of the nodal values. The form of the variations in the displacements throughout each element are prescribed by what are known as displacement functions or models. Obviously, this assumed or prescribed variation represents only approximately the actual displacement distribution of the actual or real structure. Consider for example element number 1 in FIG. 14.2. In this case, the variation of the u displacement within the element could be assumed to take the form of the polynomial

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy$$

Therefore, if the u displacements for nodes 1, 2, 3 and 4 are substituted in turn into the above equation together

with the appropriate nodal coordinates, the four unknown quantities α_1, α_2, α_3 and α_4 can be evaluated. Consequently, knowing the values of these four quantities and the four nodal displacements, the u displacement at any point within the element can be determined. From this example, it follows that for element 2 in FIG. 14.2 which has eight nodes, a displacement model of the form

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy + \alpha_5 x^2 + \alpha_6 y^2 + \alpha_7 xy^2 + \alpha_8 x^2 y$$

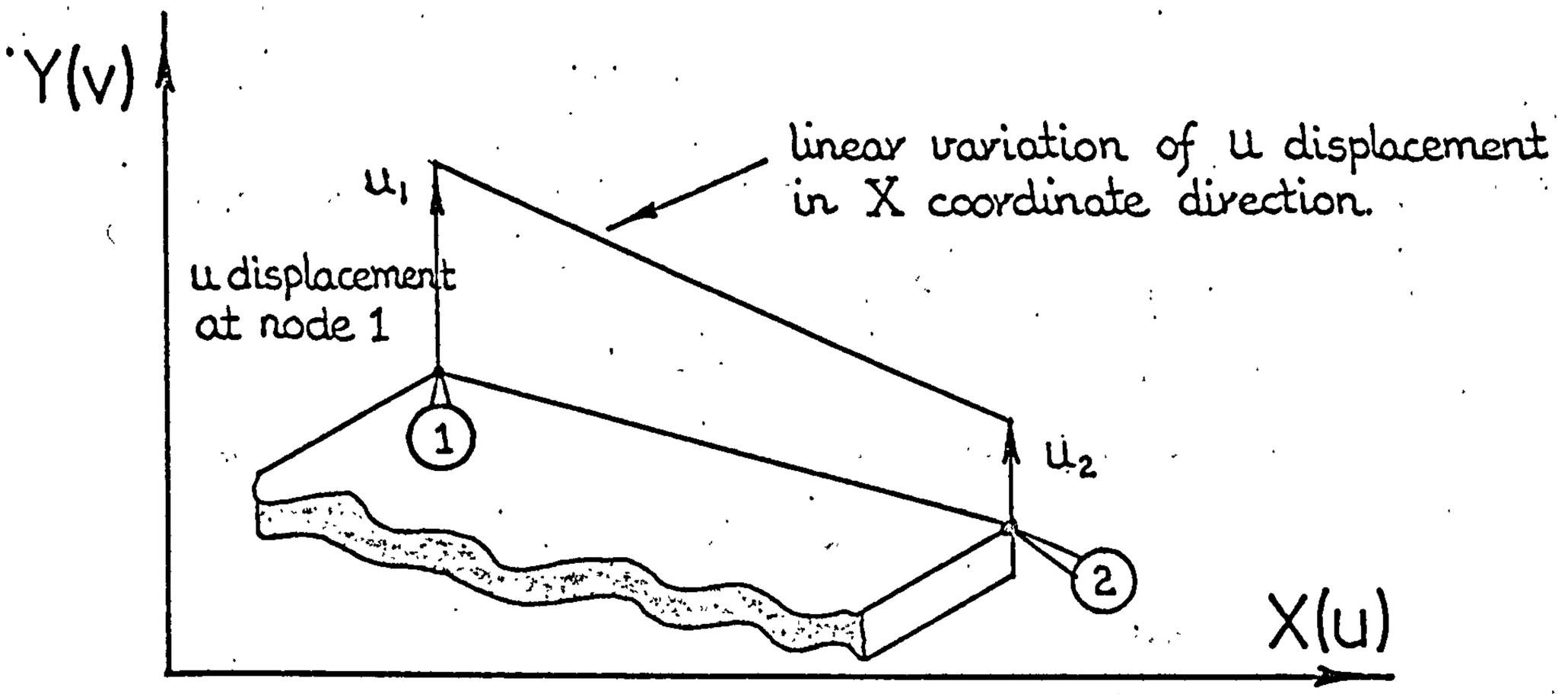
can be evaluated.

With the first displacement model, the u displacement distribution within the element in the X coordinate direction could take a linearly varying form. This condition is illustrated along the side joining nodes 1 and 2 in FIG. 14.3a. (Note that the 'variation' of the u displacement throughout the element can be

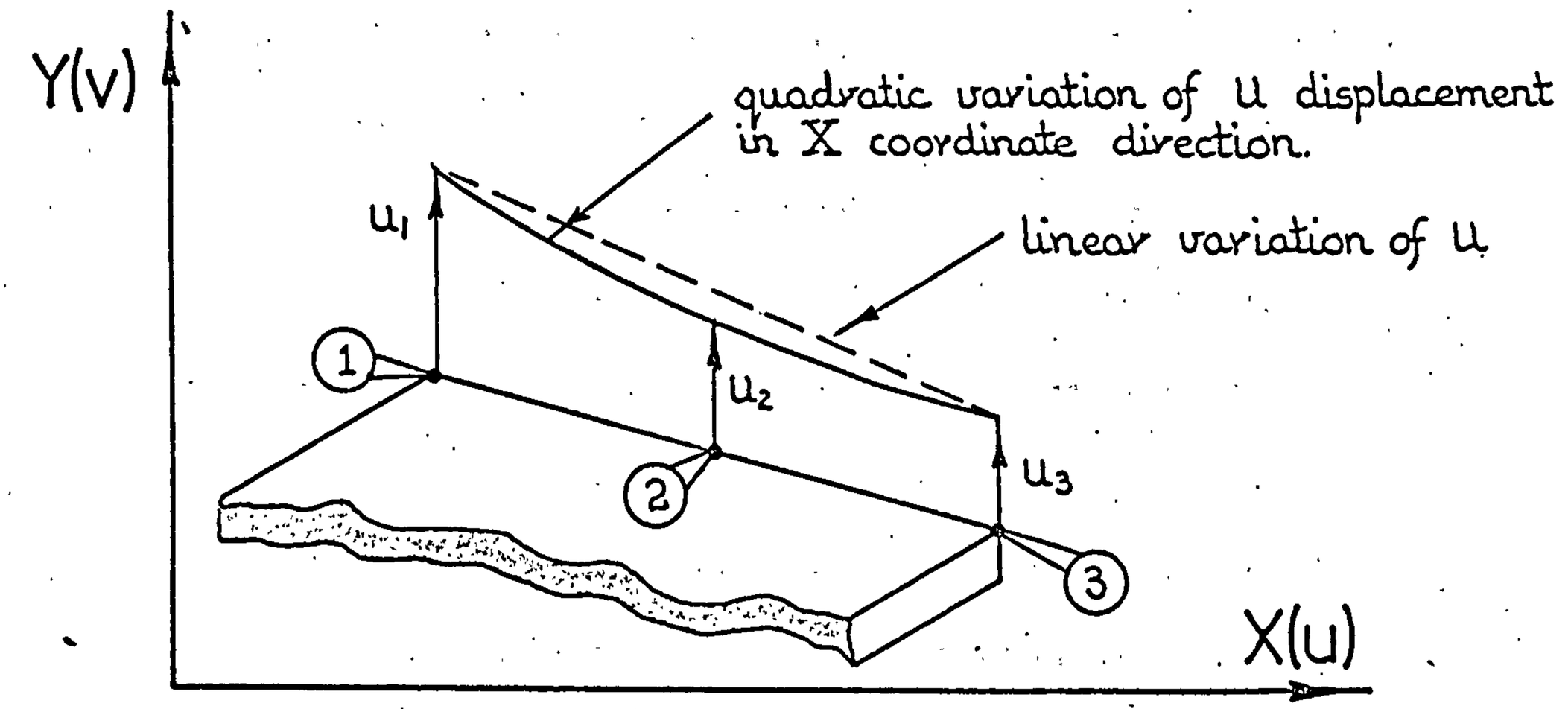
$$u = \alpha_1 \text{ provided } \alpha_2 = \alpha_3 = \alpha_4 = 0$$

This constant value of u signifies a pure rigid body movement and results in a no-strain condition.)

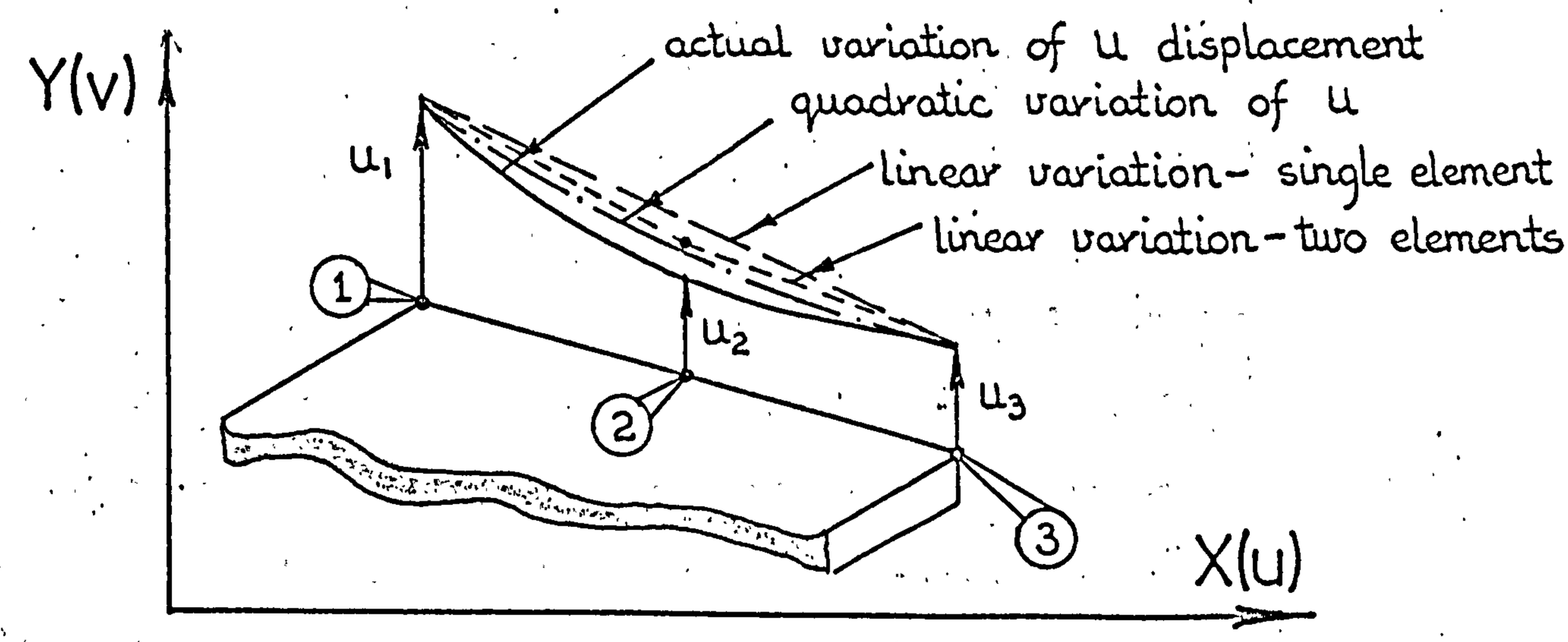
However, using the second displacement model the u displacement distribution throughout the element in the X coordinate direction, can take a quadratic varying form. FIG. 14.3b illustrates this case along the equivalent element side as before. The quadratic displacement model is obviously more refined than the linear model and can therefore represent, for the same



a) Linear displacement model.



b) Quadratic displacement model.



c) Linear and quadratic approximations to an actual displacement distribution.

FIG. 14.3 Examples of linear and quadratic displacement models.

element mesh, a more rapidly varying displacement field, see FIG. 14.3c. (Note that with the quadratic displacement model the condition of a pure rigid body movement can again be simulated provided that the αs apart from α_1 assume zero values.)

The simulation obtained by using the linear displacement model could be improved for the example discussed above by reducing the size of the element mesh, see FIG. 14.3c. Thus,the engineer has the choice of either using many of the simple 'linear' elements or fewer of the more refined 'quadratic' elements to simulate the displacement distribution. The choice depends upon the accuracy required and also upon economics. It may be cheaper from the computational aspect to use the quadratic type of displacement model. Also, it may be possible to use mixed elements, i.e. both linear and quadratic models in the same problem. However, as will be discussed later, the displacement compatibility at the junction of the two types of elements, (which mathematically is a necessary requirement), may be difficult to achieve.

It is worthwhile to emphasize here that engineering experience and a general 'feel' for the problem is required as it is uneconomic to 'over mesh' or bver model' a structure, i.e. to use too many elements to simulate the structure's displacement pattern. Also, it can be

seen from FIG. 14.3c that if the displacement variation across the element is in fact a linearly varying one in the actual structure, then the use of the higher order quadratic displacement model could not obviously improve the simulation.

In selecting the order of the displacement model for a particular finite element, various criteria have to be observed. As the displacement model chosen limits the number of the degrees of freedom of the element, the derived stiffness of the element is consequently over estimated. Thus,it follows that for any given loading regime, the simulated structure will deform less than the actual structure. However, if the subdivision of the structure is made finer, then the approximate displacement distribution obtained should CONVERGE to the actual displacement condition of the real structure. Nevertheless, in order that the finite element solution converges to the actual solution, the displacement models employed must be such that the displacements are continuous within each element. Also, the displacements along the edges or faces of the elements must be compatible with those of their neighbours. In addition, the displacement models should incorporate both rigid body (or zero strain) and constant strain states, see Desai and Abel (128) pages 80-81, and also the condition that the strains at the element interfaces are indeed finite values, Zienkiewicz (129) pages 35-36.

As well as satisfying the above conditions for convergence, other factors must be considered when selecting the displacement models. One of the more important aspects is that the model should not be dependent upon the orientation of the finite element. That is to say, the displacement model should be geometrically isotropic. One way of ensuring this is to select the variable terms of the polynomial on the basis of the Pascal triangle. FIG. 14.4 illustrates the Pascal triangle for variables in two, i.e. X and Y, dimensions. Using this figure, it can be seen that for an eight degree polynomial, either the x^3 and y^3, (or alternatively the x^2y and the y^2x), are the two additional cubic terms which should be selected to maintain geometric isotropy.

In formulating the individual element stiffness coefficients, it is necessary to express the displacements anywhere within the element in terms of the element's nodal displacements. However, there are primarily two methods of doing this, either by employing generalised coordinates or by using interpolation functions. Consider the four-noded plane stress element shown in FIG. 14.2a. Here we have two displacement variables, u corresponding to the X coordinate direction, and v corresponding to the Y coordinate direction. The displacement models selected

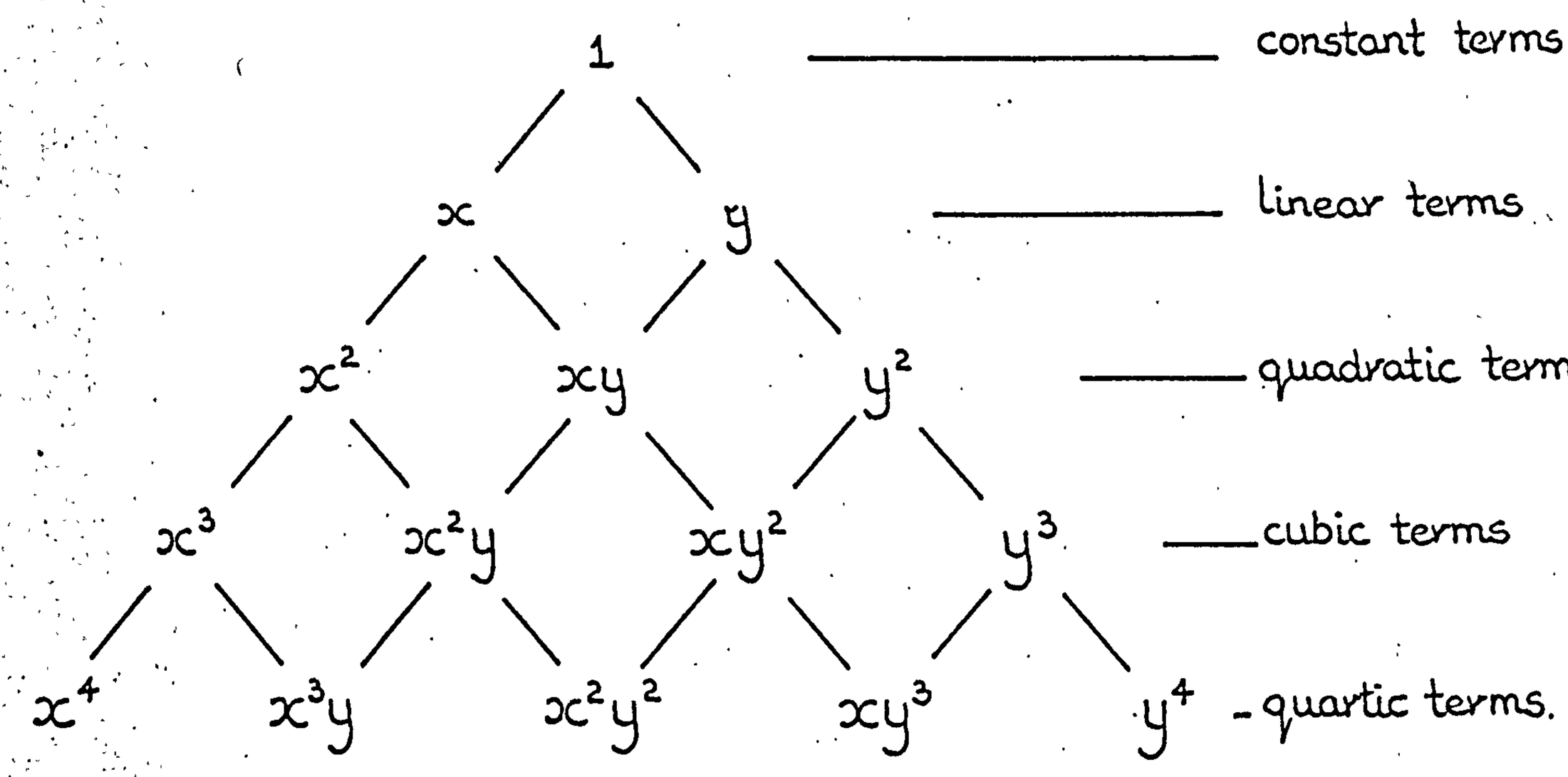


FIG. 14.4 Pascal triangle for selection of variables in two, i.e. X and Y coordinates.

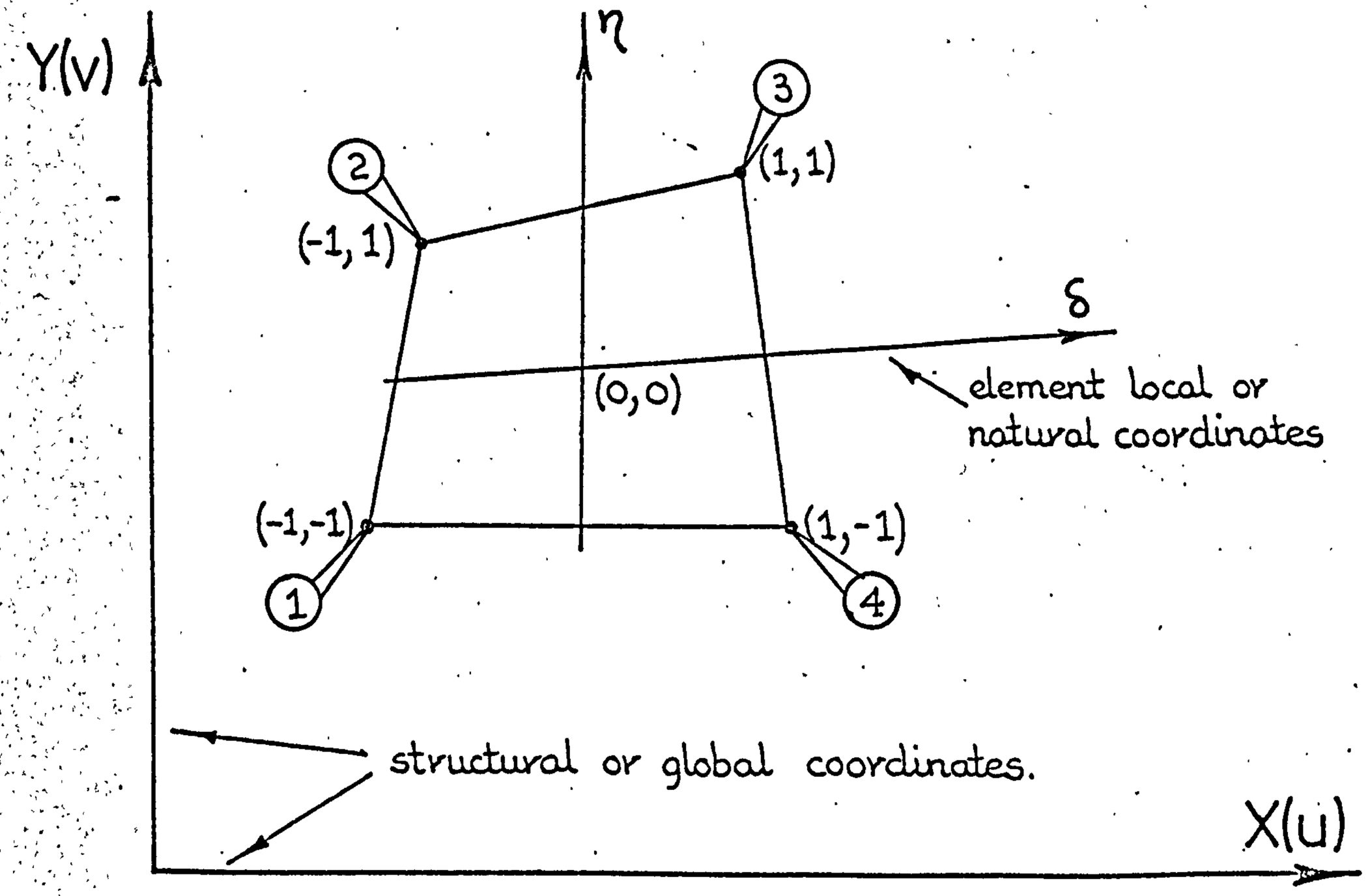


FIG. 14.5 Four-noded planestress finite element described in terms of local or natural coordinates.

must have the same number of components as the element has degrees of freedom. Consequently, as the element has four each of both the u and v degrees of freedom, the displacement models must be of the form

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 xy$$

$$\text{and } v = \alpha_5 + \alpha_6 x + \alpha_7 y + \alpha_8 xy \qquad 14.1a$$

The αs are known as the generalized coordinates.

Expressed in matrix form equation 14.1a becomes

$$\{f\} = \begin{Bmatrix} u(x,y) \\ v(x,y) \end{Bmatrix} = [C]\{\alpha\} \qquad 14.1b$$

Now by substituting the nodal coordinates into matrix [C] we can evaluate the corresponding nodal displacements

$$\begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_3 & y_3 & x_3y_3 \\ 1 & x_4 & y_4 & x_4y_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x_4 & y_4 & x_4y_4 \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \end{Bmatrix} \qquad 14.2a$$

or

$$\{q\} = [A]\{\alpha\} \qquad 14.2b$$

where $\{q\}$ is a column vector of the element's nodal displacements. Therefore, solving equation 14.2b we get

$$\{\alpha\} = [A]^{-1} \{q\} \qquad 14.3$$

and therefore by substituting this result into equation 14.1b we are able to obtain the displacements anywhere within the element in terms of the element's nodal values.

i.e.

$$\{f\} = [C] [A]^{-1} \{q\} \qquad 14.4$$

or

$$\{f\} = [N] \{q\} \qquad 14.5a$$

where

$$[N] = [C] [A]^{-1}$$

The main drawback with this generalized coordinate approach is that sometimes the inverse of the matrix [A] may not exist, that is to say [A] may be a singular matrix, Desai and Abel (128) page 85. In addition, considerable difficulty may be experienced in obtaining an inverse of [A] in general algebraic terms suitable for any element geometry. However, the interpolation function approach avoids these difficulties.

To express the displacement variation within an element in terms of its nodal values using interpolation functions, it is more convenient if element or local coordinates are employed. Consequently, the isoparametric element concept, pioneered by Zienkiewicz and his team at

Swansea has been introduced, Ergatoudis (130).

Isoparametric elements are elements whose displacement models not only express the displacement variation within the element in terms of its nodal values, but also, the coordinates of any point within the element in terms of the element's nodal coordinates. Using this approach, the four-noded plane stress quadrilateral element shown in FIG. 14.2a is described by a system of local or natural coordinates such that any point within the element can be specified by a set of dimensionless numbers whose magnitudes never exceed the value of unity, see FIG. 14.5. Thus, suitable interpolation functions * are selected such that equation 14.5a can be written as

$$\{f\} = \begin{Bmatrix} u(\delta,\eta) \\ v(\delta,\eta) \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} \quad 14.5b$$

* An interpolation function is basically a function which has the value of unity at a particular nodal point and a zero value at all the other nodal points.

where N_1, N_2, N_3 and N_4 are the interpolation functions.

For this case these are

$$N_1 = \frac{1}{4}(1-\delta)(1-\eta) \qquad N_2 = \frac{1}{4}(1-\delta)(1+\eta)$$

$$N_3 = \frac{1}{4}(1+\delta)(1+\eta) \qquad N_4 = \frac{1}{4}(1+\delta)(1-\eta)$$

Note however, that the displacement components u and v are now functions of the local element coordinates δ and η and not the structural or global coordinates X and Y. Also, because the element is isoparametric the X and Y coordinates at any point within the element can be expressed in terms of the element's nodal coordinates using the same interpolation functions. Hence,

$$x(\delta,\eta) = N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 \qquad 14.6$$

and $$y(\delta,\eta) = N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4$$

Because of the simplicity and generality of this method of defining the element's displacement variations, the interpolation function approach is more commonly employed than the generalized coordinate approach.

14.4 DERIVATION OF THE ELEMENT STIFFNESS MATRIX AND THE NODAL FORCE VECTORS

A system of nodal forces can be derived for a finite element such that they are equivalent statically to the boundary stresses and internal distributed loading acting on the element. Assuming therefore that the element behaves linearly elastically, it s characteristic relationship will be of the form,

$$\{F\} = [k]\{q\} + \{F_d\} + \{F_t\} \qquad 14.7$$

where $\{F\}$ is the column vector of element nodal forces which are equivalent statically to the boundary stresses and internal distributed loading acting on the element.

$[k]$ is the matrix of the element's stiffness coefficients.

$\{q\}$ is the column vector of the element's nodal displacements.

$\{F_d\}$ is the column vector of element nodal forces required to balance any internal distributed loading acting on the element, e.g. body weight forces.

and $\{F_t\}$ is the column vector of element nodal forces required to balance any initial internal strains in the element caused by, for example, a temperature variation.

Note that the product $[k]\{q\}$ represents the element nodal forces which are induced by the displacements of the nodes.

In order that the system of nodal forces is equivalent statically to the boundary stresses and any internal distributed loading acting on the element, it is necessary that the forces acting at a node have the same number of components as the element's nodal displacements. Similarly, the internal distributed loading $\{p\}$ say, acting on a unit volume of material, must be defined as having directions corresponding to those of the displacements $\{f\}$ at any point.

However, to make the nodal forces acting on the element equivalent statically to the actual boundary stresses and internal distributed loading, the principle of virtual work can be invoked. Basically, the principle of virtual work is a more generalized statement of the principle of the conservation of energy. It states that if a body, in static equilibrium, is given an arbitrary set of compatible small displacements, the external virtual work done by the nodal forces moving through the virtual nodal displacements is equal to the total internal virtual work done. The internal virtual work is equal to the integral of the products of all the stress components acting throughout the body and the corresponding system of virtual strains.

Now if a set of virtual displacements $\{\bar{q}\}$ are applied to the nodes of an element in static equilibrium then the virtual displacement of any point within the element can be defined by equation 14.5a or b as

$$\{f\} = [N]\{\bar{q}\} \qquad 14.8$$

Similarly, the virtual strains produced within the element can be expressed by means of equation 14.21, see section 14.8, as

$$\{\bar{\varepsilon}\} = [B]\{\bar{q}\} \qquad 14.9$$

Now the external virtual work is the sum of the products of the equivalent set of element nodal forces and the corresponding set of element virtual nodal displacements.

i.e. $W_{external} = \{\bar{q}\}^T \{F\}$ 14.10

Similarly, the internal virtual work due to the stresses and the internal distributed forces acting on the elemental volume dV of the element is

$$\frac{dW_{internal}}{dV} = \{\bar{\varepsilon}\}^T \{\sigma\} - \{f\}^T \{p\} \qquad 14.11$$

Therefore, by substituting from equations 14.8 and 14.9 we get*

$$\frac{dW_{internal}}{dV} = \{\bar{q}\}^T [B]^T \{\sigma\} - \{\bar{q}\}^T [N]^T \{p\} \qquad 14.12$$

Thus, the total internal virtual work of the element is obtained by integrating over the whole volume of the element and so

$$W_{internal} = \int \{\bar{q}\}^T [B]^T \{\sigma\}\, dV - \int \{\bar{q}\}^T [N]^T \{p\}\, dV \qquad 14.13$$

Therefore, by equating the total external virtual work with the total internal virtual work we get:-

$$\{\bar{q}\}^T \{F\} = \int \{\bar{q}\}^T [B]^T \{\sigma\}\, dV - \int \{\bar{q}\}^T [N]^T \{p\}\, dV \qquad 14.14$$

Now as this equation is valid for any set of values for the nodal displacements, remember the values chosen were selected quite arbitrarily, the multipliers can be equal.

* Note that by the rules of matrix algebra if

$$\{\bar{\varepsilon}\} = [B] \{\bar{q}\}$$

then

$$\{\bar{\varepsilon}\}^T = \{\bar{q}\}^T [B]^T$$

Therefore,

$$\{F\} = \int [B]^T \{\sigma\}\, dV - \int [N]^T \{p\}\, dV \qquad 14.15$$

Hence, by substituting for $\{\sigma\}$ from equation 14.24, see section 14.9,

$$\{F\} = \int [B]^T [D] \{\varepsilon\}\, dV - \int [B]^T [D] \{\varepsilon_o\} dV - [N]^T \{p\} dV \qquad 14.16$$

and substituting for $\{\varepsilon\}$ from equation 14.21, section 14.8

$$\{F\} = \int [B]^T [D][B]\{q\} dV - \int [B]^T [D] \{\varepsilon_o\} dV - \int [N]^T \{p\} dV \qquad 14.17$$

Therefore, by comparing equation 14.17 with equation 14.7 we see that

$$[k] = \int [B]^T [D][B] dV$$

$$\{F_d\} = -\int [N]^T \{p\}\, dV$$

and

$$\{F_t\} = -\int [B]^T [D]\ \{\varepsilon_o\}\, dV$$

It is important to note that the element stiffness matrix [k] is a square symmetric matrix*. The size of the matrix is equal to the number of nodes in the element multiplied by the number of degrees of freedom at each node. Hence, a 4-noded quadrilateral plane stress element having two degrees of freedom at each node will have a stiffness matrix of eight rows by eight columns, i.e. sixty four coefficients in all.

If an element on the boundary of the structure is subjected to a distributed external loading of say $\{g\}$ per unit area, a further loading term at the nodes will

* Not only is the elasticity matrix [D] a symmetric matrix but any triple matrix product of the form $[B]^T[D][B]$ gives a symmetric matrix.

have to be added. This will be equivalent to,

see sub-section 14.6.1,

$$\{F_b\} = - [N]^T \{g\}\, dA \qquad 14.18$$

In order to derive the element stiffness matrix and the element nodal force vectors, various integrals have to be evaluated. For elements having only linear displacement models in Cartesian coordinates, i.e. a constant strain element, the [B] matrix consists only of constant terms and hence the integrations are simple and straightforward. However, for elements having higher order displacement models, the integrations necessary to evaluate the stiffness and load matrices involve polynomials. Although it is not an easy matter to integrate these terms explicitly, it is a relatively straightforward process to carry out the integrations numerically. Indeed, the local element coordinate system adopted for the interpolation displacement model approach lends itself readily to this method of solution.

14.5 FORMATION OF THE STRUCTURAL EQUILIBRIUM EQUATIONS

The equilibrium equations for any statically loaded elastic structure take the form

$$\{R\} = [K]\{d\} \qquad 14.19$$

where

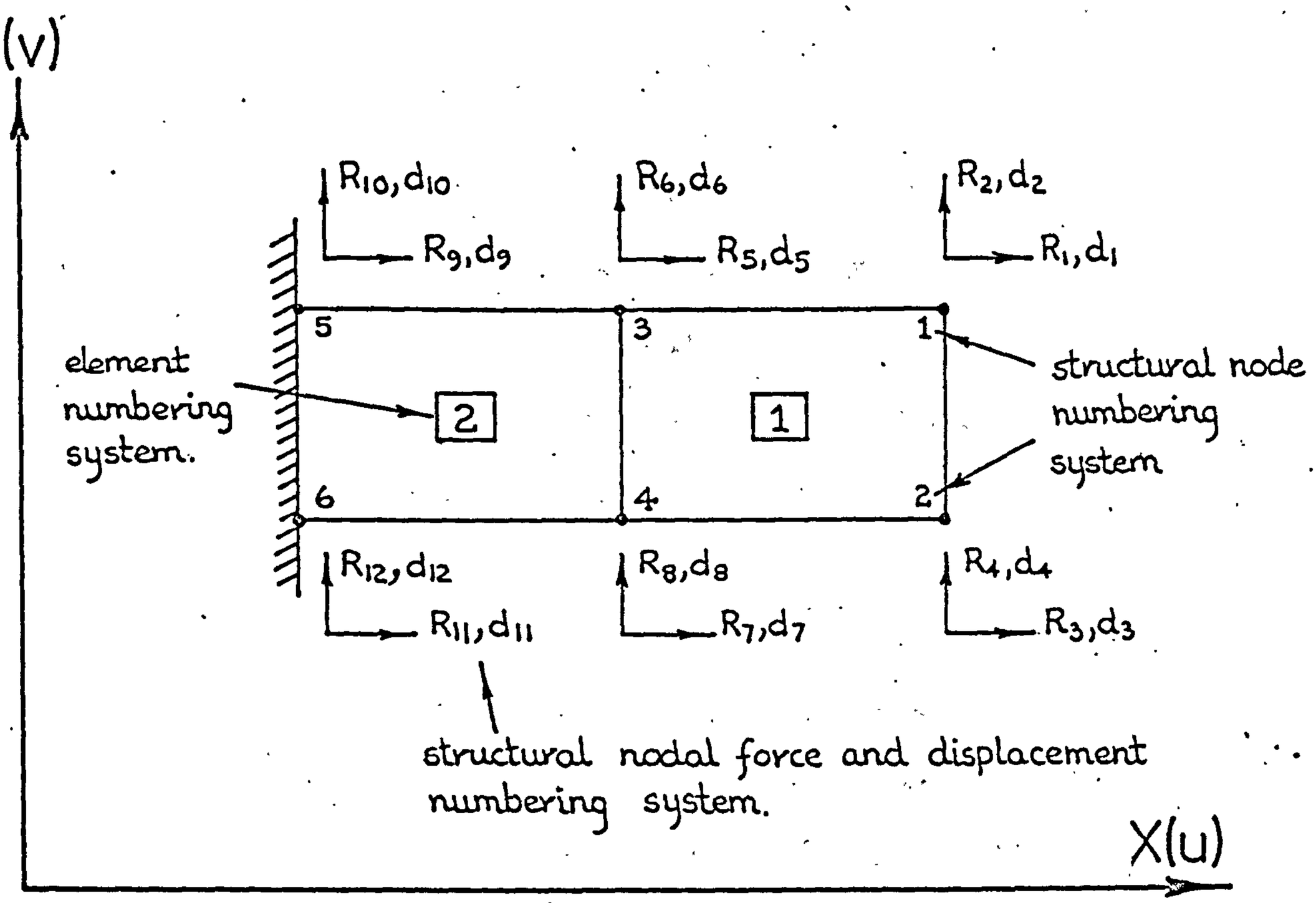
$\{R\}$ is the vector of the structural nodal forces

$[K]$ is the structural stiffness matrix or array of structural influencecoefficients

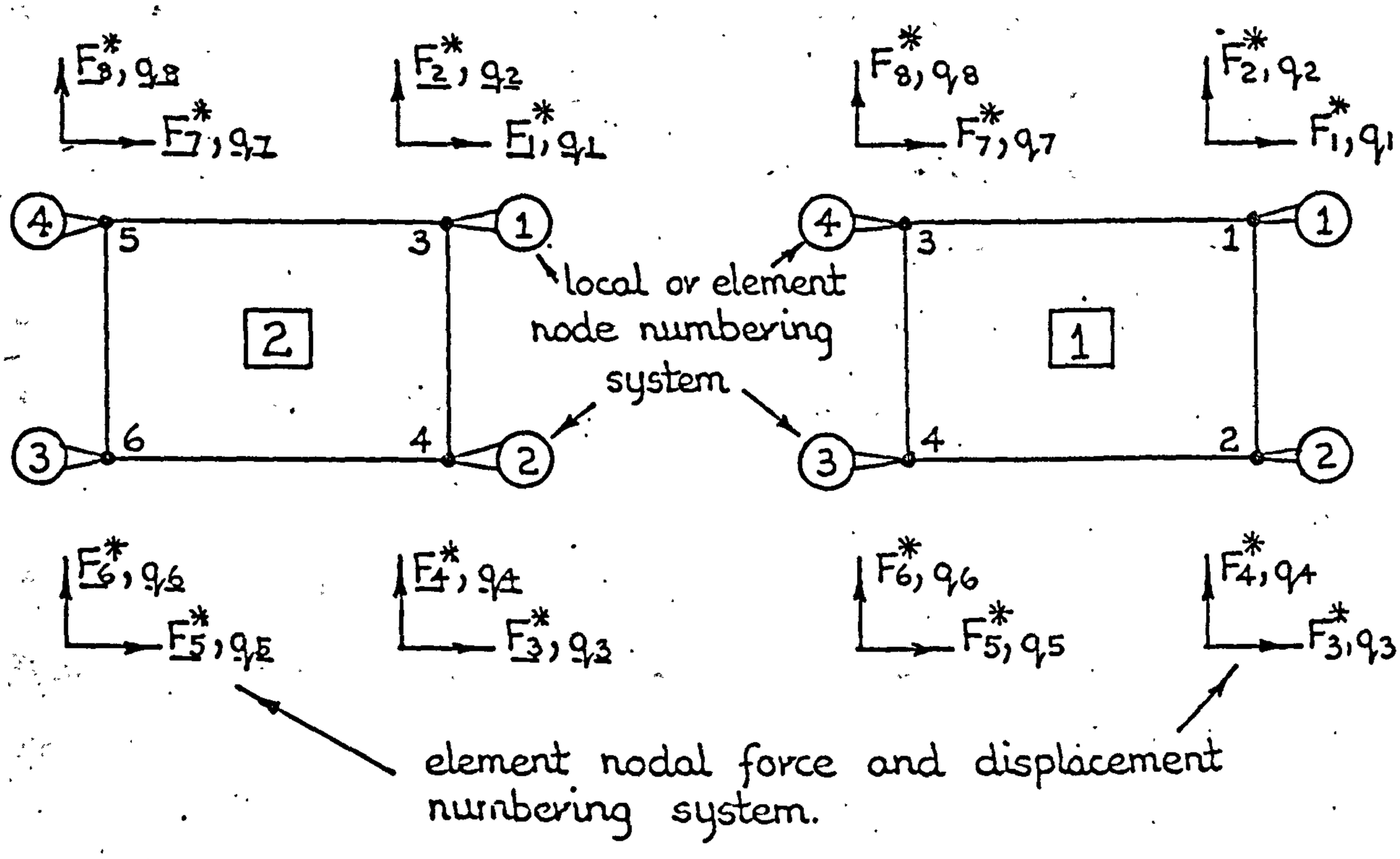
and $\{d\}$ is the vector of the structural nodal displacements.

The object of this step is to derive the $\{R\}$ and $[K]$ matrices of the entire structure from the $\{F\}$ and $[k]$ matrices of the individual finite elements from which it is comprised. However, because the structural equilibrium equations can contain an enormous number of coefficients and unknowns, the sequence in which the equations are derived and physically stored in the computer, (that is in core or on backing storage facilities such as magnetic tape and disc files), depends upon the configuration of the computer installation available. Consequently, the type of computer installation also governs the type or method employed to solve the resulting system of simultaneous equations, see section 14.7.

As an example of how the structural equilibrium equations are formed, consider the simple two-dimensional cantilever structure depicted in FIG. 14.6a. The structure consists of two, 4-noded quadrilateral plane stress finite elements with each node possessing two degrees of freedom, namely a u displacement in the X coordinate direction and a v displacement in the Y coordinate direction. The nodes of the structure are numbered 1-6 with the two elements being interconnected



a) Two element planestress cantilever model showing structural or global numbering system.



b) Elements showing local or element numbering system.

FIG. 14. 6 Simple two element two-dimensional planestress simulation of cantilever structure.

via nodes 3 and 4. Each element has the characteristic relationship expressed by equation 14.7 namely:

$$\{F\} = [k]\{q\} + \{F_d\} + \{F_t\}$$

Assume now that the $\{F_d\}$ and $\{F_t\}$ force vectors have been determined and have been added into the $\{F\}$ vector so that for each element:

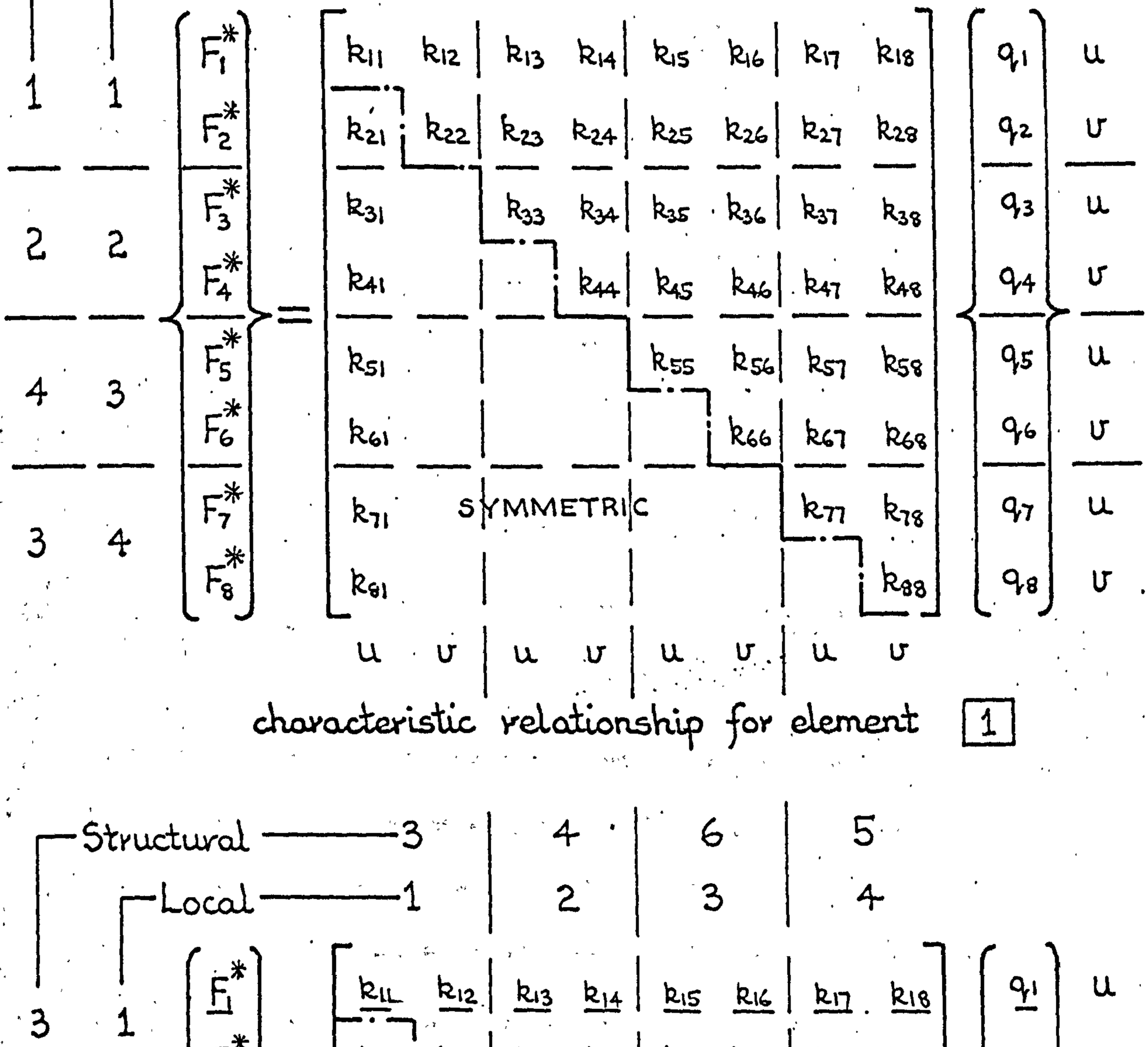
$$\{\overset{*}{F}\} = [k]\{q\} \qquad 14.20$$

FIG. 14.6b shows the two elements of the structure separated and with each element possessing it s own local or element node numbering system. The statically equivalent element nodal forces $\{\overset{*}{F}\}$ and the corresponding nodal displacements are also indicated; those for element number 2 are underlined so that they are easily distinguishable. Using equation 14.20 above, the characteristic relationship for each element can be written down as shown in FIG. 14.6c. Consider now the equilibrium and compatability of structural node number 3 in the Y coordinate direction. Obviously, for equilibrium and compatability to exist, the following conditions must apply, see FIG. 14.6a and b.

$$\overset{*}{F}_8 + \overset{*}{\underline{F}}_2 = R_6$$

and $q_8 = \underline{q_2} = d_6$

Hence, by expanding the appropriate matrix equations from FIG. 14.6c and noting that:-



characteristic relationship for element 1

characteristic relationship for element 2

FIG. 14.6c Characteristic relationships for the two planestress quadrilateral elements of FIG. 14.6a.

$q_1 = d_1$

$q_2 = d_2$

$q_3 = d_3$

$q_4 = d_4$

$q_5 = \underline{q_3} = d_7$

$q_6 = \underline{q_4} = d_8$

$q_7 = \underline{q_1} = d_5$

$q_8 = \underline{q_2} = d_6$

$\underline{q_5} = d_{11}$

$\underline{q_6} = d_{12}$

$\underline{q_7} = d_9$

$\underline{q_8} = d_{10}$

we find that:-

$$R_6 = k_{81}d_1 + k_{82}d_2 + k_{83}d_3 + k_{84}d_4 + (k_{87}+\underline{k_{21}})d_5 +$$
$$(k_{88}+\underline{k_{22}})d_6 + (k_{85}+\underline{k_{23}})d_7 + (k_{86}+\underline{k_{24}})d_8 +$$
$$\underline{k_{27}}d_9 + \underline{k_{28}}d_{10} + \underline{k_{25}}d_{11} + \underline{k_{26}}d_{12}$$

FIG.14.6d presents in matrix form this and the other structural equilibrium equations for the two element cantilever structure shown in FIG. 14.6a.

It can be seen that the formation of the structural equilibrium equations, (equation number 14.19),is a relatively simple operation once the individual element stiffness coefficients and equivalent nodal load vectors have been determined. Each of the element's stiffness coefficients and nodal force components are merely added or 'dumped' into the appropriate position in the structural stiffness matrix or nodal force vector. The appropriate position is easily determined from the global or structural node numbering arrangement and the local or element node numbering system.

Structural			1		2		3		4		5		6			
1	R_1	F_1^*	k_{11}	k_{12}	k_{13}	k_{14}	k_{17}	k_{18}	k_{15}	k_{16}	0	0	0	0	d_1	u
	R_2	F_2^*	k_{21}	k_{22}	k_{23}	k_{24}	k_{27}	k_{28}	k_{25}	k_{26}	0	0	0	0	d_2	v
2	R_3	F_3^*	k_{31}	k_{32}	k_{33}	k_{34}	k_{37}	k_{38}	k_{35}	k_{36}	0	0	0	0	d_3	u
	R_4	F_4^*	k_{41}	k_{42}	k_{43}	k_{44}	k_{47}	k_{48}	k_{45}	k_{46}	0	0	0	0	d_4	v
3	R_5	$F_7^* + F_1^*$	k_{71}	k_{72}	k_{73}	k_{74}	$k_{77}+k_{11}$	$k_{78}+k_{12}$	$k_{75}+k_{13}$	$k_{76}+k_{14}$	k_{17}	k_{18}	k_{15}	k_{16}	d_5	u
	R_6	$F_8^* + F_2^*$	k_{81}	k_{82}	k_{83}	k_{84}	$k_{87}+k_{21}$	$k_{88}+k_{22}$	$k_{85}+k_{23}$	$k_{86}+k_{24}$	k_{27}	k_{28}	k_{25}	k_{26}	d_6	v
4	R_7	$F_5^* + F_3^*$	k_{51}	k_{52}	k_{53}	k_{54}	$k_{51}+k_{31}$	$k_{58}+k_{32}$	$k_{55}+k_{33}$	$k_{56}+k_{34}$	k_{37}	k_{38}	k_{35}	k_{36}	d_7	u
	R_8	$F_6^* + F_4^*$	k_{61}	k_{62}	k_{63}	k_{64}	$k_{67}+k_{41}$	$k_{68}+k_{42}$	$k_{65}+k_{43}$	$k_{66}+k_{44}$	k_{47}	k_{48}	k_{45}	k_{46}	d_8	v
5	R_9	F_7^*	0	0	0	0	k_{71}	k_{72}	k_{73}	k_{74}	k_{77}	k_{78}	k_{75}	k_{76}	d_9	u
	R_{10}	F_8^*	0	0	0	0	k_{81}	k_{82}	k_{83}	k_{84}	k_{87}	k_{88}	k_{85}	k_{86}	d_{10}	v
6	R_{11}	F_5^*	0	0	0	0	k_{51}	k_{52}	k_{53}	k_{54}	k_{57}	k_{58}	k_{55}	k_{56}	d_{11}	u
	R_{12}	F_6^*	0	0	0	0	k_{61}	k_{62}	k_{63}	k_{64}	k_{67}	k_{68}	k_{65}	k_{66}	d_{12}	v
			u	v	u	v	u	v	u	v	u	v	u	v		

FIG. 14.6d Structural equilibrium equations $\{R\}=[K]\{d\}$ for the two element cantilever structure shown in FIG. 14.6a.

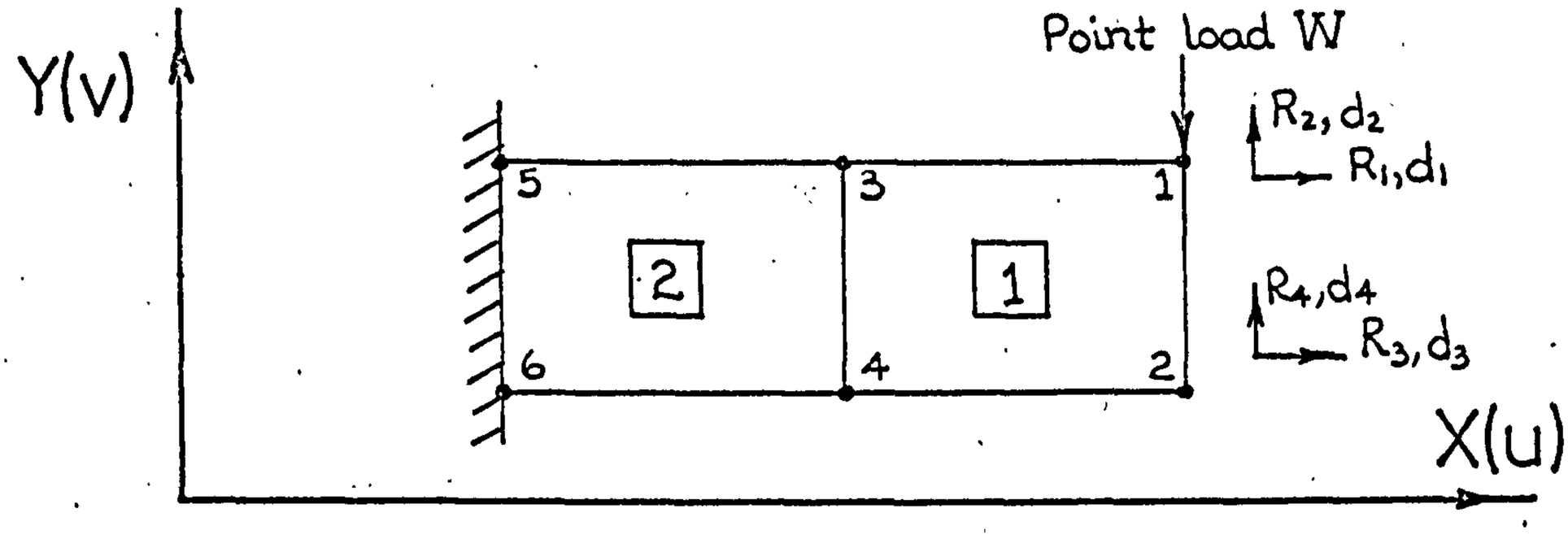
Two features of the structural stiffness matrix, which are shown in FIG. 14.6d, are of vital importance. First, as each element stiffness matrix [k] is symmetrical, see section 14.4 i.e. $k_{21} = k_{12}$ or $\underline{k_{81}} = \underline{k_{18}}$, the structural stiffness matrix [K] is also symmetric, i.e. $K_{65} = k_{87} + \underline{k_{21}} = K_{56} = k_{78} + \underline{k_{12}}$. Secondly, although it is not so apparent from the small cantilever example illustrated, the array of the structural stiffness coefficients are confined within a band of[K] and thereby constitute a banded stiffness matrix. The top right hand corner of [K] in FIG. 14.6d can be seen to contain all zero terms. The width of the band of non-zero terms in [K] can be determined ab initio from the structural node numbering system. Therefore, as it is important from the equation solution time aspect to keep the width of the band of coefficients to a minimum, see sub-section 14.7.1, careful and judicious numbering of the structural nodes can have enormous economic consequences.

14.6 APPLICATION OF THE STRUCTURAL BOUNDARY CONDITIONS

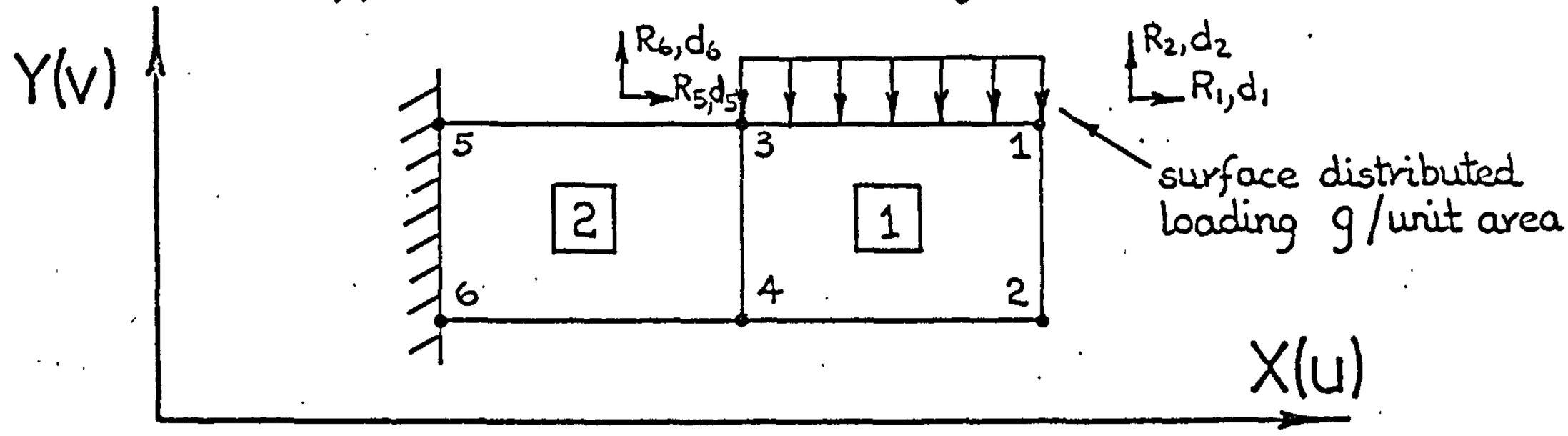
In this step of the finite element analysis procedure, the boundary conditions of the structure are applied to the finite element model. These consist basically of two types, namely, boundary loading and geometric or kinematic boundary conditions.

14.6.1 Boundary Loading Conditions

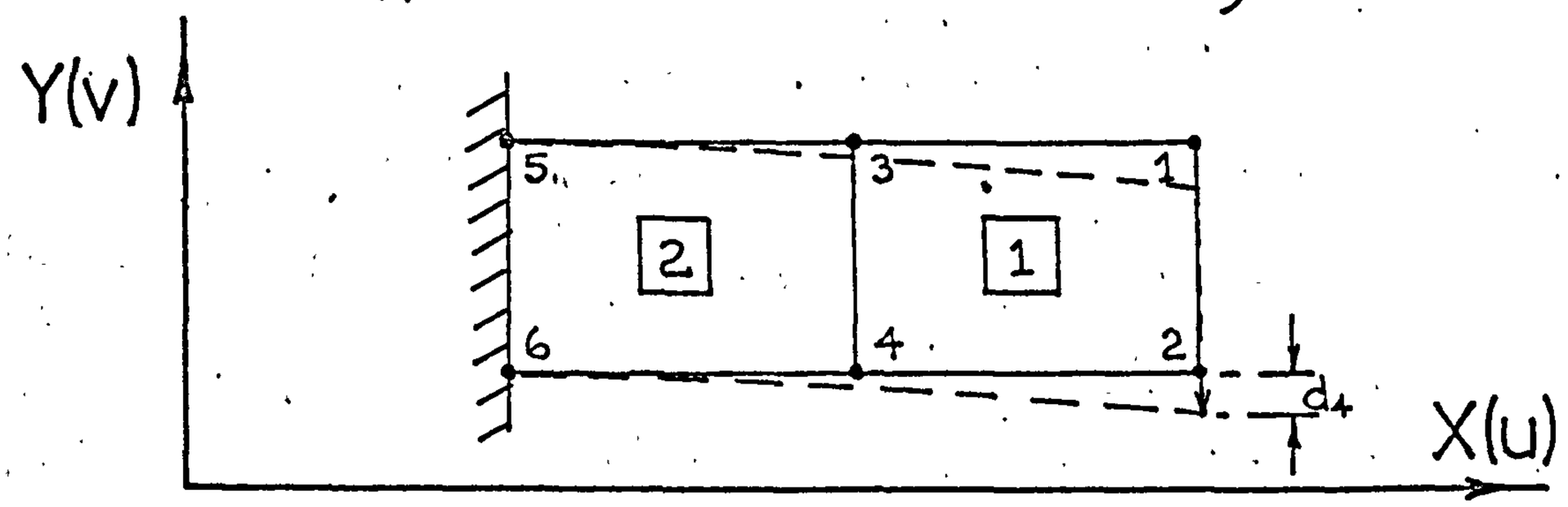
Boundary loading type boundary conditions can be sub-divided into two groups. These are 1) point loads and 2) distributed loads. Point load boundary conditions are the easiest of the boundary conditions to apply. As an example, consider again the 2 element cantilever structure discussed in the previous section. Suppose now that a known point load W is applied to the structure as shown in FIG. 14.7a. In the finite element analysis, this load is applied to the model simply by adding the value of W, (taking due account of the sign convention of the structural coordinate axis system), into the appropriate location of the structural nodal force vector $\{R\}$. Thus, for the case at hand R_2 will become equal to ${}^{*}F_2 + (-W)$. However, if the boundary loading is distributed, such as for example due to fluid pressure or floor loading, the total load has to be apportioned or 'lumped' between the adjacent nodes. For the case illustrated in FIG. 14.7b, where the distributed load is purely vertical, the total load can simply be divided between nodes 1 and 3 in the Y coordinate direction. Hence, the two resulting point loads, 'equivalent' to the total distributed loading would be applied to the finite element model structure as before, i.e. to R_2 and R_6 in $\{R\}$.



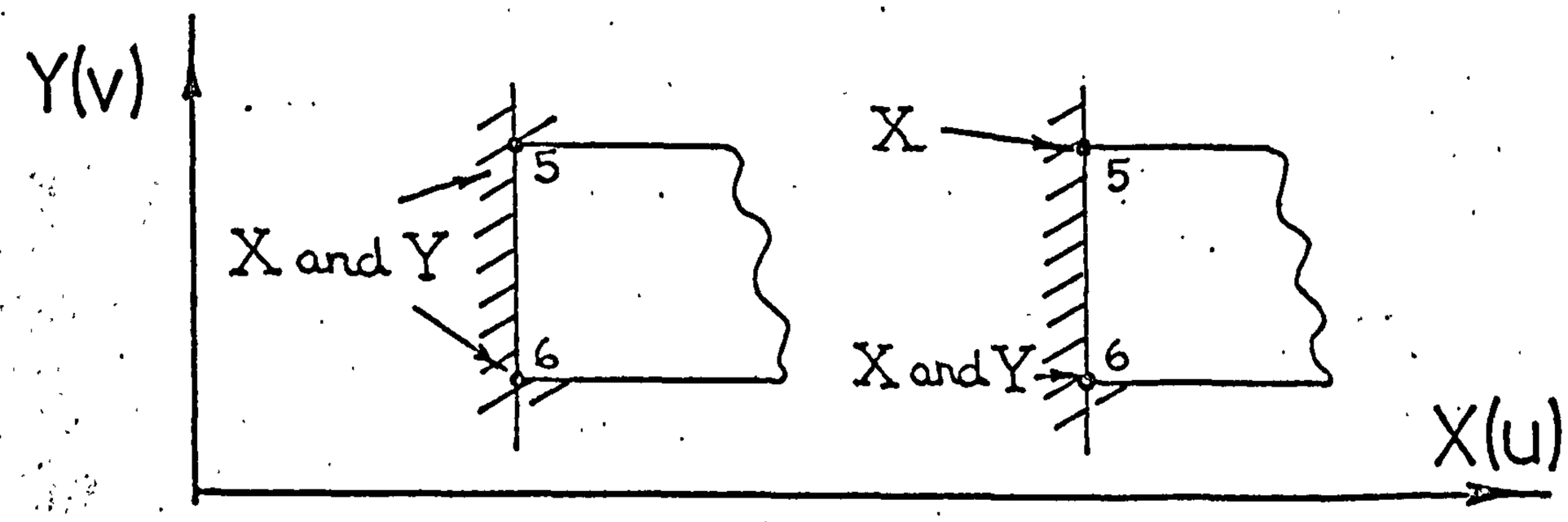
a) Applied point load boundary condition.



b) Applied distributed load boundary condition.



c) Applied nodal displacement boundary condition.



d) Applied kinematic constraint boundary conditions.

FIG. 14.7 Four basic types of structural boundary conditions.

If however, the more refined element displacement models are employed, the distributed loading cannot be evenly distributed between the adjacent nodes. In these cases, it is necessary to employ equation 14.18 and to carry out the appropriate integration,

i.e. $\{F_b\} = -\int [N]^T \{g\}\, dA$

(Note that if the above integration is carried out for the linearly varying displacement element and the structure is loaded as shown in FIG. 14.7b, the two equivalent nodal forces will, in fact, be equal to one half of the total distributed loading.)

14.6.2 Geometric Boundary Conditions

As with the boundary loading boundary conditions, the geometric boundary conditions can be one of two types. They are known as boundary or 'known' applied displacements and kinematic constraints respectively. Suppose that instead of applying a point or distributed load to the cantilever structure, a 'known' displacement is applied to node number 2 in the Y coordinate direction as illustrated in FIG. 14.7c. Consequently, d_4 is now a known displacement and so the number of unknown displacements is reduced by one. Clearly, it can be seen by expanding the first equation of the matrix shown in FIG. 14.6d and rearranging that

$$R_1 - k_{14}d_4 = k_{11}d_1 + k_{12}d_2 + k_{13}d_3 + k_{17}d_5 + k_{18}d_6 + k_{15}d_7 + k_{16}d_8$$

Hence, the structural nodal forces $\{R\}$, can be modified by multiplying the known applied displacement component with the appropriate structural stiffness terms as shown above. This obviously reduces the set of equilibrium equations down to eleven equations with eleven unknown nodal displacement components.

Once the remaining eleven unknown displacements have been determined, it is a simple matter to calculate the reaction or nodal force R_4 which was responsible for the applied known displacement d_4. This is obtained by substituting all the now known nodal displacements into equation four of the matrix arrangement of FIG. 14.6d, and multiplying out.

Although theoretically, a system of twelve equations containing twelve unknowns can be solved, unless a minimum number of prescribed displacements or kinematic constraints are applied to the cantilever structure, the values of the nodal displacements cannot be uniquely determined. Physically, this is because the structure can move freely as a rigid body and therefore it possesses an infinite number of displaced configurations which will satisfy the equilibrium equations. Mathematically, this will be manifest by the fact that the structural stiffness matrix [K], shown in FIG. 14.6d, will be a singular matrix, i.e. it possesses no inverse. Consequently, it is necessary to prescribe a minimum number of kinematic constraints which will prevent all possible rigid body movements of the structure.

For the simple cantilever problem there are three possible rigid body movements. Horizontal and vertical movements and also a rigid body rotation in the XY plane. FIG. 14.7d shows two alternative ways of eliminating these three rigid body movements, namely constraining nodes 5 and 6 in both the X and Y coordinate directions or by constraining node 6 in both the X and Y directions and node 5 in the X coordinate direction only. For the former case, d_9, d_{10}, d_{11} and d_{12} will all be ascribed zero values for the displacements, whereas for the latter case, zero values will only be ascribed to d_9, d_{11} and d_{12}. Either of these two alternatives will remove the rigid body displacement modes and will render the matrix equation of FIG. 14.6d amenable to solution by making [K] non-singular. As before, once the remaining unknown nodal displacements have been determined, the vertical and horizontal 'support' reactions can be calculated at the constrained nodes by back substitution into the equilibrium equations and multiplying out.

It is worthwhile to point out before leaving this sub-section, that the methodology for applying the kinematic boundary conditions computationally, depends upon the type of equation solution routine employed, see following section.

14.7 SOLUTION OF THE STRUCTURAL EQUILIBRIUM EQUATIONS

The solution of the structural equilibrium equations is a very important step in the finite element analysis procedure. For large problems, the total number of equations to be solved can be several hundred. Also, and especially for three-dimensional structures, the number of coefficients per equation can approach two to three hundred. Consequently, enormous computer storage capacity is generally required to store all the coefficients of the structural stiffness matrix [K] and considerable amounts of computational time involved for the subsequent solution of the equilibrium equations.

As an example of the storage capacity required for a simple two-dimensional problem using the 4-noded quadrilateral plane stress element, consider again the cantilever structure shown in FIG. 14.6a. Suppose that instead of the two element mesh shown in the figure, the structure was subdivided into a 10x5 element mesh. Hence, the finite element model would therefore consist of fifty elements, would have sixty six structural nodes and would possess one hundred and thirty two degrees of freedom. Thus, the overall size of the structural stiffness matrix [K], including all the zero terms, would be one hundred and thirty two terms square giving a total number of 17424 coefficients. However, because these coefficients are real numbers, each coefficient requires

two words of computer storage. Consequently, even for this very small problem, the computer storage required for [K] is approximately 35000 words.

For all but the smallest engineering problems the capacity required to store the structural stiffness coefficients exceeds the 'core' storage capacity of even today's largest machines. Consequently, external or peripheral devices have to be employed to store the coefficients, e.g. magnetic tape or disc files. This in itself again increases the solution time because of the extra and necessary transfer and search times involved. Because of the economic and storage capacity problems associated with the finite element method, much effort has been expended in trying to improve the efficiency of the solution of the equilibrium equations. Indeed, it has been shown that the type of solution routine adopted should not only be governed by the size and type of the problem being analysed, but also by the particular computer installation employed, Brooks and Brotton (131).

There are two basic approaches to the solution of large systems of linear simultaneous equations. These are commonly known as direct and indirect methods respectively. The direct approach attempts to obtain an 'exact' solution of the equations, (within the limits of round-off accuracy), and includes the elimination methods of Gauss and Cholesky. The indirect approach on the other hand, aims only at an

approximate solution and relies upon successive corrections being made to an original estimate of the actual solution. However, because generally the true solution is approached asymptotically, a point has to be selected at which the correcting or iterative process is terminated.

14.7.1 Direct Approach - Gaussian Elimination

The Gauss elimination method consists of two distinct phases. In the first phase, the structural stiffness matrix [K] is reduced to an upper triangular matrix by eliminating all the coefficients which fall below the leading diagonal line. The second phase then consists of a back-substitution process in which all the unknown quantities are determined. As an example, consider the trivial non-singular set of structural equilibrium equations:-

$$\begin{Bmatrix} R_1 \\ R_2 \end{Bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} \cdots$$

The general procedure is to make the coefficients of one of the unknowns equal, such that by subtracting the two modified equations, a third equation is obtained which only contains one unknown. Hence, by dividing the first equation in the set above by K_{11} and then subsequently multiplying this equation by K_{21} and subtracting the result from the second equation we have:-

$$\begin{Bmatrix} \frac{R_1}{K_{11}} \\ R_2 - K_{21}\frac{R_1}{K_{11}} \end{Bmatrix} - \begin{bmatrix} 1 & \frac{K_{12}}{K_{11}} \\ 0 & K_{22} - K_{21}\frac{K_{12}}{K_{11}} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix}$$

The second equation in the above, now only contains one unknown quantity d_2, therefore, d_2 can be determined. Consequently, by back-substituting d_2 in the first of the equations, d_1 can subsequently be evaluated.

The procedure outlined above is not restricted to single term processing. Indeed, K_{11}, K_{12} .....K_{22} can be submatrices. There are two main advantages with the direct elimination methods. These are:-

1) The solution time required can be determined at the outset and provided that the equations are well conditioned, an accurate solution is always obtained.* Conditioning is not generally a problem with elastic analyses as with this class of structure, the main diagonal terms of [K] are always positive and are usually much larger than the off-diagonal coefficients.**

2) The solution to several different load cases,

* Solution time is directly proportional to the number of equations and directly proportional to the square of the bandwidth.

** If the main diagonal terms are not predominant, pivoting can be carried out which rearranges the equations such that the larger coefficients then fall on the main diagonal. However, this procedure is undesirable as it obviously destroys the banded nature of the [K] matrix.

i.e. $\{R\}$ vectors, can be obtained simultaneously with only a very modest increase in the computational time.

The main disadvantage with this method is that a vast amount of storage capacity is required to store all the coefficients of the structural stiffness matrix [K]. However, various techniques can be employed to reduce the number of coefficients required. Because the [K] matrix is always symmetrical, only those terms which fall on or above the main diagonal line need be stored. Also, because of the banded nature of [K] only those terms which are contained within this band are required. The terms outside the bandwidth are all zero-valued and consequently play no part in the analysis.

14.7.2 Indirect Approach - Conjugate Gradients

The conjugate gradient method basically minimizes a quadratic error function associated with the structural equilibrium equations, i.e. $[K]\{d\} = \{R\}$, Yettram and Hirst (132). Using an initial estimate $\{d_o\}$, the method looks for the minimum value along the steepest path towards the true solution. From each new position the subsequent direction of search is taken conjugate to its predecessor. Theoretically, the maximum number of corrections or iterations required to obtain convergence to the true solution is equal to the number of equations in the system. However, for well conditioned matrices the convergence should be faster and so far fewer iterations should be required.

The flow diagram of the conjugate gradient method is shown in FIG. 14.8. It can be seen from this figure that exit from the iterative process can be achieved by one of two ways. If convergence is achieved, (i.e. the residual falls below a prescribed acceptable level), before the specified number of iterations has been completed, the iterative process is automatically terminated.

The great advantage of the method is that the structural stiffness matrix [K] is not required in its entirity at any one time. In fact, the product $[K]\{d_o\}$ and subsequently the $[K]\{p_i\}$ products can be evaluated on a piecewise or sequential element basis. Consequently, although each element's stiffness matrix [k] is required during each iteration and therefore should be stored*, the method should not require the same storage capacity as the direct Gauss approach. Iterative methods are attractive also for non-linear problems, where repeated solutions of a similar structural configuration are required. In such cases, a good initial 'guess' to the nodal displacements can be derived from the solution of a previous analysis.

The conjugate gradient method however, has its disadvantages. The principal one being that it is as yet, impossible to predict the number of cycles required for convergence. It is probably due to this uncertainty and

* This would be particularly attractive for regular type structures containing many of the same element.

To Solve $\{R\} = [K]\{d\}$

Take initial estimate of $\{d\}$ to be $\{d_0\}$

BEGIN

Determine initial residual or initial direction of search. $\{p_0\} = \{r_0\} = \{R\} - [K]\{d_0\}$

Calculate the direction of travel $\alpha_i = \dfrac{\{r_i\}^T\{r_i\}}{\{p_i\}^T[K]\{p_i\}}$

Determination of new approximation to the nodal displacements. $\{d_{i+1}\} = \{d_i\} + \alpha_i\{p_i\}$

Determine the associated new residual vector $\{r_{i+1}\} = \{r_i\} - \alpha_i[K]\{p_i\}$

Is residual small enough? Yes

No

Calculate new direction of search $\beta_i = \dfrac{\{r_{i+1}\}^T\{r_{i+1}\}}{\{r_i\}^T\{r_i\}}$

Obtain new direction of search $\{p_{i+1}\} = \{r_{i+1}\} + \beta_i\{p_i\}$

Are number of iterations required completed? No

Yes

END

FIG. 14.8 Flow diagram of the conjugate gradient method.

the obvious inability to be able to specify accurately the computational time required for a specific analysis, that this and the other indirect methods have not become more popular. Another disadvantage with the conjugate gradient method lies in the fact that only one load case can be considered at a time.

14.8 DERIVATION OF THE ELEMENT STRAINS

One important quantity which the engineer sometimes requires to know, is the strain distribution throughout the structure he is investigating. In the finite element method, the state of strain anywhere within an element can be evaluated once the nodal displacements of the element have been determined. Of course, the strain-displacement relationship depends upon the element's particular material behaviour. However, for two-dimensional plane stress elements, and assuming linear elastic material behaviour, the strain-displacement relationships are:-

$$\varepsilon_{xx} = \frac{\partial u}{\partial x}$$

$$\varepsilon_{yy} = \frac{\partial v}{\partial y}$$

and

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

Consequently, irrespective of whether the generalized coordinate or interpolation approach is adopted, the element strains $\{\varepsilon\}$, can be expressed in terms of the element nodal displacements $\{q\}$ by differentiating either equation 14.5a or 14.5b.

Thus, the element strains can be expressed by the matrix equation:-

$$\{\varepsilon\} = [B]\{q\} \qquad 14.21$$

where the [B] matrix is made up from the terms of the [N] matrix which have been differentiated with respect to the X and Y structural coordinate system. Of course, for the case of equation 14.5b, the displacement variation is expressed in terms of the local or element coordinates, δ and η. Hence, it is necessary in this case to transform the local element coordinate system into the global or structural X-Y coordinate system.

14.9 DERIVATION OF THE ELEMENT STRESSES

Once the element strain components have been evaluated, it is a relatively simple step to derive the corresponding element stress components. The only necessary additional requiste being a knowledge of the material's constitutive relationship.

For the two-dimensional plane stress case and assuming linear elastic isotropic material behaviour, the strain components are related to the stress components by:-

$$\varepsilon_{xx} = \frac{\sigma_{xx}}{E} - \frac{\mu\,\sigma_{yy}}{E}$$

$$\varepsilon_{yy} = \frac{-\mu\,\sigma_{xx}}{E} + \frac{\sigma_{yy}}{E}$$

$$\gamma_{xy} = \frac{\tau_{xy}}{G}$$

(Note that for this particular material behaviour only, two material properties are independent, i.e. $G = \frac{E}{2(1+\mu)}$)

Thus in matrix form:-

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E} & -\frac{\mu}{E} & 0 \\ -\frac{\mu}{E} & \frac{1}{E} & 0 \\ 0 & 0 & \frac{1}{G} \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix}$$

or

$$\{\varepsilon\} = [M]\{\sigma\} \qquad 14.22$$

Therefore, by inverting [M] we can express the element's stress components in terms of the corresponding strain components. Hence,

$$\{\sigma\} = [M]^{-1}\{\varepsilon\}$$

or

$$\{\sigma\} = [D]\{\varepsilon\} \qquad 14.23$$

The matrix [D], which is obviously equal to $[M]^{-1}$, is the elasticity matrix of the material. For the two-dimensional plane stress isotropic case considered above

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1-\mu^2)} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{(1-\mu)}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix}$$

For anisotropic material behaviour, the properties making up the elasticity matrix must refer to the same relative coordinate axis system in which the stress and strain components are considered to act.

If an element is subjected to initial strains $\{\varepsilon_o\}$, due to, say a temperature change, then the stresses within the element will be caused by the difference between the actual total strains and these initial strains. Thus equation 14.23 becomes:-

$$\{\sigma\} = [D](\{\varepsilon\} - \{\varepsilon_o\}) \qquad 14.24$$

14.10 DERIVATION OF THE ELEMENT NODAL FORCES

If the force distribution is required on any particular boundary or surface of the structure being analysed, it is a relatively straightforward process to determine this by simply calculating and summing the nodal forces of the elements adjacent to that boundary or surface. Therefore, by using equation 14.7

$$\{F\} = [k]\{q\} + \{F_d\} + \{F_t\}.$$

Of course, $[k]$, $\{F_d\}$ and $\{F_t\}$ will have to be re-calculated for the appropriate elements, (if they have not been kept in store), and the appropriate nodal displacements 'picked' out of the now known structural nodal displacement vector $\{d\}$.

If this process were to be carried out for all the elements in the structure, it would in fact provide a very useful check on the accuracy of the solution.

Consider FIG. 14.9 where two elements, i and j meet at node number n. Obviously, for equilibrium at this node

$$\{R\}_n - \{F\}_n = \{0\} \qquad 14.25$$

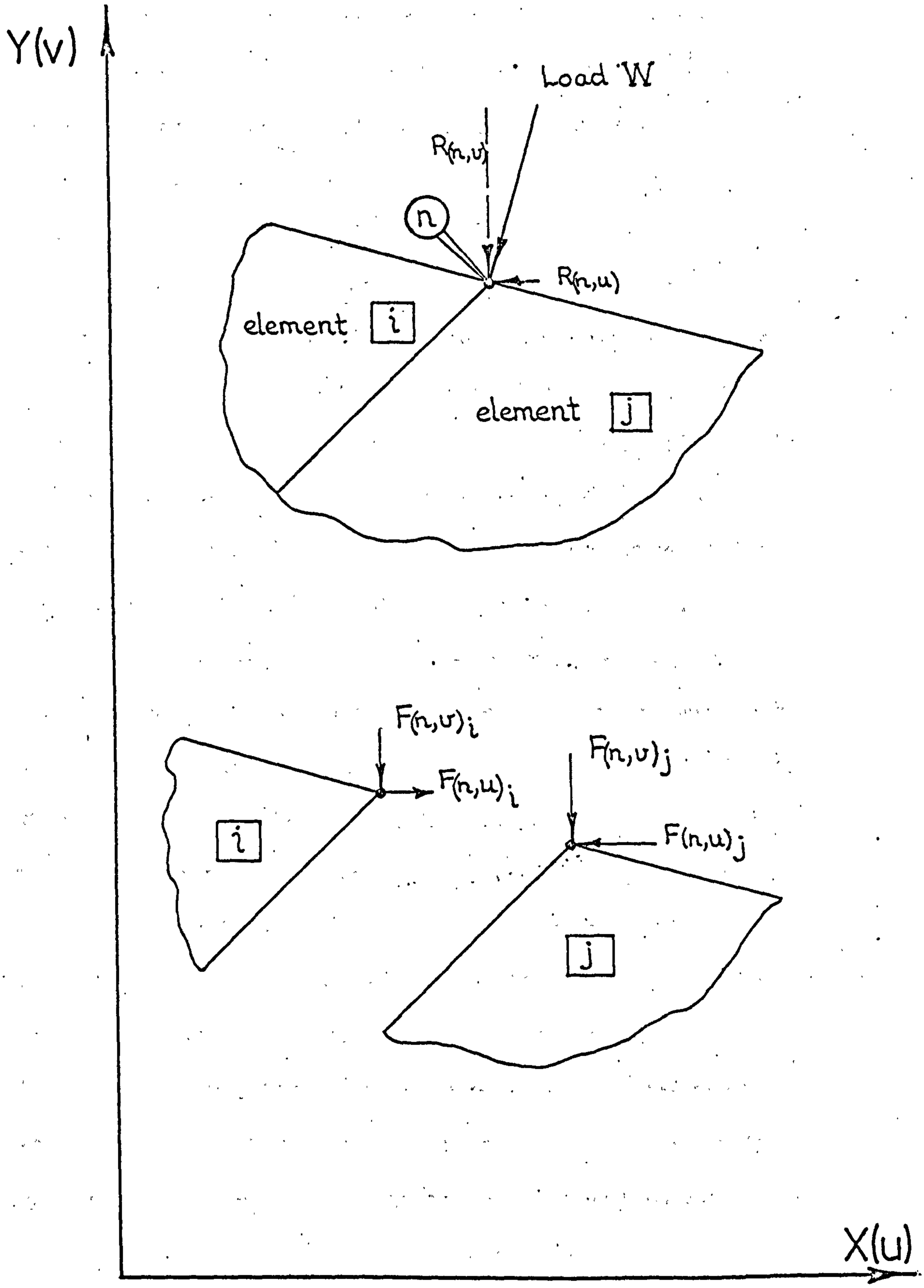


FIG. 14.9 Check on the accuracy or 'out of equilibrium' of finite element results using the element nodal forces.

The size of the residual of this equation, at this and all the other nodes, gives an indication of the accuracy or 'out of equilibrium' of the solution obtained from the analysis.

14.11 PROBLEM ORIENTATED DATA CHECK

It will have become apparent that it is necessary to feed into the finite element analysis program all the separate finite elements' nodal coordinate and connectivity data. Although other items such as the elements' material property data and the various boundary conditions have also to be input, the nodal coordinates and connectivity make up at least 95 % of the input data required. Of course, the chances of wrongly determining a coordinate value or making data preparation punching errors are very great. Consequently, it is necessary to check the data which has been punched for each problem as obviously an incorrect dimension or nodal connection will yield an incorrect solution. After all, it is possible for a computer to continue an analysis quite oblivious to the fact that a nodal dimension may be wildly out.

For all the work reported herein, simple programs were written to check the node coordinate data and node numbering systems. This was achieved by employing the I. C. L's computer library plot procedures, I.C.L. Manual (133) and the computer installation's graph plotter facility. The data check process consisted of feeding into the computer the complete pack of data cards produced for the

particular problem together with the appropriate plot program. Hence, after suitably scaling the coordinate data,the computer and plotter produced a replica of the finite element model. The plotter could also be programmed to number the nodal points. Therefore, by examining the computer plot and the original meshed structural configuration, data errors could easily be detected. FIG. 14.10 shows part of a computer plot of the bridge construction analysed in Chapter Ten. The figure, which is drawn to a scale six times full size, clearly shows up three coordinate errors, namely at node numbers 237, 271 and 451. The plot also shows the directions assumed for the grain in the cortical bone elements. However, the node numbering system has been omitted from the plot for clarity. Even so, the numbering and writing capabilities of the plotter are demonstrated by the clearly labelled and dimensioned Cartesian coordinate axis system employed.

14.12 FINITE ELEMENT ANALYSIS PROGRAM CHECK

Another aspect which must be considered is how to check the actual finite element analysis programs themselves. Clearly, the most logical way is to compare the finite element results with those obtained using some other proven method. Here, all the finite element programs developed are checked by solving problems which have 'known' analytical solutions. However, certain

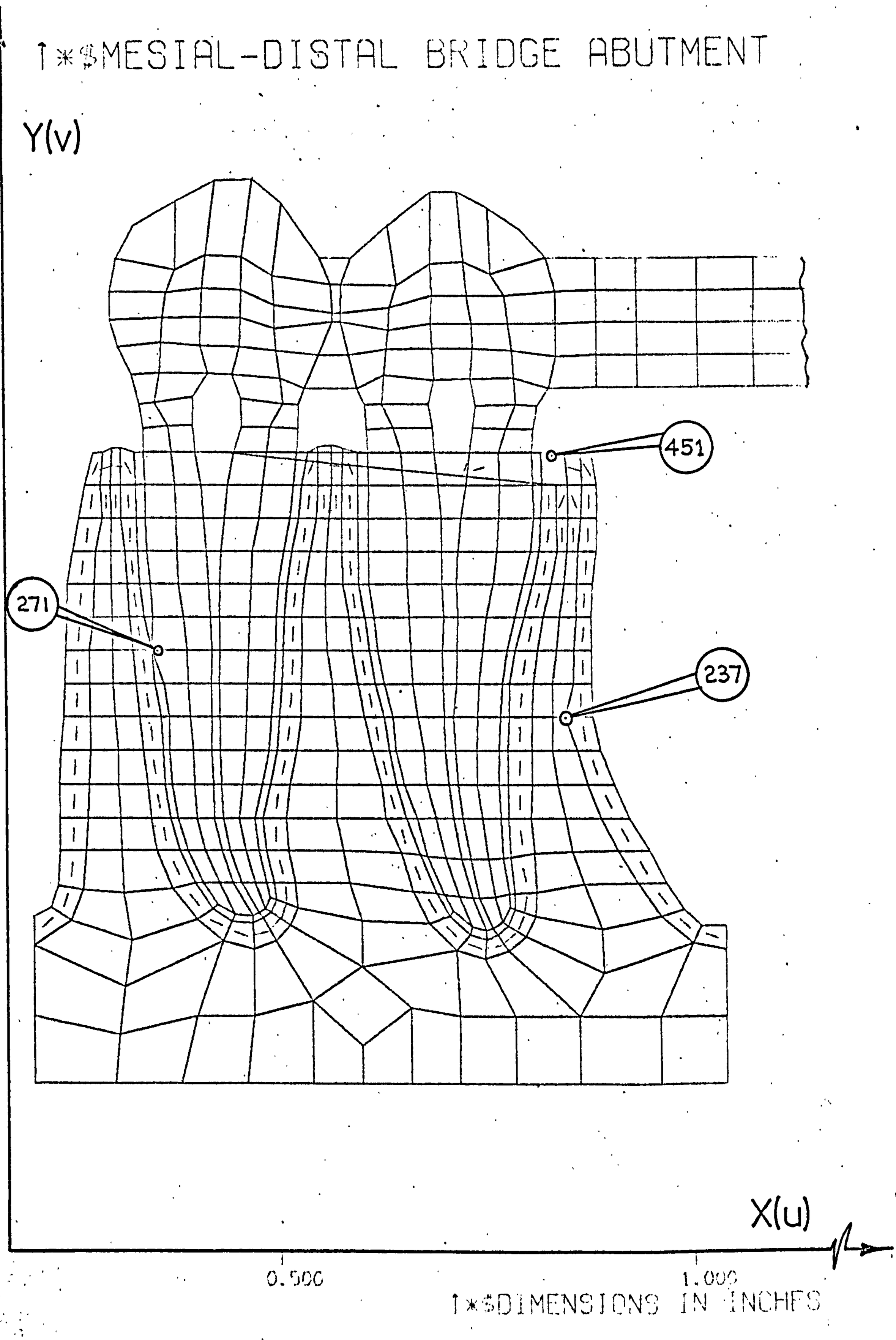


FIG. 14.10 Computer plot of 2-D repvesentation of the 1st and 2nd mandibular premolars with bridge type restoration. Plot shows clearly the thvee data coordinate errors.

facets of the programs cannot be satisfactorily tested because analytical solutions do not yet exist for problems involving orthotropic materials.

CHAPTER FIFTEEN

AXISYMMETRIC FINITE ELEMENT ANALYSIS PROGRAM

15 AXISYMMETRIC FINITE ELEMENT ANALYSIS PROGRAM

Structures which possess rotational symmetry about an axis and are loaded and constrained axisymmetrically about this axis can be simulated using axisymmetric finite elements. For the dental analysis problem investigated in Chapter Seven section 7.4, the second mandibular premolar with its class 1 amalgam restoration was assumed to possess this rotational symmetry, see FIG. 7.12. The flow diagram of the axisymmetric finite element computer program is shown in FIG. 15.1; the notation in the boxes indicates the equivalent Algol variable declarations given in the program listing in Appendix Three.

15.1 STRUCTURE DISCRETIZATION.

To facilitate the grading of the finite element mesh, i.e. to have a fine element subdivision in areas of special interest and a coarse meshwork in more remote areas, the triangular type of finite element is most suitable, see FIG. 15.10. Therefore, for the axisymmetric program, the 3-noded triangular finite element was selected. A typical element is shown in FIG. 15.2 together with the local node numbering system ijm and the corresponding global Cartesian nodal coordinates. Note that the nodes in this case are actually rings and that the element is in the form of an annulus. Hence, the volume of material associated with the element is

FIG. 15.1 Flow diagram of the axisymmetric finite element analysis program.

that of a body of revolution. Consequently, all the necessary integrations have to be carried out on this basis.

15.2 DISPLACEMENT MODELS.

Due to the assumed rotational symmetry of the body, there is no displacement generated in the circumferential (theta) direction, see FIG. 15.2. Consequently, each node has only two degrees of freedom, namely the u and v displacements associated with the RZ plane. Therefore, equation 14.1 for this element will be :-

$$\begin{aligned} u(r,z) &= \alpha_1 + \alpha_2 r + \alpha_3 z \\ \text{and } v(r,z) &= \alpha_4 + \alpha_5 r + \alpha_6 z \end{aligned} \qquad 15.1a$$

This linear displacement field ensures continuity of displacement between the elements since 'lines' that are initially straight before deformation remain straight after deformation. Also, the displacement model incorporates both the necessary rigid body and constant strain states.

Employing the generalized coordinate approach, equation 14.2a becomes :-

$$\begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_m \\ v_m \end{Bmatrix} = \begin{bmatrix} 1 & r_i & z_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & r_i & z_i \\ 1 & r_j & z_j & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & r_j & z_j \\ 1 & r_m & z_m & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & r_m & z_m \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{Bmatrix} \qquad 15.1b$$

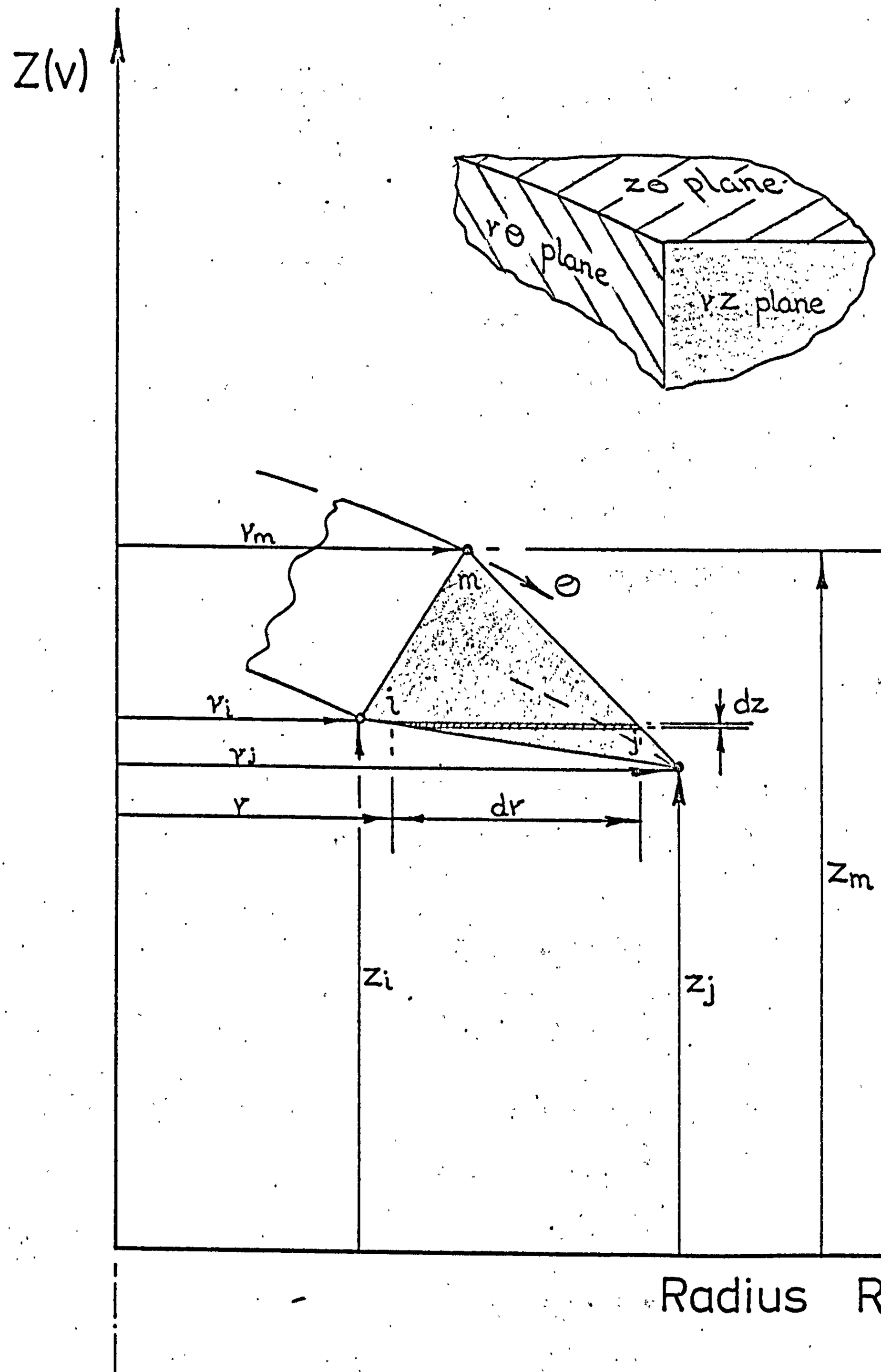


FIG. 15.2 Typical triangular shaped axisymmetric finite element i.j.m.

Consequently, on solving the above equation (by inverting the [A] matrix), we obtain equation 14.3. Hence, by substituting this result into equation 14.1b ,the displacements anywhere within the element can be determined, i.e. via equation 14.4. Hence,

$$u(r,z) = \frac{1}{2\Delta}\left\{(a_i+b_ir+c_iz)u_i + (a_j+b_jr+c_jz)u_j + (a_m+b_mr+c_mz)u_m\right\}$$

$$\text{and } v(r,z) = \frac{1}{2\Delta}\left\{(a_i+b_ir+c_iz)v_i + (a_j+b_jr+c_jz)v_j + (a_m+b_mr+c_mz)v_m\right\} \qquad 15.2a$$

$$\text{or } u(r,z) = N_iu_i + N_ju_j + N_mu_m$$

$$\text{and } v(r,z) = N_iv_i + N_jv_j + N_mv_m \qquad 15.2b$$

(Note that here the N functions are expressed in terms of the Cartesian coordinate directions.)

In the above

$$2\Delta = \text{determinant} \begin{vmatrix} 1 & r_i & z_i \\ 1 & r_j & z_j \\ 1 & r_m & z_m \end{vmatrix} \quad \ldots\ldots$$

and is equal to twice the cross-sectional area of the triangular element ijm in theRZ plane.

Also, $a_i = r_jz_m - r_mz_j$ $\qquad$ $c_j = r_i - r_m$

$b_i = z_j - z_m$ $\qquad$ $a_m = r_iz_j - r_jz_i$

$c_i = r_m - r_j$ $\qquad$ $b_m = z_i - z_j$

$a_j = r_mz_i - r_iz_m$ $\qquad$ $c_m = r_j - r_i$

$b_j = z_m - z_i$

15.3 DERIVATION OF THE FINITE ELEMENT STIFFNESS MATRIX AND NODAL THERMAL LOAD VECTOR.

For the dental structural analysis problems, the effect of the body weight force component on the stress distribution is insignificant in comparison with all the other force components, i.e. $\{F_d\} = \{0\}$. (Generally, the body weight or body force components are only significant in structures such as massive concrete gravity dams.) Consequently, equation 14.7 reduces to :-

$$\{F\} = [k]\{q\} + \{F_t\} \qquad 15.3$$

There are four strain components associated with the axisymmetric finite element, see Timoshenko and Goodier (134) page 343. These are :-

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_{zz} \\ \varepsilon_{rr} \\ \varepsilon_{\theta\theta} \\ \gamma_{rz} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial v}{\partial r} \\ \frac{\partial u}{\partial r} \\ \frac{u}{r} \\ \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} \end{Bmatrix}$$

vertical strain

horizontal or radial strain

hoop or circumferential strain

shear strain in the rz plane

Hence the [B] matrix in equation 14.21 becomes, using equation 15.2b

$$\begin{Bmatrix} \varepsilon_{zz} \\ \varepsilon_{rr} \\ \varepsilon_{\theta\theta} \\ \gamma_{rz} \end{Bmatrix} = \begin{bmatrix} 0 & \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_j}{\partial z} & 0 & \frac{\partial N_m}{\partial z} \\ \frac{\partial N_i}{\partial r} & 0 & \frac{\partial N_j}{\partial r} & 0 & \frac{\partial N_m}{\partial r} & 0 \\ \frac{N_i}{r} & 0 & \frac{N_j}{r} & 0 & \frac{N_m}{r} & 0 \\ \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial r} & \frac{\partial N_j}{\partial z} & \frac{\partial N_j}{\partial r} & \frac{\partial N_m}{\partial z} & \frac{\partial N_m}{\partial r} \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_m \\ v_m \end{Bmatrix} \quad 15.4$$

where $$\frac{\partial N_i}{\partial z} = \frac{1}{2\Delta} c_i$$

$$\frac{\partial N_i}{\partial r} = \frac{1}{2\Delta} b_i$$

and $$\frac{N_i}{r} = \frac{1}{2\Delta r} (a_i + b_i r + c_i z) \text{ etc.}$$

Thus

$$[B] = \frac{1}{2\Delta} \begin{bmatrix} 0 & c_i & 0 & c_j & 0 & c_m \\ b_i & 0 & b_j & 0 & b_m & 0 \\ \frac{N_i}{r} & 0 & \frac{N_j}{r} & 0 & \frac{N_m}{r} & 0 \\ c_i & b_i & c_j & b_j & c_m & b_m \end{bmatrix}$$

It can be seen that the [B] matrix does not consist purely of constant terms. Indeed, the hoop strain component is a function of the position r.

The corresponding stress components are related to the strains by the elasticity matrix, equation 14.23.

Hence, for an elastic isotropic material, the [D] matrix for the element of revolution is, Zienkiewicz (129) page 79 :-

$$\begin{Bmatrix} \sigma_{zz} \\ \sigma_{rr} \\ \sigma_{\theta\theta} \\ \tau_{rz} \end{Bmatrix} = \frac{E(1-\mu)}{(1+\mu)(1-2\mu)} \begin{bmatrix} 1 & \frac{\mu}{1-\mu} & \frac{\mu}{1-\mu} & 0 \\ \frac{\mu}{1-\mu} & 1 & \frac{\mu}{1-\mu} & 0 \\ \frac{\mu}{1-\mu} & \frac{\mu}{1-\mu} & 1 & 0 \\ 0 & 0 & 0 & \frac{1-2\mu}{2(1-\mu)} \end{bmatrix} \begin{Bmatrix} \varepsilon_{zz} \\ \varepsilon_{rr} \\ \varepsilon_{\theta\theta} \\ \gamma_{rz} \end{Bmatrix} \qquad 15.5$$

If an axisymmetric finite element of an isotropic material is subjected to an initial temperature increase of T degrees above the datum temperature, the initial thermal strain components due to the associated thermal expansion within the element will be, see Zienkiewicz (129) page 76,

$$\{\varepsilon_o\} = \begin{Bmatrix} \varepsilon_{zz_o} \\ \varepsilon_{rr_o} \\ \varepsilon_{\theta\theta_o} \\ \gamma_{rz_o} \end{Bmatrix} = \begin{Bmatrix} \alpha T \\ \alpha T \\ \alpha T \\ 0 \end{Bmatrix} \qquad 15.6$$

where α is the material's coefficient of linear thermal expansion in units compatible with those of T degrees.

Using the [B], [D] and $\{\varepsilon_o\}$ matrices developed above, the element stiffness matrix [k] and the equivalent nodal thermal force vector $\{F_t\}$ can be determined. In both instances, the integrations are taken over the

volume of the element which for the axisymmetric case is equal to :-

$$\int dV = \int 2\pi\, r\, dr\, dz \qquad 15.7$$

Again, the integration of dV is, like that for the [B] matrix, a function of the variable r. Consequently, the integrations cannot be carried out simply. However, an approximate procedure is to evaluate the integrations and the [B] matrix at the element's centroid where

$$\bar{r} = \frac{1}{3}(r_i + r_j + r_m)$$

$$\text{and} \quad \bar{z} = \frac{1}{3}(z_i + z_j + z_m) \qquad 15.8$$

Hence,

$$[k] \doteq 2\pi \int [B]^T[D][B]\bar{r}\, dr\, dz = 2\pi[B]^T[D][B]\bar{r}\,\Delta$$

and 15.9

$$F_t \doteq -2\pi \int [B]^T[D]\,\{\epsilon_o\}\,\bar{r}\, dr\, dz = -2\pi[B]^T[D]\bar{r}\,\Delta\{\epsilon_o\}$$

The flow diagram for deriving the [k] and $\{F_t\}$ matrices is shown in FIG. 15.3. The notation used in this figure refers to the equivalent Algol variables employed in the computer program, see listing in Appendix Three.

15.4 FORMATION OF THE STRUCTURAL EQUILIBRIUM EQUATIONS.

It was decided to solve the structural equilibrium equations by the direct Gauss elimination method, see 14.7.1. Because of the limitation in the size of the computer installation available, the 'frontal' solution technique was not a viable proposition, Irons (135),

CONTAINS

PULLOUTS

BEGIN

Set up Cartesian nodal coordinates TRI[ ]

Calculate a_i, b_i, c_i ------ $\bar{r}, \bar{z}$ terms; AI, BI, ------ ZBAR

Form $[\bar{B}]$ matrix

Calculate area of element ~ AREA

Calculate elasticity matrix [D] (if necessary)

Form $[\bar{B}]^T[D]$ and save for *

Form $[\bar{B}]^T[D][\bar{B}]$

Calculate $2\pi \bar{r}$ AREA ~ FACT and save for **

Form element stiffness matrix $[k] = \overset{**}{FACT} \times [\bar{B}]^T[D][\bar{B}]$

Dump [k] into Modified Structural Stiffness Matrix [MBK]

Is element temperature above datum?

No

Yes

Form $\overset{**}{FACT} \times \overset{*}{[\bar{B}]}^T[D]$

Form element thermal strain vector {ETS}

Calculate element thermal force vector

$$\{ETF\} = \overset{**}{FACT} \times \overset{*}{[\bar{B}]}^T[D] \times \{ETS\}$$

FIG. 15. 3 Flow diagram for the determination of the [k] and $\{F_t\}$ matrices in the axisymmetric finite element analysis program.

and so the conventional elimination method was employed. Consequently, it was necessary, before commencing the solution process, to derive and store all the non-zero stiffness coefficients of the structural stiffness matrix [K]. However, because only approximately 20000 words of core storage space was available after the analysis program itself had been entered into the computer, the structural equilibrium equations were stored on a magnetic disc backing store facility. Nevertheless, because the structural nodal force and displacement vectors are comparatively small matrices, it was possible to store them internally in the core of the computer.

The Gauss elimination method of equation solution only operates on the stiffness coefficients of a single column of the [K] matrix at each stage of the forward elimination phase. Hence, because the structural stiffness matrix is and always remains symmetrical, the coefficients of the column under the pivotal term (which are to be eliminated in order to produce the lower triangle of zero terms), are the same as the coefficients which appear in the pivotal equation, see FIG. 14.6d. Consequently, there is no need to duplicate these terms and so only the coefficients which fall on or above the main diagonal of the [K] matrix are actually stored. Thus, the total number of storage locations required is almost reduced by half.

Another way of conserving storage space is to condense or modify the arrangement of the structural stiffness matrix.

This can be achieved by sliding all the rows of the upper triangle of the stiffness coefficients until all the main diagonal terms of the [K] matrix are made to coincide with the first column of the modified form of the structural stiffness matrix. As an example, the structural stiffness matrix for the cantilever problem depicted in FIG. 14.6d is shown in this modified form in FIG. 15.4. It can be seen from this figure that the modified arrangement of the [K] matrix now only requires a rectangular matrix of the order of 12 rows by 8 columns.

Because Algol library procedures were available for reading and writing to disc backing files, the rows of coefficients of the [K] matrix in the modified arrangement were stored sequentially in nodal block matrix form. For the axisymmetric element, each node has only two degrees of freedom. Consequently, each node is associated with a 2 x 2 submatrix block of stiffness coefficients. The number of column blocks (NCB) in the modified arrangement of [K], i.e. [MBK], is equal to one plus the maximum node number difference (MANND) which exists in any one element of the structure. Also, the total number of row blocks (NRB) which exist in [MBK] is equal to the total number of nodal points (NONOP). Hence, the space required on the disc to store [MBK] can be determined ab initio and the necessary initialising (erasure of a previous user's data) carried out before each element's stiffness contributions are added into the appropriate location. The total number

Structural

1	R_1	k_{11}	k_{12}	k_{13}	k_{14}	k_{17}	k_{18}	k_{15}	k_{16}	d_1	u
	R_2	k_{22}	k_{23}	k_{24}	k_{27}	k_{28}	k_{25}	k_{26}	0	d_2	v
2	R_3	k_{33}	k_{34}	k_{37}	k_{38}	k_{35}	k_{36}	0	0	d_3	u
	R_4	k_{44}	k_{47}	k_{48}	k_{45}	k_{46}	0	0	0	d_4	v
3	R_5	$k_{77}+k_{11}$	$k_{78}+k_{12}$	$k_{75}+k_{13}$	$k_{76}+k_{14}$	k_{17}	k_{18}	k_{15}	k_{16}	d_5	u
	R_6	$k_{88}+k_{22}$	$k_{85}+k_{23}$	$k_{86}+k_{24}$	k_{27}	k_{28}	k_{25}	k_{26}	0	d_6	v
4	R_7	$k_{55}+k_{33}$	$k_{56}+k_{34}$	k_{37}	k_{38}	k_{35}	k_{36}	0	0	d_7	u
	R_8	$k_{66}+k_{44}$	k_{47}	k_{48}	k_{45}	k_{46}	0	0	0	d_8	v
5	R_9	k_{77}	k_{78}	k_{75}	k_{76}	0	0	0	0	d_9	u
	R_{10}	k_{88}	k_{85}	k_{86}	0	0	0	0	0	d_{10}	v
6	R_{11}	k_{55}	k_{56}	0	0	0	0	0	0	d_{11}	u
	R_{12}	k_{66}	0	0	0	0	0	0	0	d_{12}	v

FIG. 15.4 Condensed or modified arrangement $\{R\} = [MBK]\{d\}$ of the structural equilibrium equations shown in FIG. 14.6d.

of locations required to store [MBK] is equal to

(2 x NRB) x (2 x NCB)

or NRMBK x NCMBK

where the Number of Rows in Modified Big K = 2 x NRB etc.

The required coefficients of [k] are dumped into [MBK] by re-calling, in turn, from the disc backing file, the whole structural row block associated with the three nodes ijm of each element. Each row block is retrieved from the disc file and put temporarily in the array [WMBK]. When all the coefficients have been added into this array, it is simply returned to its original position in the disc file. The program is written such that the first node i must be the lowest node number of the element. However, if node number j is greater than node number m, then the appropriate coefficients of [k] falling below its main diagonal have to be dumped into [MBK]. Note though, that in this case, the submatrix block must be transposed before it is dumped.

Because the nodal force vector is stored 'in core', it is a simple matter to dump any nodal thermal force components associated with any particular element, into the appropriate position in this vector. However, in the axisymmetric program, the element thermal nodal forces $\{ETF\}$ are initially dumped into a structural thermal force vector $\{STF\}$. Subsequently, the resulting $\{STF\}$ vector for the whole structure is added to the total nodal force vector $\{F\}$ at the time when the boundary conditions are being applied.

15.5 APPLICATION OF THE STRUCTURAL BOUNDARY CONDITIONS.

Only two types of boundary conditions were allowed for in the axisymmetric analysis program. These were:-

1) Boundary point loads*

and 2) Kinematic constraints.

(Although provision in part is indicated in the program listing for applied nodal displacements, this feature was not fully implemented.)

For all the degrees of freedom of the nodes which have no applied point loads, the corresponding locations in the structural nodal force vector $\{F\}$ must have zero values. (In the program, the structural thermal force vector $\{STF\}$ is added into $\{F\}$ at a later stage.) Consequently, as the number of nodes where actual point loads are applied is small, the whole of the $\{F\}$ vector is initially zeroed. The applied loads are then subsequently added into the appropriate locations.

For all the degrees of freedom of the nodes which are unconstrained and are 'free' to move, the corresponding locations in the structural nodal displacement vector $\{X\}$

* For an axisymmetric element, a node consists of a complete ring. Hence, the point load must be equal to the equivalent total load divided by the circumference of the ring. This obviously applies to all nodes except those which coincide with the axis of rotational symmetry.

must have zero values. As before, because the number of nodes where constraints are applied is small compared with the total number of degrees of freedom in the structural displacement vector, the whole of the $\{X\}$ vector is initially zeroed. The degrees of freedom which are actually constrained are then subsequently amended. This is achieved by reading into the appropriate location in the $\{X\}$ vector the value of 0.000001. The reason for this is so that the positions in the [MBK] matrix,where modifications have to be carried out in order to render it non-singular and therefore solvable, can be identified. This modifying process is carried out to [MBK] immediately after all the constraints have been read in.

As an example for discussion purposes, consider the complete structural stiffness matrix [K] shown in FIG. 14.6d for the two-dimensional cantilever. Suppose that for this problem node number 6 is to be constrained in both the X and Y coordinate directions and node 5 in the X direction only. Hence, d_9, d_{11}.and d_{12} are therefore all zero values. For the notation used in this program X_9, X_{11} and X_{12} are made equal to zero and hence become known quantities. Consequently,the structural equilibrium equations corresponding to these constrained degrees of freedom can be eliminated.

It is easier, when employing a digital computer, to effectively eliminate the appropriate equations by

simply zeroing all the stiffness coefficients which would otherwise operate on the constrained displacements. The set of equations are then subsequently solved as if nothing had happened. However, in order to prevent the computer from trying to divide by zero pivotal terms when it comes to the constrained equations, a 1 is placed in the appropriate pivotal positions. FIG. 15.5 illustrates the eliminated equations using this technique for the constrained cantilever structure discussed above. Note that the appropriate locations in the nodal force vector have been zeroed and that the [K] matrix remains symmetric. Thus, the modified arrangement of [K], i.e. [MBK] is still amenable to this form of solution. FIG. 15:6: gives the modified form of [K] for exactly the same boundary constraints depicted in FIG. 15.5.

15.6 SOLUTION OF THE STRUCTURAL EQUILIBRIUM EQUATIONS.

The equilibrium equations were solved in the axisymmetric analysis program using the direct Gauss elimination method outlined in 14.7.1. However, because the number of degrees of freedom per node was two, the procedure developed was for block processing. Hence, all operations are performed on 2 x 2 sub-matrix blocks. The flow diagram for the procedure is shown in FIG. 15.7, see also the listing of the procedure DBLOKGAUSS 2 given in Appendix Three.

The procedure developed,looks for zero sub-matrix blocks using the [BUG] matrix and is subsequently able

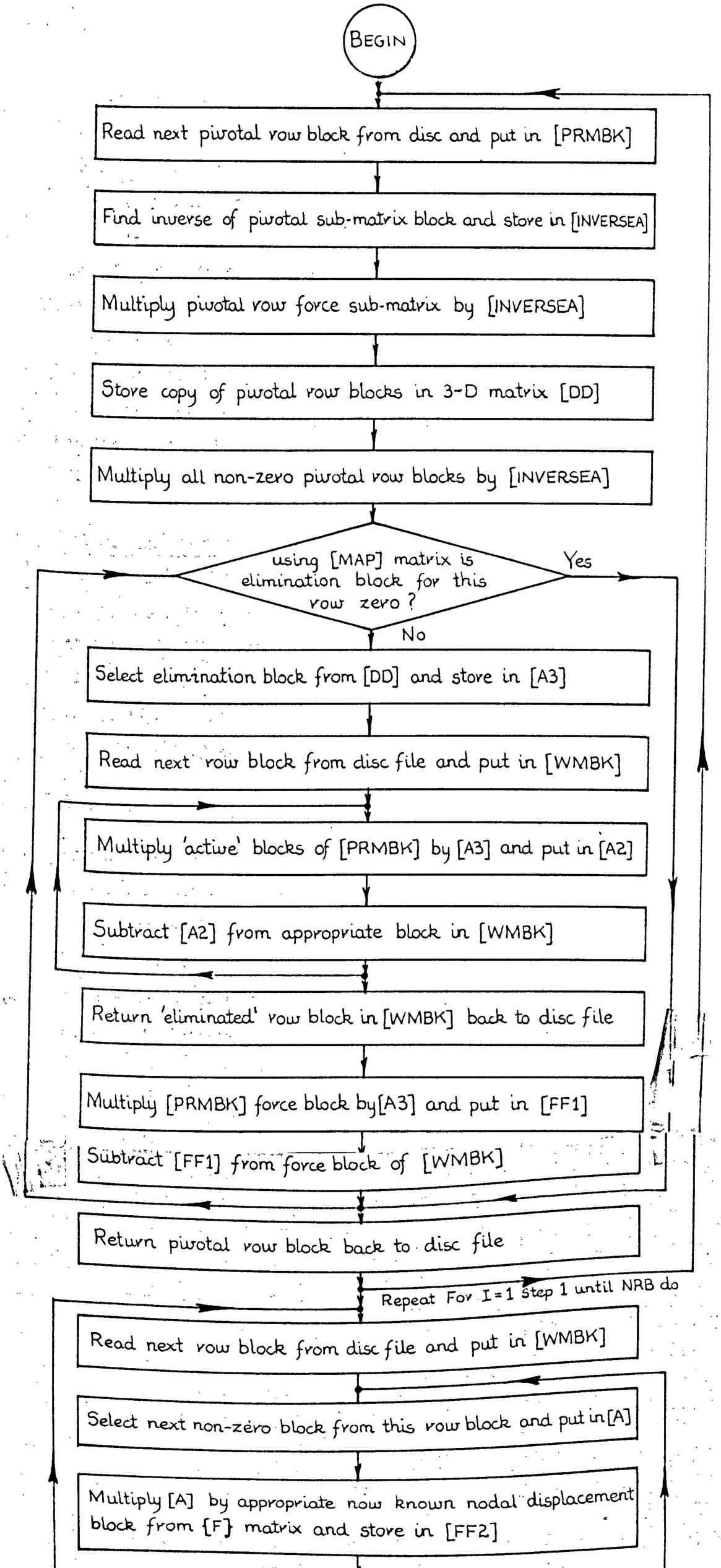
Structural		1		2		3		4		5		6			
1	R_1	k_{11}	k_{12}	k_{13}	k_{14}	k_{17}	k_{18}	k_{15}	k_{16}	0	0	0	0	d_1	u
	R_2	k_{21}	k_{22}	k_{23}	k_{24}	k_{27}	k_{28}	k_{25}	k_{26}	0	0	0	0	d_2	v
2	R_3	k_{31}	k_{32}	k_{33}	k_{34}	k_{37}	k_{38}	k_{35}	k_{36}	0	0	0	0	d_3	u
	R_4	k_{41}	k_{42}	k_{43}	k_{44}	k_{47}	k_{48}	k_{45}	k_{46}	0	0	0	0	d_4	v
3	R_5	k_{71}	k_{72}	k_{73}	k_{74}	$k_{77}+k_{11}$	$k_{78}+k_{12}$	$k_{75}+k_{13}$	$k_{76}+k_{14}$	0	k_{18}	0	0	d_5	u
	R_6	k_{81}	k_{82}	k_{83}	k_{84}	$k_{87}+k_{21}$	$k_{88}+k_{22}$	$k_{85}+k_{23}$	$k_{86}+k_{24}$	0	k_{28}	0	0	d_6	v
4	R_7	k_{51}	k_{52}	k_{53}	k_{54}	$k_{57}+k_{31}$	$k_{58}+k_{32}$	$k_{55}+k_{33}$	$k_{56}+k_{34}$	0	k_{38}	0	0	d_7	u
	R_8	k_{61}	k_{62}	k_{63}	k_{64}	$k_{67}+k_{41}$	$k_{68}+k_{42}$	$k_{65}+k_{43}$	$k_{66}+k_{44}$	0	k_{48}	0	0	d_8	v
5	0	0	0	0	0	0	0	0	0	1	0	0	0	0.000001	u
	R_{10}	0	0	0	0	k_{81}	k_{82}	k_{63}	k_{64}	0	k_{88}	0	0	d_{10}	v
6	0	0	0	0	0	0	0	0	0	0	0	1	0	0.000001	u
	0	0	0	0	0	0	0	0	0	0	0	0	1	0.000001	v
		u	v	u	v	u	v	u	v	u	v	u	v		

FIG. 15.5 The [K] matrix shown in FIG. 14.6d, modified to take account of the three constraints namely, d_9, d_{11} and d_{12}.

Structural

1	R_1		k_{11}	k_{12}	k_{13}	k_{14}	k_{17}	k_{18}	k_{15}	k_{16}		d_1		u
	R_2		k_{22}	k_{23}	k_{24}	k_{27}	k_{28}	k_{25}	k_{26}	0		d_2		v
2	R_3		k_{33}	k_{34}	k_{37}	k_{38}	k_{35}	k_{36}	0	0		d_3		u
	R_4		k_{44}	k_{47}	k_{48}	k_{45}	k_{46}	0	0	0		d_4		v
3	R_5		$k_{77}+k_{11}$	$k_{78}+k_{12}$	$k_{75}+k_{13}$	$k_{76}+k_{14}$	0	k_{18}	0	0		d_5		u
	R_6	=	$k_{88}+k_{22}$	$k_{85}+k_{23}$	$k_{86}+k_{24}$	0	k_{28}	0	0	0		d_6		v
4	R_7		$k_{55}+k_{33}$	$k_{56}+k_{34}$	0	k_{38}	0	0	0	0		d_7		u
	R_8		$k_{66}+k_{44}$	0	k_{48}	0	0	0	0	0		d_8		v
5	0		1	0	0	0	0	0	0	0		0·000001		u
	R_{10}		k_{88}	0	0	0	0	0	0	0		d_{10}		v
6	0		1	0	0	0	0	0	0	0		0·000001		u
	0		1	0	0	0	0	0	0	0		0·000001		v

FIG. 15.6 The [MBK] matrix shown in FIG. 15.4 modified to take account of the three constraints namely, d_9, d_{11} and d_{12}


BEGIN
Read next pivotal row block from disc and put in [PRMBK]
Find inverse of pivotal sub-matrix block and store in [INVERSEA]
Multiply pivotal row force sub-matrix by [INVERSEA]
Store copy of pivotal row blocks in 3-D matrix [DD]
Multiply all non-zero pivotal row blocks by [INVERSEA]
using [MAP] matrix is elimination block for this row zero ?
Yes
No
Select elimination block from [DD] and store in [A3]
Read next row block from disc file and put in [WMBK]
Multiply 'active' blocks of [PRMBK] by [A3] and put in [A2]
Subtract [A2] from appropriate block in [WMBK]
Return 'eliminated' row block in [WMBK] back to disc file
Multiply [PRMBK] force block by [A3] and put in [FF1]
Subtract [FF1] from force block of [WMBK]
Return pivotal row block back to disc file
Repeat For I = 1 step 1 until NRB do
Read next row block from disc file and put in [WMBK]
Select next non-zero block from this row block and put in [A]
Multiply [A] by appropriate now known nodal displacement block from {F} matrix and store in [FF2]

Subtract [FF1] from force block of [WMBK]

Return pivotal row block back to disc file

Repeat For I = 1 Step 1 until NRB do

Read next row block from disc file and put in [WMBK]

Select next non-zero block from this row block and put in [A]

Multiply [A] by appropriate now known nodal displacement block from {F} matrix and store in [FF2]

Subtract [FF2] from force block for this row.

Repeat For I = (NRB-1) Step -1 until 1 do.

END

FIG. 15.7 Flow diagram of the direct Gauss elimination 'Procedure' DBLOKGAUSS2. See listing in Appendix Three.

to avoid all operations on these blocks by referring to the [MAP] matrix. The [MAP] matrix simply tells the procedure with a zero if the sub-matrix block is null; a one indicating that the sub-matrix block is non-zero.

Although the storage and form of [MBK] required for the sub-matrix block approach is similar to that for single term processing, it must be remembered that the first column of [MBK] now consists of pivotal sub-matrix blocks. To illustrate this point, the arrangement of the [MBK] matrix shown in FIG. 15.6 in the equivalent sub-matrix block form is given in FIG. 15.8.

15. 7. DETERMINATION OF THE ELEMENT STRAIN AND STRESS COMPONENTS.

The strain and stress components are determined at the centroid of each of the finite elements. This gives a fairly representative or average estimate of the state of strain and stress acting in the element. Hence, by employing equation 14.21 and evaluating the [B] matrix at the centroidal point, the total strain vector $\{\varepsilon\}$ can be derived for each element using the appropriate nodal displacements.

Before the corresponding centroidal stress components can be evaluated, the element's thermal strains must first be deducted from the total strain vector. This done, the element's stresses can be determined using the appropriate material elasticity matrix and equation 14.24.

Structural

1	R_1	k_{11}	k_{12}	k_{13}	k_{14}	k_{17}	k_{18}	k_{15}	k_{16}	d_1	u
	R_2	k_{21}	k_{22}	k_{23}	k_{24}	k_{27}	k_{28}	k_{25}	k_{26}	d_2	v
2	R_3	k_{33}	k_{34}	k_{37}	k_{38}	k_{35}	k_{36}	0	0	d_3	u
	R_4	k_{43}	k_{44}	k_{47}	k_{48}	k_{45}	k_{46}	0	0	d_4	v
3	R_5	$k_{11}+k_{77}$	$k_{12}+k_{78}$	$k_{13}+k_{75}$	$k_{14}+k_{76}$	0	k_{18}	0	0	d_5	u
	R_6	$k_{21}+k_{87}$	$k_{22}+k_{88}$	$k_{23}+k_{85}$	$k_{24}+k_{86}$	0	k_{28}	0	0	d_6	v
4	R_7	$k_{33}+k_{55}$	$k_{34}+k_{56}$	0	k_{38}	0	0	0	0	d_7	u
	R_8	$k_{43}+k_{65}$	$k_{44}+k_{66}$	0	k_{48}	0	0	0	0	d_8	v
5	0	1	0	0	0	0	0	0	0	0·000001	u
	R_{10}	0	k_{88}	0	0	0	0	0	0	d_{10}	v
6	0	1	0	0	0	0	0	0	0	0·000001	u
	0	0	1	0	0	0	0	0	0	0·000001	v

FIG. 15.8 The [MBK] matrix shown in FIG. 15.6 in 2×2 sub-matrix block form. (Matrix modified to take account of the three constraints namely d_9, d_{11} and d_{12}.)

Both strain and stress are second order tensor quantities. Consequently, the strain and stress values determined, (which are the components corresponding to the directions of the Cartesian coordinate reference axis system selected for the problem), are not necessarily the maximum values actually present in the element. Hence, it is often these maximum or principal values which are required. These are determined for the axisymmetric element by employing the standard equations given in Chou and Pagano (85).

For principal strains, see page 44

$$\varepsilon \begin{matrix} max \\ min \end{matrix} = \frac{\varepsilon_{zz} + \varepsilon_{rr}}{2} \pm \frac{1}{2}\sqrt{(\varepsilon_{zz} - \varepsilon_{rr})^2 + \gamma_{rz}^2}$$

$$\text{and} \quad \gamma \begin{matrix} max \\ min \end{matrix} = \pm \sqrt{(\varepsilon_{rr} - \varepsilon_{zz})^2 + \gamma_{rz}^2}$$

and for principal stresses, see pages 10 and 11

$$\sigma \begin{matrix} max \\ min \end{matrix} = \frac{\sigma_{rr} + \sigma_{zz}}{2} \pm \sqrt{\left(\frac{\sigma_{rr} - \sigma_{zz}}{2}\right)^2 + \tau_{rz}^2} \qquad 15.10$$

$$\text{and} \quad \tau \begin{matrix} max \\ min \end{matrix} = \pm \sqrt{\left(\frac{\sigma_{rr} - \sigma_{zz}}{2}\right)^2 + \tau_{rz}^2}$$

The appropriate directions in which these principal quantities act are also determined from the Cartesian coordinate stress and strain values. These are also derived in the program. The sign convention adopted is fully explained in the comment given in the program listing, see Appendix Three.

15.8 AXISYMMETRIC DATA CHECK

The coordinate data for the axisymmetric program was checked by plotting the finite element subdivisions with the I.C.L. 1903A computer's plotting facility as outlined in 14.11. FIG. 7.12 shows a typical plot obtained using the axisymmetric data plot program (PLOTMESHAX3T). A listing of the program is given in Appendix Four.

The plot program only allows for the construction and labelling of the Cartesian coordinate axes and the subsequent drawing of the finite element mesh to some predetermined scale. The flow diagram of the program is given in FIG. 15.9.

15.9 AXISYMMETRIC PROGRAM TEST PROBLEMS

Several problems which have known and accepted solutions were analysed using the axisymmetric finite element analysis program. The finite element results were then subsequently compared with the 'known' solutions in order to examine the correctness of the design and logic of the computer program. Only two of the test problems will be discussed here.

The first problem examined was the well-known Boussinesq structure consisting of an infinite elastic half-space and loaded with a single point load, see Timoshenko and Goodier (134) page 364. Obviously, for finite element model purposes the infinite half-space had

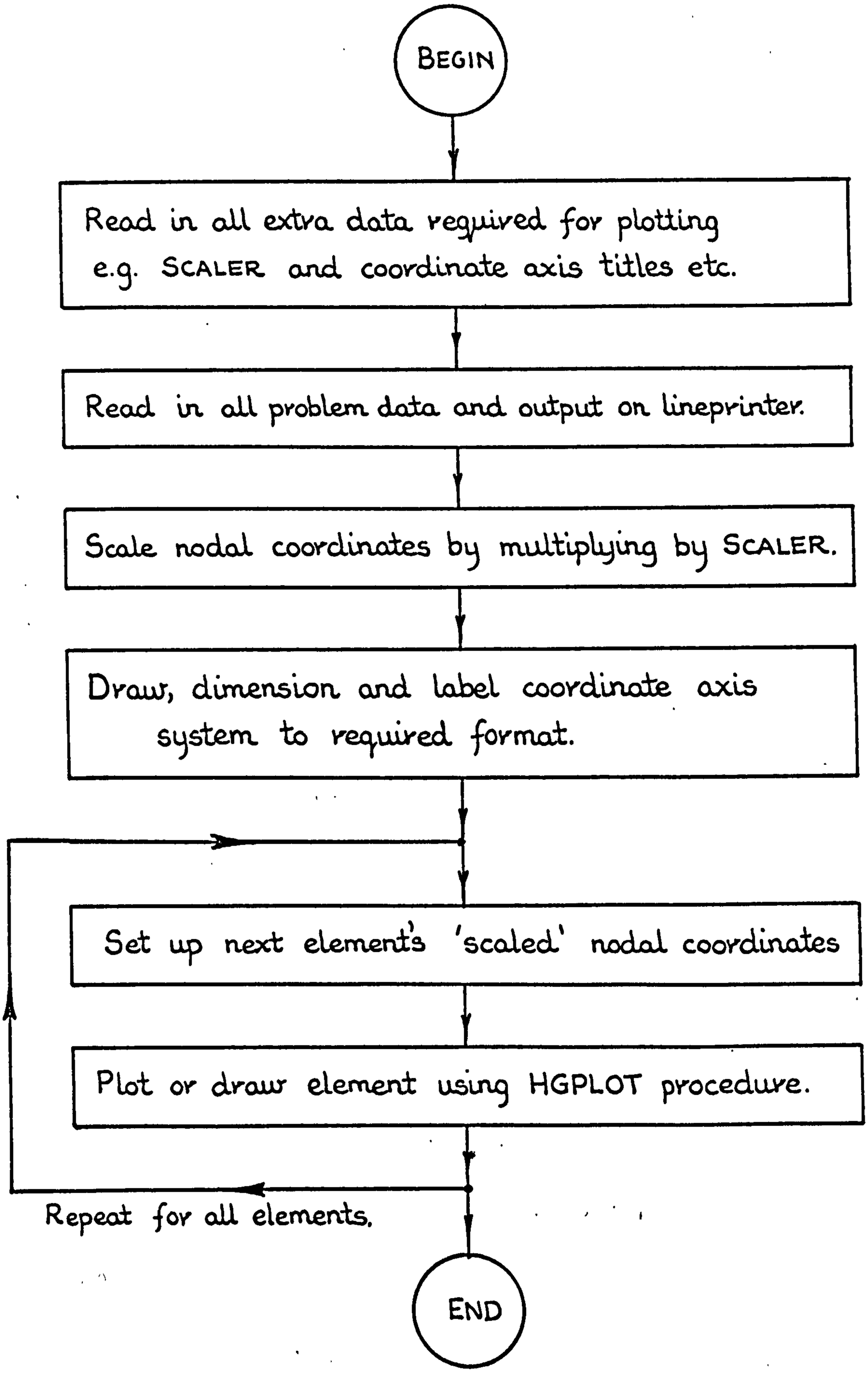


FIG. 15.9 Flow diagram of the axisymmetric finite element plot program. See Appendix Four.

to be restricted to one of finite size ; in this case to one of 14 inches radius by 14 inches deep. The finite element model is shown in FIG. 15.10. As can be seen from the figure, the structure was modelled by 186 elements having 114 nodes. The boundary conditions employed and the material properties are also shown in the figure.

FIG. 15.11 shows both the axisymmetric finite element and the 'exact' solutions for the vertical displacement of both the top surface of the half-space and at a depth of 2 inches. Similarly, FIG. 15.12 gives the vertical stress distribution obtained at two levels in the structure, one at a depth of 1 inch and the other at a depth of 3 inches. To obtain the values of the stresses at each point, the centroidal values of the elements adjacent to the point were averaged.

It can be seen that both the displacements and stresses obtained from the finite element analysis agree very favourably with the analytical solutions.

The second problem presented here was specifically selected to test the thermal analysis section of the program. For the Bousinesq problem, the whole structure was at ambient temperature and consequently, no thermal stresses were induced. The basic problem consisted of a disc 1 inch thick and having outside and inside diameters of 36 and 6 inches respectively. It was

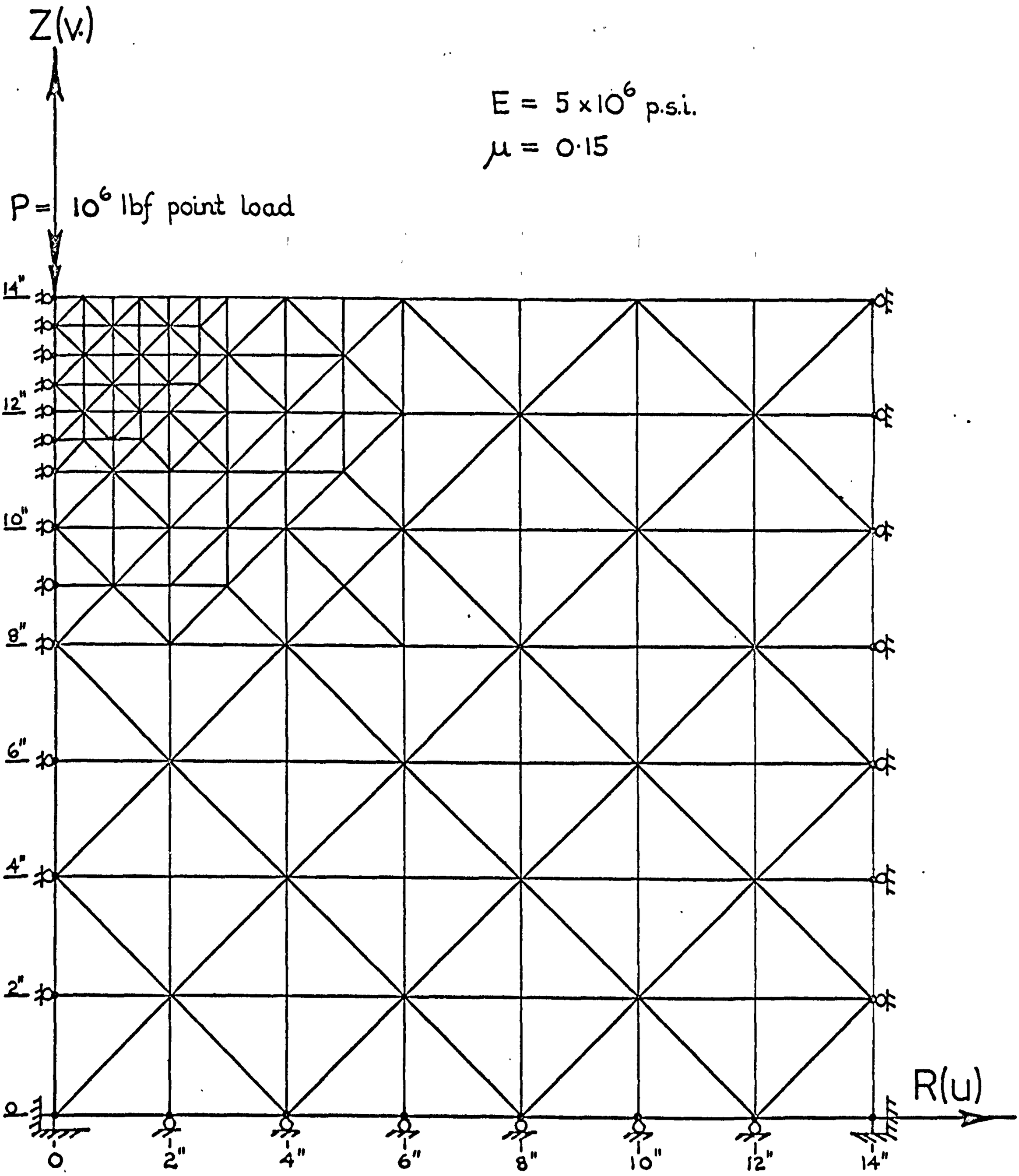


FIG. 15. 10 Axisymmetric finite element representation of Bousinesq structure. Model comprised of 186 elements.

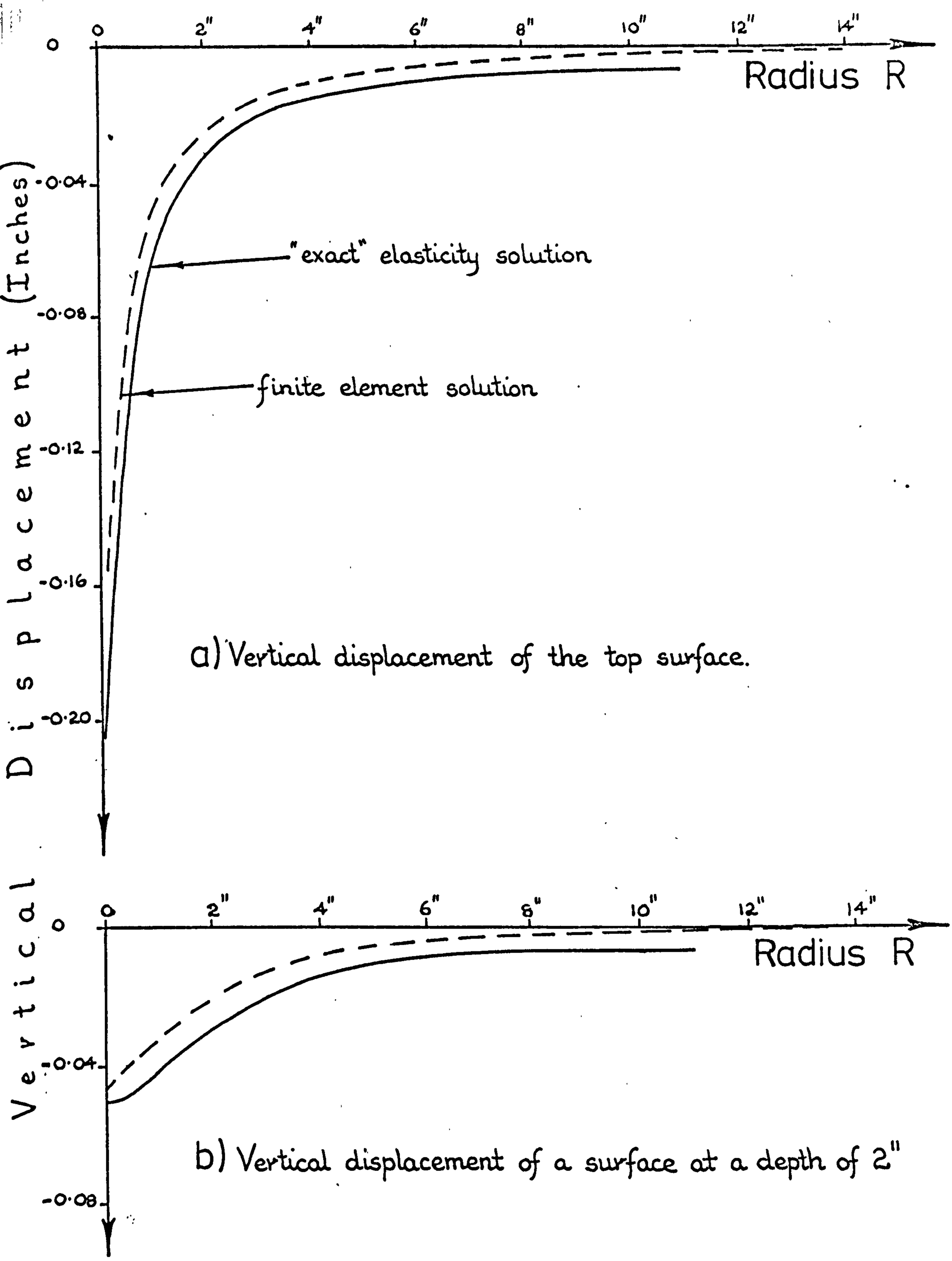


FIG. 15.11 Comparison of the vertical displacement distribution for the Bousinesq structure obtained from the axisymmetric finite element analysis and the "exact" elasticity theory, Timoshenko and Goodier (134).

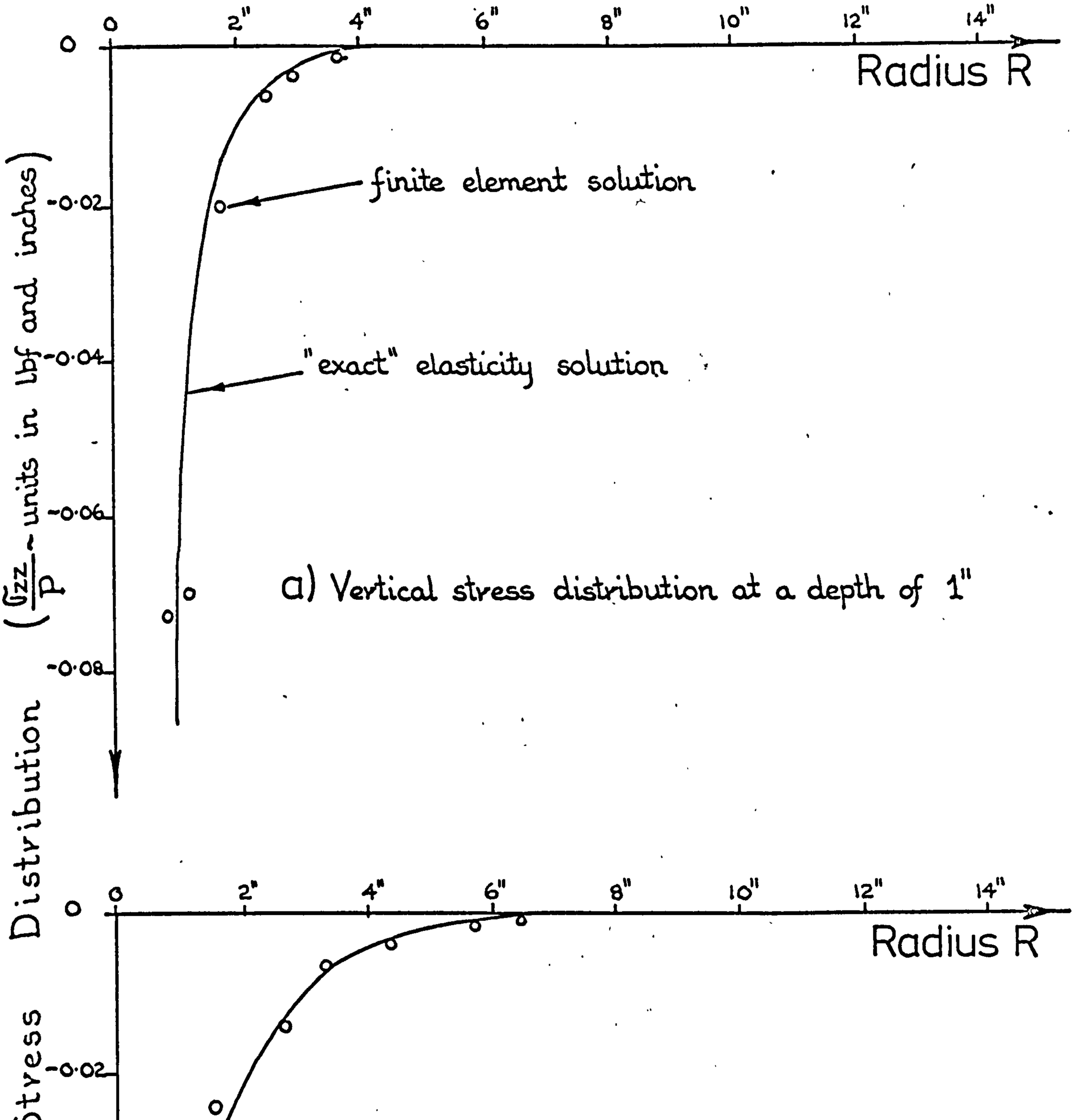
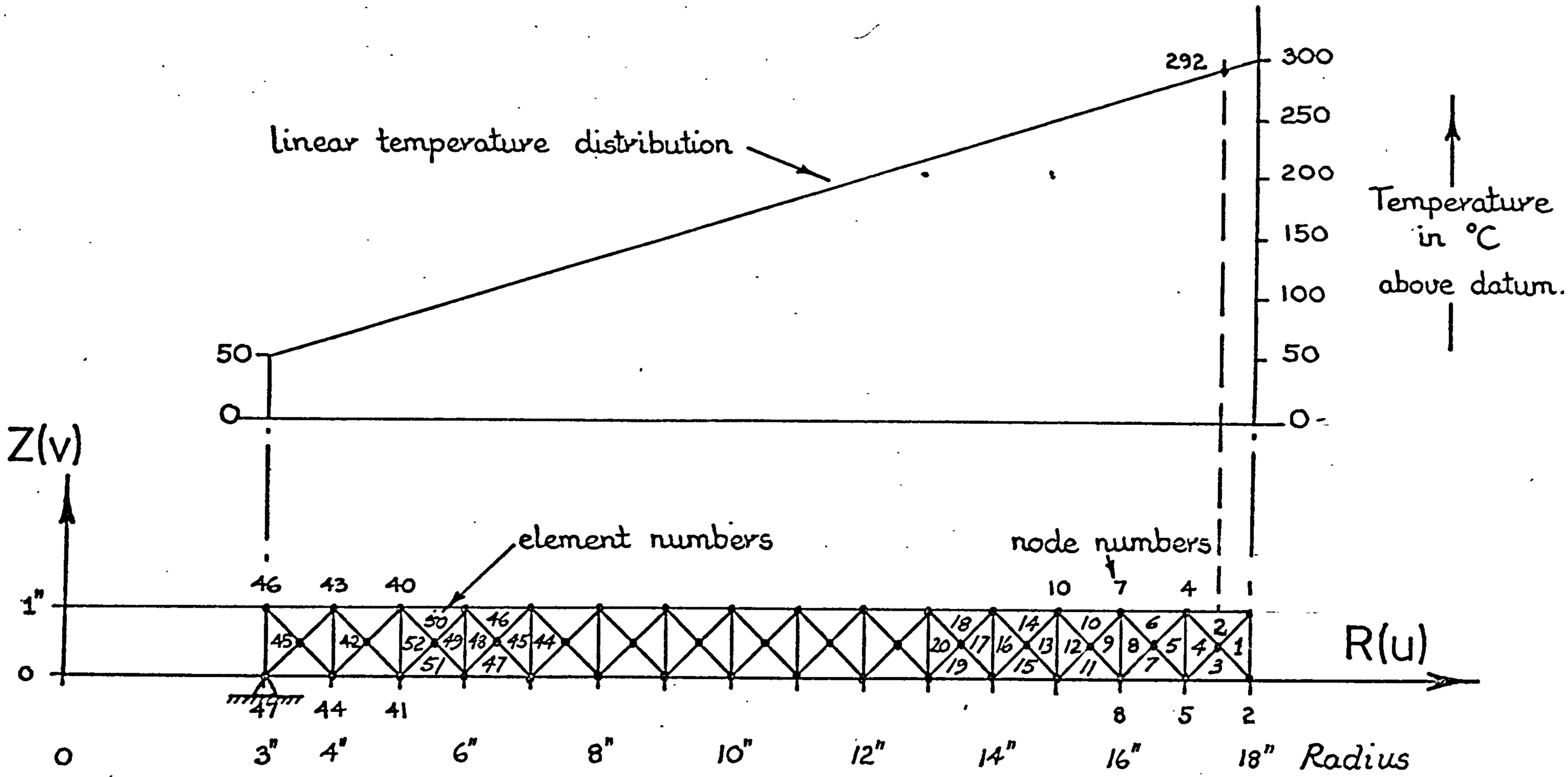


FIG. 15.12 Comparison of the vertical (compressive) stress distribution for the Bousinesq structure obtained from the axisymmetric finite element analysis and the "exact" elasticity theory, Timoshenko and Goodier (134).

assumed that the disc was subjected to a purely linear temperature gradient, having a temperature at the inner surface of 50 degrees C and at the outer surface 300 degrees C. The finite element mesh of the structure is shown in FIG. 15.13. This figure also shows how the average temperature of the elements above that of the datum temperature were derived. FIG.15.14 gives the complete layout of the data required for this problem together with the corresponding coding of the program listing. It must be pointed out that the dummy load of 0.01bf , is only for programming convenience. Although the Algol language allows the use of 'dynamic arrays' a load pointer vector of zero length is not a legitimate declaration.

A sample of the stress and strain components obtained from the computer output is given in FIG. 15.15. While FIG. 15.16 gives the analytical and finite element radial displacement distribution across the disc, FIG. 15.17 shows a comparison of the corresponding radial and hoop stress distributions. The analytical results were obtained from the formulae given in Timoshenko and Goodier (134) page 407. Again, there is very close agreement between the finite element and the analytical results.

From the problemsinvestigated,it was possible to derive an equation through which the computational time



$$E = 30 \times 10^{6} \text{ p.s.i.}$$
$$\mu = 0.3$$
$$\alpha = 11 \times 10^{-6} /°C$$

FIG. 15. 13 Axisymmetric finite element representation of a 1" thick disc, 6" I.D. and 36" O.D. Model comprised of 60 elements.

Algol Listing Variable	Data.
Job No.	101
Copystring	'DISC 1 INCH THICK 6 INCH BORE 36 INCH O.D. TEMP. 300 C O.D. TEMP. 50 C I.D. LINEAR TEMPERATURE DISTRIBUTION.'
Nelem	60
Nonop	47
Lo	1
Co	1
Manmd	3
Nomat	1

Mat. No.

	E	Mu	Excof
1	30&6	0·3	11&-6

Coord

	R	Z
1	18·0	1·0
2	18·0	0·0
⋮	⋮	⋮
46	3·0	1·0
47	3·0	0·0

Non

Elem Nº.	Mat. Nº.	Temp Nº.	Node Nos.(Anticlock) i	j	m
1	1	292	1	3	2
2	1	288	1	4	3
⋮	⋮	⋮	⋮	⋮	⋮
59	1	58	44	45	47
60	1	52	45	46	47

Co

	Equ. Nº.	Constraint
1	94	0·000001

Lo

	Equ. Nº.	Load
1	92	0·0

****

FIG. 15. 14 Data layout for axisymmetric disc problem.

***** ELEMENT CENTROIDAL STRESS COMPONENTS *****

ELEMENT NO.	STRESS Z STRAIN Z	STRESS R STRAIN R	STRESS THETA STRAIN THETA	STRESS RZ STRAIN RZ	SIGMA MAX EPSILON MAX	SIGMA MIN EPSILON MIN	TOR MAX GAMMA MAX	ALPHA BETA
1	-8.94035& 1	-3.45241& 2	-3.03249& 4	-7.65091& -4	-8.94035& 1	-3.45241& 2	-1.27919& 2	-0.0
	3.03721& -4	2.92635& -4	-1.00648& -3	-3.31539&-11	3.51572& -3	3.50463& -3	1.10863& -5	-0.0
2	-1.15255& 3	2.54847& 2	-2.98759& 4	1.24365& 3	9.80088& 2	-1.87779& 3	1.42894& 3	-120.2
	2.57793& -4	3.18780& -4	-9.86888& -4	5.38917& -5	3.51821& -3	3.39437& -3	1.23841& -4	-120.2
3	-1.15255& 3	2.54845& 2	-2.98759& 4	-1.24365& 3	9.80086& 2	-1.87779& 3	1.42894& 3	120.2
	2.57793& -4	3.18780& -4	-9.86888& -4	-5.38917& -5	3.51821& -3	3.39437& -3	1.23841& -4	120.2
4	2.42905& 2	3.31353& 3	-2.69903& 4	7.50169& -4	3.31353& 3	2.42905& 2	1.53531& 3	-90.0
	2.44864& -4	3.77925& -4	-9.35241& -4	3.25073&-11	3.46892& -3	3.33586& -3	1.33060& -4	-90.0
5	-1.85166& 2	-4.17936& 2	-2.79857& 4	-8.65605& -4	-1.85166& 2	-4.17936& 2	-1.16385& 2	-0.0
	2.77864& -4	2.67778& -4	-9.26827& -4	-3.75096&-11	3.33586& -3	3.32578& -3	1.00867& -5	-0.0
6	1.06071& 3	3.61698& 3	-2.44849& 4	1.41690& 3	4.24705& 3	4.30647& 2	1.90820& 3	-114.0
	2.44036& -4	3.54808& -4	-8.62940& -4	6.13990& -5	3.35211& -3	3.18673& -3	1.65377& -4	-114.0
7	1.06071& 3	3.61698& 3	-2.44849& 4	-1.41690& 3	4.24705& 3	4.30645& 2	1.90820& 3	114.0
	2.44036& -4	3.54808& -4	-8.62941& -4	-6.13990& -5	3.35211& -3	3.18673& -3	1.65377& -4	114.0
8	-1.83562& 3	3.50968& 3	-2.51492& 4	7.84439& -4	3.50968& 3	-1.83562& 3	2.67265& 3	-90.0
	1.55208& -4	3.86838& -4	-8.55049& -4	3.39924&-11	3.32384& -3	3.09221& -3	2.31630& -4	-90.0
9	6.16308& 2	2.09347& 3	-2.32599& 4	-9.41852& -4	2.09347& 3	6.16308& 2	7.38582& 2	90.0
	2.32208& -4	2.96219& -4	-8.02429& -4	-4.08136&-11	3.15622& -3	3.09221& -3	6.40105& -5	90.0
10	2.51720& 2	4.47914& 3	-2.13814& 4	1.41245& 3	4.90763& 3	-1.76774& 2	2.54220& 3	-106.9
	1.77413& -4	3.60601& -4	-7.60021& -4	6.12064& -5	3.17317& -3	2.95284& -3	2.20324& -4	-106.9

FIG. 15.15 Sample of the computer output obtained for the axisymmetric disc problem.

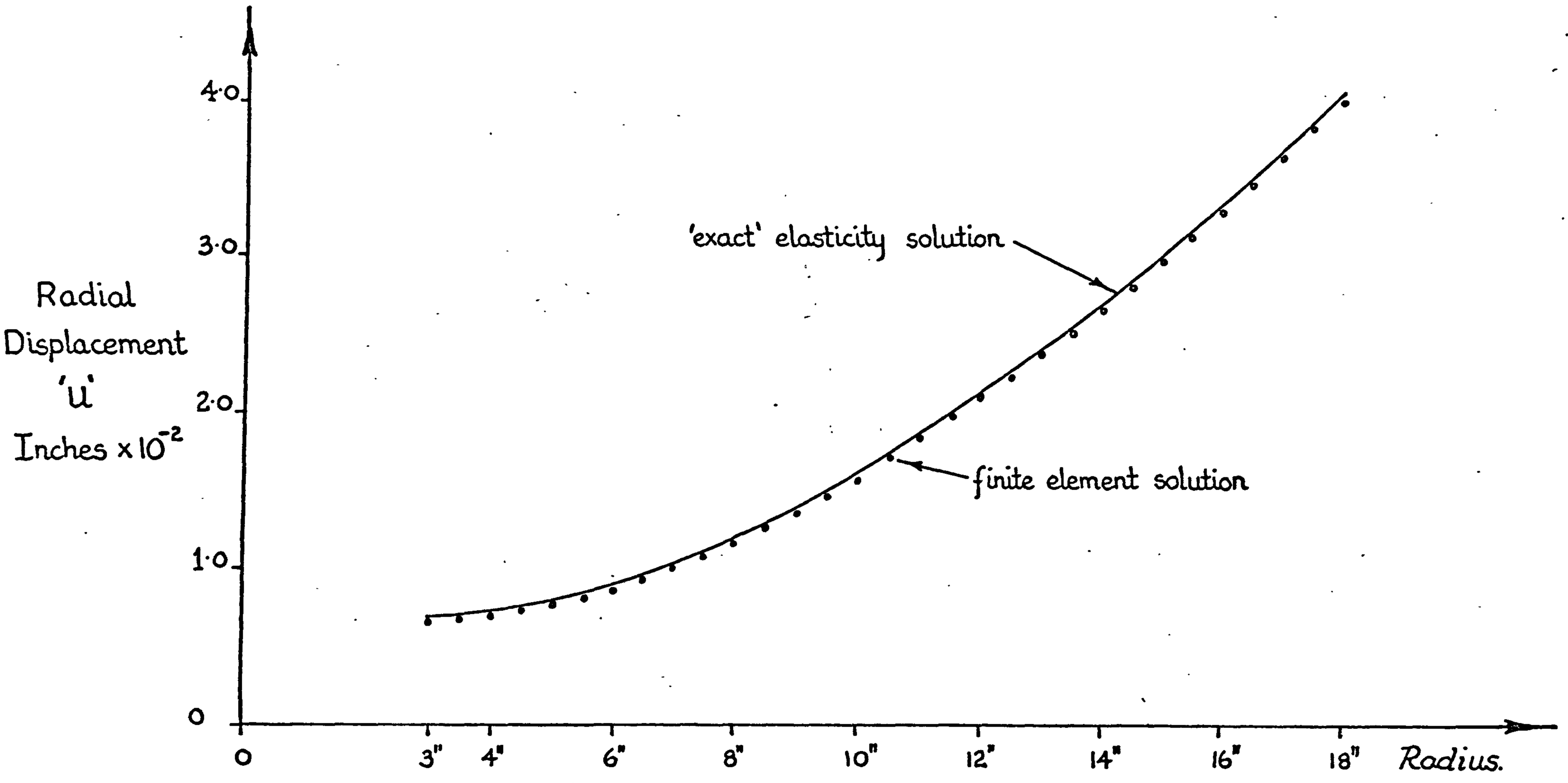


FIG. 15. 16 Comparison of the radial (u) displacement distribution for the disc structure obtained from the axisymmetric finite element analysis and the "exact" elasticity theory, Timoshenko and Goodier (134).

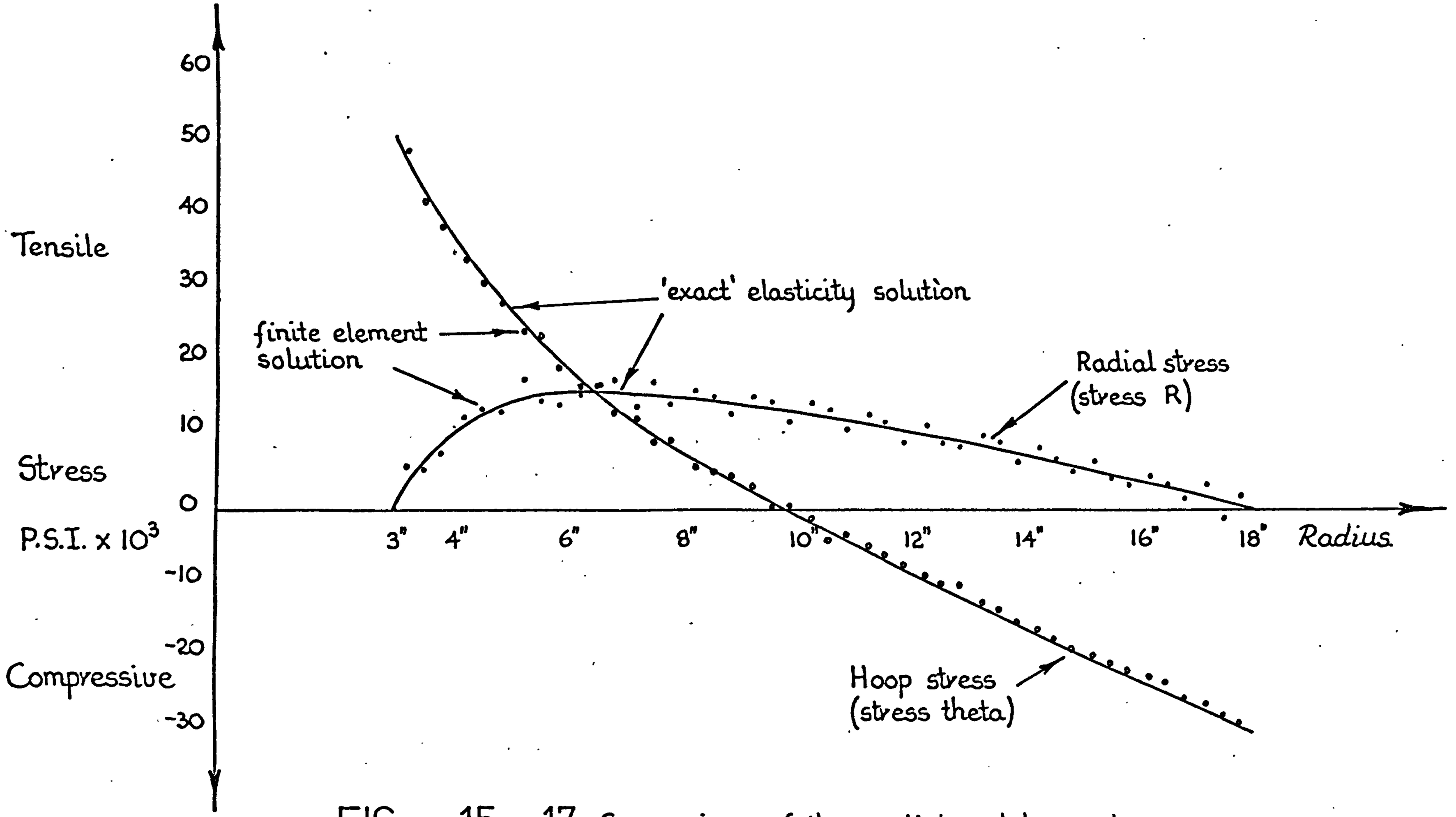


FIG. 15. 17 Comparison of the radial and hoop stress distribution for the disc structure obtained from the axisymmetric finite element analysis and the "exact" elasticity theory, Timoshenko and Goodier (134).

required for a specific structure could be estimated.

For the axisymmetric program this was found to be:-

$$\text{Time (seconds)} = \text{NELEM} + 0.01125\ ([\text{MANND} + 1]^2 \times \text{NONOP}) \quad 15.11$$

CHAPTER SIXTEEN

PLANE STRESS AND PLANE STRAIN FINITE ELEMENT ANALYSIS PROGRAMS

16 PLANE STRESS AND PLANE STRAIN FINITE ELEMENT ANALYSIS PROGRAMS.

Structures which are subjected to stresses or strains in only one plane are said to be in a condition of plane stress or plane strain respectively. Hence, the systems of stresses or strains are therefore only essentially two-dimensional in character, although in each case either a strain or a stress component can exist in the direction perpendicular to the two-dimensional plane.

From the finite element point of view, these two-dimensional stress and strain systems can be simulated by using two-dimensional plane stress and plane strain finite elements. Indeed for some structures, reasonable engineering answers can be obtained for physically three-dimensional structures using these two-dimensional simulations.

The technique for two-dimensional finite element analysis follows again the general procedural layout of FIG. 14.1. Also, because the two-dimensional element has, like the axisymmetric finite element, two degrees of freedom per node, the analysis programs are in many respects identical. The flow diagram for the plane stress finite element analysis program is given in FIG. 16.1. Again, the variables shown in the matrix brackets correspond to the Algol variable declarations of the computer program, a listing of which is given in Appendix Five. Because

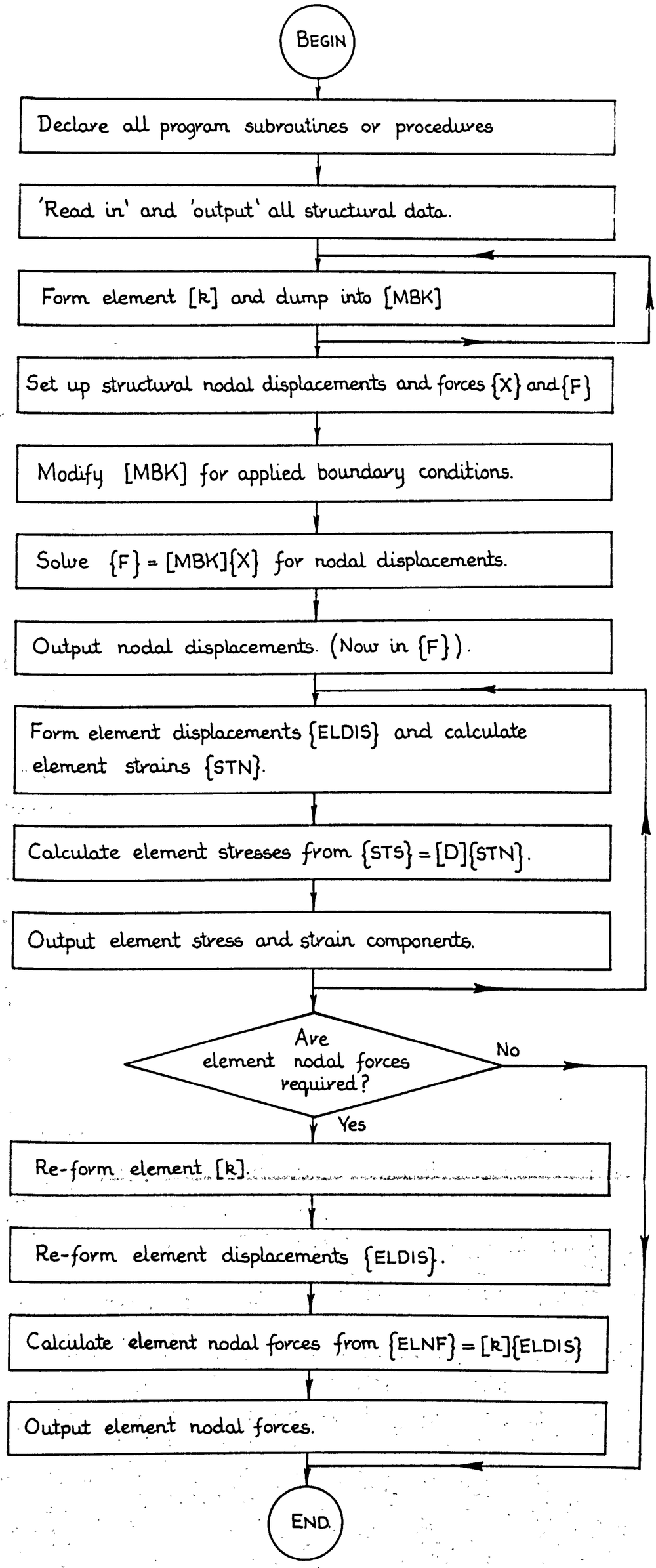


FIG. 16. 1 Flow diagram of the plane stress finite element analysis program.

the plane stress and plane strain programs are very similar, only the former will be dealt with here in detail. However, the major differences between the two will be pointed out in the discussion.

16.1 STRUCTURE DISCRETIZATION

The arbitary shaped constant thickness quadrilateral finite element shown in FIG. 16.2 was selected for the two-dimensional finite element analysis programs. Although, as was stated earlier, the triangular shaped element enables an easier grading of the structural sub-division, the extra node afforded by the quadrilateral element allows for a more refined and far superior form of displacement model to be employed.

16.2 DISPLACEMENT MODELS

For both the plane stress and plane strain cases, each node of the quadrilateral element was assumed to have only two degrees of freedom. That is the u and v displacement components associated with either the Cartesian coordinate directions X and Y or with the element local coordinate axes δ and η. Hence, the displacement of any point within the element can be expressed in terms of the local element coordinates and the nodal displacements by:-

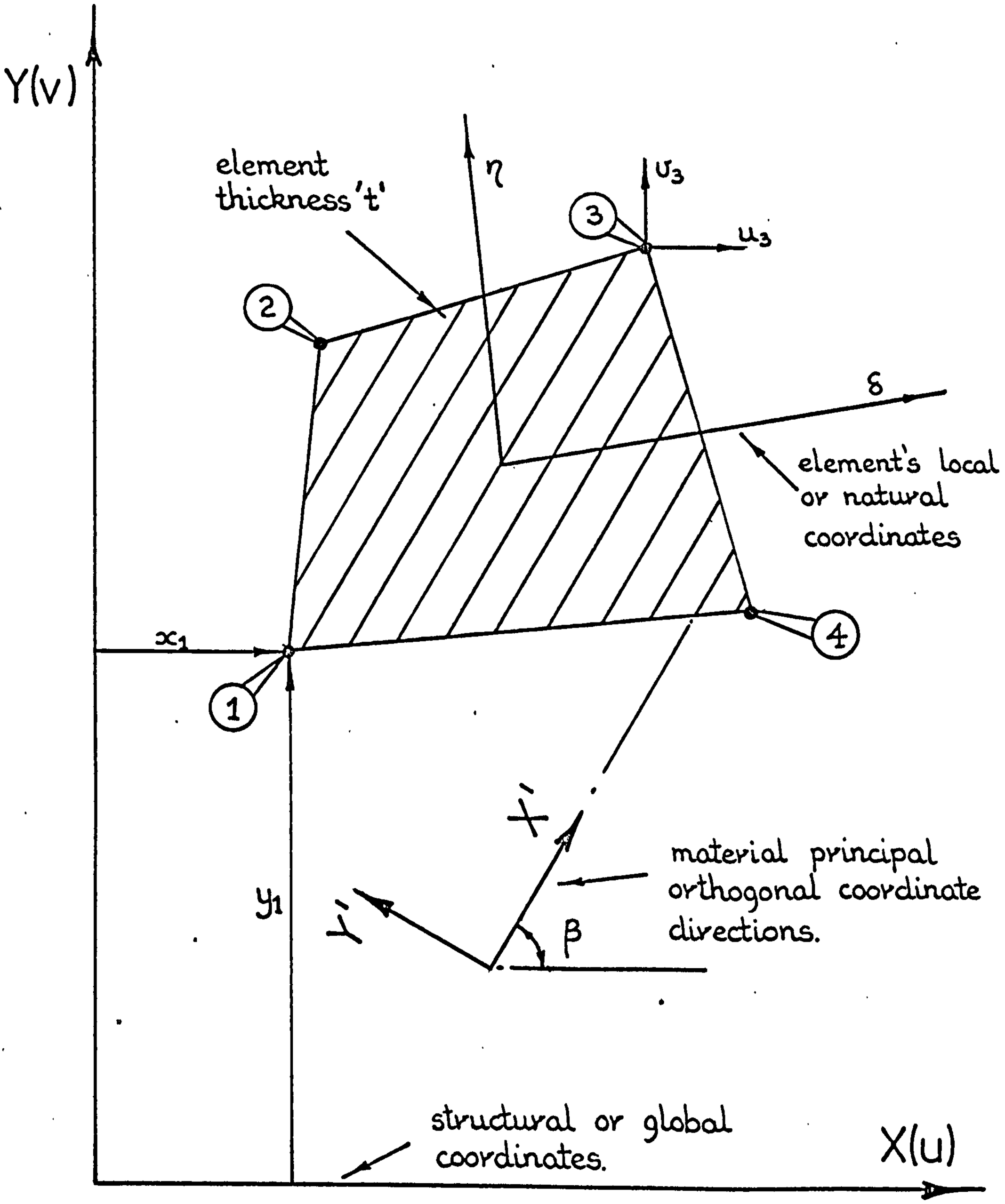


FIG. 16.2 Typical 4-noded plane stress orthotropic finite element 1.2.3.4. showing both the material and the element's local or natural coordinate directions.

$$u(\delta,\eta) = N_1u_1 + N_2u_2 + N_3u_3 + N_4u_4$$

and $$v(\delta,\eta) = N_1v_1 + N_2v_2 + N_3v_3 + N_4v_4 \qquad 16.1$$

(Note that the N functions which are given in section 14.3, are also expressed in terms of the element's local coordinates.)

16.3 DERIVATION OF THE PLANE STRESS AND PLANE STRAIN FINITE ELEMENT STIFFNESS MATRICES.

Neither body weight nor thermal loading effects were included in the two-dimensional analysis programs. Consequently equation 14.7 reduces to:-

$$\{F\} = [k]\{q\} \qquad 16.2$$

For both the plane stress and plane strain cases, there are only three strain components which contribute to the internal work of the element. These are:-

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} \qquad 16.3$$

ε_{xx} — horizontal strain

ε_{yy} — vertical strain

γ_{xy} — shear strain in the XY plane.

Although for the plane stress case, the strain perpendicular to the plane of the element is non-zero, by definition, the corresponding normal stress component is zero. Consequently, the product of these stress and strain components does not contribute to the total internal work of the element. A similar argument can also be applied to the plane strain situation. However, in this case, the normal strain component is the zero quantity with the normal stress component being non-zero.

The [B] matrix required in equation 14.21 cannot be determined directly from equations 16.1 because the element displacements are expressed in terms of the element's local coordinates δ and η. Consequently, these local coordinates must be transformed such that the derivatives required can be obtained in terms of the global or structural coordinates X and Y. Hence by equation 14.5a :-

$$\{f\} = [N_1 \quad N_2 \quad N_3 \quad N_4] \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} \qquad 16.4$$

And differentiating

$$\begin{Bmatrix} \frac{\partial f}{\partial \delta} \\ \frac{\partial f}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial \delta} & \frac{\partial N_2}{\partial \delta} & \frac{\partial N_3}{\partial \delta} & \frac{\partial N_4}{\partial \delta} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix}$$

or

$$\begin{Bmatrix} \frac{\partial f}{\partial \delta} \\ \frac{\partial f}{\partial \eta} \end{Bmatrix} = [\lambda] \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} \qquad 16.5$$

Now $$\frac{\partial f}{\partial \delta} = \frac{\partial x}{\partial \delta}\cdot\frac{\partial f}{\partial x} + \frac{\partial y}{\partial \delta}\cdot\frac{\partial f}{\partial y}$$

and $$\frac{\partial f}{\partial \eta} = \frac{\partial x}{\partial \eta}\cdot\frac{\partial f}{\partial x} + \frac{\partial y}{\partial \eta}\cdot\frac{\partial f}{\partial y}$$

or

$$\begin{Bmatrix} \frac{\partial f}{\partial \delta} \\ \frac{\partial f}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \delta} & \frac{\partial y}{\partial \delta} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{Bmatrix}$$

and

$$\begin{Bmatrix} \frac{\partial f}{\partial \delta} \\ \frac{\partial f}{\partial \eta} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{Bmatrix} \qquad 16.6$$

where [J] is the Jacobian transformation matrix. Hence, the quantities required, i.e. $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$ etc, can be obtained

by inverting equation 16.6 so that

$$\begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial f}{\partial \delta} \\ \frac{\partial f}{\partial \eta} \end{Bmatrix} \qquad 16.7$$

By invoking the isoparametric concept, the X and Y Cartesian coordinates of any point within the element can also be expressed in terms of the element's displacement models as shown by equation 14.6. Hence, by differentiating equation 14.6 and rearranging, we obtain

$$\begin{bmatrix} \frac{\partial x}{\partial \delta} & \frac{\partial y}{\partial \delta} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial \delta} & \frac{\partial N_2}{\partial \delta} & \frac{\partial N_3}{\partial \delta} & \frac{\partial N_4}{\partial \delta} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

which we recognise to be equivalent to

$$[J] = [\lambda] \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} \qquad 16.8$$

Hence, $[J]^{-1}$ can easily be determined by inverting the product of the $[\lambda]$ matrix and the element's nodal coordinates. Therefore, substituting equation 16.5 into equation 16.7 we see that

$$\begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{Bmatrix} = [J]^{-1}\,[\lambda] \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} \qquad 16.9$$

Thus, for the two-dimensional case where the nodal displacement components are u and v respectively, equation 16.9 becomes :-

$$\begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} = [J]^{-1}\,[\lambda] \begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \\ u_3 & v_3 \\ u_4 & v_4 \end{bmatrix} \qquad 16.10$$

Hence, the [B] matrix in equation 14.21 can simply be obtained from the terms of the $[J]^{-1}\,[\lambda]$ matrix product.

For the plane stress case, the stress components in a linear elastic orthotropic material are related to the strain components by :-

$$\varepsilon_{xx} = \frac{\sigma_{xx}}{Ex} - \frac{\mu_{yx}\,\sigma_{yy}}{Ey}$$

$$\varepsilon_{yy} = \frac{-\mu_{xy}\,\sigma_{xx}}{Ex} + \frac{\sigma_{yy}}{Ey}$$

$$\gamma_{xy} = \frac{\tau_{xy}}{Gxy}$$

and rearranging into the form of equation 14.23

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{Ex}{1 - \mu_{xy}\mu_{yx}} & \frac{\mu_{yx}\ Ex}{1 - \mu_{xy}\mu_{yx}} & 0 \\ \frac{\mu_{xy}\ Ey}{1 - \mu_{xy}\mu_{yx}} & \frac{Ey}{1 - \mu_{xy}\mu_{yx}} & 0 \\ 0 & 0 & Gxy \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} \qquad 16.11$$

or

$$\{\sigma\} = [D1]\{\varepsilon\} \qquad 16.12$$

Now the elasticity matrix [D1] is only applicable provided that the orthotropic material axes coincide with those of the global axes X and Y, for example, property Ex is the Young's modulus of the material in the X coordinate direction. Suppose therefore that the material axes about which the mechanical properties are known, differ from the global axis system as shown by the general element in FIG. 16.2. In order to determine the elasticity matrix for this element with respect to the global directions, another coordinate transformation has to be carried out. This can be derived in one of two ways. For the general case it can be carried out using the Cartesian tensor approach. However, here it will be achieved by employing the two-dimensional principal stress and strain relationships.

The stresses on any plane of a two-dimensional structure can be evaluated from the known stress components on any other plane by the well known principal stress relationships, Chou and Pagano (85) page 8.

$$\begin{Bmatrix} \sigma_{xx}' \\ \sigma_{yy}' \\ \tau_{xy}' \end{Bmatrix} = \begin{bmatrix} \cos^2\beta & \sin^2\beta & 2\sin\beta\cos\beta \\ \sin^2\beta & \cos^2\beta & -2\sin\beta\cos\beta \\ -\sin\beta\cos\beta & \sin\beta\cos\beta & (\cos^2\beta-\sin^2\beta) \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} \quad 16.13$$

or $$\{\sigma'\} = [T]\{\sigma\} \quad 16.14$$

Here, the primed components refer to the primed orthotropic material axes of the element shown in FIG.16.2. The angle β is assumed positive when measured in an anticlockwise sense from the global X axis to the materials X' axis. The strain components on the primed axes can also be deduced from those on the global axes by a similar relationship, Chou and Pagano (85) pages 43-44.

$$\begin{Bmatrix} \varepsilon_{xx}' \\ \varepsilon_{yy}' \\ \gamma_{xy}' \end{Bmatrix} = \begin{bmatrix} \cos^2\beta & \sin^2\beta & \sin\beta\cos\beta \\ \sin^2\beta & \cos^2\beta & -\sin\beta\cos\beta \\ -2\sin\beta\cos\beta & 2\sin\beta\cos\beta & (\cos^2\beta-\sin^2\beta) \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} \quad 16.15$$

Unfortunately, the axes transformation matrix is not as it stands, the same for the strain components as it is for the stresses. However, if instead of using the engineering shear strain component γ_{xy}, we use the tensorial shear strain ε_{xy} (which is equal to $\underline{\gamma_{xy}}$) we obtain:-

$$\begin{Bmatrix} \varepsilon_{xx}' \\ \varepsilon_{yy}' \\ \varepsilon_{xy}' \end{Bmatrix} = \begin{bmatrix} \cos^2\beta & \sin^2\beta & 2\sin\beta\cos\beta \\ \sin^2\beta & \cos^2\beta & -2\sin\beta\cos\beta \\ -\sin\beta\cos\beta & \sin\beta\cos\beta & (\cos^2\beta-\sin^2\beta) \end{bmatrix}^2 \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{Bmatrix} \quad 16.16$$

or $$\{\underline{\varepsilon}'\} = [T]\{\underline{\varepsilon}\} \qquad 16.17$$

(The underline in the above is to signify tensorial strain vectors.) If the elasticity matrix [D1] in equation 16.12 is now modified so that the strain vector is also in tensor form, i.e. by multiplyirg Gxy by 2, then the stress components in the element's primed axis system can be written using equation 16.12 as :-

$$\{\sigma'\} = [\underline{D1}]\{\underline{\varepsilon}'\} \qquad 16.18$$

But substituting from equations 16.14 and 16.17 we have

$$[T]\{\sigma\} = [\underline{D1}][T]\{\underline{\varepsilon}\}$$

and $$\{\sigma\} = [T]^{-1}[\underline{D1}][T]\{\underline{\varepsilon}\} \qquad 16.19$$

or $$\{\sigma\} = [\underline{D}]\{\underline{\varepsilon}\}$$

where $$[\underline{D}] = [T]^{-1}[\underline{D1}][T]$$

Finally, if the [D] matrix is modified so that the tensor strain vector $\{\underline{\varepsilon}\}$ can be replaced with the engineering strain vector $\{\varepsilon\}$ we obtain the elasticity matrix for the element in terms of the global coordinate axis system, i.e.

$$\{\sigma\} = [D]\{\varepsilon\}$$

To obtain the [D] matrix, the third column of [D] must simply be divided by 2.

The element stiffness matrix [k] can be derived now that the [B] and [D] matrices have been evaluated, i.e.

$$[k] = \int [B]^T [D][B]\, dVol$$

For this case the integral is

$$\int dVol = \int t\,dx\,dy$$

However, the [B] matrix has been expressed in terms of the element's local coordinate system. Consequently, the

integral must be evaluated with respect to the δ and η coordinates. Thus :-

$$\int dVol = \int_{-1}^{1}\int_{-1}^{1} t \frac{\partial(x,y)}{\partial(\delta,\eta)} \, d\delta \, d\eta$$

where $\frac{\partial(x,y)}{\partial(\delta,\eta)}$ is equal to the determinant of the [J] matrix. Hence, the element stiffness matrix becomes:-

$$[k] = \int_{-1}^{1}\int_{-1}^{1} [B]^T [D] [B] \, t \, \det[J] \, d\delta \, d\eta \qquad 16.20$$

This integral can be easily evaluated using the Gauss quadrature formula, Kopal (136). For the plane stress program, 3 by 3 integrating points were employed and hence equation 16.20 is evaluated at nine discrete points, see Appendix One.

The flow diagram for the derivation of [k] in terms of the corresponding Algol variables used in the program is given in FIG. 16.3.

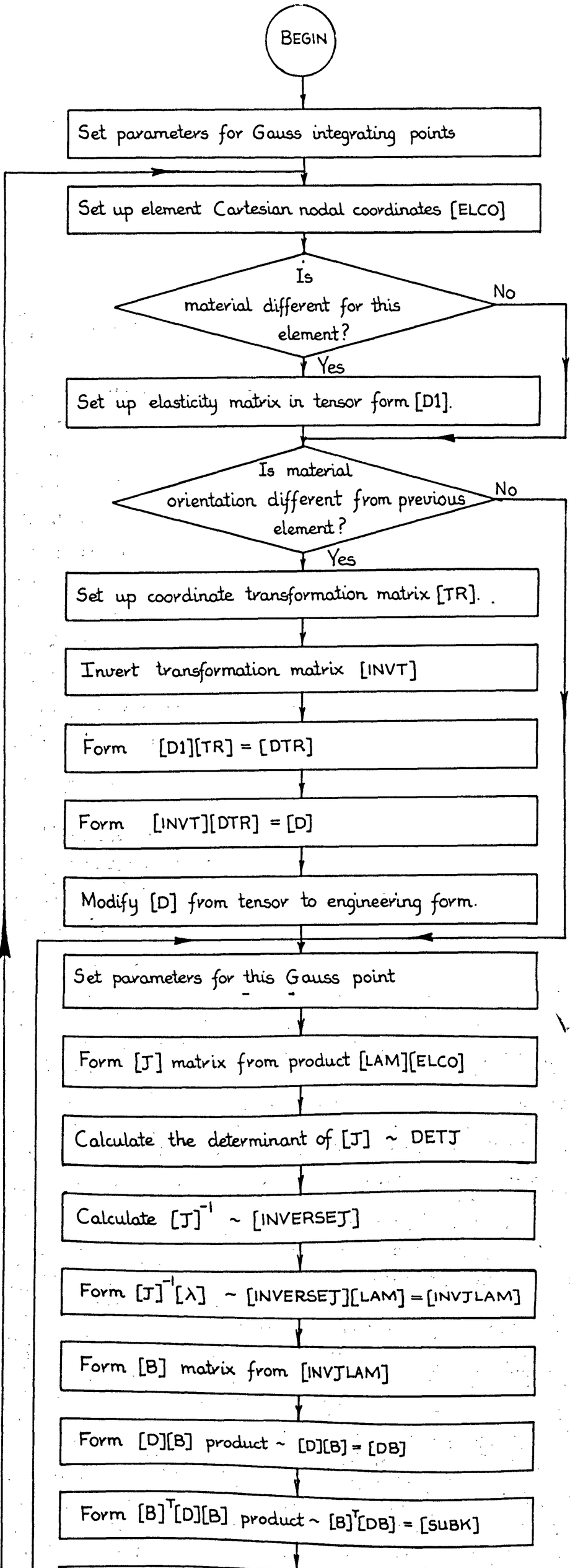
The procedure for determining the stiffness matrix of the corresponding plane strain element is very similar to that for the plane stress case previously discussed. However, for the plane strain element, the strain-stress relationships are :-

$$\varepsilon_{xx} = \frac{\sigma_{xx}}{Ex} - \frac{\mu_{yx}}{Ey}\sigma_{yy} - \frac{\mu_{zx}}{Ez}\sigma_{zz}$$

$$\varepsilon_{yy} = \frac{-\mu_{xy}\sigma_{xx}}{Ex} + \frac{\sigma_{yy}}{Ey} - \frac{\mu_{zy}}{Ez}\sigma_{zz}$$

$$\gamma_{xy} = \frac{\tau_{xy}}{Gxy}$$

because σ_{zz} is not zero. However, ε_{zz} is, by definition zero so that :-


BEGIN
Set parameters for Gauss integrating points
Set up element Cartesian nodal coordinates [ELCO]
Is material different for this element?
No
Yes
Set up elasticity matrix in tensor form [D1].
Is material orientation different from previous element?
No
Yes
Set up coordinate transformation matrix [TR].
Invert transformation matrix [INVT]
Form [D1][TR] = [DTR]
Form [INVT][DTR] = [D]
Modify [D] from tensor to engineering form.
Set parameters for this Gauss point
Form [J] matrix from product [LAM][ELCO]
Calculate the determinant of [J] ~ DETJ
Calculate $[J]^{-1}$ ~ [INVERSEJ]
Form $[J]^{-1}[\lambda]$ ~ [INVERSEJ][LAM] = [INVJLAM]
Form [B] matrix from [INVJLAM]
Form [D][B] product ~ [D][B] = [DB]
Form $[B]^T[D][B]$ product ~ $[B]^T$[DB] = [SUBK]

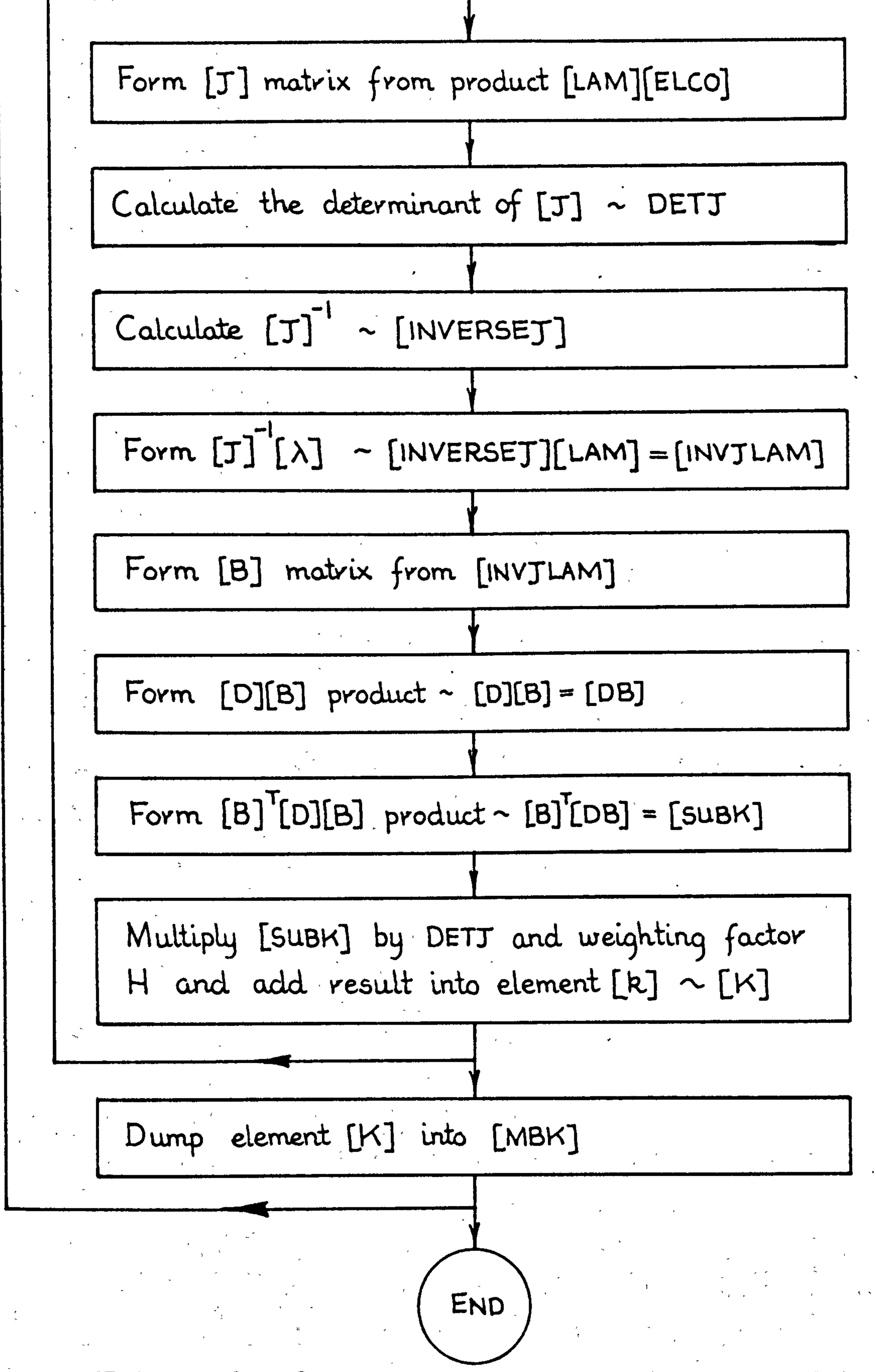


FIG. 16. 3 Flow diagram for the determination of the [k] matrix in the plane stress finite element analysis program.

$$\varepsilon_{zz} = 0 = -\frac{\mu_{xz}}{Ex}\sigma_{xx} - \frac{\mu_{yz}}{Ey}\sigma_{yy} + \frac{\sigma_{zz}}{Ez}$$

Consequently, σ_{zz} is expressed in terms of σ_{xx} and σ_{yy} and the appropriate modifications made to the [D1] matrix. The procedure is then identical to that followed for the plane stress element.

16.4 FORMATION OF THE STRUCTURAL EQUILIBRIUM EQUATIONS.

The two-dimensional finite element, like the axisymmetric element discussed in Chapter Fifteen, possesses only two degrees of freedom per node. Consequently, the same equation solution procedure DBLOKGAUSS2 was employed to solve the structural equilibrium equations. Therefore, the equilibrium equations were arranged and stored on the disc backing store file in the same manner as they were for the axisymmetric case. The only significant difference between the formation of the equilibrium equations for the axisymmetric and plane stress cases was that the latter element had four nodes whereas the former only had three. Hence, the ''dumping'' process had to be slightly modified so as to take this additional node into account.

16.5 APPLICATION OF THE STRUCTURAL BOUNDARY CONDITIONS.

As for the axisymmetric program, only boundary point loads and kinematic constraint type boundary conditions were allowed for in the plane stress analysis program.

Consequently, they were applied and the corresponding modifications made to the [MBK] matrix in exactly the same manner as before. Of course, in this case, the element's nodes are really lines extending through the total thickness 't' of the element. Thus, the nodal point loads and constraints are assumed to act over the whole thickness of the element.

16.6 SOLUTION OF THE STRUCTURAL EQUILIBRIUM EQUATIONS.

Again, due to the similarity between the axisymmetric and plane stress situations, exactly the same equation solution procedure was employed, see 15.6.

16.7 DETERMINATION OF THE ELEMENT STRAIN AND STRESS COMPONENTS.

Although the strain and stress components can be determined at any point within the plane stress element, the position selected was again at the element's centroid. Thus, the [B] matrix is evaluated for the values of $\delta = \eta = 0$ and the element strain vector obtained using equation 14.21. The corresponding stress components are then subsequently obtained by equation 14.23, after of course, the appropriate elasticity matrix for the element has been evaluated.

The principal strains and stresses are again determined for each element using equation 15.10. (For the plane stress case the z subscript is merely

replaced by y and the r subscript by x.) The directions in which these principal values occur are also determined. However, it must be remembered that with orthotropic materials the direction of the principal stresses do not necessarily coincide with the directions of the principal strains.

16.8 DETERMINATION OF THE ELEMENT NODAL FORCES.

For some of the dental structural analyses, it was desired to determine the force distribution occuring on the socket walls of teeth, subjected to lateral and/or intrusively applied loads. Although this could be obtained from the principal stresses, the method employed here was to determine the equivalent nodal forces acting on the elements adjacent to the socket wall.

In the plane stress program, the number of elements for which the nodal forces are required is read into NELNF; the numbers of the individual elements themselves are read subsequently into the array [NOELF]. The program then loops for each of these elements, deriving both the element stiffness matrix [k] and the nodal displacement vector [ELDIS]. The element nodal forces are then derived by equation 16.2.

It is obvious that this aspect of the program is inefficient as the stiffness matrices of the elements at which the equivalent nodal forces are required, have already been determined during the formation of the structural equilibrium equations. Consequently, it

would be advantageous, if storage space were available, to store the appropriate [k]'s so that they could be recalled for the subsequent nodal force calculation.

16.9 PLANE STRESS AND PLANE STRAIN DATA CHECKS.

The punched data for these programs was checked using the plotting technique discussed in 14.11. The flow diagram for the process is similar to that shown in FIG. 15.9. Various versions of the program were written to include either node numbering, element numbering or the indication of the principal material property direction, i.e. X' in FIG. 16.2 representing the direction of the major Young's Modulus Ex. Due to the size restriction of the plotting facility itself and the limited symbolic scales available, all these features are not desirable for the purposes of clarity on a single computer plot.

A computer listing of a typical plot program is given in Appendix Six.

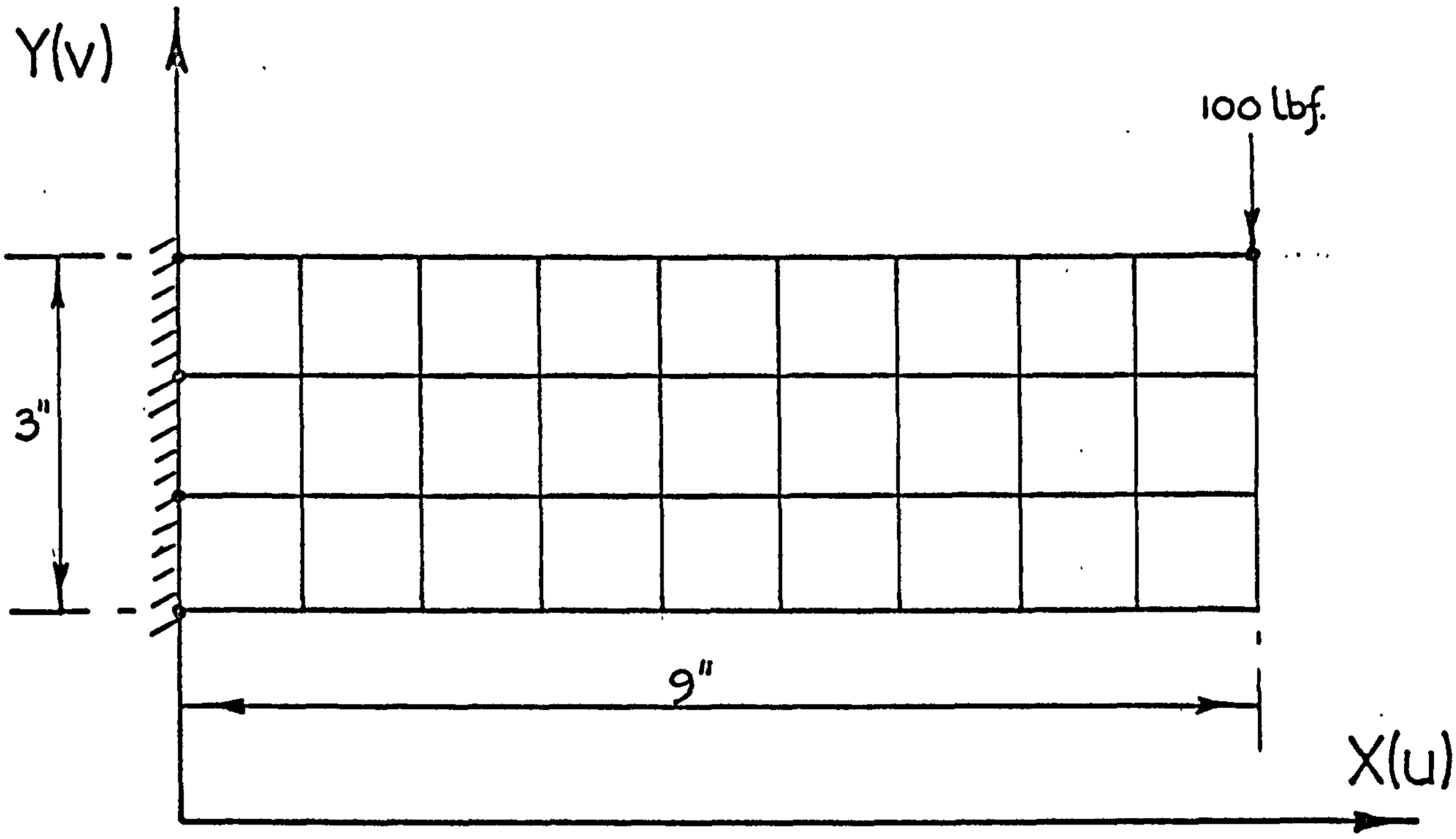
16.10 PLANE STRESS PROGRAM TEST PROBLEMS.

Various problems involving isotropic materials and having known solutions were examined using the plane stress finite element analysis program. However, known solutions to problems involving orthotropic materials are very scarce. Indeed, only one set of results, obtained from an experimental analysis employing electrical resistance strain gauges could be found, Greszczuk (83).

Consequently, the orthotropic feature of the program could not be rigorously tested. Only two of the test problems investigated will be discussed here.

The first problem which involved only isotropic materials, was the simple cantilever structure having a length to depth aspect ratio of 3 to 1, shown in FIG. 16.4. This problem has been investigated by many authors and FIG. 16.5 gives a comparison of their results for the tip deflection and those obtained using the plane stress program. The structure was analysed using a 9 by 3 ''normal'' element mesh as shown in FIG. 16.4a and by the 65 element ''cross-grained'' mesh shown in FIG. 16.4b. As can be seen, the tip deflections obtained from both these meshes are quite accurate results although that of the more normal mesh arrangement is superior to that of the cross-grained mesh. This implies that even though a greater number of elements and degrees of freedom are employed, the cross-grained arrangement represents a much stiffer idealisation.

The second problem was specifically chosen to test the orthotropic material section of the program. However, the only published work which could be found was by Greszczuk (83), who examined the differences in the directions of the principal stresses and strains in a simple tensile specimen of a fibre glass composite material. Greszczuk varied the direction of the unidirectional glass fibre reinforcement in his

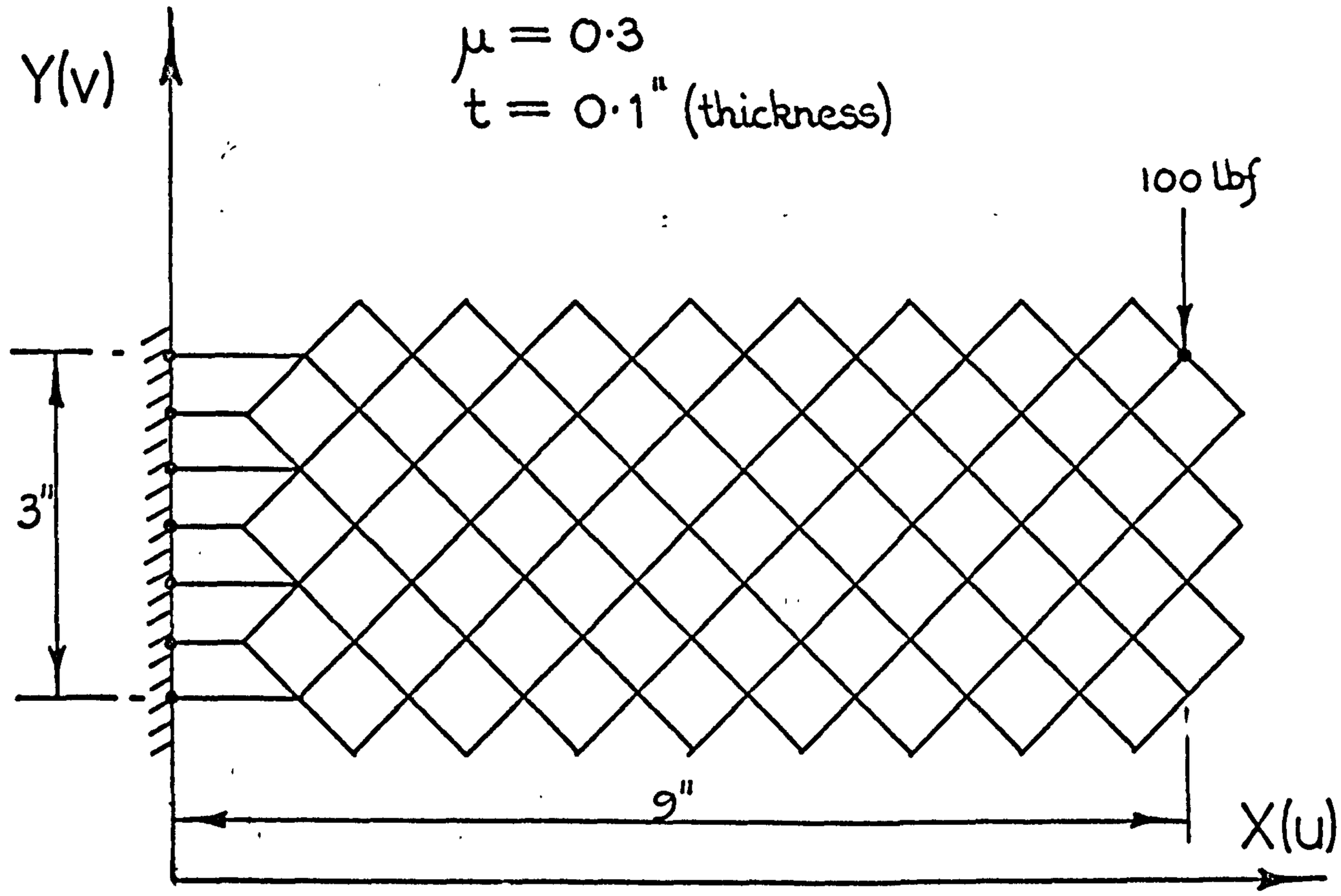


a) 9 x 3 'normal' element mesh

$E = 10^6$ p.s.i.

$\mu = 0.3$

$t = 0.1"$ (thickness)



b) 65 (approx. 10 x 4) 'cross-grained' element mesh.

FIG. 16. 4 Two-dimensional representations of cantilever structure having 3 to 1 aspect ratio.

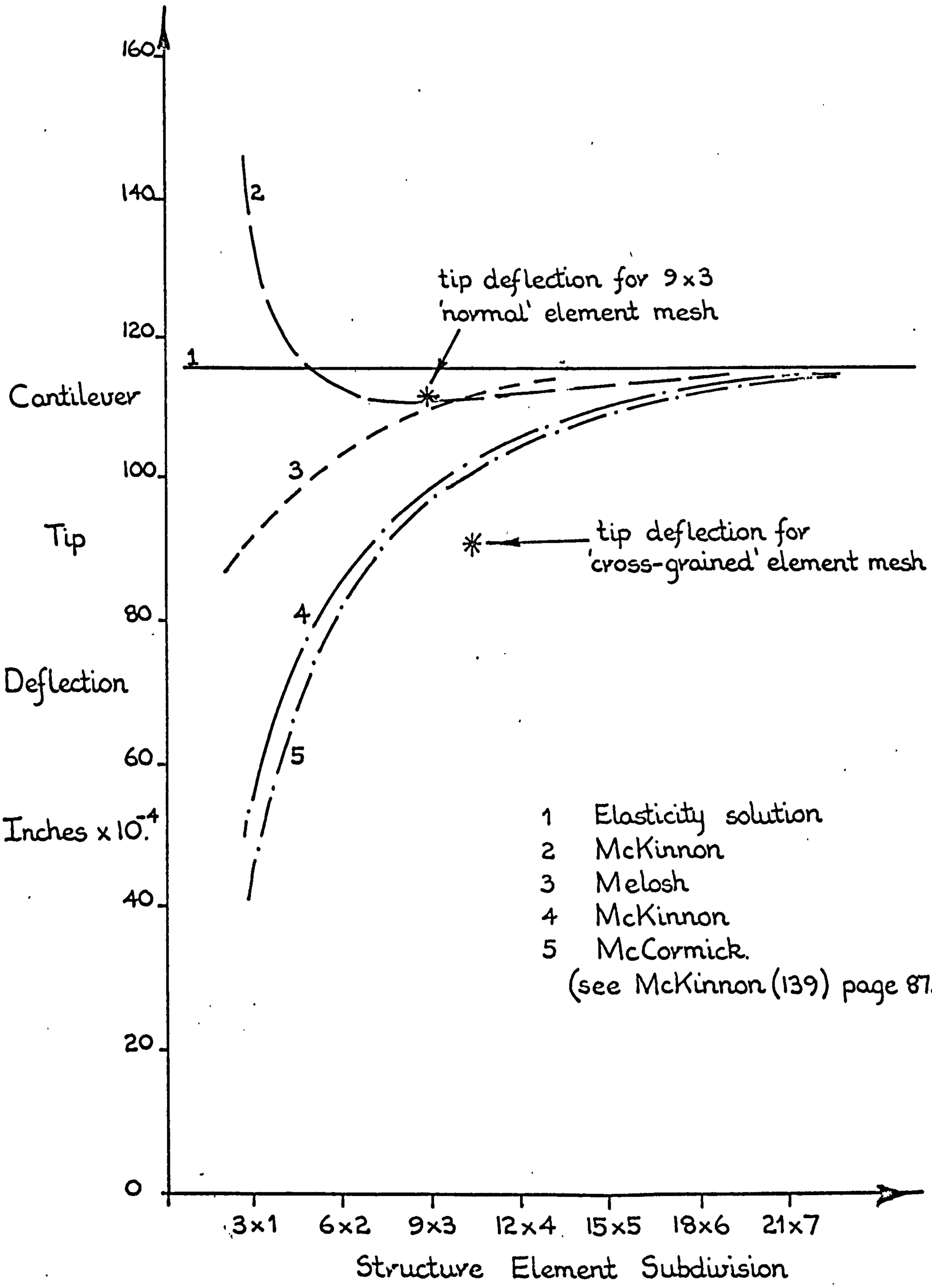
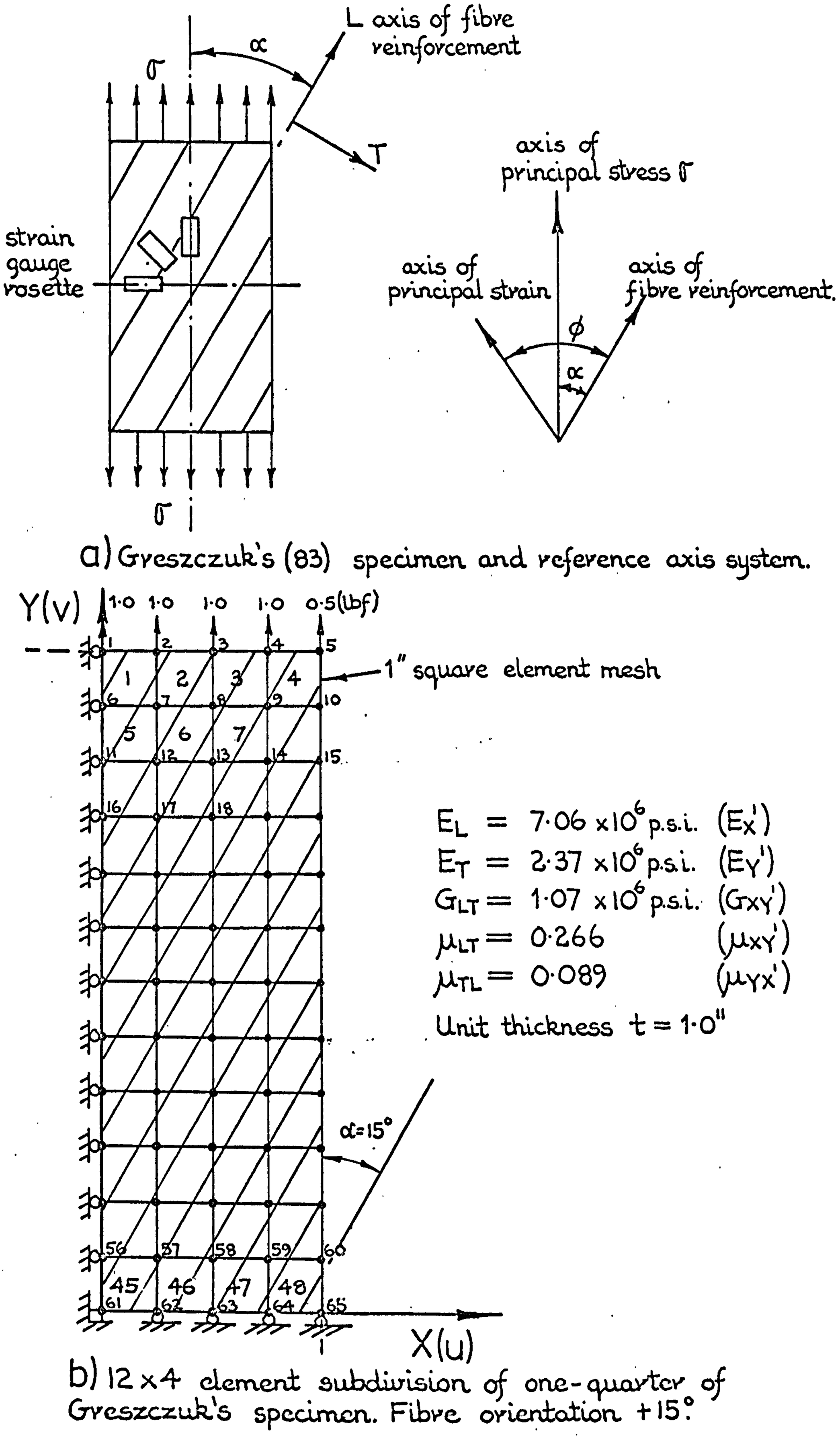


FIG. 16.5 Comparison of tip deflection results for the cantilever structure shown in FIG. 16.4 and those obtained by other authors.

specimens and determined the directions of the principal strains by means of a strain gauge rosette, see FIG. 16.6. This figure also gives the 48 element, 65 node,one quarter finite element idealisation while FIG. 16.7 gives an abridged layout of the data required for the problem, together with the corresponding coding used in the computer program. FIG. 16.8 gives Greszczuk's and the finite element results for the differences in the directions of the principal stresses and strains for various fibre reinforcement orientations. The results are indeed in very close agreement.

Due to the similarity in the axisymmetric and two-dimensional programs, an estimate of the computional time required for a plane stress or plane strain problem can be obtained from equation 15.11.



a) Greszczuk's (83) specimen and reference axis system.

b) 12 x 4 element subdivision of one-quarter of Greszczuk's specimen. Fibre orientation +15°.

FIG. 16. 6 Two-dimensional representation of orthotropic tensile specimen.

Algol Listing Variable	Data
Job No.	7
Copystring	'ORTHOTROPIC TENSILE TEST SPECIMEN FIBRE ORIENTATION +15 DEGREES ANTICLOCK TO +VE X AXIS.'
Nelem	48
Nonop	65
Lo	5
Co	18
Mannd	6
Nomat	1
Nothick	1
Nelnf	4

Mat. No.

	EX	MU-XY	EY	MU-YX	GXY
1	7·06&6	0·266	2·37&6	0·089	1·07&6

Elthick

	Thickness
1	1·0

Coord

	X	Y
1	0·0	12·0
2	1·0	12·0
:	:	:
64	3·0	0·0
65	4·0	0·0

Non

Elem. No.	Mat. No.	Orien. No.	Thick. No.	Node Nos. (Clockwise) i	j	k	m
1	1	75	1	1	2	7	6
2	1	75	1	2	3	8	7
:	:	:	:	:	:	:	:
47	1	75	1	58	59	64	63
48	1	75	1	59	60	65	64

Co

	Equ No.	Constraint
1	1	0·000001
:	:	:
18	130	0·000001

Lo

	Equ.No.	Load
1	2	1·0
:	:	:
5	10	0·5

Noelf

	Elem. No.
1	45
:	:
4	48
	****

FIG. 16. 7 Data layout for two-dimensional orthotropic test specimen problem, see FIG. 16.6.

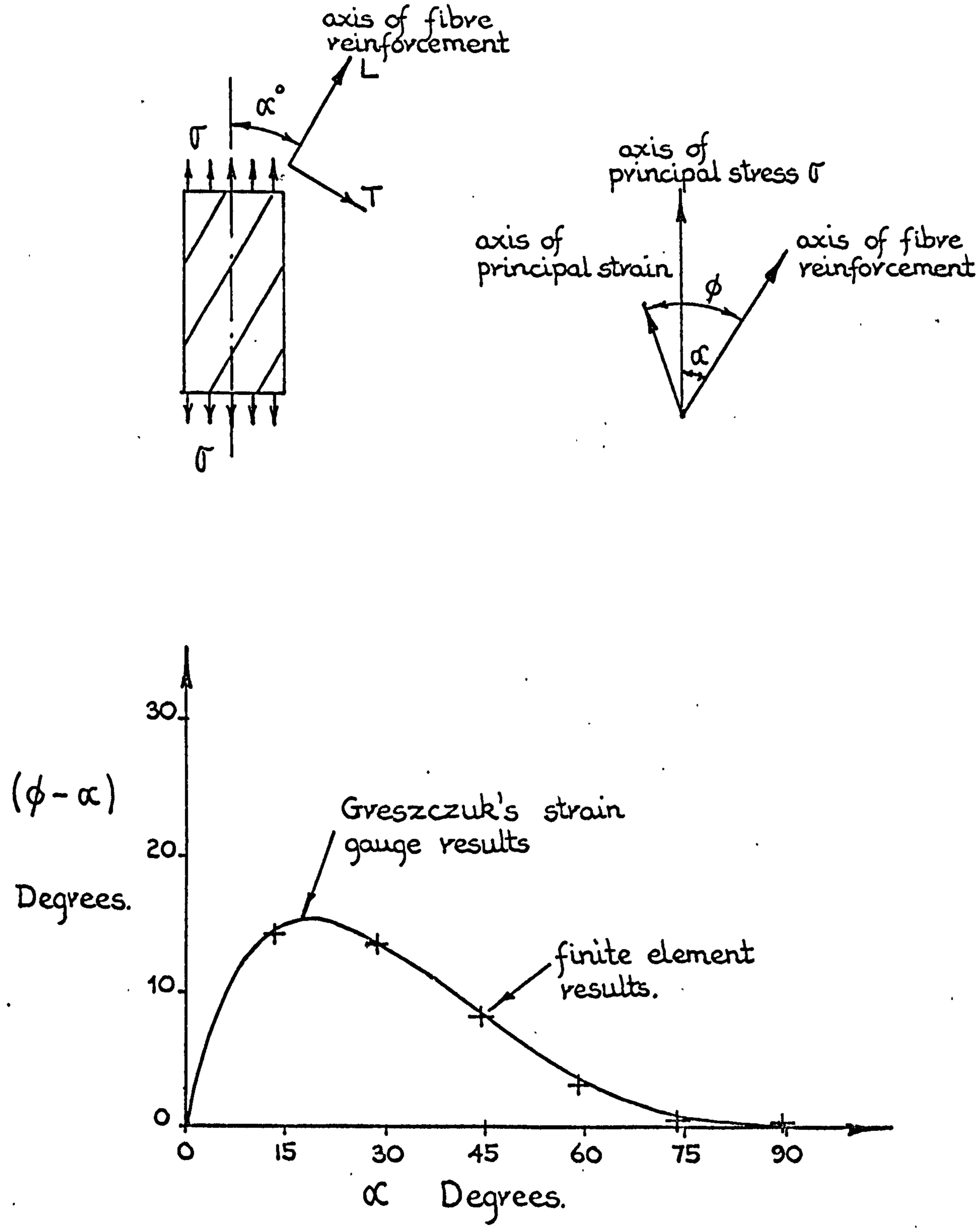


FIG. 16. 8 Comparison of the finite element and Greszczuk's (83) strain gauge results for the differences in the directions of the principal stresses and strains for various fibre reinforcement orientations.

CHAPTER SEVENTEEN

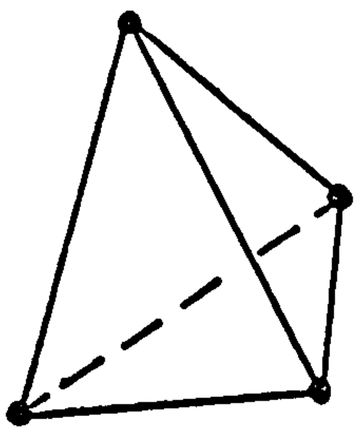
THREE-DIMENSIONAL FINITE ELEMENT ANALYSIS PROGRAMS

17 THREE-DIMENSIONAL FINITE ELEMENT ANALYSIS PROGRAMS

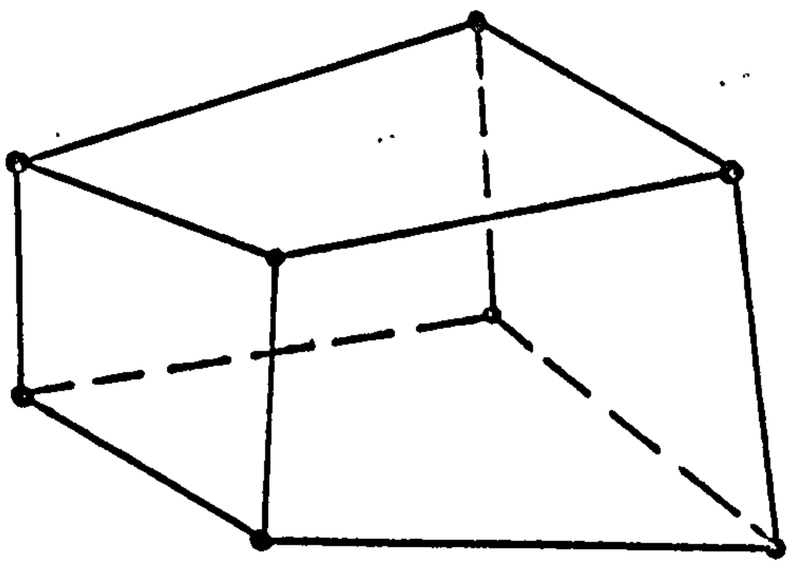
Structures that are generally three-dimensional in character, can in some cases be adequately represented and analysed using simple two-dimensional models. Even so, many problems arise where a full three-dimensional analysis has to be carried out if 'acceptable' solutions are to be obtained. In the finite element approach, these three-dimensional structures must be represented by three-dimensional type finite element models.

The simplest three-dimensional finite element, (which corresponds to the 3-noded triangular element used in two-dimensional and axisymmetric analyses), is the 4-noded tetrahedron type element shown in FIG. 17.1a. However, it has been found that this element is inefficient with respect to the computational time required and the accuracy of the obtained solution when compared with the hexahedra type of elements, Clough (137). Consequently, the 8-noded and 20-noded hexahedra finite elements also shown in FIG. 17.1 were adopted for the work reported herein.

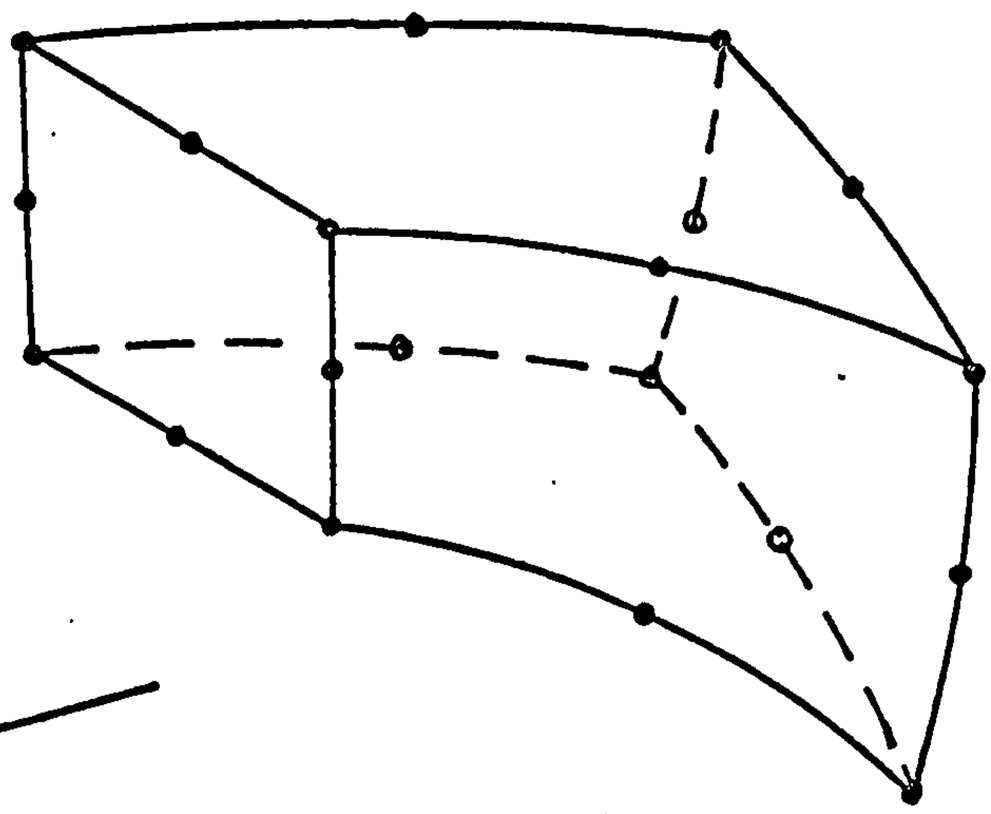
The computational time required for fully three-dimensional analyses is several orders of magnitude greater than that required for an equivalent



a) Tetrahedron 3-D finite element.



b) 8-noded hexahedra 3-D finite element.



c) 20-noded hexahedra 3-D finite element.

FIG. 17. 1 Some typical three-dimensional finite elements.

two-dimensional or axisymmetric analysis having either a similar number of elements or nodal points. This is because of the increase in the time required to form the element stiffness matrices and due to the increased bandwidth of the resulting structural equilibrium equations. Consequently, much effort has been expended to find ways of improving the economics of three-dimensional finite element analyses.

Here, the method of conjugate gradients has been employed to solve the structural equilibrium equations obtained from both 8 and 20-noded finite element idealisations. However, it was found that this method, at least, for the dental structural type of problems encountered, was inferior to that of the direct Gaussian elimination method employed for the axisymmetric and two-dimensional programs. For completeness however, a brief discussion of the work carried out is included in Appendix Two.

The technique employed for the three-dimensional analyses using the direct Gaussian elimination approach follows again the general procedural layout of FIG. 14.1. The two three-dimensional elements employed, namely the 8 and 20-noded elements shown in FIG. 17.1, were each incorporated into a separate program. Consequently, both programs follow the general pattern of the plane stress program discussed in the previous chapter.

Again, because both the 8 and 20-noded programs were very similar, only the 20-noded version will be discussed in detail. However, the main differences between the two programs will be indicated. The flow diagram for the 20-noded program, expressed in terms of the corresponding Algol variables utilized in the program listing given in Appendix Seven, is shown in FIG. 17.2.

17.1 STRUCTURE DISCRETIZATION

FIG. 17.3 shows the arbitrary shaped 8 and 20-noded isoparametric hexahedra type of finite elements selected for the three-dimensional finite element analysis programs. The figure also shows the correspondence between the global Cartesian XYZ and the orthogonal orthotropic material $\bar{X}\bar{Y}\bar{Z}$ axis systems. The elements' local coordinate directions are also shown together with the element node numbering sequence systems employed.

17.2 DISPLACEMENT MODELS

The nodes for both the 8 and 20-noded finite elements were each ascribed three degrees of freedom, namely, u, v and w displacements associated with the global XYZ Cartesian coordinate axis directions respectively. Consequently, the displacement of any point within either of the elements can be expressed

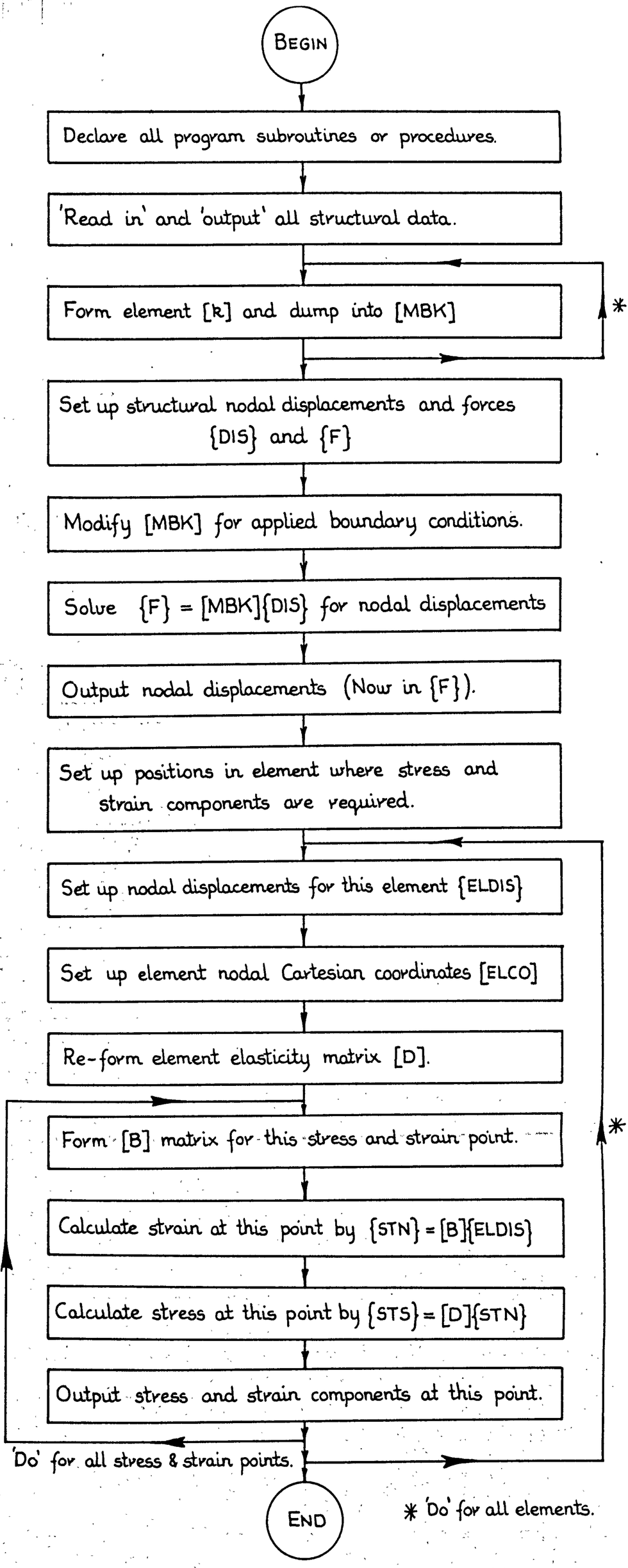
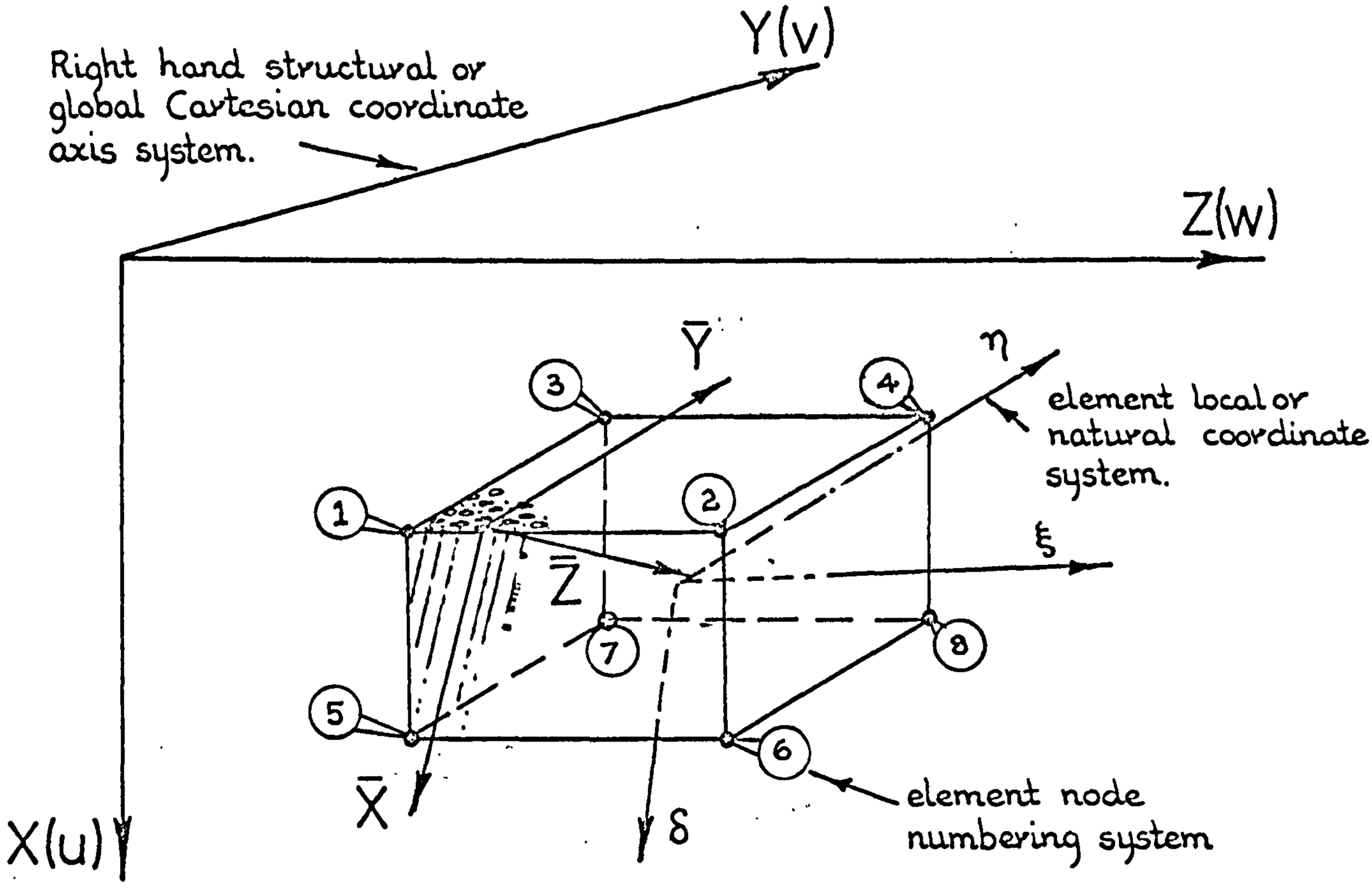
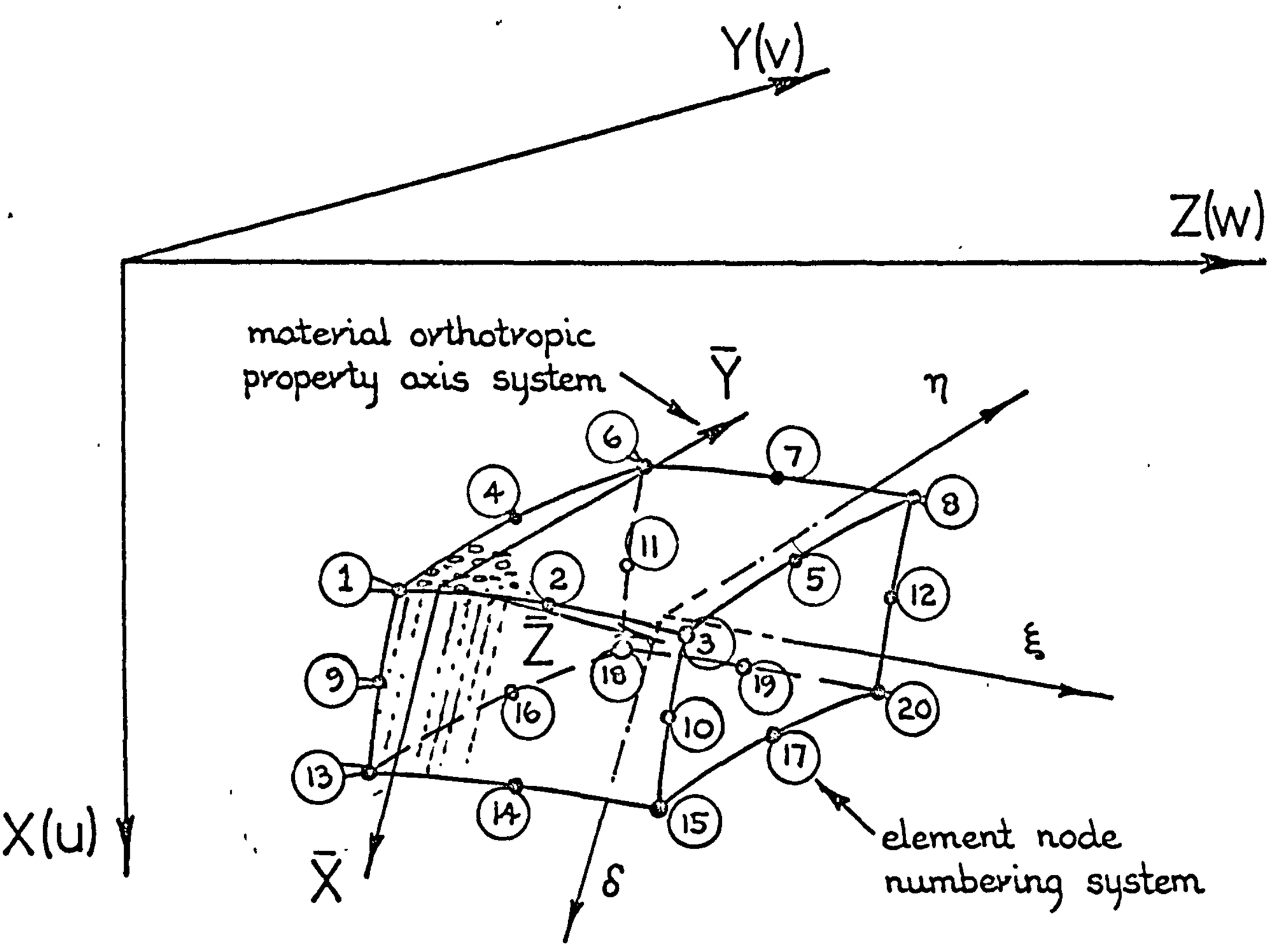


FIG. 17. 2 Flow diagram of the three-dimensional finite element analysis program.



a) 8-noded hexahedra orthotropic finite element.



b) 20-noded hexahedra orthotropic finite element.

FIG. 17.3 Orthotropic 3-D finite elements showing material and element coordinate axis systems together with the node numbering sequences.

in terms of the local coordinates and the nodal displacement values by :-

$$u(\delta,\eta,\xi) = N_1u_1 + N_2u_2 + \ldots N_mu_m$$

$$v(\delta,\eta,\xi) = N_1v_1 + N_2v_2 + \ldots N_mv_m \qquad 17.1$$

$$w(\delta,\eta,\xi) = N_1w_1 + N_2w_2 + \ldots N_mw_m$$

While each of the N functions is expressed in terms of the element's local coordinates, the number of functions required is equal to the element's corresponding number of nodal points, i.e. m = 8 or 20. The N functions for both of the elements employed, and expressed in terms of the node numbering sequence system shown in FIG. 17.3, are given in FIG. 17.4.

17.3 DERIVATION OF THE 8-NODED AND 20-NODED FINITE ELEMENT STIFFNESS MATRICES

As for the two-dimensional analysis programs, both body weight and thermal loading effects were excluded. Consequently, equation 14.7 reduces down to equation 16.2.

For three dimensional stress analysis, there are six separate strain components which contribute to the internal work of the element.

$$N_1 = \frac{1}{8}(1-\delta)(1-\eta)(1-\xi) \qquad N_5 = \frac{1}{8}(1+\delta)(1-\eta)(1-\xi)$$

$$N_2 = \frac{1}{8}(1-\delta)(1-\eta)(1+\xi) \qquad N_6 = \frac{1}{8}(1+\delta)(1-\eta)(1+\xi)$$

$$N_3 = \frac{1}{8}(1-\delta)(1+\eta)(1-\xi) \qquad N_7 = \frac{1}{8}(1+\delta)(1+\eta)(1-\xi)$$

$$N_4 = \frac{1}{8}(1-\delta)(1+\eta)(1+\xi) \qquad N_8 = \frac{1}{8}(1+\delta)(1+\eta)(1+\xi)$$

a) The N functions for the 8-noded element.

$$N_1 = \frac{1}{8}(1-\delta)(1-\eta)(1-\xi)(-\delta-\eta-\xi-2) \qquad N_{11} = \frac{1}{4}(1-\delta^2)(1+\eta)(1-\xi)$$

$$N_2 = \frac{1}{4}(1-\xi^2)(1-\delta)(1-\eta) \qquad N_{12} = \frac{1}{4}(1-\delta^2)(1+\eta)(1+\xi)$$

$$N_3 = \frac{1}{8}(1-\delta)(1-\eta)(1+\xi)(-\delta-\eta+\xi-2) \qquad N_{13} = \frac{1}{8}(1+\delta)(1-\eta)(1-\xi)(\delta-\eta-\xi-2)$$

$$N_4 = \frac{1}{4}(1-\eta^2)(1-\delta)(1-\xi) \qquad N_{14} = \frac{1}{4}(1-\xi^2)(1+\delta)(1-\eta)$$

$$N_5 = \frac{1}{4}(1-\eta^2)(1-\delta)(1+\xi) \qquad N_{15} = \frac{1}{8}(1+\delta)(1-\eta)(1+\xi)(\delta-\eta+\xi-2)$$

$$N_6 = \frac{1}{8}(1-\delta)(1+\eta)(1-\xi)(-\delta+\eta-\xi-2) \qquad N_{16} = \frac{1}{4}(1-\eta^2)(1+\delta)(1-\xi)$$

$$N_7 = \frac{1}{4}(1-\xi^2)(1-\delta)(1+\eta) \qquad N_{17} = \frac{1}{4}(1-\eta^2)(1+\delta)(1+\xi)$$

$$N_8 = \frac{1}{8}(1-\delta)(1+\eta)(1+\xi)(-\delta+\eta+\xi-2) \qquad N_{18} = \frac{1}{8}(1+\delta)(1+\eta)(1-\xi)(\delta+\eta-\xi-2)$$

$$N_9 = \frac{1}{4}(1-\delta^2)(1-\eta)(1-\xi) \qquad N_{19} = \frac{1}{4}(1-\xi^2)(1+\delta)(1+\eta)$$

$$N_{10} = \frac{1}{4}(1-\delta^2)(1-\eta)(1+\xi) \qquad N_{20} = \frac{1}{8}(1+\delta)(1+\eta)(1+\xi)(\delta+\eta+\xi-2)$$

b) The N functions for the 20-noded element.

FIG. 17. 4 The N functions for the 8 and 20-noded finite elements shown in FIG. 17.3.

These are:-

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \end{Bmatrix}$$

Normal strain in X direction
" " Y "
" " Z "
Shear strain in XY plane
" " YZ "
" " ZX "

17.2

Following the argument of section 15.3, we obtain for the three-dimensional case:-

$$\begin{Bmatrix} \frac{\partial f}{\partial \delta} \\ \frac{\partial f}{\partial \eta} \\ \frac{\partial f}{\partial \xi} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial \delta} & \frac{\partial N_2}{\partial \delta} & \cdots & \frac{\partial N_m}{\partial \delta} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \cdots & \frac{\partial N_m}{\partial \eta} \\ \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \cdots & \frac{\partial N_m}{\partial \xi} \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ \cdot \\ \cdot \\ \cdot \\ q_m \end{Bmatrix}$$

or

$$\begin{Bmatrix} \frac{\partial f}{\partial \delta} \\ \frac{\partial f}{\partial \eta} \\ \frac{\partial f}{\partial \xi} \end{Bmatrix} = [\lambda] \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ \cdot \\ \cdot \\ q_m \end{Bmatrix} \qquad 17.3$$

where $[\lambda]$ is either a 3 by 8 or a 3 by 20 matrix depending upon whether m = 8 or 20.

Hence,

$$\begin{Bmatrix} \frac{\partial f}{\partial \delta} \\ \frac{\partial f}{\partial \eta} \\ \frac{\partial f}{\partial \xi} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \delta} & \frac{\partial y}{\partial \delta} & \frac{\partial z}{\partial \delta} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \end{bmatrix} \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{Bmatrix}$$

or

$$\begin{Bmatrix} \frac{\partial f}{\partial \delta} \\ \frac{\partial f}{\partial \eta} \\ \frac{\partial f}{\partial \xi} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{Bmatrix} \qquad 17.4$$

where [J] is the equivalent Jacobian transformation matrix for the three-dimensional case. Employing equation 14.6 and rearranging, we obtain:-

$$\begin{bmatrix} \frac{\partial x}{\partial \delta} & \frac{\partial y}{\partial \delta} & \frac{\partial z}{\partial \delta} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial \delta} & \frac{\partial N_2}{\partial \delta} & \cdots & \frac{\partial N_m}{\partial \delta} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \cdots & \frac{\partial N_m}{\partial \eta} \\ \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \cdots & \frac{\partial N_m}{\partial \xi} \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x_m & y_m & z_m \end{bmatrix}$$

which is equivalent to:-

$$[J] = [\lambda] \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x_m & y_m & z_m \end{bmatrix} \qquad 17.5$$

Consequently, by inverting the result of the above matrix product, $[J]^{-1}$ can be obtained for either the 8 or 20-noded finite elements. In this case

$$[J]^{-1} = \begin{bmatrix} \frac{\partial\delta}{\partial x} & \frac{\partial\eta}{\partial x} & \frac{\partial\xi}{\partial x} \\ \frac{\partial\delta}{\partial y} & \frac{\partial\eta}{\partial y} & \frac{\partial\xi}{\partial y} \\ \frac{\partial\delta}{\partial z} & \frac{\partial\eta}{\partial z} & \frac{\partial\xi}{\partial z} \end{bmatrix} \qquad 17.6a$$

which is equivalent to:-

$$[J]^{-1} = \frac{1}{\det[J]} \begin{bmatrix} \frac{\partial y\partial z}{\partial\eta\partial\xi} - \frac{\partial y\partial z}{\partial\xi\partial\eta} & \frac{\partial y\partial z}{\partial\xi\partial\delta} - \frac{\partial y\partial z}{\partial\delta\partial\xi} & \frac{\partial y\partial z}{\partial\delta\partial\eta} - \frac{\partial y\partial z}{\partial\eta\partial\delta} \\ \frac{\partial x\partial z}{\partial\xi\partial\eta} - \frac{\partial x\partial z}{\partial\eta\partial\xi} & \frac{\partial x\partial z}{\partial\delta\partial\xi} - \frac{\partial x\partial z}{\partial\xi\partial\delta} & \frac{\partial x\partial z}{\partial\eta\partial\delta} - \frac{\partial x\partial z}{\partial\delta\partial\eta} \\ \frac{\partial x\partial y}{\partial\eta\partial\xi} - \frac{\partial x\partial y}{\partial\xi\partial\eta} & \frac{\partial x\partial y}{\partial\xi\partial\delta} - \frac{\partial x\partial y}{\partial\delta\partial\xi} & \frac{\partial x\partial y}{\partial\delta\partial\eta} - \frac{\partial x\partial y}{\partial\eta\partial\delta} \end{bmatrix} \qquad 17.6b$$

Hence, on substituting equation 17.3 into equation 17.4 and rearranging we have

$$\begin{Bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{Bmatrix} = [J]^{-1}[\lambda] \begin{Bmatrix} q_1 \\ q_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ q_m \end{Bmatrix} \qquad 17.7$$

which for the three-dimensional case having u, v and w displacement components becomes

$$\begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix} = \underset{(\text{INVJLAM})}{[J]^{-1}[\lambda]} \begin{bmatrix} u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ u_m & v_m & w_m \end{bmatrix} \qquad 17.8$$

Consequently, the [B] matrix in equation 14.21 can be obtained for either the 8 or 20-noded elements from the appropriate $[J]^{-1}[\lambda]$, (i.e. INVJLAM) matrix product.

The six independent stress components of a linear elastic orthotropic material are related to the corresponding strain components shown in equation 17.2 by equation 14.22, that is:-

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} \frac{1}{Ex} & \frac{-\mu_{yx}}{Ey} & \frac{-\mu_{zx}}{Ez} & 0 & 0 & 0 \\ \frac{-\mu_{xy}}{Ex} & \frac{1}{Ey} & \frac{-\mu_{zy}}{Ez} & 0 & 0 & 0 \\ \frac{-\mu_{xz}}{Ex} & \frac{-\mu_{yz}}{Ey} & \frac{1}{Ez} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{Gxy} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{Gyz} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{Gzx} \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix}$$

Inverting the above, we obtain

$$\{\sigma\} = [M]^{-1} \{\varepsilon\}$$

where the $[M]^{-1}$ elasticity matrix is equivalent to the [D1] matrix given by equation 16.12 for the plane stress case. Consequently, the $[M]^{-1}$ matrix shown above is only applicable provided that the $\bar{X}$ $\bar{Y}$ $\bar{Z}$ orthotropic material axes coincide with the structural XYZ coordinate axes. Therefore, for the general element whose material axes are arbitrarily orientated, the appropriate transformation has to be performed. For the three-dimensional case this is carried out more easily using the Cartesian tensor approach.

Consider a position vector orientated in the XYZ Cartesian axes system to have the coordinates x_1, x_2 and x_3 with respect to the X, Y and Z coordinate directions respectively. In a new $\bar{X}$ $\bar{Y}$ $\bar{Z}$ Cartesian coordinate system

having the same origin as the old XYZ system, the coordinates of the position vector will be:-

$$\bar{x}_1 = a_{11}x_1 + a_{21}x_2 + a_{31}x_3$$

$$\bar{x}_2 = a_{12}x_1 + a_{22}x_2 + a_{32}x_3$$

$$\bar{x}_3 = a_{13}x_1 + a_{23}x_2 + a_{33}x_3$$

or in matrix form

$$\{\bar{x}\} = [t_r]\{x\} \qquad 17.9$$

where the a_{ij}s are known as the direction cosines, see FIG. 17.5. In indicial form, equation 17.9 can be written as

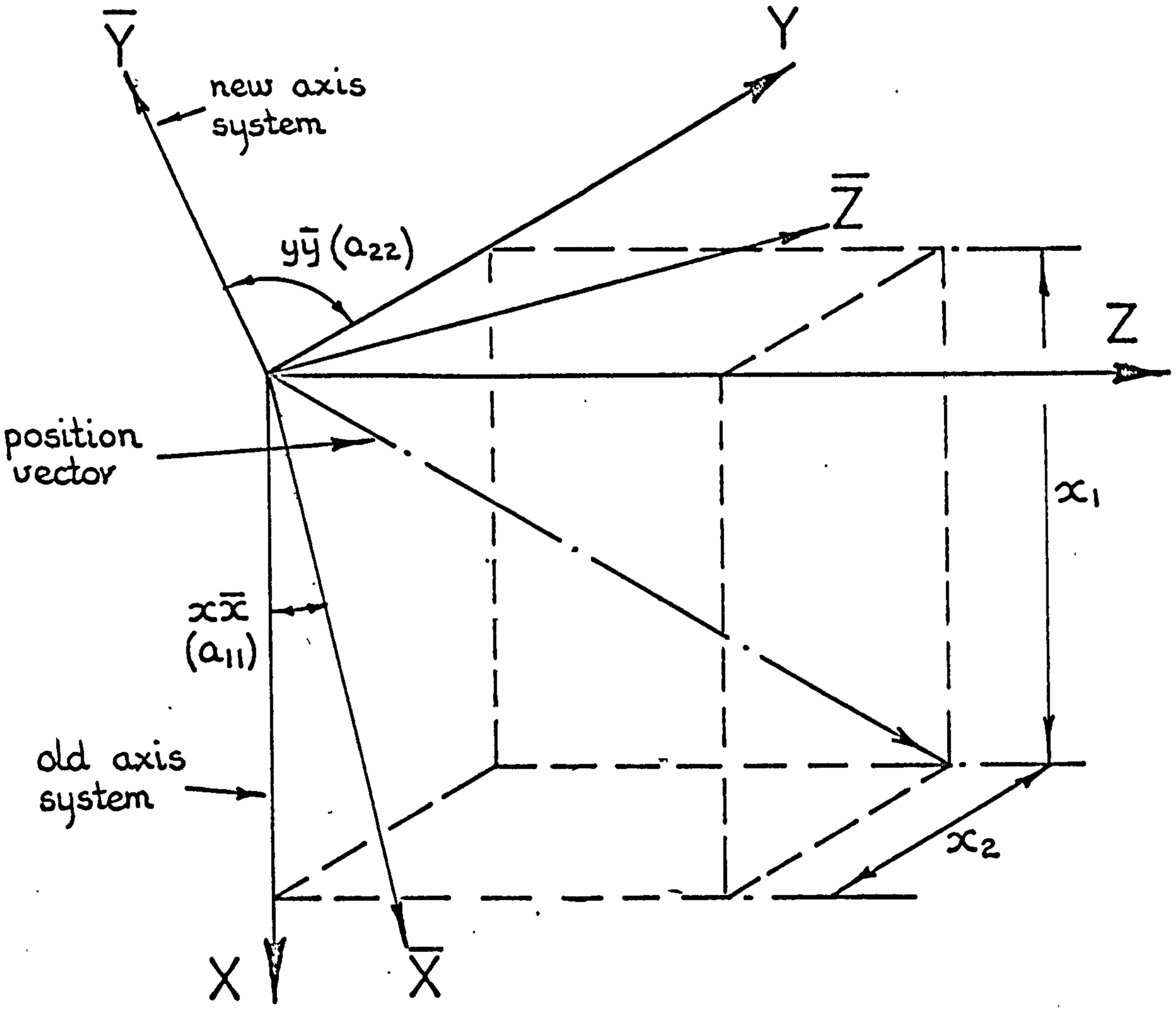
$$\bar{x}_j = a_{ij}x_i \qquad i,j = 1,2,3$$

Now if the strain components $\{\varepsilon\}$ are also known in the XYZ coordinate system, then, because strain is a second order tensor quantity, the corresponding tensorial strain components can be expressed in the $\bar{X}$ $\bar{Y}$ $\bar{Z}$ coordinate system by the equivalent indicial equation:-

$$\bar{\varepsilon}_{jl} = a_{ij}a_{kl}\varepsilon_{ik} \qquad i,j,k,l = 1,2,3$$

Therefore, on expanding the above equation and by collecting the terms, remembering that $\varepsilon_{21}=\varepsilon_{12}$ etc, we obtain the matrix equation 17.10 which is shown in FIG. 17.6. Hence, by substituting x, y and z for 1, 2 and 3 respectively and by expressing the strain components in terms of the engineering strains, i.e. $\gamma_{xy}= 2\varepsilon_{xy}$ etc, we obtain equation 17.11 shown in FIG. 17.7, i.e.

$$\{\bar{\varepsilon}\} = [TR]\{\varepsilon\} \qquad 17.11$$



$$\cos(y\bar{x}) = a_{21}$$

		NEW AXES		
		$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$
OLD AXES	x_1	a_{11}	a_{12}	a_{13}
	x_2	a_{21}	a_{22}	a_{23}
	x_3	a_{31}	a_{32}	a_{33}

FIG. 17.5 Direction cosines associated with the re-orientation of a position vector in 3-D space.

$$
\begin{Bmatrix} \bar{\epsilon}_{11} \\ \bar{\epsilon}_{22} \\ \bar{\epsilon}_{33} \\ \bar{\epsilon}_{12} \\ \bar{\epsilon}_{23} \\ \bar{\epsilon}_{31} \end{Bmatrix} =
\begin{bmatrix}
a_{11}a_{11} & a_{21}a_{21} & a_{31}a_{31} & 2a_{11}a_{21} & 2a_{21}a_{31} & 2a_{11}a_{31} \\
a_{12}a_{12} & a_{22}a_{22} & a_{32}a_{32} & 2a_{12}a_{22} & 2a_{22}a_{32} & 2a_{32}a_{12} \\
a_{13}a_{13} & a_{23}a_{23} & a_{33}a_{33} & 2a_{13}a_{23} & 2a_{23}a_{33} & 2a_{33}a_{13} \\
a_{11}a_{12} & a_{21}a_{22} & a_{31}a_{32} & (a_{11}a_{22}+a_{21}a_{12}) & (a_{21}a_{32}+a_{31}a_{22}) & (a_{31}a_{12}+a_{11}a_{32}) \\
a_{12}a_{13} & a_{22}a_{23} & a_{32}a_{33} & (a_{12}a_{23}+a_{22}a_{13}) & (a_{22}a_{33}+a_{32}a_{23}) & (a_{32}a_{13}+a_{12}a_{33}) \\
a_{13}a_{11} & a_{23}a_{21} & a_{33}a_{31} & (a_{13}a_{21}+a_{23}a_{11}) & (a_{23}a_{31}+a_{33}a_{21}) & (a_{33}a_{11}+a_{13}a_{31})
\end{bmatrix}
\begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{12} \\ \epsilon_{23} \\ \epsilon_{31} \end{Bmatrix}
$$

— 17.10

FIG. 17. 6 Strain transformation from old $\{\epsilon\}$ to new $\{\bar{\epsilon}\}$ axis systems. Strains expressed in tensor form.

$$
\begin{Bmatrix} \bar{\epsilon}_{xx} \\ \bar{\epsilon}_{yy} \\ \bar{\epsilon}_{zz} \\ \bar{\gamma}_{xy} \\ \bar{\gamma}_{yz} \\ \bar{\gamma}_{zx} \end{Bmatrix} =
\begin{bmatrix}
a_{11}a_{11} & a_{21}a_{21} & a_{31}a_{31} & a_{11}a_{21} & a_{21}a_{31} & a_{11}a_{31} \\
a_{12}a_{12} & a_{22}a_{22} & a_{32}a_{32} & a_{12}a_{22} & a_{22}a_{32} & a_{32}a_{12} \\
a_{13}a_{13} & a_{23}a_{23} & a_{33}a_{33} & a_{13}a_{23} & a_{23}a_{33} & a_{33}a_{13} \\
2a_{11}a_{12} & 2a_{21}a_{22} & 2a_{31}a_{32} & (a_{11}a_{22}+a_{21}a_{12}) & (a_{21}a_{32}+a_{31}a_{22}) & (a_{31}a_{12}+a_{11}a_{32}) \\
2a_{12}a_{13} & 2a_{22}a_{23} & 2a_{32}a_{33} & (a_{12}a_{23}+a_{22}a_{13}) & (a_{22}a_{33}+a_{32}a_{23}) & (a_{32}a_{13}+a_{12}a_{33}) \\
2a_{13}a_{11} & 2a_{23}a_{21} & 2a_{33}a_{31} & (a_{13}a_{21}+a_{23}a_{11}) & (a_{23}a_{31}+a_{33}a_{21}) & (a_{33}a_{11}+a_{13}a_{31})
\end{bmatrix}
\begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix}
$$

FIG. 17.7 Strain transformation from old $\{\epsilon\}$ to new $\{\bar{\epsilon}\}$ axis systems. Strains expressed in 'engineering' form.

Referring to the arbitrarily orientated $\bar{X}\ \bar{Y}\ \bar{Z}$ coordinate axis system, Hooke's generalized law for an elastic orthotropic material can be written as

$$\{\bar{\sigma}\} = [\bar{D}]\{\bar{\varepsilon}\} \qquad 17.12$$

Similarly, for the global or structural XYZ coordinate axis system, Hooke's generalized law can be written in the form given by equation 14.23 i.e.

$$\{\sigma\} = [D]\{\varepsilon\}$$

Since the stress vectors $\{\bar{\sigma}\}$ and $\{\sigma\}$ correspond to the strain vectors $\{\bar{\varepsilon}\}$ and $\{\varepsilon\}$ respectively, for equality of the internal work of both systems

$$\{\bar{\varepsilon}\}^T\{\bar{\sigma}\} = \{\varepsilon\}^T\{\sigma\} \qquad 17.13$$

Hence, by substituting from equations 17.11, 17.12 and 14.23 we obtain

$$\{\varepsilon\}^T[TR]^T[\bar{D}][TR]\{\varepsilon\} = \{\varepsilon\}^T[D]\{\varepsilon\}$$

where $$[D] = [TR]^T[\bar{D}][TR] \qquad 17.14$$

Thus, the elasticity matrix [D] for a finite element in terms of the structural XYZ coordinate axis system can be determined from the known elasticity matrix $[\bar{D}]$ in an arbitrary $\bar{X}\ \bar{Y}\ \bar{Z}$ coordinate axis system, and the corresponding table of direction cosines.

The element stiffness matrix [k] can now be evaluated as before by using the appropriate [B] matrix for the 8 or 20-noded element and the elasticity matrix [D], i.e.

$$[k] = \int [B]^T[D][B]\, dVol$$

For the three-dimensional case

$$\int dVol = \int dxdydz$$

However, the [B] matrices have been expressed in terms of the element's local coordinate systems and sc the integrals must be evaluated with respect to these coordinate systems, i.e.

$$\int dVol = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \frac{\partial(x,y,z)}{\partial(\delta,\eta,\xi)} \, d\delta d\eta d\xi$$

where $\frac{\partial(x,y,z)}{\partial(\delta,\eta,\xi)}$ is equal to the determinant of the [J] matrix.

Thus

$$[k] = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} [B]^T[D][B]\det[J] \, d\delta d\eta d\xi \qquad 17.15$$

As before, the integral is evaluated by using the Gauss quadrature formula. Although 2 by 2 by 2 integrating points are sufficient for the simple 8-noded element, a higher order integrating rule is required for the 20-noded element, particularly, when it is in a distorted form, i.e. when it s shape departs from that of a cube. However, because the computational time required to form the element stiffness matrix is directly proportional to the number of integrating points, 3 by 3 by 3 integrating points proved to be too expensive. Consequently, the 14 point integrating rule developed by Irons (138), was employed in order to combine economy with sufficient accuracy, see Appendix One.

A further point worthy of note is the procedure BTDB which has been incorporated into the three-dimensional

analysis program. A large proportion of the computational time required to form the element stiffness matrix is taken to carry out the matrix triple product $[B]^T[D][B]$. However, because the [k] matrix is always symmetric, there is no need to compute the coefficients below the main diagonal line. Indeed, procedure BTDB carries out the matrix triple product only on those terms falling on or above the main diagonal line. Consequently, this saves almost 50 °/o of the computer time required to form the full product. Any terms which are required below the main diagonal for the subsequent block dumping process are simply reflected from the appropriate coefficients determined above the main diagonal.

17.4 FORMATION OF THE STRUCTURAL EQUILIBRIUM EQUATIONS

The direct Gauss elimination method of equation solution previously described, was again adopted for the three-dimensional finite element analysis programs. Therefore, it was necessary to form the complete modified arrangement of the structural stiffness matrix [MBK]. Of course, for the three-dimensional case, the number of degrees of freedom per node has been increased from two up to three. Consequently, the solution procedure and dump processes had to be modified from those used in the axisymmetric and two-dimensional programs in order to handle the equilibrium equations in a 3 by 3 submatrix block form. Nevertheless, apart from this, the formation

of the structural equilibrium equations followed exactly the same process as that previously described for the axisymmetric program.

It must be emphasized that the formation and dumping processes incorporated into the 8 and 20-noded finite element analysis programs, determine the structural node numbering system that can be employed. In fact, the dumping process requires the node numbers of each of the elements in the structure to progressively increase when the element node numbering sequence depicted in FIG. 17.3 is implemented. However, this sequence can generally be achieved if the structure is numbered in the Z, Y and X coordinate directions respectively, i.e. the node numbering sequence of the structure follows the same pattern as the node numbering sequence of the elements, see FIG. 17.3. Although this node numbering sequence may not be attainable for hollow ring type structures, the reason for adopting this restriction was simply for one of economy. This method eliminated the need for any 'IF' statements in the programs. These would otherwise be required in order to check whether one node number was greater than another so that the appropriate terms of the element stiffness matrix could be dumped.

17.5 APPLICATION OF THE STRUCTURAL BOUNDARY CONDITIONS

Three types of boundary conditions were required for the three-dimensional finite element analysis programs. In addition to the kinematic constraint and boundary point loading conditions previously described for the axisymmetric and two-dimensional programs, it was required to incorporate the boundary type surface loading condition as well.

The nodal constraints of the three-dimensional finite element models were implemented in the programs in a similar manner to those of the axisymmetric and two-dimensional cases. The only difference being in the way in which the data for the nodes, which have one or more of their three degrees of freedom constrained, is read in. The data simply consists of the node number followed by either a 1 or a zero for each of the X, Y and Z coordinate axis directions. Constrained degrees of freedom are indicated by a 1 while unconstrained degrees of freedom are indicated by a zero. The program searches for the 1s and subsequently,assigns the appropriate degree of freedom in the structural displacement vector, the value of 0.000001. As before, this value is used as a pointer which enables the program to carry out the necessary modifications required to the [MBK] matrix.

The nodes where point forces are to be applied are handled in a similar manner. The node numbers are simply

read in as before but in this case they are followed by the actual values of the point loads applied to the structure. Thus, the node number is followed by three entries which correspond to the X, Y and Z coordinate directions respectively. A zero entry is used to signify a zero load component. The actual non-zero values are simply added into the appropriate location of the structural nodal force vector $\{F\}$.

As discussed in 14.6.1, it is a relatively easy matter to apportion the distributed boundary loading applied to the surfaces of the simple types of finite element. This is achieved by dividing the total load applied to the element surface equally between that surface's nodes. While this is also true for the 8-noded three-dimensional element, equation 14.18 must be employed for the 20-noded element if an 'equivalent' or 'consistent' system of nodal point loads is to be derived.

Consider the 20-noded finite element shown in FIG. 17.8 to be loaded with a uniform pressure of g per unit area over the surface corresponding to $\delta=-1$. Using equation 14.18 we have:-

$$\{F_b\} = \int [N]^T g \, dA$$

However, as can be seen from FIG. 17.8, the surface of the element for $\delta=-1$ does not lie parallel to the YZ plane. Consequently, the uniform normal surface loading will produce an equivalent set of X, Y and Z nodal forces.

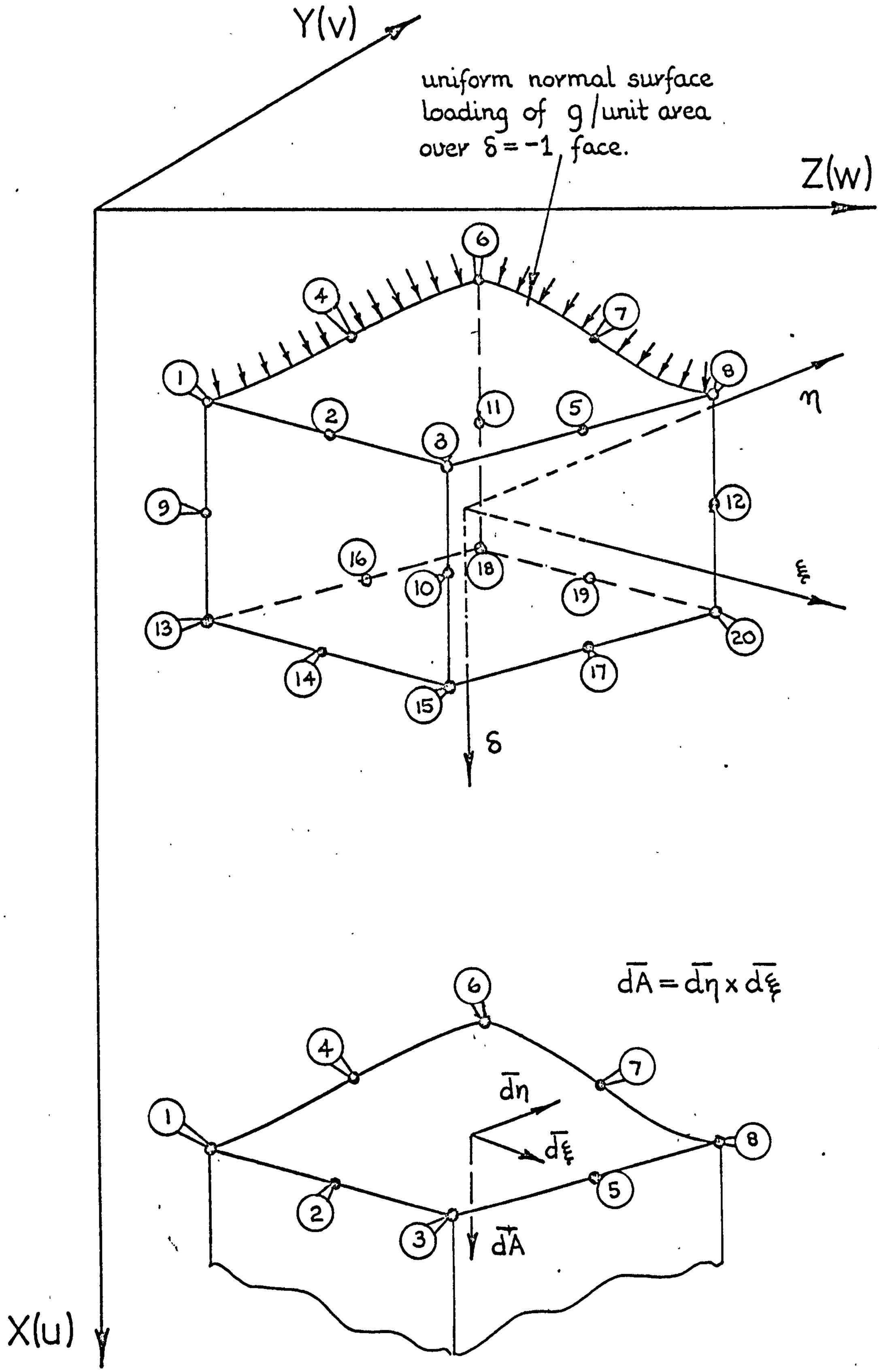


FIG. 17. 8 Uniform normal surface loading applied to the 20-noded finite element on the $\delta = -1$ face.

Therefore, before these forces can be evaluated, the respective X, Y and Z projected or vector areas associated with the surface must be determined. Again, consider FIG. 17.8 and let the area of the surface be represented by the vector $\bar{\bar{dA}}$, acting in a direction normal to the surface. It can be seen that the base vectors $\bar{\bar{d\eta}}$ and $\bar{\bar{d\xi}}$ will lie on the surface as shown and so the vector area $\bar{\bar{dA}}$ will be given by the cross product of the two base vectors:-

i.e. $$\bar{\bar{dA}} = \bar{\bar{d\eta}} \times \bar{\bar{d\xi}}$$

or expressed in component form

$$\bar{\bar{dA}} = \begin{Bmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \\ \frac{\partial z}{\partial \eta} \end{Bmatrix} \times \begin{Bmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \xi} \\ \frac{\partial z}{\partial \xi} \end{Bmatrix} d\eta d\xi$$

which is equal to

$$\bar{\bar{dA}} = \begin{Bmatrix} \frac{\partial y \partial z}{\partial \eta \partial \xi} - \frac{\partial z \partial y}{\partial \eta \partial \xi} \\ \frac{\partial z \partial x}{\partial \eta \partial \xi} - \frac{\partial x \partial z}{\partial \eta \partial \xi} \\ \frac{\partial x \partial y}{\partial \eta \partial \xi} - \frac{\partial y \partial x}{\partial \eta \partial \xi} \end{Bmatrix} d\eta d\xi \qquad 17.16$$

Comparing equation 17.16 with equation 17.6b, it can be seen that the columns of the $[J]^{-1}$ matrix, which are respectively the δ, η an ξ contravariant base vectors, represent vector areas. Thus, the vector area of the $\delta = -1$ face can be alternatively expressed by

$\bar{\bar{dA}}$ = det[J].(δ contravariant base vector). $d\eta d\xi$

Therefore, the equivalent system of nodal forces for the 20-noded element loaded by a uniform pressure on the $\delta=-1$ face as illustrated in FIG. 17.8 becomes:-

$$\begin{Bmatrix} F_{1x} \\ F_{1y} \\ F_{1z} \\ F_{2x} \\ F_{2y} \\ F_{2z} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ F_{8x} \\ F_{8y} \\ F_{8z} \end{Bmatrix} = \int \begin{bmatrix} N_1 & 0 & 0 \\ 0 & N_1 & 0 \\ 0 & 0 & N_1 \\ N_2 & 0 & 0 \\ 0 & N_2 & 0 \\ 0 & 0 & N_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ N_8 & 0 & 0 \\ 0 & N_8 & 0 \\ 0 & 0 & N_8 \end{bmatrix} \begin{Bmatrix} g \det[J] \text{ INVERSEJ } [1,1] \\ g \det[J] \text{ INVERSEJ } [2,1] \\ g \det[J] \text{ INVERSEJ } [3,1] \end{Bmatrix} d\eta d\xi \qquad 17.17$$

This can be evaluated using numerical integration over the face for $\delta=-1$, see Appendix One. If g was not uniform over the surface of the element, then the variation would, of course, also have to be considered in the above.

For the finite element analysis programs discussed here, the equivalent system of nodal forces for a particular element subjected to a surface loading, was determined separately outside the main analysis programs. Subsequently, once the equivalent nodal point forces had been determined, they were applied to the appropriate nodes of the structure at the time of analysis in exactly the same manner as were the actual boundary point loads.

17.6 SOLUTION OF THE STRUCTURAL EQUILIBRIUM EQUATIONS

As previously mentioned, the Gauss elimination method in 3 by 3 sub-matrix block form, was employed to solve the structural equilibrium equations for the three-dimensional finite element analysis programs. The procedure is called DBLOKGAUSS in the computer listing of the 20-noded program given in Appendix Seven.

17.7 DETERMINATION OF THE ELEMENT STRAIN AND STRESS COMPONENTS

The six strain components given by equation 17.2 together with the corresponding stresses, were determined for each of the 8-noded elements at the elements' centroid, i.e. at $\delta=\eta=\xi=0$. Thus, after evaluating the [B] matrix at these local coordinate values, the strain vector for each of the elements was obtained by equation 14.21. Subsequently, after the appropriate elasticity matrix was evaluated with respect to the global coordinate directions, the corresponding stress components were determined by applying equation 14.23.

For the 20-noded elements, exactly the same process was adopted as described above. However, in this case the strain and stress components were evaluated at fifteen separate positions in each element. The program numbering sequence of these positions is clearly shown in FIG. 17.9.

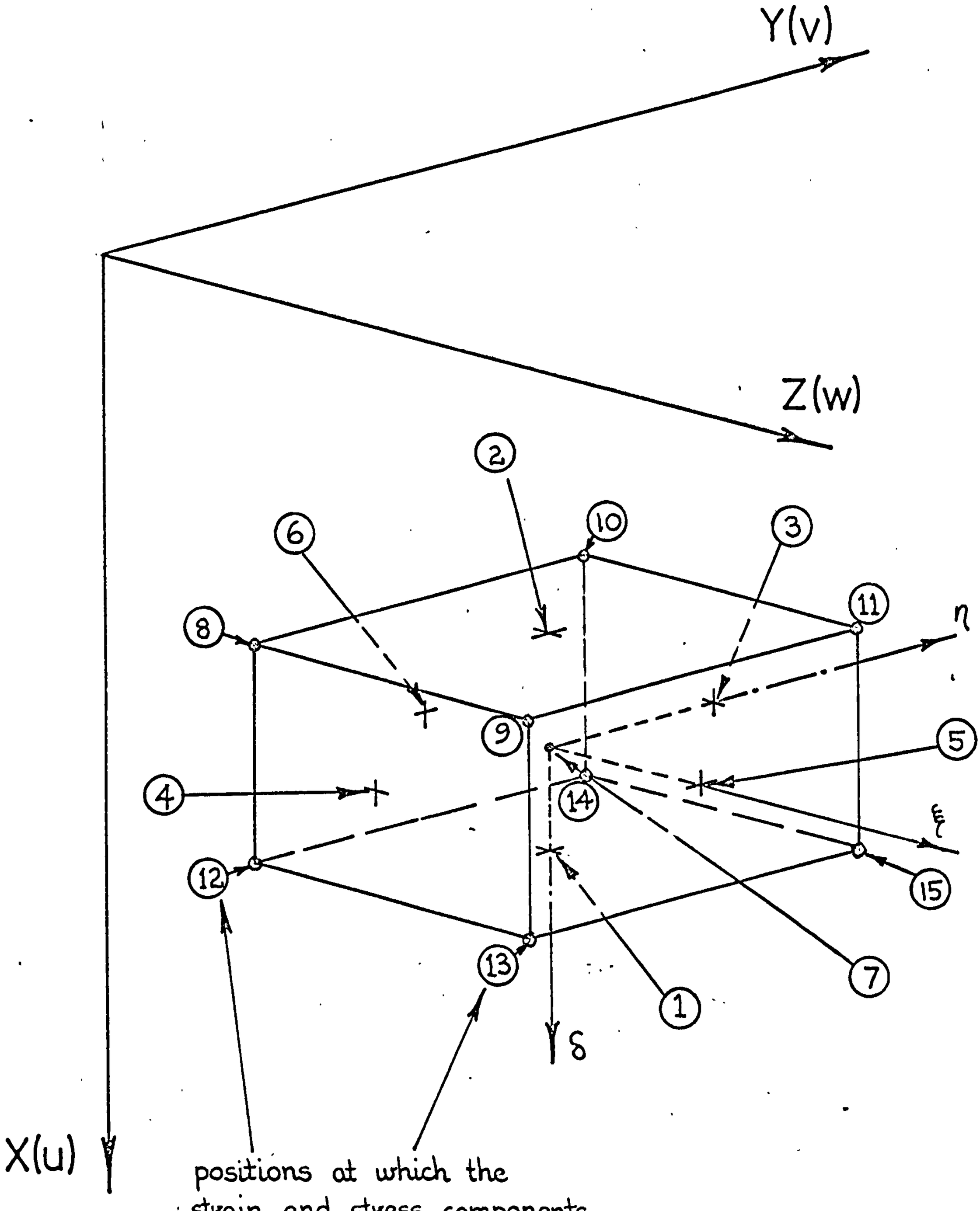


FIG. 17. 9 The program numbering sequence of the positions at which the strain and stress components are evaluated.

17.8 THREE-DIMENSIONAL DATA CHECK

As for the axisymmetric and two-dimensional programs,the three-dimensional data card packs were checked using the same procedure discussed in 14.11. However, because of the obvious difficulty of displaying a three-dimensional object on a two-dimensional drawing, the three-dimensional finite element model structures were drawn on the plotter in isometric type projection, see FIG. 9.5b. This was achieved by simply selecting the projection required and determining the corresponding nine direction cosines between the XYZ Cartesian global axis system of the structure and the X', Y', Z' Cartesian coordinate system of the graph plotter, see FIG. 17.10. (As can be seen from the figure, the Z' axis of the plotter is taken as being normal to the plotter paper.) Hence, by using the coordinate transformation matrix given by equation 17.9, each of the structural nodal coordinates was transformed to the plotter's 'new' coordinate system. The transformed nodal coordinates (multiplied by some suitable scaling factor), were then plotted by assigning the transformed X structural nodal coordinates to the plotter's X' axis and the transformed Z structural coordinates to the plotter's Y' coordinate axis. Node and/or element numbering was subsequently carried out as before, see again FIG. 9.5b.

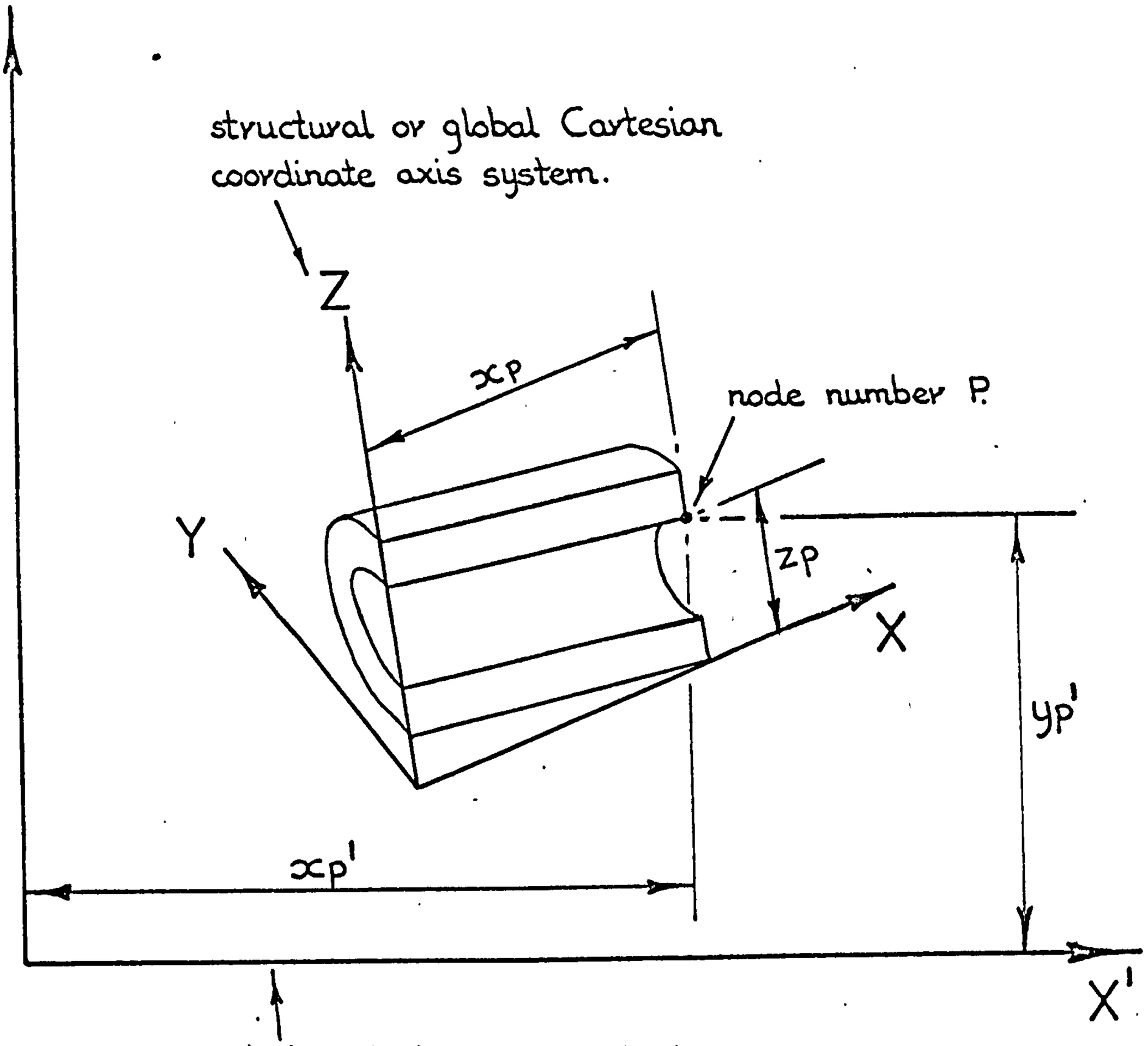


FIG. 17. 10 Method of displaying 3-D structures using the computer plotting facility.

The flow diagram for the three-dimensional data plotting program is given in FIG. 17.11.

17.9 THREE-DIMENSIONAL PROGRAM TEST PROBLEMS

There is a dearth of published results on genuine three-dimensional structures which were suitable for testing the three-dimensional finite element analysis programs. Indeed,no results at all could be traced in which orthotropic materials were involved. Consequently, the working of the orthotropic material feature of the programs could not be examined. Consequently, various simple structures such as an end loaded cantilever and a tensile test specimen involving isotropic materials, were analysed using both the 8 and 20-noded analysis programs. The deflections, stresses and strains obtained were in all cases in very close agreement with the generally accepted values derived by using conventional analytical techniques. As a more exacting test, the Bousinesq structure which was examined using the axisymmetric program, see 15.9 was also analysed. However, because the results for this problem, obtained by using both the 8 and 20-noded finite elements and employing the conjugate gradient method of equation solution, are included in Appendix Two, the results will not be discussed here. Instead, another formidable problem for which results were available will be presented.

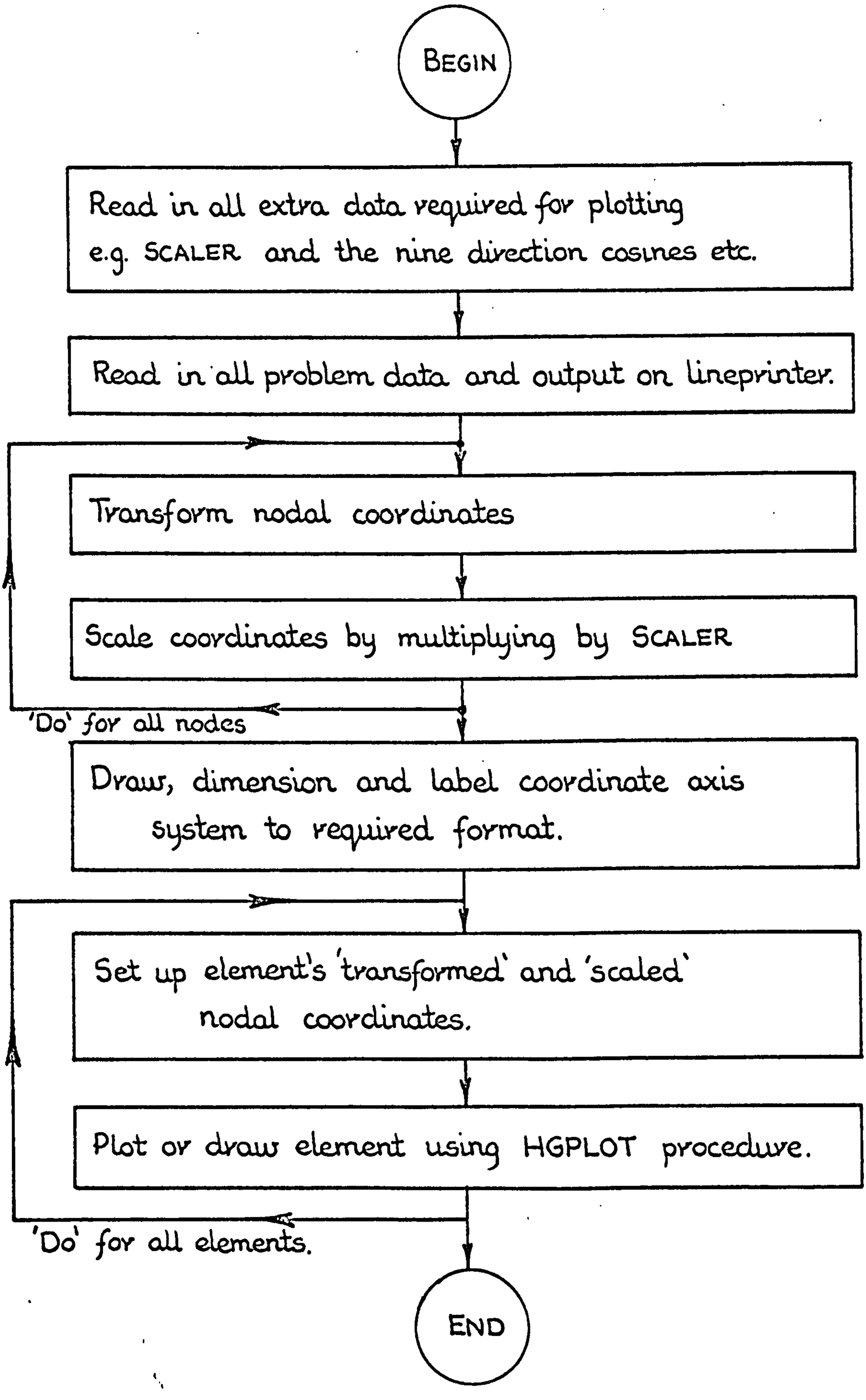


FIG. 17. 11 Flow diagram of the 3-D finite element plot program. See Appendix Eight.

The problem consisted of a 24 inch square by 6 inches thick plate, simply supported around the perimeter and loaded centrally with a single vertical point load. Hence, because of the double symmetry (like that of the Bousinesq structure), only one quarter of the plate needed to be modelled. The structure was modelled by using eight 20-noded elements as shown in FIG. 17.12. (Note that the structural node numbering sequence follows the required ZYX pattern.) FIG. 17.13 shows a comparision between the displacements of the plate obtained by using the program and those obtained by McKinnon (139) and by Zienkiewicz (140). As can be seen, the agreement between the sets of results is very good. Of course, the displacement, strain and stress values obtained from the finite element analysis program, were identical to the results supplied by Zienkiewicz down to the sixth decimal place. This is haraly surprising however, as Zienkiewicz used exactly the same finite element idealisation as employed here.

As before, it was possible to derive a formula by which the computational time required to analyse a specific structure could be estimated. For the 20-noded finite element program employing the fourteen point integration rule this was found to be:-

$$\text{Time(secs)} = 184 \times \text{NELEM} + 0.014([\text{MANND}+1]^2 \times \text{NONOP}) \qquad 17.18$$

N.B. In the above NELEM should be the number of different element [k]s which have to be derived.

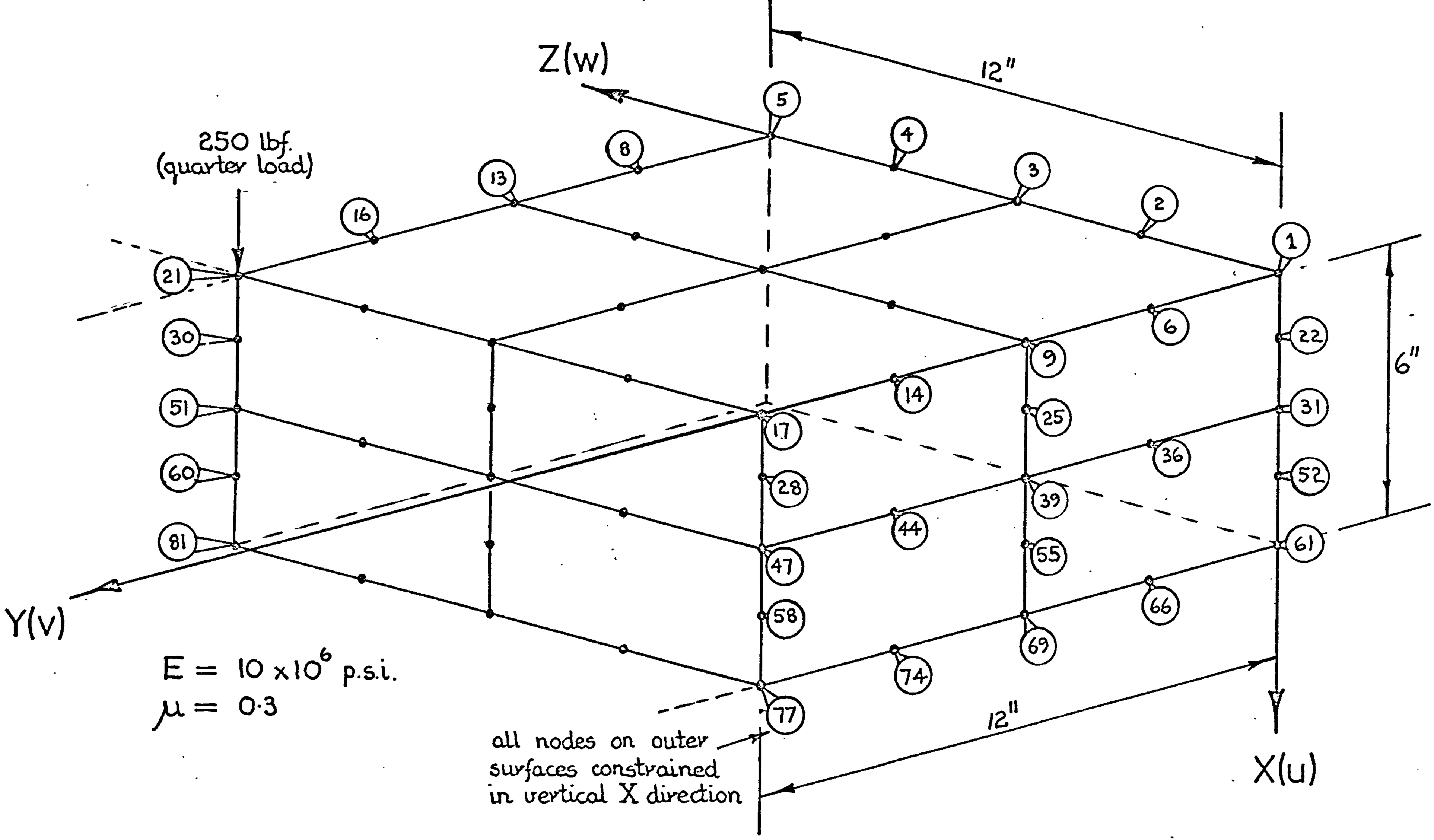


FIG. 17. 12 Finite element representation of the thick plate problem. Model comprised of eight 20-noded elements.

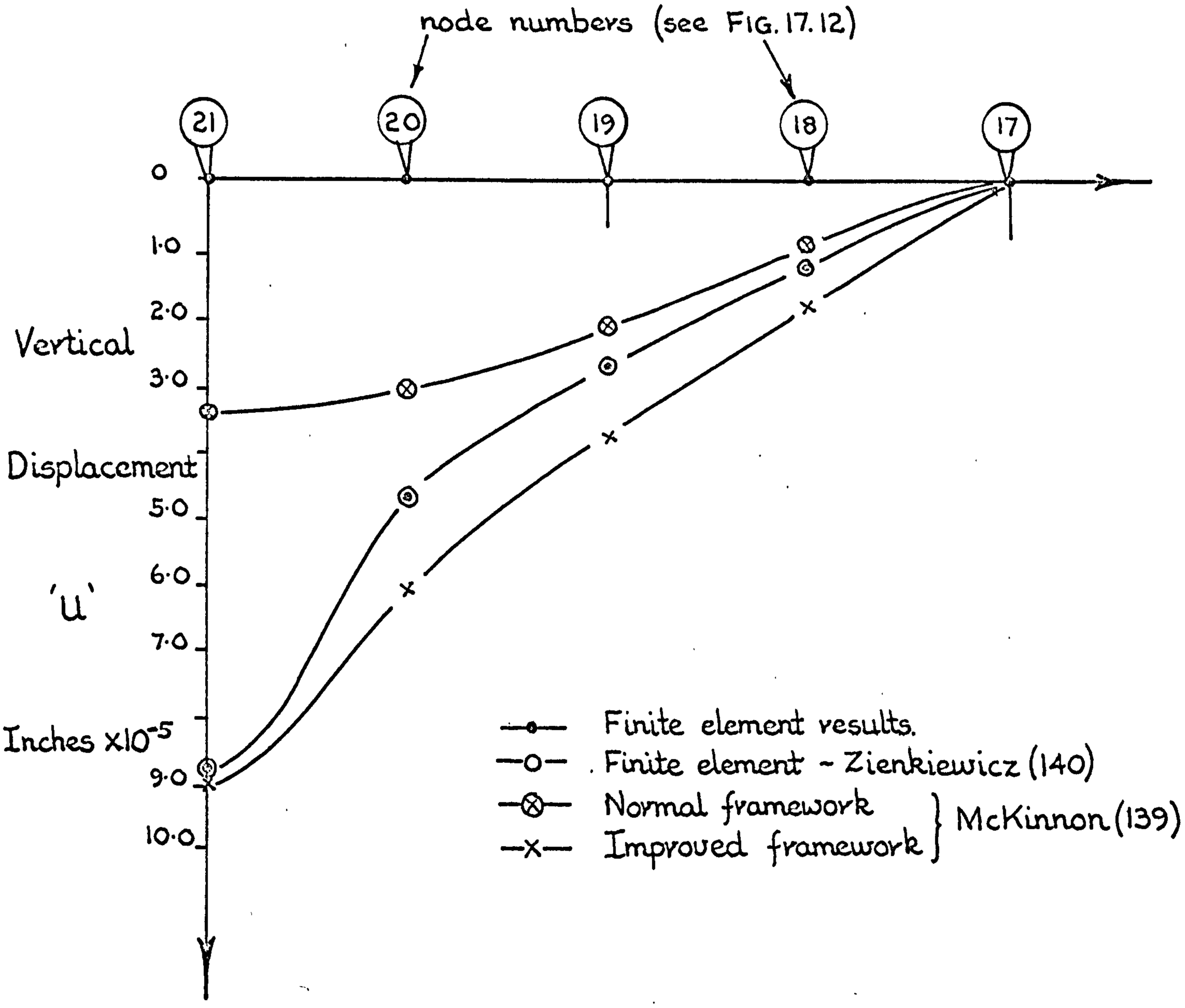


FIG 17 13 Comparison of various vertical displacement results obtained for the thick plate problem shown in FIG. 17.12

CHAPTER EIGHTEEN

COMPARISON OF FINITE ELEMENT MODELS AND GENERAL CONCLUSIONS

18 COMPARISON OF FINITE ELEMENT MODELS AND GENERAL CONCLUSIONS

Many structures, and in particular many of the dental structures met with in this project, are three-dimensional in nature. However, full three-dimensional finite element analyses are very expensive in terms of the computational time required. Indeed, the computer installation employed for the work reported herein was in fact unsuitable for three-dimensional analysis due to the small amount of core storage space available. Consequently, it was found necessary to examine some of the dental structural problems by employing the cheaper less-demanding axisymmetric and two-dimensional finite element model simulations. Consequently, the aim here was to examine a simple dental type structure using various finite element simulations in order to ascertain the limitations of these simplified models.

18.1 FINITE ELEMENT MODELS AND TEST PROCEDURE

The structure employed for this comparative study was the shoulderless mesiooclusodistal (M.O.D.) inlay configuration which was examined in section 5.5 in Volume One. The structure, which consisted of a photoelastic resin superstructure supported on an aluminium base, was represented by various finite element model simulations, the details of which are given in

FIG. 18.1. As indicated by the figure, the models designed were such that the grade of the element meshes produced were very comparable. It must be realised however, that for the axisymmetric model and for the three-dimensional model proper, the M.O.D. restoration reverted to that of a full crown type restoration.

18.2 RESULTS

Both the photoelastic restoration and the aluminium base materials were assumed to be isotropic. The mechanical properties employed were:-

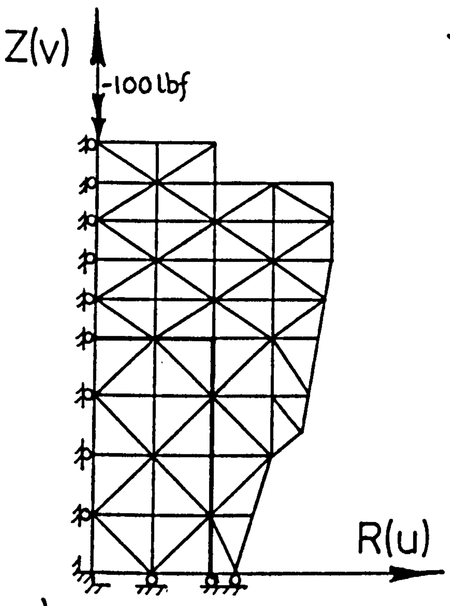
Photoelastic resin $E = 6.15 \times 10^5$ p.s.i., $\mu = 0.365$

Aluminium $E = 10.5 \times 10^6$ p.s.i., $\mu = 0.33$

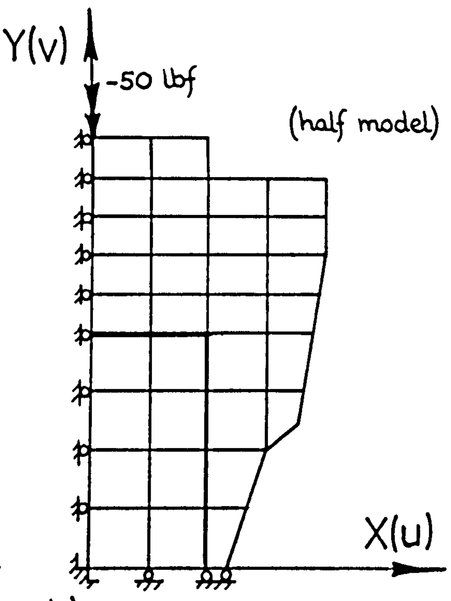
A single point load, equivalent to a 100 lbf load for the whole structure, was applied centrally to the finite element models, as shown in FIG. 18.1. The models, in addition to being constrained on the appropriate planes or axes of symmetry, were all constrained distally in the vertical direction. The vertical displacement and stress distributions obtained for the models at both a high and a low level are given in FIGS. 18.2 and 18.3 respectively.

18.3 DISCUSSION OF RESULTS

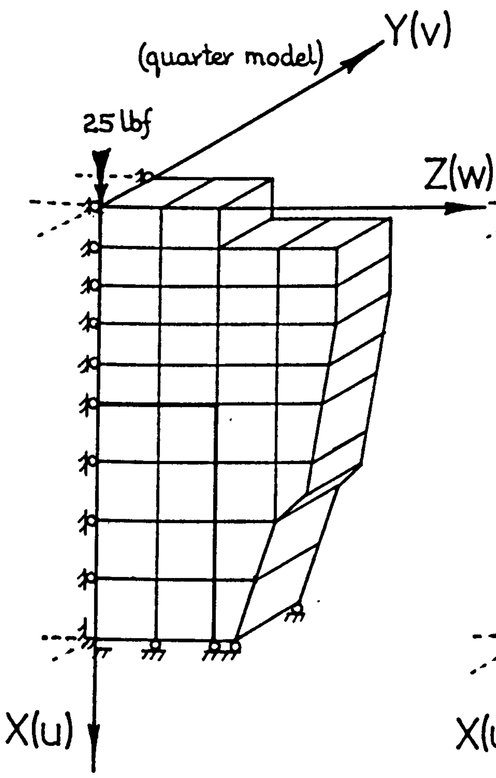
As expected, there were large differences between the magnitudes of the results obtained using the various finite element models. However, apart from the displacement distributions for the pseudo and the three-dimensional models, the displacement and stress



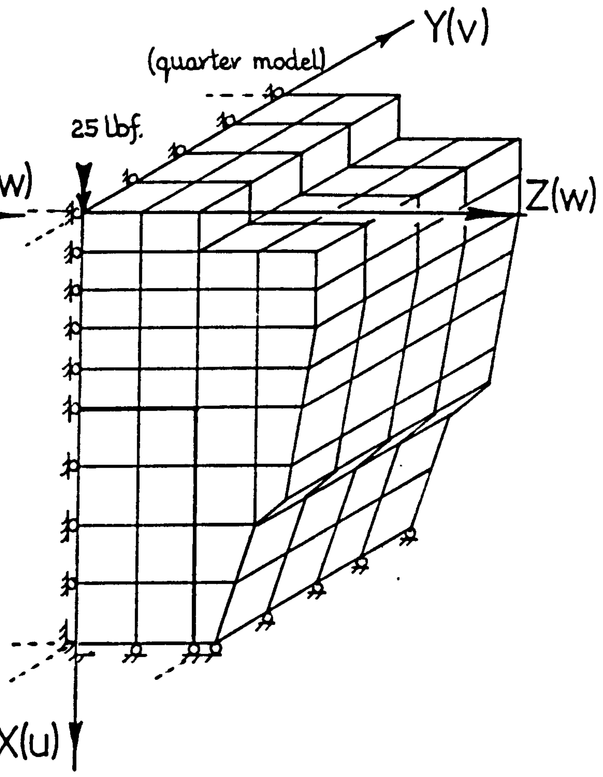
a) Axisymmetric model
64, 3-noded elements



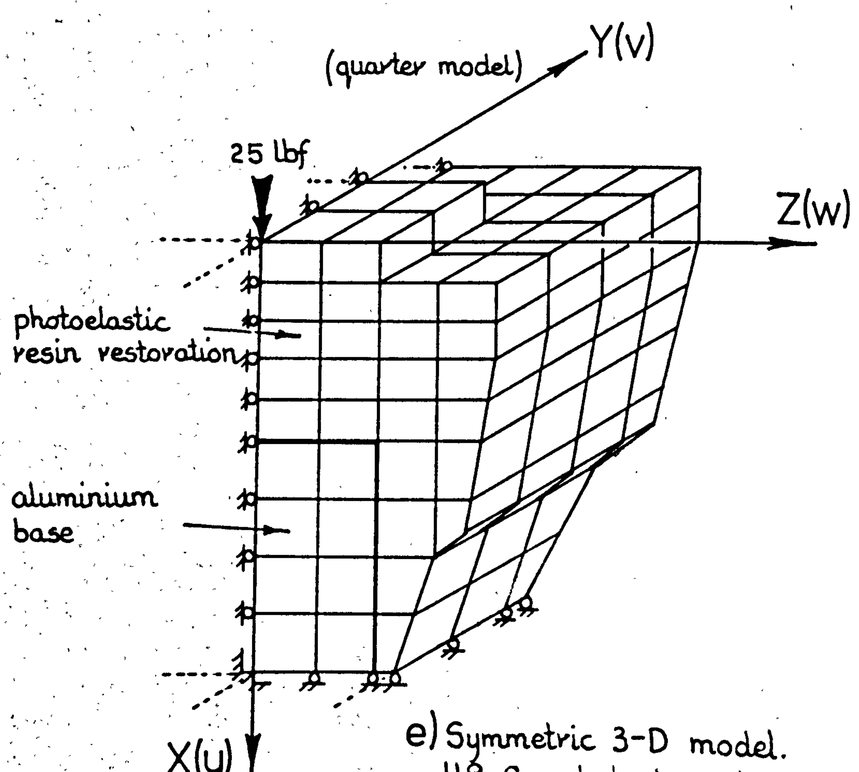
b) Plane stress/strain 2-D model.
32, 4-noded elements 0·375" thick



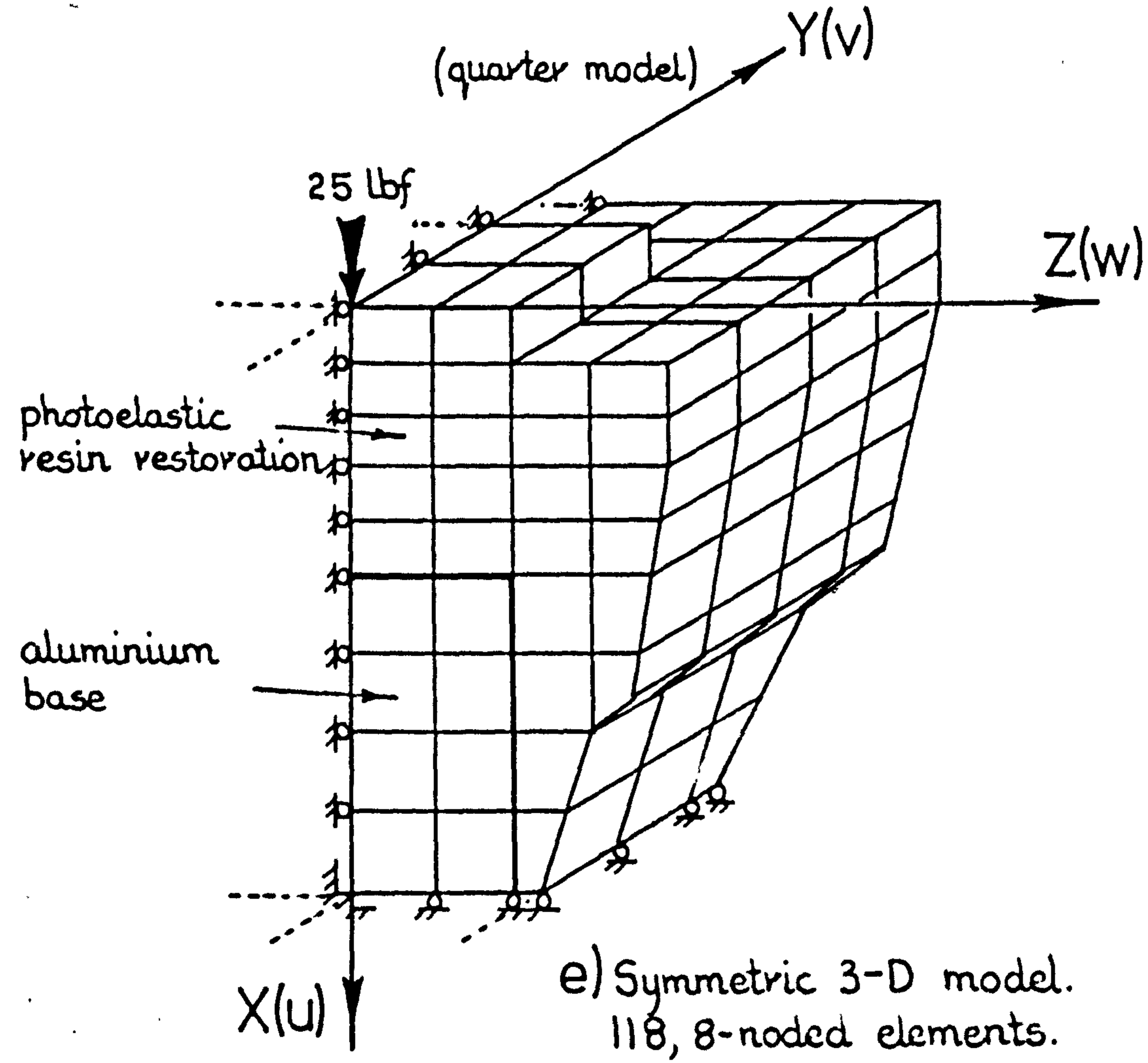
c) Pseudo 3-D model
32, 8-noded elements.
0·375" thick.



d) Prismatic 3-D model
128, 8-noded elements



e) Symmetric 3-D model.



e) Symmetric 3-D model.
118, 8-noded elements.

FIG. 18. 1 Various comparative finite element models of an M.O.D. type restoration, see Chapter Five.

—•— Planestress

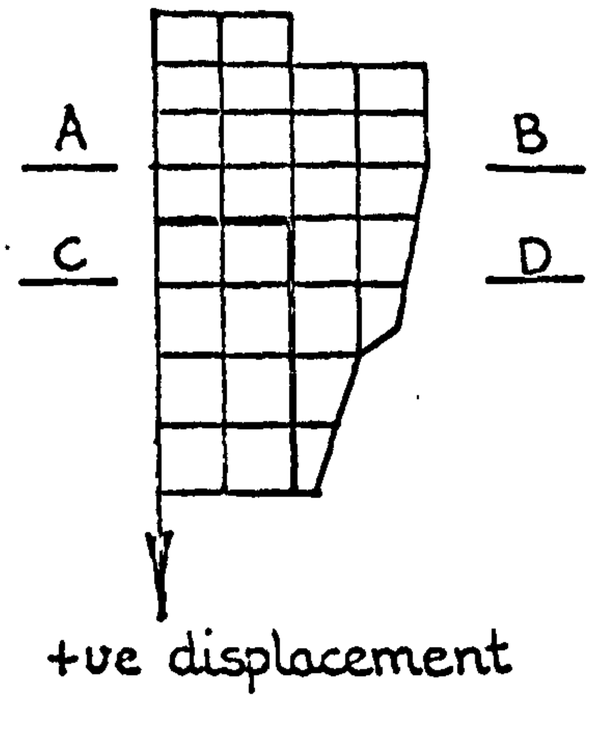
o Planestrain

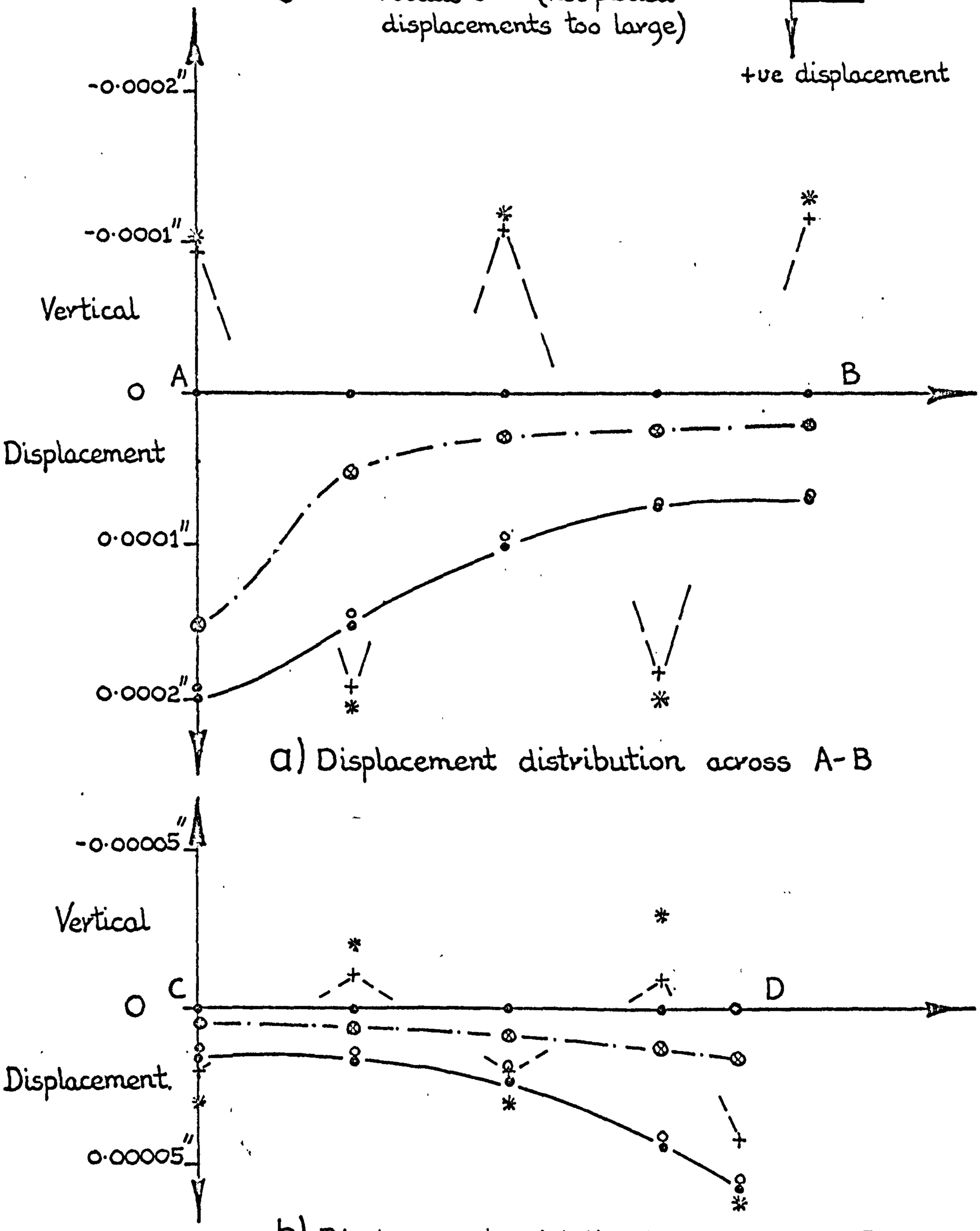
—⊗— Axisymmetric

\+ 3-D prismatic

* 3-D

◐ Pseudo 3-D (not plotted displacements too large)





a) Displacement distribution across A-B

b) Displacement distribution across C-D

FIG. 18.2 Comparison of the vertical displacement distributions obtained for the various models shown in FIG. 18.1.

—•— Planestress

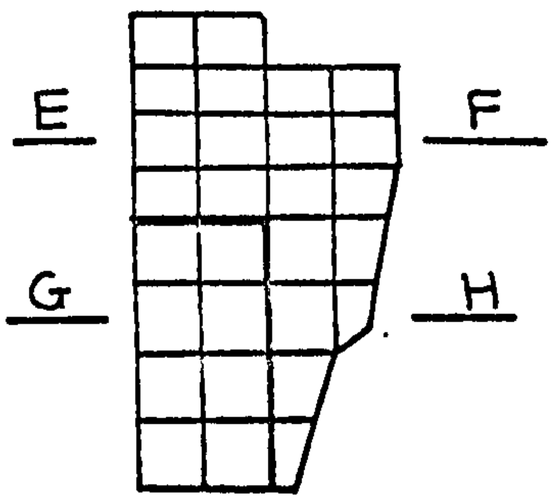
o Planestrain

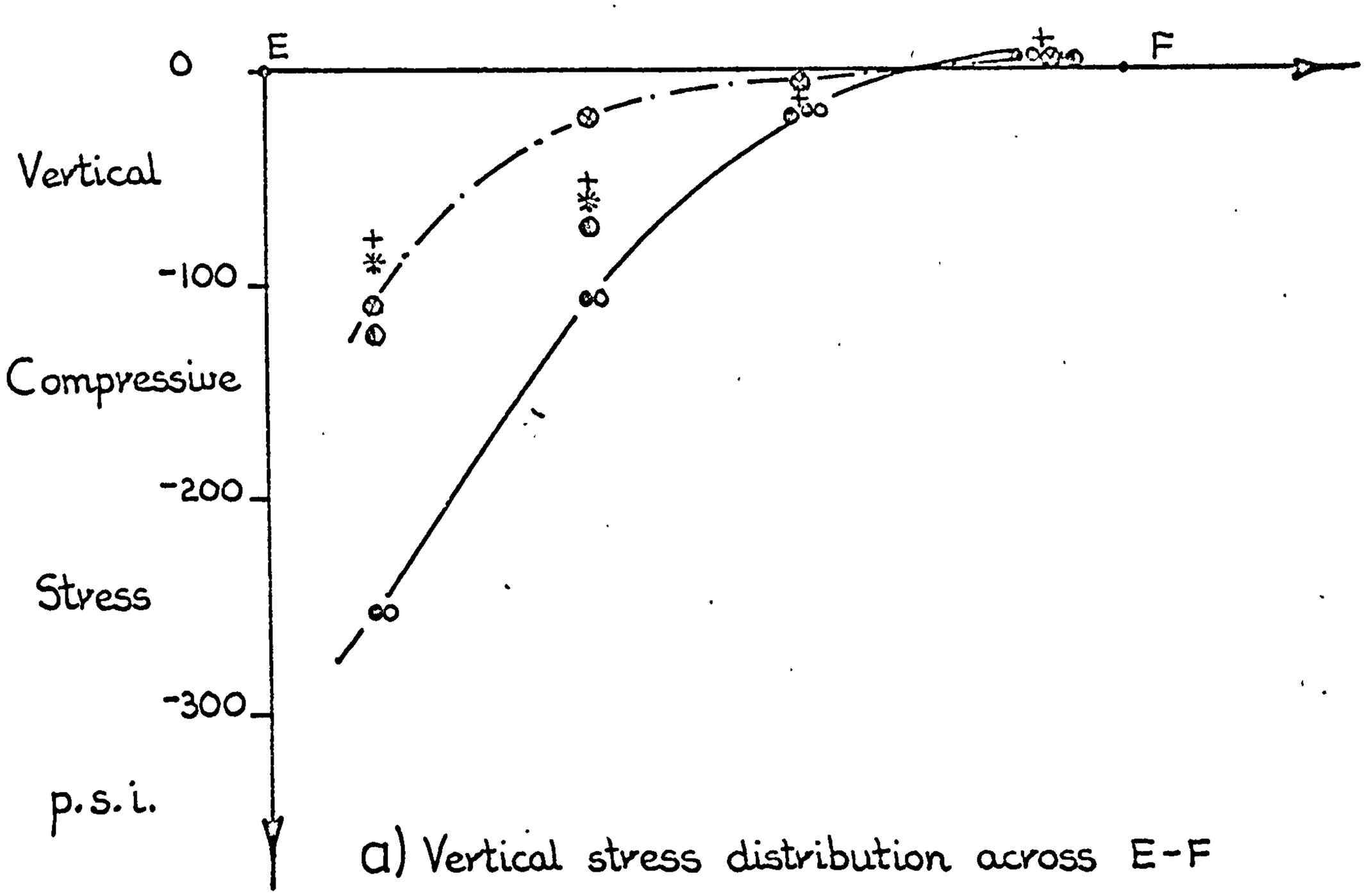
—⊗— Axisymmetric

\+ 3-D prismatic

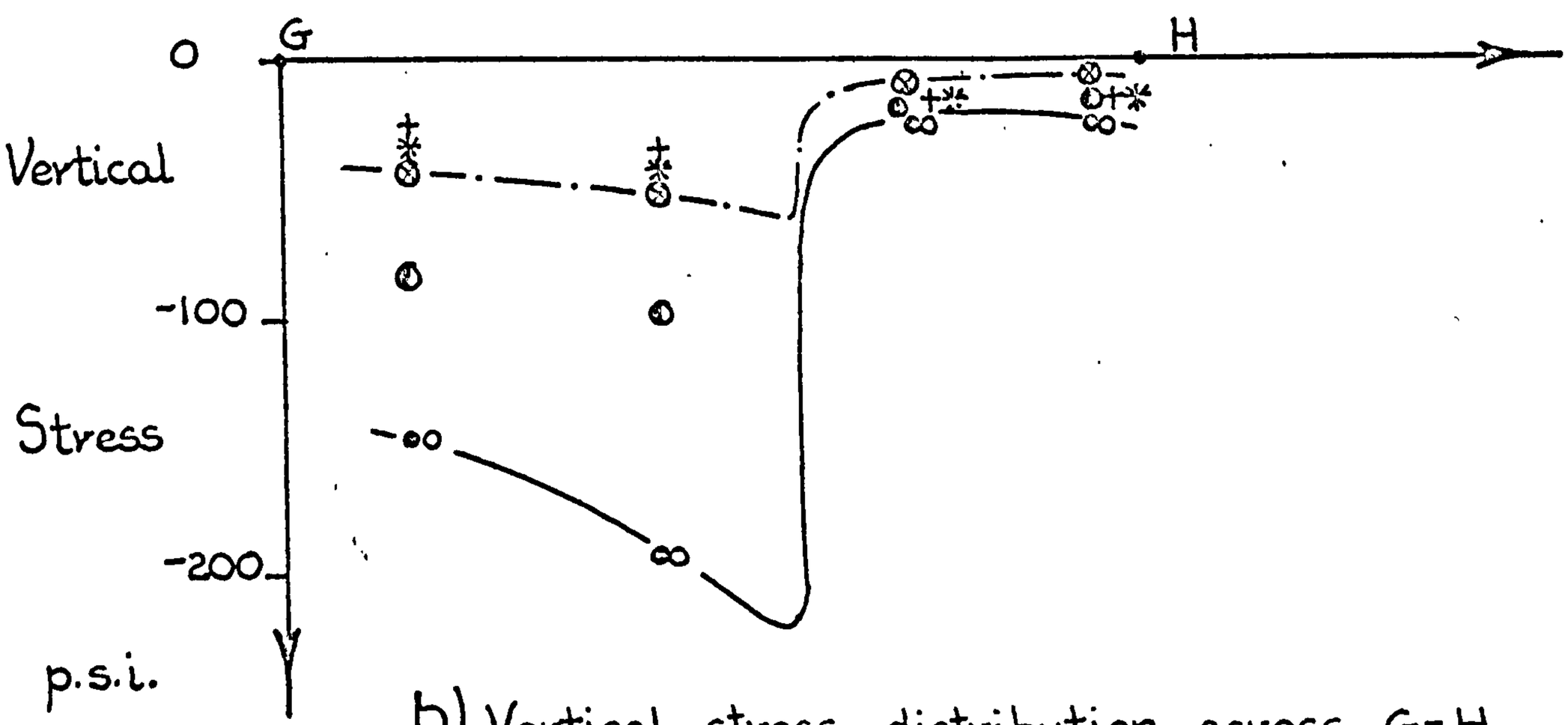
✳ 3-D

◉ Pseudo 3-D





a) Vertical stress distribution across E-F



b) Vertical stress distribution across G-H.

FIG. 18. 3 Comparison of the vertical stress distributions obtained for the various models shown in FIG. 18.1.

distribution patterns obtained were very similar in form. Of course, because the magnitudesof both the displacements and the stresses are directly prr portional to the thickness of the two-dimensional models, one would expect a considerable variation between the values obtained from these models and those obtained from an equivalent three-dimensional model. For this reason, the two-dimensional slice type model using three-dimensional elements seemed to offer an interesting compromise between two and full three-dimensional analyses. However, as for the plane stress and plane strain cases, the displacement and stress components perpendicular to the plane of the model are not effectively interpreted. Consequently, a compromise between the single layer and prismatic three-dimensional models might be a more promising approach.

Although a certain amount of "smoothing" of finite element results has to be generally undertaken, the erratic displacement distributions obtained for the pseudo and full three-dimensional simulations using the 8-noded finite elements were very surprising. Indeed, it is very difficult to "smooth" the values obtained into a reasonably accurate and useful form. Even so the stress distributions determined were smooth and seemed to be reasonably correct. It is worthy to note however, that the displacement patterns obtained for structures using

the 20-noded elements have always been smooth.

18.4 GENERAL CONCLUSIONS

The finite element method of stress analysis is a versatile method which allows structures of intricate shape and form, and having complex material behaviour, to be analysed. Indeed, the method can provide solutions to problems which have hitherto been intractable. However, for the most efficient implementation of the method, a large sophisticated computer complex is required. In terms of economics full three-dimensional analyses are orders of magnitude more costly to perform than are equivalent axisymmetric and two-dimensional analyses, (FIG. 18.4 gives a comparison of the cost of the various types of analyses carried out earlier in the chapter). Consequently, it is advantageous to represent three-dimensional structures where possible by the less expensive, simpler type of finite element model. Although the absolute values of the displacements and stresses may be exaggerated by using these simpler models, with care, the trend or pattern of the distributions can often be obtained. However, irrespective of which ever type of element is employed, it is essential to have some means of checking the sometimes voluminous data if analyses of incorrectly defined structures are to be avoided. The plotting technique employed here was simple, cheap and very flexible and although it did not eliminate the human element completely, proved to be very satisfactory.

Type of Analysis	Type of Element	No. of Elements	No. of Nodes	No. of Deg. of Fr.	Half Bandwdh.	Computer Time(Sec.)
Axisymmetric	3-noded triangles	64	46	92	14	65
Plane stress	4-noded rectangles	32	46	92	14	55
Plane strain	4-noded rectangles	32	46	92	14	55
Pseudo 3-D	8-noded hexahedra	32	92	276	51	710
Prismatic 3-D	8-noded hexahedra	128	230	690	96	4812
3-D	8-noded hexahedra	118	216	648	96	4466

FIG. 18. 4 Comparison of the computer times required for analysing the various finite element models of the M.O.D. type structure shown in FIG. 18.1.

N.B. All programs employed the direct Gauss elimination technique in submatrix block form with the stiffness coefficients stored in a disc backing store file

CHAPTER NINETEEN

FURTHER DEVELOPMENTS AND RESEARCH

19 FURTHER DEVELOPMENTS AND RESEARCH

19.1 INTRODUCTION

In the past, analyses which have been concerned with biological structures have generally considered the tissues to undergo only small strains and to behave as linear elastic isotropic materials. However, it is well known that many of the tissues undergo large deformations and are highly directionally orientated structures, so as to provide optimum mechanical properties to meet the functional demands placed upon them.

Although anisotropic material behaviour and the associated mechanical properties of tissues are neither fully understood nor documented, the work reported in this thesis has attempted to take into account these directional variations. Even so, to do this it has been necessary to assume that the tissues undergo only small displacements and behave as linear elastic orthotropic materials. However, it is becoming more apparent that even this model is still a very long way away from simulating actual tissue behaviour. Evidence is being accumulated which suggests that the tissues of the body are more nearly represented by linear viscoelastic material behaviour, Sedlin (21), Wills *et al* (81) and Edwards (141). Consequently, as the tissues' properties for this type of material behaviour become better known, methods of analysis which can incorporate and utilise this new data will be required.

The finite element method of stress analysis can be adapted for the analysis of structures involving visco-elastic material behaviour. Consequently, with this added facility,the method has an even wider field of application in the area of Bioengineering. While the next section outlines how viscoelasticity can be incorporated into the finite element method, the final section of the chapter discusses some other non-dental medical problems where the method could or has been fruitfully applied.

19.2 EXTENSION OF THE FINITE ELEMENT METHOD OF ANALYSIS TO PROBLEMS INVOLVING VISCOELASTIC MATERIAL BEHAVIOUR

For problems involving small strains and linear elastic material behaviour, the constitutive equation assumes that the stress-strain relationship is independent of time. However, for materials which exhibit linear viscoelastic behaviour, the strains and stresses are dependent upon the loading history of the material, see Chapter Three and Lee (142). Nevertheless, even though the constitutive equation for a viscoelastic material is dependent on time, the general conditions of displacement continuity and equilibrium still have to be satisfied, Zienkiewicz (129). Consequently, the only relationship which has to be modified in the finite element procedure outlined in Chapter Fourteen, is equation 14.24, namely:-

$$\{\sigma\} = [D](\{\varepsilon\} - \{\varepsilon_o\})$$

The equivalent relationship for a linear viscoelastic material can take the form of either

$$[D] = [D(\{\varepsilon\})]$$

$$\{\sigma\} = f(\{\varepsilon\})$$

$$\text{or} \{\varepsilon\} = f(\{\sigma\})$$

Therefore, by varying one or more of $[D]$, $\{\sigma\}$ or $\{\varepsilon_o\}$ an iterative type solution to the structural equilibrium equations given by equation 14.19 can be obtained. Obviously, the quantity which is varied, depends upon the nature or physical law defining the constitutive equation for the material. If adjustments are made to the $[D]$ matrix the process is known as one of variable stiffness whereas if $\{\varepsilon_o\}$ or $\{\sigma\}$ are adjusted, then the approach is termed either one of 'initial strain' or 'initial stress'.

The main disadvantage with the variable stiffness approach is that at each step in the solution process the structural stiffness matrix has to be reformed and a completely new solution found to the resulting equilibrium equations. For problems involving creep however, it is usual to determine the increments of the strains in terms of the corresponding stresses. Consequently, equality between the elastic relationship given by equation 14.24 and the corresponding viscoelastic constitutive relationship, is achieved by making adjustments to the value of $\{\varepsilon_o\}$. Hence, as $\{\varepsilon_o\}$ affects the value of the nodal forces $\{R\}$,

the iterative procedure is one of solving

$$[K]\{d\} - R(\{d\}) = 0$$

Zienkiewicz et al (143).

Initially, a solution is sought to

$$\{d_o\} = [K_o]^{-1} \{R_o\}$$

in which $\{R_o\}$ are the actual loads applied to the structure. The element elastic strains and stresses are then determined in the usual way. Consequently, by assuming these stresses to remain constant throughout the time interval Δt, the true or actual strains are subsequently determined from the constitutive equation:-

$$\{\varepsilon\} = f(\{\sigma\})$$

Equality between these actual strains and the elastic strains, determined by solving the equilibrium equations, is obtained by adjusting the value of $\{\varepsilon_o\}$ in equation 14.24. The additional nodal forces required to balance these 'initial strains' $\{\varepsilon_o\}$ are then computed as shown by equation 14.17, in a manner similar to that used for the thermal effects in the axisymmetric program discussed in 15.3. Hence, by taking the new overall nodal force vector $\{R_1\}$, the new nodal displacements are obtained from

$$\{d_1\} = [K_o]^{-1} \{R_1\}$$

This iterative type process is then repeated for all time increments as indicated by the flow diagram shown in FIG. 19.1. Of course, the question arises as to what values must be ascribed to the elastic constants.

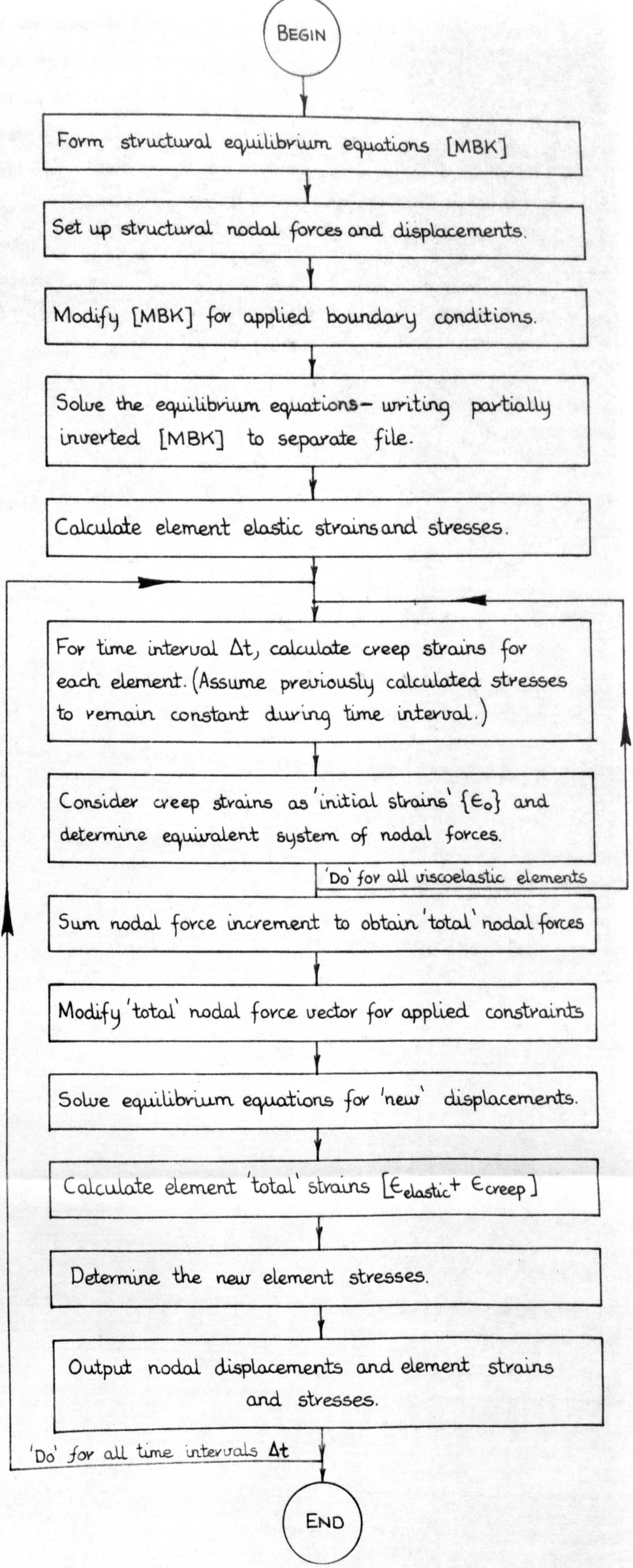


FIG. 19.1 Flow diagram for a 'linear viscoelastic' finite element analysis program.

If the material behaves in an essentially elastic manner with only small departures from linearity, then the original values of the properties are probably accurate enough. However, if it is apparent that non-linearity occurs at all stress levels, it may be necessary to make adjustments to the elastic properties during the iterative process. It is of course, advantageous if the same structural stiffness matrix can be used at every stage of the solution process. If the matrix can be left in a partially inverted state after the first solution, then each additional iteration can be carried out in only a fraction of the time required for the first solution.

An alternative approach to the one outlined above, is to determine only the increments in the nodal displacements resulting from the additional nodal forces required to balance the 'initial strains' $\{\varepsilon_o\}$, i.e.

$$\{\Delta d_1\} = [K_o]^{-1} \{\Delta R_1\} \quad \text{etc.}$$

In this case, the additional forces $\{\Delta R\}$ required at each stage of the iterative process, can be considered simply as being the unbalanced residual forces acting on the structure.

19.3 SCOPE AND APPLICATION OF THE FINITE ELEMENT METHOD TO OTHER MEDICAL PROBLEMS

The finite element method of analysis has been limited in this thesis almost exclusively to problems in the field of dentistry. Although suggestions for further research in this field were proposed in Chapter Twelve, it is

apparent that the method can be applied to a wide variety of other medical problems. In fact, Ghista (144), has employed the method to estimate the level of the stresses which can occur in the left ventricle of the heart during the cardiac cycle, and Matthews and West (145), to study the mechanical behaviour of the lung loaded by the action of it s own weight. This work, which is still in it s early stages, has like nearly all work in Biomechanics, been hampered by the paucity of data available concerning the mechanical properties of the tissues involved.

The finite element method may also have other applications in the cardiovascular field in helping to understand how the changes which occur in the blood vessels, affect the course of heart disease. The method could be employed to study blood flow in the arteries and to look at the effects on the flow and flow patterns caused by the hardening of the vessel walls as a result of the disease process.

Another area of medicine in which the finite element method of analysis could play a significant role, is in the field of orthopaedics. Indeed, there are numerous situations where the method could be applied to investigate skeletal structural problems. An obvious example lies in the area of implant surgery. Here, the majority of total joint replacements rely upon a filler material or cement for the fixation of the prosthesis into the marrow cavity of the bone. In the case of the hip replacement for

example, it is not known how the forces applied to the head of the prosthesis are transfered into the cortex of the femur. It may be that high stress concentrations are created in the bone in some instances. Subsequently, this may lead to the 'melting away' of the bone tissue in these areas and to the eventual loosening of the prosthesis.

The stems of many currently available hip prostheses sometimes fail in service as a result of fatigue. The finite element method may therefore prove to be a very useful technique for investigating new designs of prostheses' stems in order to reduce the very high bending stresses induced in these components during normal locomotion.

The foregoing discussion is by no means an exhaustive list of all the possible areas in medicine and biology where the finite element method could be fruitfully employed. However, as with any other process, method or technique which attempts to bridge the disciplines of medicine and engineering, the extent of the successful application of the finite element method to problems in the biomedical field depends finally upon the interchange of ideas and the cooperation of the various parties involved. It is hoped that this thesis is a small step in this direction.

SELECTED REFERENCES

SELECTED REFERENCES

1. SKINNER, E.W. and PHILLIPS, R.W.
The Science of Dental Materials.
W.B. Saunders Co., 6th Ed. 1967.

2. SWANSON, S.A.V. and FREEMAN, M.A.R.
Is Bone Hydraulically Strengthened ?
Med. and Biol. Engng. 4 : 433-438, 1966.

3. McLEISH, R.D. and HABBOOBI, S.
Strain Gauge Techniques for Cadaveric Bone.
Engng. in Med. 1(2) : 36-40 and 47, 1971.

4. EVANS, F.G. and LEBOW, M.
Regional Differences in Some of the Physical
Properties of the Human Femur.
J. App. Phys. 3 : 563-572, 1951.

5. TYLDESLEY, W.R.
The Mechanical Properties of Human Enamel
and Dentine.
Br. Dent. J. 106(8) : 269-277, 1959.

6. STANFORD, J.W. et al.
Determination of Some Compressive Properties of
Human Enamel and Dentine.
J.A.D.A. 57 : 487-495, 1958.

7. McELHANEY, J.H.
Dynamic Response of Bone and Muscle Tissue.
J. App. Phys. 21 : 1231-1236, 1966.

8. STANFORD, J.W., et al.
Compressive Properties of Hard Tooth Tissues and Some Restorative Materials.
J.A.D.A. 60 : 746-756, 1960.

9. CRAIG, R.G., et al.
Compressive Properties of Enamel, Dental Cements and Gold.
J. Dent. Res. 40(5) : 936-945, 1961.

10. HAINES, D.J.
Physical Properties of Human Tooth Enamel and Enamel Sheath Material Under Load.
J. Biomech. 1 : 117-125, 1968.

11. LEES, S. and ROLLINS, F.R.
Anisotropy in Hard Dental Tissues.
J. Biomech. 5 : 557-566, 1972.

12. PEYTON, F.A., et al.
Physical Properties of Dentine.
J. Dent. Res. 31 : 366-370, 1952.

13. CRAIG, R.G. and PEYTON, F.A.
Elastic and Mechanical Properties of Human Dentine.
J. Dent. Res. 37 : 710-718, 1958.

14. RENSON, C.E.
An Experimental Study of the Physical Properties of Human Dentine.
A Thesis submitted to the University of London for the degree of Doctor of Philosophy, November 1970.

15. DYMENT, M.L. and SYNGE, J.L.
The Elasticity of the Periodontal Membrane.
Oral Health (Toronto). 25 : 105-109, 1935.

16. McELHANEY, J.H., et al.
Mechanical Properties of Cranial Bone.
J. Biomech. 3 : 495-511, 1970.

17. WOOD, J.L.
Dynamic Response of Human Cranial Bone.
J. Biomech. 4 : 1-12, 1971.

18. KRAUS, H.
On the Mechanical Properties and Behaviour of
Human Compact Bone.
Advances in Biomedical Engineering and
Medical Physics.
Editor S.N. Levine, Volume 2, 1968.

19. WELCH, D.O.
The Composite Structure of Bone and it's Response
to Mechanical Stress.
Recent Advances in Engineering Science.
Editor A.C. Eringen, Volume 5, Part 1, 1968.

20. SWANSON, S.A.V.
Biomechanical Characteristics of Bone.
Advances in Biomedical Engineering.
Editor R.M. Kenedi, Volume 1, 1971.

21. SEDLIN, E.D.
A Rheologic Model for Cortical Bone.
Acta. Ortho. Scand. Supplement No. 83, 1965.

22. DEMPSTER, W.T. and LIDDICOAT, R.T.
Compact Bone as a Non-isotropic Material.
Am. J. Anat. 91 : 331-362, 1952.

23. BONFIELD, W. and LI, C.H.
Deformation and Fracture of Bone.
J. of App. Physics. 37(2) : 869-875, 1966.

24. BONFIELD, W. and LI, C.H.
Anisotropy of Non-elastic Flow in Bone.
J. of App. Physics. 38(6) : 2450-2455, 1967.

25. CARTWRIGHT, A.G.
The Effects of Histological Variation on the
Tensile Properties of Cortical Bone.
A Thesis submitted to the University of Surrey for
the degree of Doctor of Philosophy, September 1971.

26. YOKOO, S.
Compression Test of the Cancellated Bone.
J. Kyoto Pref. Med. Univ. 51 : 273-276, 1952.

27. HOWELL, A.H. and BRUDEVOLD, F.
Vertical Forces Used During Chewing Food.
J. Dent. Res. 29(2) : 133-136, 1950.

28. YURKSTAS, A. and CURBY, W.A.
Force Analysis of Prosthetic Appliances
During Function.
J. Pros. Den. 3(1) : 82-87, 1953.

29. CHARNLEY, J.
Biomechanics in Orthopaedic Surgery.
Biomechanics and Related Bio-Engineering Topics
Proceedings of a Symposium held in Glasgow in 1964.
Editor R.M. Kenedi, Pergamon: 1965.

30. PICKARD, H.M.
A Manual of Operative Dentistry.
Oxford University Press. 3rd Edition, 1970.

31. SYNGE, J.L.
The Tightness of the Teeth, Considered as a Problem concerning the Equilibrium of a Thin Incompressible Elastic Membrane.
Phil. Trans. Roy. Soc. (A)231 : 435-477, 1933.

32. HAY, G.E.
Stress in the Periodontal Membrane.
Oral Health. 29 : 257-261, 1939.

33. HAACK, D.C. and HAFT, E.E.
An Analysis of Stresses in a Model of the Periodontal Ligament.
Int. J. Engng. Sci. 10 : 1093-1106, 1972.

34. GABEL, A.B.
A Mathematical Analysis of the Function of the Fibres of the Periodontal Membrane.
J. of Periodontology. 27 : 191-198, 1956.

35. LEDLEY, R.S.
Theoretical Analysis of Displacement and Force Distribution for the Tissue-Bearing Surface of Dentures.
J. Dent. Res. 47(2) : 318-322, 1968.

36. LEDLEY, R.S. and HUANG, H.K.
Linear Model of Tooth Displacement by Applied Forces.
J. Dent. Res. 47 : 427-432, 1968.

37. LEDLEY, R.S. and HUANG, H.K.
Numerical Experiments with a Linear Force-Displacement Tooth Model.
J. Dent. Res. 48 : 32-37, 1969.

38. TSAO, D.H.
Designing Occlusal Rests using Mathematical Principles.
J. Prosth. Dent. 23(2) : 154-163, 1970.

39. BURSTONE, C.J.
Mechanics.
Vistas in Orthodontics, Chap. 5.
Ed. Kraus, B.S. and Riedel, R.A., H. Kimpton, 1962.

40. CHRISTIANSEN, R.L. and BURSTONE, C.J.
Centres of Rotation within the Periodontal Space.
Amer. J. Orthod. 55(4) : 353-369, 1969.

41. HOPPENSTAND, D.C. and McCONNELL, D.
Mechanical Failure of Amalgam Restorations with Zinc Phosphate and Zinc Oxide/Eugenol Cement Bases.
J. Dent. Res. 39 : 899-905, 1960.

42. MAHLER, D.B., et al.
Evaluation of Techniques for Analysing Cavity Design for Amalgam Restorations.
J. Dent. Res. 40(3) : 497-503, 1961.

43. HAACK, D.C. and WEINSTEIN, S.
Geometry and Mechanics as Related to Tooth Movement Studied by Means of Two-dimensional Models.
J.A.D.A. 66 : 157-164, 1963.

44. DEMPSTER, W.T. and DUDDLES, R.A.
Tooth Statics.
J.A.D.A. 68 : 652-666, 1964.

45. HENDERSON, D. et al.
The Cantilever Type of Posterior Fixed Partial Dentures.
A Laboratory Study.
J. Pros. Dent. 24 : 47-67, 1970.

46. DICKSON, R.L.
Mechanical Analysis of Posterior Teeth in Centric Closure.
J. Pros. Dent. 27(4) : 358-363, 1972.

47. HOLISTER, G.S.
Experimental Stress Analysis.
Pub. Cambridge University Press 1967.

48. FROCHT, M.M.
Photoelasticity. Vols. 1 and 2.
Pub. Wiley, New York. 1957.

49. MAHLER, D.B. and PEYTON, F.A.
Photoelasticity as a Research Technique for Analysing Stresses in Dental Structures.
J. Dent. Res. 34 : 831-838, 1955.

50. MAHLER, D.B.
An Analysis of Stresses in a Dental Amalgam Restoration.
J. Dent. Res. 37 : 516-526, 1958.

51. CRAIG, R.G. et al.
Experimental Stress Analysis of Dental Restorations.
Part 1. Two-dimensional Photoelastic Stress Analysis of Inlays.
J. Pros. Dent. 17(3) : 277-291, 1967.

52. CRAIG, R.G. et al.
Experimental Stress Analysis of Dental Restorations.
Part 2. Two-dimensional Photoelastic Stress Analysis of Crowns.
J. Pros. Dent. 17(3) : 292-302, 1967.

53. EL-EBRASHI, M.K. et al.
Experimental Stress Analysis of Dental Restorations.
Part 3. The Concept of the Geometry of Proximal Margins.
J. Pros. Dent. 22(3) : 333-345, 1969.

54. EL-EBRASHI, M.K. et al.
Experimental Stress Analysis of Dental Restorations.
Part 4. The Concept of Parallelism of Axial Walls.
J. Pros. Dent. 22(3) : 346-353, 1969.

55. EL-EBRASHI, M.K. et al.
Experimental Stress Analysis of Dental Restorations.
Part 5. The Concept of Occlusal Reduction and Pins.
J. Pros. Dent. 22(5) : 565-577, 1969.

56. EL-EBRASHI, M.K. et al.
Experimental Stress Analysis of Dental Restorations.
Part 6. The Concept of Proximal Reduction in Compound Restorations.
J. Pros. Dent. 22(6) : 663-670, 1969.

57. EL-EBRASHI, M.K. et al.
Experimental Stress Analysis of Dental Restorations.
Part 7. Structural Design and Stress Analysis of Fixed Partial Dentures.
J. Pros. Dent. 23(2) : 177-186, 1970.

58. TANNER, A.N.
Factors Affecting the Design of Photoelastic Models for Two-dimensional Analysis.
J. Pros. Dent. 27(1) : 48-62, 1972.

59. NALLY, J.N. et al.
Experimental Stress Analysis of Dental Restorations.
Part 9. Two-dimensional Photoelastic Stress Analysis
of Porcelain Bonded to Gold Crowns.
J. Pros. Dent. 25(3) : 307-316, 1971.

60. LEHMAN, M.L. and MEYER, M.L.
Relationship of Dental Caries and Stress: Concentrations
in Teeth as Revealed by Photoelastic Tests.
J. Dent. Res. 45(6) : 1706-1714, 1966.

61. JOHNSON, E.W. et al.
Stress Pattern Variations in Operatively Prepared Human
Teeth, Studied by Three-dimensional Photoelasticity.
J. Dent. Res. 47(4) : 548-558, 1968.

62. LEHMAN, M.L.
Stress Distribution in the Alveolar Bone.
J. Biomech. 1 : 139-145, 1968.

63. FARAH, J.W. and CRAIG, R.G.
Reflection Photoelastic Stress Analysis of a Dental
Bridge.
J. Dent. Res. 50(5) : 1253-1259, 1971.

64. SHARRY, J.J. et al.
Influence of Artificial Tooth Forms on Bone Deformation
Beneath Complete Dentures.
J. Dent. Res. 39(2) : 253-266, 1960.

65. CRAIG, R.G. and PEYTON, F.A.
Measurement of Stresses in Fixed-Bridge Restorations
Using a Brittle Coating Technique.
J. Dent. Res. 44(4) : 756-762, 1965.

66. TILLITSON, E.W. et al.
Experimental Stress Analysis of Dental Restorations.
Part 8. Surface Strains on Gold and Chromium Fixed Partial Dentures.
J. Pros. Dent. 24(2) : 174-180, 1970.

67. BISPLINGHOFF, R.L., MAR, J.W. and PIAN, T.H.H.
Statics of Deformable Solids.
Pub. Addison-Wesley, 1965.

68. MUHLEMANN, H.R.
Tooth Mobility.
J. Periodont. Part 1 25(JAN) : 22-29, 1954.
J. Periodont. Part 2 25(APR) : 125-128, 1954.
J. Periodont. Part 3 25(APR) : 128-137,153, 1954.

69. MUHLEMANN, H.R.
10 Years of Tooth-Mobility Measurements.
J. Periodont. 31 : 110-122, 1960.

70. PARFITT, G.J.
Measurement of the Physiological Mobility of Individual Teeth in an Axial Direction.
J. Dent. Res. 39(3) : 608-618, 1960.

71. PARFITT, G.J.
The Dynamics of a Tooth in Function.
J. Periodont. 32 : 102-107, 1961.

72. PICTON, D.C.A.
Tilting Movements of the Teeth During Biting.
Arch. Oral Biol. 7 : 151-159, 1962.

73. PICTON, D.C.A.
Distortion of the Jaws During Biting.
Arch. Oral Biol. 7 : 573-580, 1962.

74. PICTON, D.C.A.
The Effect of Repeated Thrusts on Normal Axial Tooth Mobility.
Arch. Oral Biol. 9 : 55-63, 1964.

75. PICTON, D.C.A.
Some Implications of Normal Tooth Mobility During Mastication.
Arch. Oral Biol. 9 : 565-573, 1964.

76. ANDERSON, D.J.
Measurement of Stress in Mastication.
Part 1. J. Dent. Res. 35 : 664-670, 1956.
Part 2. J. Dent. Res. 35 : 671-673, 1956.

77. PICTON, D.C.A.
On the Part Played by the Socket in Tooth Support.
Arch. Oral Biol. 10 : 945-955, 1965.

78. PICTON, D.C.A. and DAVIES, W.I.R.
Dimensional Changes in the Periodontal Membrane of Monkeys (Macara lrus) due to Horizontal Thrusts applied to the Teeth.
Arch. Oral Biol. 12 : 1635-1643, 1967.

79. PICTON, D.C.A.
The Effect on Tooth Mobility of Trauma to the Mesial and Distal Regions of the Periodontal Membrane in Monkeys.
Helv. Odont. Acta. 11 : 105-112, 1967.

80. PICTON, D.C.A. and SLATTER, J.M.
The Effect on Horizontal Tooth Mobility of Experimental Trauma to the Periodontal Membrane in Regions of Tension or Compression in Monkeys.
J. Periodont. Res. 7 : 35-41, 1972.

81. WILLS, D.J., PICTON, D.C.A., and DAVIES, W.I.R.
An Investigation of the Viscoelastic Properties of the Periodontium in Monkeys.
J. Periodont. Res. 7 : 42-51, 1972.

82. BIEN, S.M.
Hydrodynamic Damping of Tooth Movement.
J. Dent. Res. 45(3) : 907-914, 1966.

83. GRESZCZUK, L.B.
Effect of Material Orthotropy on the Directions of the Principal Stresses and Strains.
From Orientation Effects in the Mechanical Behaviour of Anisotropic Structural Materials.
A.S.T.M., S.T.P. No. 405 : 1-13, 1965.

84. WHEELER, R.C.
Tooth Form. A Manual on Drawing and Carving.
Pub. W.B. Saunders Co., 1939.

85. CHOU, P.C. and PAGANO, N.J.
Elasticity.
Pub. D. Van Nostrand Co., 1967.

86. BRADEN, M.
Heat Conduction in Teeth and the Effect of Lining Materials.
J. Dent. Res. 43 : 315-322, 1964.

87. THRESHER, R.W. and SAITO, G.E.
The Stress Analysis of Human Teeth.
J. Biomech. 6 : 443-449, 1973.

88. FARAH, J.W. et al.
Photoelastic and Finite Element Stress Analysis of a Restored Axisymmetric First Molar.
J. Biomech. 6 : 511-520, 1973.

89. UTLEY, R.K.
The Activity of Alveolar Bone Incident to Orthodontic Tooth Movement as Studied by Oxytetracycline-induced Fluorescence.
Am. J. of Orthod. 54(3) : 167-201, 1968.

90. HERMANSON, P.C.
Alveolar Bone Remodelling Incident to Tooth Movement.
Angle Orthod. 42(2) : 107-115, 1972.

91. STEPHENS, C.D.
A Preliminary Report of a Cephalometric Appraisal of Incisor Movement During Orthodontic Treatment.
British J. of Orthod. 1(2) : 44, 1974.

92. WOLFF, J.
Ueber die innere Architectur der Knochen und ihre Bedeutung fiir die frage vom Knochenwachstum.
Virchow's Arch Path. Anat. 50 : 389-453, 1870.

93. WOLFF, J.
Das Gasetz der Transformation der Knochen.
Berlin (Monograph). 1892.

94. THOMPSON, D'Arcy W.
On Growth and Form.
Pub. Cambridge Univ. Press. 1952.

95. FELL, H.B.
In Biochemistry and Physiology of Bone.
Editor G.H. Bourne, 1st Edition.
Pub. Academic Press.

96. BASSETT, C.A.L.
Electrical Effects in Bone.
Scientific America. 213 : 18-25, 1965.

97. CHAMAY, A. and TSCHANTZ, P.
Mechanical Influences in Bone Remodelling.
Experimental Research on Wolff's Law.
J. Biomech. 5 : 173-180, 1972.

98. EPKER, B.N. and FROST, H.M.
Correlation of Bone Resorption and Formation with the Physical Behaviour of Loaded Bone.
J. Dent. Res. 44(1) : 33-41, 1965.

99. BAUMRIND, S.
A Reconsideration of the Propriety of the " Pressure-Tension" Hypothesis.
Am. J. Orthod. 55(1) : 12-22, 1969.

100. REITAN, K.
Some Factors Determining the Evaluation of Forces in Orthodontics.
Am. J. Orthod. 43(1) : 32-45, 1957.

101. REITAN, K.
Bone Formation and Resorption during Reversed Tooth Movement.
From Vistas in Orthodontics.
Editors B.S. Kraus and R.H. Reidel.
Pub. H. Kimpton 1962.

102. REITAN, K.
Effects of Force Magnitude and Direction of Tooth Movement on Different Alveolar Bone Types.
Angle Orthod. 34(4) : 244-255, 1964.

103. REITAN, K.
Clinical and Histological Observations on Tooth Movement During and After Orthodontic Treatment.
Am. J. of Orthod. 53(10) : 721-745, 1967.

104. ACKERMAN, J.L. et al.
The Effects of Quantified Pressure on Bone.
Am. J. Orthod. 52(1) : 34-46, 1966.

105. WEINSTEIN, S.
Minimal Forces in Tooth Movement.
Am. J. Orthod. 53(12) : 881-903, 1967.

106. FUKADA, E. and YASUDA, I.
On the Piezoelectric Effect of Bone.
J. Physical Soc. of Japan. 12(10) : 1158-1162, 1957.

107. SHAMOS, M.H. and LAVINE, L.S.
Physical Bases for the Bioelectric Effects in
Mineralised Tissues.
Clin. Orthop. 35 : 177-188, 1964.

108. BRADEN, M. et al.
Electrical and Piezoelectric Properties of Dental
Hard Tissues.
Nature. 212 : 1565-1566, 1966.

109. BASSETT, C.A.L. and BECKER, R.O.
Generation of Electrical Potentials by Bone in
Response to Mechanical Stress.
Science. 137 : 1063-1064, 1962.

110. COCHRAN, G.V.B. et al.
Stress Generated Electrical Potentials in the
Mandible and Teeth.
Arch. Oral Biol. 12 : 917-920, 1967.

111. McELHANEY, J.H.
The Charge Distribution on the Human Femur Due to Load.
J. Bone and Joint Surgery. 49(8) : 1561-1571, 1967.

112. GJELSVIK, A.
Bone Remodelling and Piezoelectricity (1).
J. Biomech. 6 : 69-77, 1973.

113. GJELSVIK, A.
Bone Remodelling and Piezoelectricity (2).
J. Biomech. 6 : 187-193, 1973.

114. KOCH, J.C.
The Laws of Bone Architecture.
Am. J. Anat. 21 : 177-298, 1917.

115. FESSLER, H.
Load Distribution in a Model of a Hip Joint.
J. Bone and Joint Surgery. 39(1) : 145-153, 1951.

116. WILLIAMS, J.F. and SVENSSON, N.L.
An Experimental Stress Analysis of the Neck of the Femur.
Med. and Biol. Eng. 9 : 479-493, 1971.

117. PAUL, J.P.
Forces Transmitted by Joints in the Human Body.
Proc. Inst. Mech. Eng. 181(3J) : 8-15, 1967.

118. RYBICKI, E.F. et al.
On the Mathematical Analysis of Stress in the Human Femur.
J. Biomech. 5 : 203-215, 1972.

119. STOREY, E.
Bone Changes Associated with Tooth Movement.
A Radiographic Study.
The Australian J. of Dent. 57(2) : 57-64, 1953.

120. EDWARDS, J.G.
A Study of the Periodontium During Orthodontic
Rotation of Teeth.
Am. J. of Orthod. 54(6) : 441-476, 1968.

121. TAPPEN, N.C.
Main Patterns and Individual Differences in Baboon Skull Split-lines and Theories of Causes of Split-line Orientation in Bone.
Am. J. Phys. Anthrop. 33(1) : 61-71, 1970.

122. CURREY, J.D.
The Adaption of Bones to Stress.
J. Theoret. Biol. 20 : 91-106, 1968.

123. FROST, H.M.
The Laws of Bone Structure.
Pub. Charles C. Thomas, Springfield.

124. OXNARD, C.E.
Tensile Forces in Skeletal Structures.
J. of Morphology. 134 : 425-436, 1971.

125. HRENNIKOFF, A.
Solution of Problems of Elasticity by the Framework Method.
J. App. Mech. 63 : A-169-175, 1941.

126. YETTRAM, A.L. and ROBBINS, K.
Space-Framework Method for Three-dimensional Solids.
J. Eng. Mech. Div. A.S.C.E. EM 6, 21-36, 1967.

127. TURNER, M.J. et al.
Stiffness and Deflection Analysis of Complex Structures.
J. Aero. Sci. 23 : 805-823, 1956.

128. DESAI, C.S. and ABEL, J.F.
Introduction to the Finite Element Method.
Pub. Van Nostrand Reinhold Co., 1972.

129. ZIENKIEWICZ, O.C.
The Finite Element Method in Engineering Science.
Pub. McGraw-Hill, 1971.

130. ERGATOUDIS, J.G.
Isoparametric Elements in Two and Three-dimensional Analysis.
Ph. D. Thesis, University of Wales, Swansea, 1968.

131. BROOKS, D.F. and BROTTON, D.M.
Computer Systems for Analysis of Large Frameworks.
J. of the Structural Division, Proc. A.S.C.E.
93(ST6) : 1-23, 1967.

132. YETTRAM, A.L. and HIRST, M.J.S.
The Solution of Structural Equilibrium Equations by the Conjugate Gradient Method with Particular Reference to Plane Stress Analysis.
Int. J. Num. Meth. Eng. 3 : 349-360, 1971.

133. I.C.L. Technical Publication 4087,
Graph Plotter 1900 series, 1970.

134. TIMOSHENKO, S. and GOODIER, J.N.
Theory of Elasticity.
Pub. McGraw-Hill, 2nd Edition, 1951.

135. IRONS, B.M.
A Frontal Solution Program for Finite Element Analysis.
Int. J. Num. Meth. Eng. 2 : 5-32, 1970.

136. KOPAL, Z.
Numerical Analysis.
Pub. Chapman and Hall, London 1955.

137. CLOUGH, R.W.
Comparison of Three-dimensional Finite Elements.
From Proceedings of a Symposium on the Application of Finite Element Methods in Civil Engineering.
A.S.C.E. Nov. 13-14 : 1969.

138. IRONS, B.M.
Quadrature Rules for Brick Based Finite Elements.
Int. J. Num. Meth. Eng. 3 : 295-296, 1971.

139. McKINNON, V.
Matrix Analysis of Plates and Plated Grillages Using Equivalent Framework Idealisations.
Ph. D. Thesis, Leeds University, 1970.

140. ZIENKIEWICZ, O.C.
Private Communication.

141. EDWARDS, J.
Physical Characteristics of Articular Cartilage.
Paper 6. Proc. Inst. Mech. Eng. 181(3J) 16-24, 1967.

142. LEE, E.H.
Stress Analysis in Viscoelastic Materials.
J. App. Physics. 27(7) : 665-672, 1956.

143. ZIENKIEWICZ, O.C. et al.
A Numerical Method of Viscoelastic Stress Analysis.
Int. J. Mech. Sci. 10 : 807-827, 1968.

144. GHISTA, D.N. et al.
Computerised Left Ventricular Mechanics and Control System Analysis Models Relevant for Cardiac Diagnosis.
Comput. Biol. Med. 3 : 27-46, 1973.

145. MATTHEWS, F.L. and WEST, J.B.
Finite Element Displacement Analysis of a Lung.
J. Biomech. 5 : 591-600, 1972.

GENERAL REFERENCES

GENERAL REFERENCES

ANDERSON, G.M.
Practical Orthodontics.
Henry Kimpton, London. 7th Ed. 1948.

HANCOX, N.M.
Biology of Bone.
Cambridge University Press. 1972.

HEMLEY, S.
Orthodontic Theory and Practice.
Grune and Stratton, N.Y. 2nd Ed. 1953.

PEYTON, F.A., et al.
Restorative Dental Materials.
Henry Kimpton, London. 2nd Ed. 1964.

SCOTT, J.H. and SYMONS, N.B.B.
Introduction to Dental Anatomy.
E. and S. Livingstone, Edinburgh and London. 3rd Ed. 1961.

SICHER, H.
Oral Anatomy.
Henry Kimpton, London. 1949.

TYLMAN, S.D., and TYLMAN, S.G.
Theory and Practice of Crown and Bridge Prosthodontics.
Mosby Co., St. Louis. 4th Ed. 1960.

Mechanisms of Hard Tissue Destruction.
Symposium of the American Association for the Advancement of Science. Dec., 29th and 30th, 1962.
Editor R.F. Sognnaes.
Publication no. 75 of the A.A.A.S.

The American Textbook of Operative Dentistry.
Edited by A.B. Gabel.
Henry Kimpton, London. 9th Ed. 1954.

APPENDIX ONE

NUMERICAL INTEGRATION IN TWO AND THREE DIMENSIONS USING THE GAUSS QUADRATURE FORMULA

APPENDIX ONE

Numerical Integration in Two and Three-Dimensions using the Gauss Quadrature Formula

In order to determine the finite element stiffness matrices and boundary surface load vectors, integrals of the form

$$[k] = \int_{-1}^{1}\int_{-1}^{1} [B]^T[D][B]t\det[J]d\delta d\eta \quad \text{(equation 16.20)}$$

$$[k] = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} [B]^T[D][B]\det[J]d\delta d\eta d\xi \quad \text{(equation 17.15)}$$

and

$$\{F_b\} = \int_{-1}^{1}\int_{-1}^{1} [N]^T \{g\}\, d\eta d\xi \quad \text{(equation 17.17)}$$

have to be evaluated. However, the function $f(\delta)$ shown in FIG. A1.1, can be integrated approximately by using the Gaussian Quadrature formula, Kopal (136), i.e.

$$\int_{-1}^{1} f(\delta)d\delta = H_1 f(\delta_1) + H_2 f(\delta_2) + \ldots . H_n f(\delta_n)$$

or

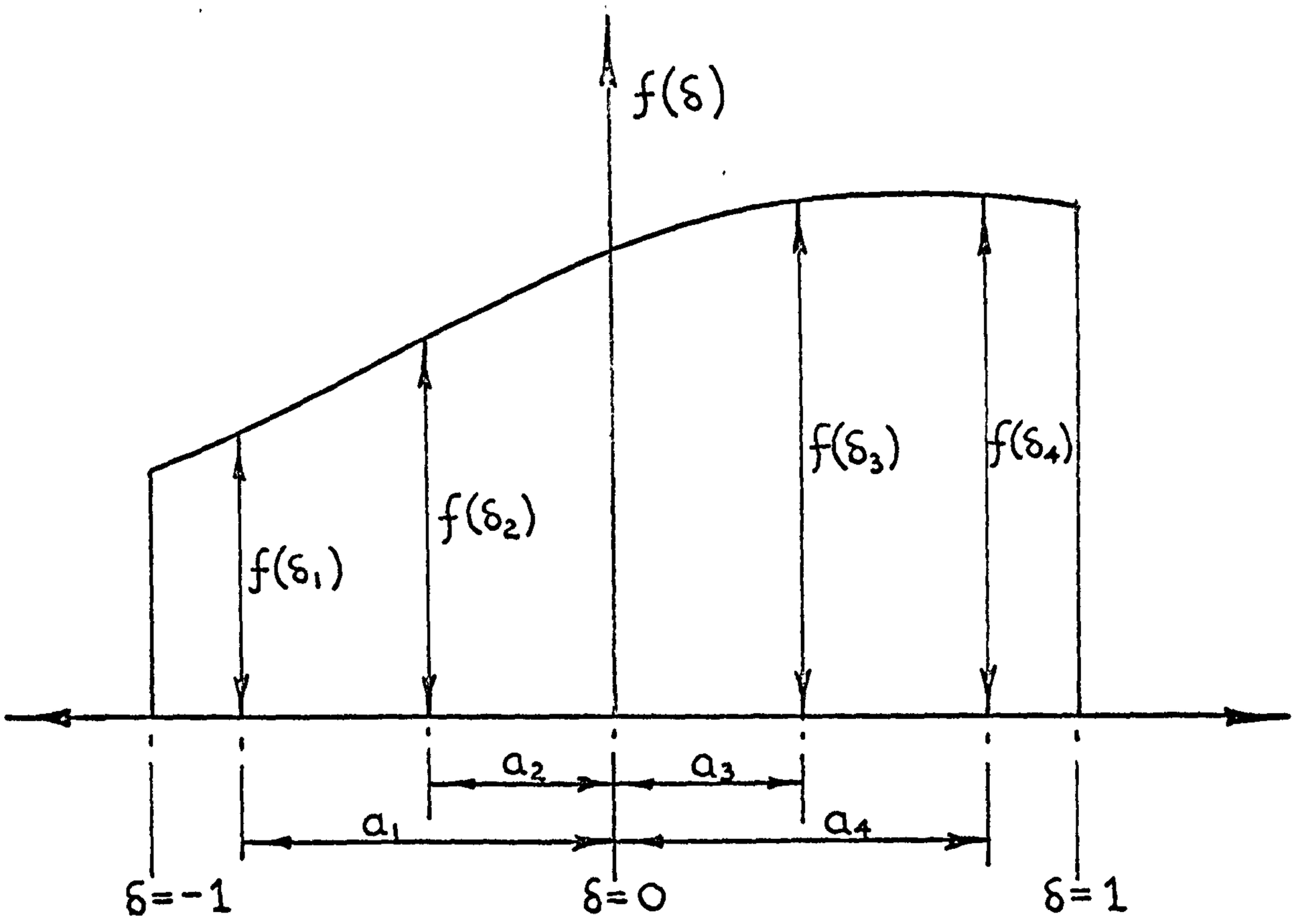
$$\int_{-1}^{1} f(\delta)d\delta = \sum_{i=1}^{i=n} H_i f(\delta_i) \ldots \ldots \ldots \text{A1.1}$$

where n is the total number of sampling points.

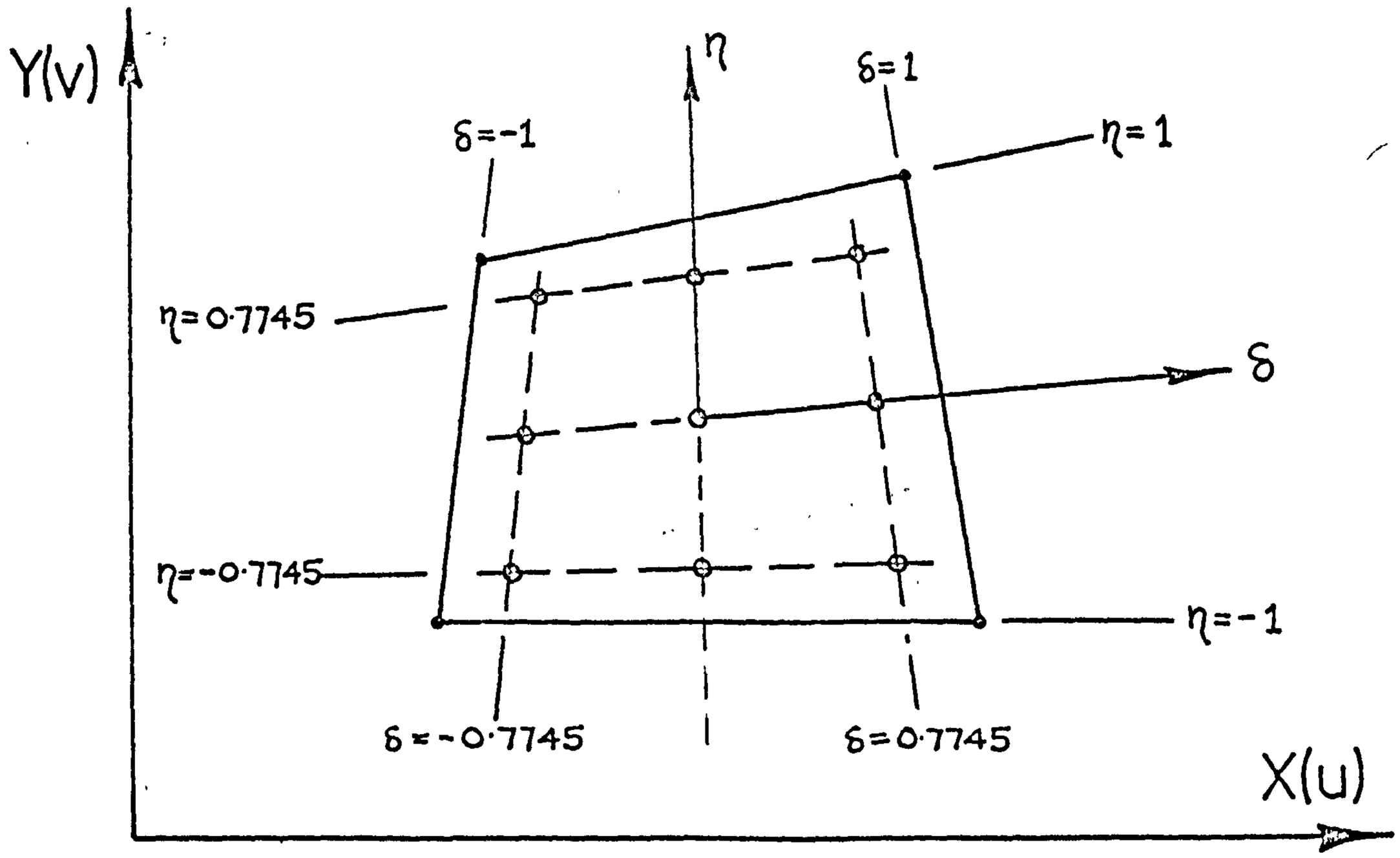
$f(\delta_i)$ is the value of the function at the sampling point i.

and H_i is a weighting coefficient for the sampling point i.

If the value of the function varies steeply, then an increase in the number of sampling points used in the



a) Sampling points for the function f(δ).



b) 3x3 Gauss integration points for 2-D element.

FIG. A1.1 Sampling points for numerical integration in one and two dimensions.

evaluation of the integral will obviously improve the accuracy of the solution obtained.

For the two-dimensional finite element, the most obvious way of obtaining the integral

$$[k] = \int_{-1}^{1}\int_{-1}^{1}[B]^T[D][B]t\det[J]d\delta d\eta$$

or

$$[k] = \int_{-1}^{1}\int_{-1}^{1}f(\delta,\eta)d\delta d\eta$$

is to evaluate the inner integral by making δ equal to a constant, i.e.

$$\int_{-1}^{1}f(\delta,\eta)d\eta = \sum_{j=1}^{j=n}H_j f(\delta,\eta_j)$$

which we will call $\phi(\delta)$. Hence, by evaluating the outer integral in a similar manner we have

$$\int_{-1}^{1}\phi(\delta)d\delta = \sum_{i=1}^{i=n}H_i\phi(\delta_i)$$

Therefore,

$$\int_{-1}^{1}\int_{-1}^{1}f(\delta,\eta)d\delta d\eta = \sum_{i=1}^{i=n}H_i \sum_{j=1}^{j=n}H_j f(\delta_i,\eta_j)$$

or

$$\int_{-1}^{1}\int_{-1}^{1}f(\delta,\eta)d\delta d\eta = \sum_{i=1}^{i=n}\sum_{j=1}^{j=n}H_i H_j f(\delta_i,\eta_j)$$

Similarly, for three-dimensions

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}f(\delta,\eta,\xi)d\delta d\eta d\xi = \sum_{i=1}^{i=n}\sum_{j=1}^{j=n}\sum_{l=1}^{l=n}H_i H_j H_l f(\delta_i\eta_j\xi_l)$$

The sampling points at which to evaluate the function in order to achieve the greatest accuracy have been determined and are tabulated by Kopal (136) together with the corresponding weighting factors.

For example, for three Gauss points, i.e. n=3,

Sampling Point	Weighting Factor
$\pm$ a	H
0.77459667	0.55555556
0.00000000	0.88888889

For a two-dimensional finite element therefore, employing this three point rule, 3 by 3 evaluations of the function $f(\delta,\eta)$ would be required, see FIG. A1.1b. In three-dimensions however, and in particular for the 20-noded finite elements, the corresponding 3 by 3 by 3 Gauss points requiring 27 evaluations of the function $f(\delta,\eta,\xi)$, i.e. $[B]^T[D][B]\det[J]$, proved to be far too expensive in terms of computation time. Consequently, the much less expensive but reasonably accurate 14 point integrating rule described by Irons (138) was implemented.

For this scheme, the integral becomes:-

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\delta,\eta,\xi)\,d\delta\,d\eta\,d\xi = HH_6\{f(a,0,0) + f(-a,0,0) + f(0,a,0) + f(0,-a,0) + f(0,0,a) + f(0,0,-a)\} + HH_8\{f(b,b,b) + f(-b,-b,-b) + f(b,-b,b) + f(-b,-b,b) + f(b,b,-b) + f(-b,b,-b) + f(b,-b,-b) + f(-b,b,b)\}$$

where

Sampling Point	Combined Weighting Factor HH (equals $H_i x H_j x H_l$)
a = 0.795822426	HH_6 = 0.886426593
b = 0.758786911	HH_8 = 0.335180055

APPENDIX TWO

THE CONJUGATE GRADIENT METHOD OF EQUATION SOLUTION

APPENDIX TWO

The Conjugate Gradient Method of Equation Solution

The great advantage of the conjugate gradient method of equation solution is that it is not necessary to form either the complete or the modified arrangement of the structural stiffness matrix. Consequently, although each of the finite element's stiffness matrices have to be stored throughout the solution process, (as they are required for each itera+ion), for structures such as the thick plate and Bousinesq problems where all the elements are the same, the amount of storage space required for the element [k]s is relatively small. For such problems therefore, the amount of storage space required for the conjugate gradient method, should not be as great as that required for the direct Gauss elimination method. Hence, for a computer installation having a relatively small core storage capacity, the method seems to be most attractive.

As discussed earlier, the conjugate gradient method proceeds by making adjustments to the displacement vector, until the residual forces acting on the structure are reduced to an insignificantly low level, see FIG. 14.8. Consequently, the iterative procedure is terminated once

the residual force vector has converged to this preselected low level. However, certain difficulties arise because

1). It is difficult to predetermine the required level of the residual forces for a specific problem such that the solution obtained is "acceptable". (The acceptable residual size is in some way governed by the boundary conditions applied to the structure).

and 2). It is impossible to predetermine the number of iterations required to achieve the predetermined residual force level.

From the computer operational aspect therefore, the method is far from satisfactory as the computational time required for a specific problem cannot be determined ab initio. In order to overcome this drawback, the complete problem analysis procedure was broken down into five separate sections and programs as shown by FIG. A2.1. Consequently, it was possible using this scheme to predetermine approximately the computational time required for each particular section. In addition, it was possible with the major section INTERSOLV, to tailor the number of iterations requested so as to utilize the amount of computer time available. It was also possible using this scheme to examine the displacements, (and also the element stresses by running the STRESS program), after each run of INTERSOLV to see if the solution had converged. If the degree of convergence was not considered sufficient then the INTERSOLV program was run again and the process repeated.

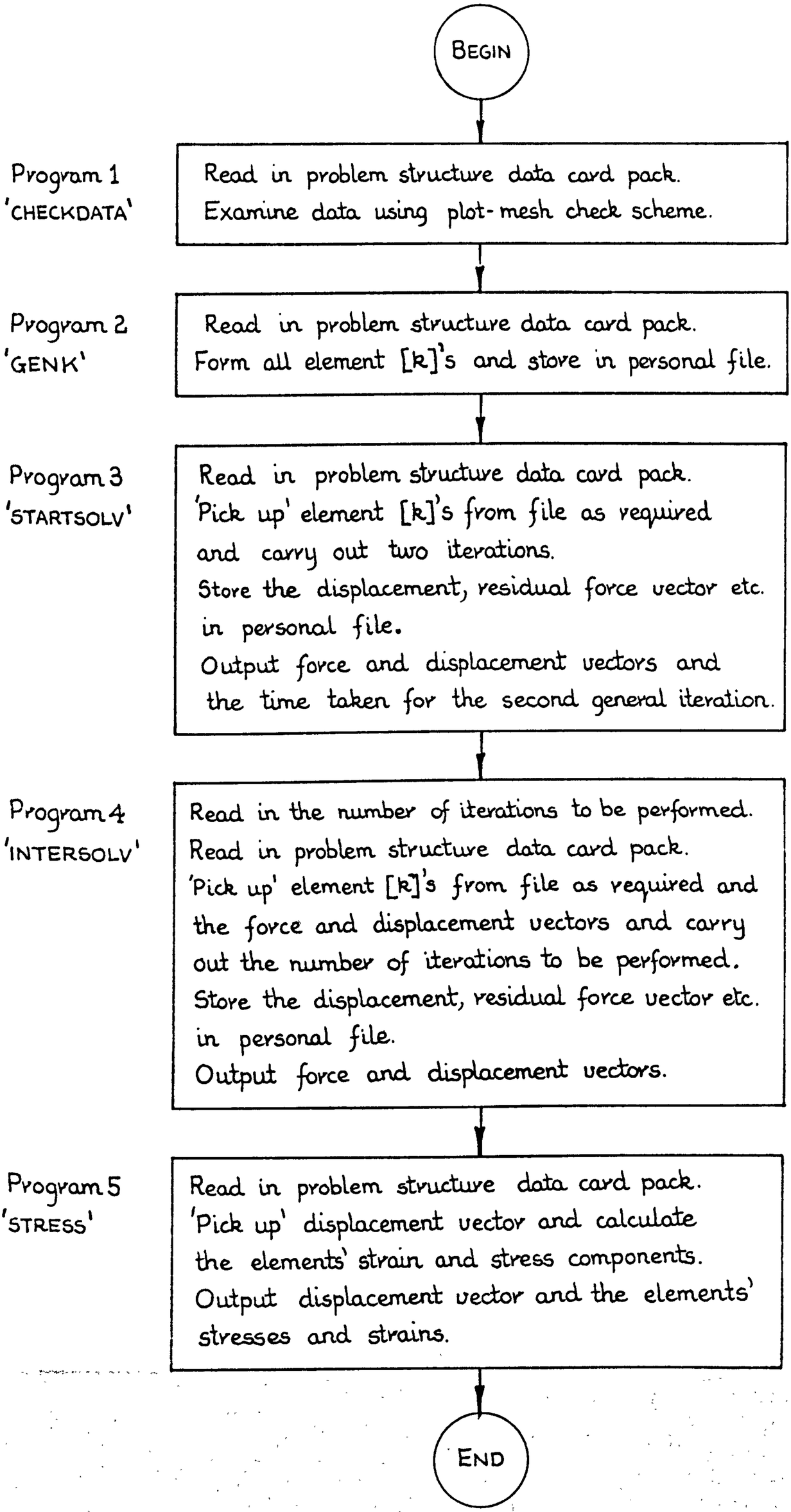


FIG. A2. 1 Flow diagram of the 'conjugate gradient' finite element analysis program suite.

The conjugate gradient method was examined by testing the process described above on two problems. The first was the thick plate problem discussed previously in 17.9. The structure, which consisted of eight identical 20-noded finite elements, is shown in FIG. 17.12. Using the conjugate gradient method, eighty iterations were required to obtain nodal displacements and element strain and stress components of the same order of accuracy, i.e. to six decimal places, as those obtained by using the direct Gauss elimination method. Therefore, as each iteration took 28 seconds to complete, on comparing the solution times required by the two methods we have:-

Conjugate Gradient Method	2268 seconds
Gauss Elimination Method	1814 seconds

Consequently, although six places of decimals may not be necessary for acceptable engineering solutions, it is apparent that the conjugate gradient method was approximately 25% more expensive for this particular problem. In addition, during the analysis the intermediate residual force and displacement vectors were "lost" due to a computer malfunction after 30 iterations, and so the whole analysis had to be restarted from the beginning. In fact, the accidental erasure or loss of files was found to make the whole analysis scheme a very hazardous operation and could result in a considerable waste of computer time.

The square of the sum of the residual forces, which gives an indication of the 'out of equilibrium' of the solution and which was printed out after each iteration, is plotted in FIG. A2.2. It can be seen that apart from small perturbations, the value falls fairly rapidly after 30 iterations. At the final solution stage, i.e. after 80 iterations, the value is seen to be approximately 10^{-5}.

The second problem examined was the same Bousinesq structure which was discussed in 15.9. The one-quarter, 14 inch cube model of the elastic half-space, was meshed by using seven by seven by seven 8-noded elements and by using three by three by three 20-noded elements, as illustrated in FIG. A2.3. Again, this problem was ideal from the point of view of the storage space required as all the elements were identical in each of the 8 or the 20-noded element idealisations.

It was found that the solution to both finite element models had converged when the size of the residual had attained a value of approximately 10^{3}. As can be seen from FIG. A2.4, while this value was attained for the 8-noded element model after only 50 iterations, the 20-noded element model required a further 30 iterations. However, as each iteration for the 8-noded element model took 110 seconds to complete and that for the 20-noded element model only 36 seconds, the latter element mesh was the more efficient. Even so, as can be seen from FIG. A2.5, the vertical displacement distribution obtained from the

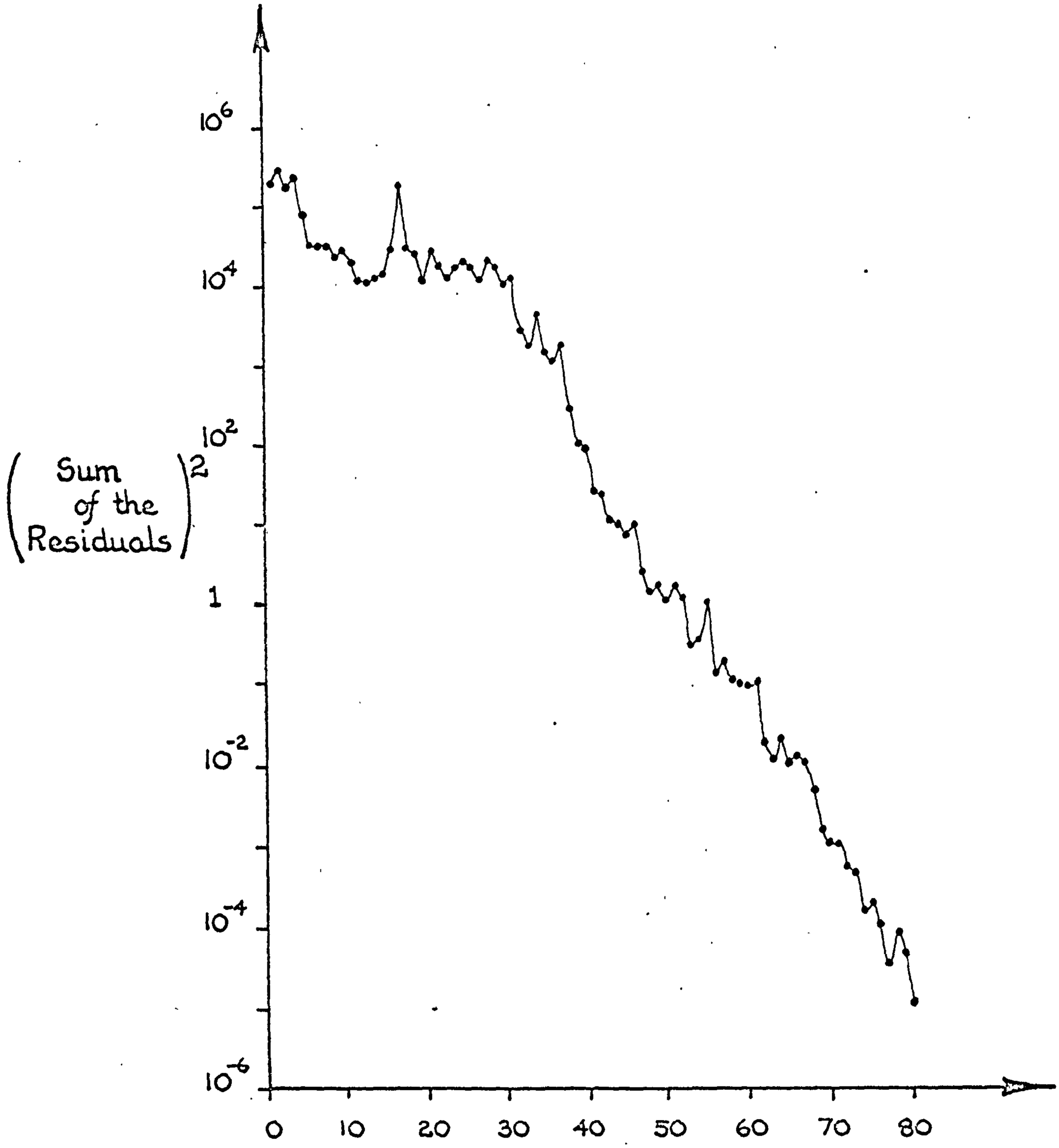
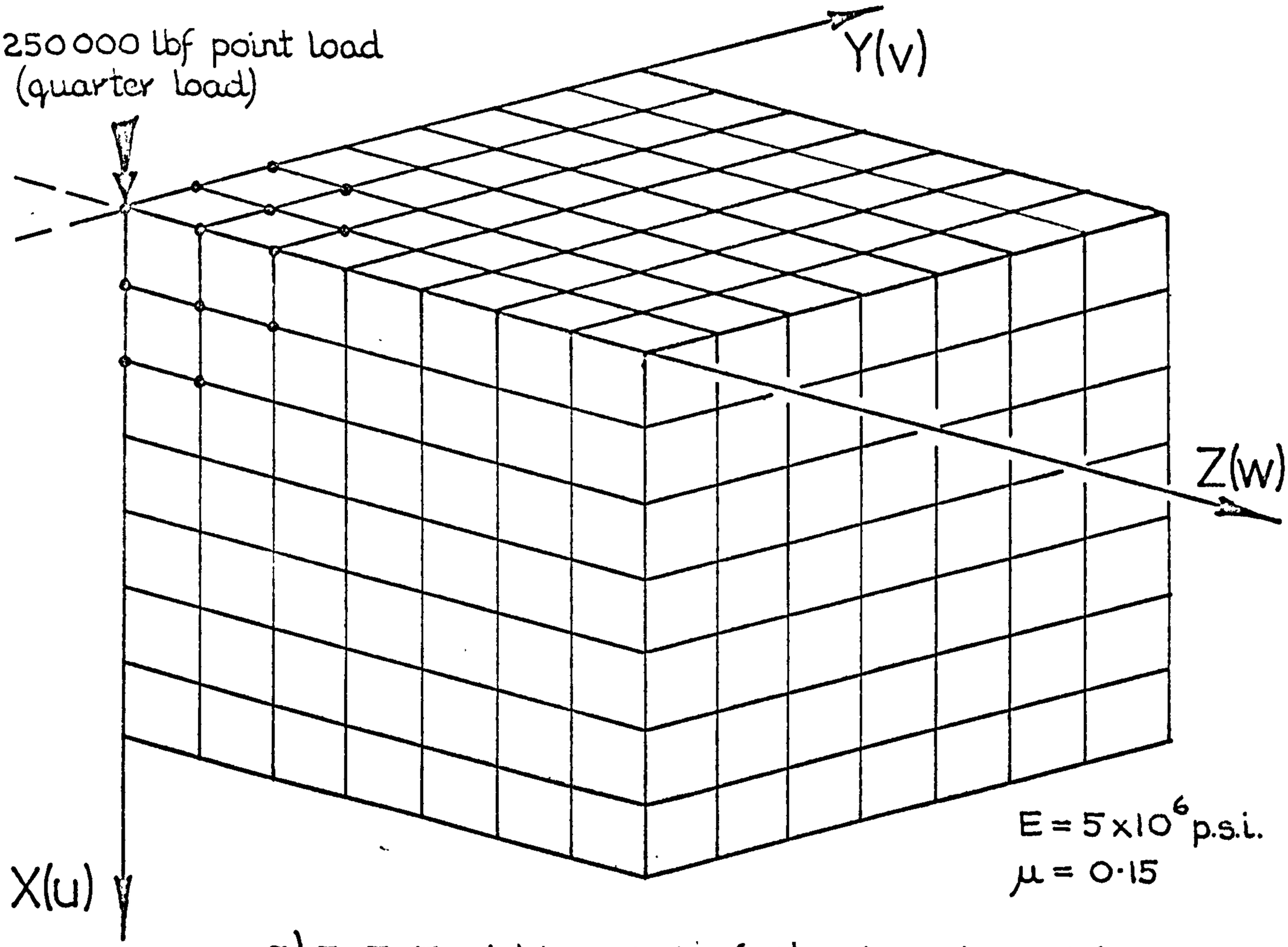
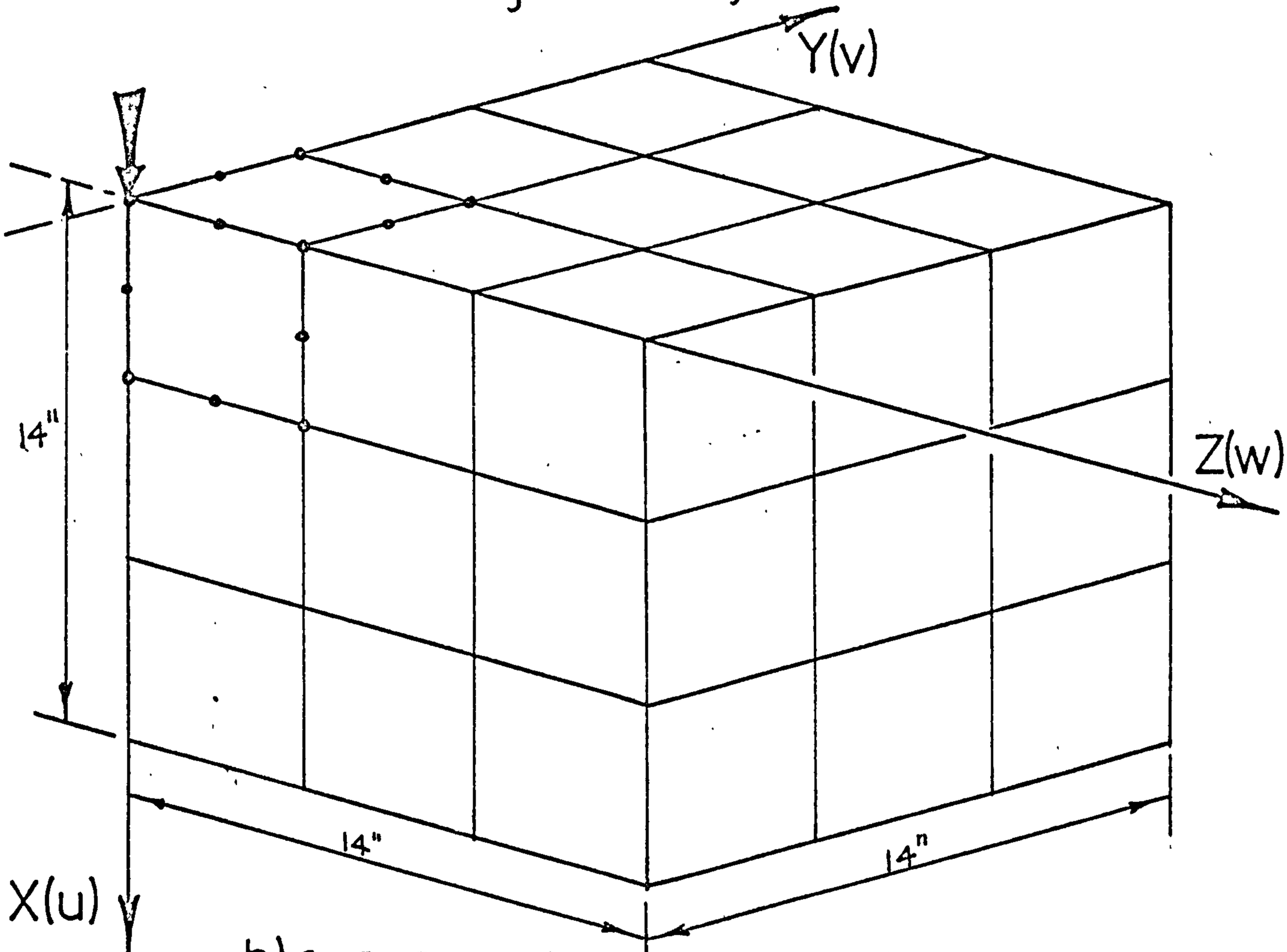


FIG. A2. 2 Plot showing how the conjugate gradient solution to the thick plate problem shown in FIG. 17.12 converges with the number of iterations completed.



a) 7x7x7 eight-noded finite element model.



b) 3x3x3 twenty-noded finite element model.

FIG. A2. 3 Three-dimensional finite element representations of the Bousinesq structure.

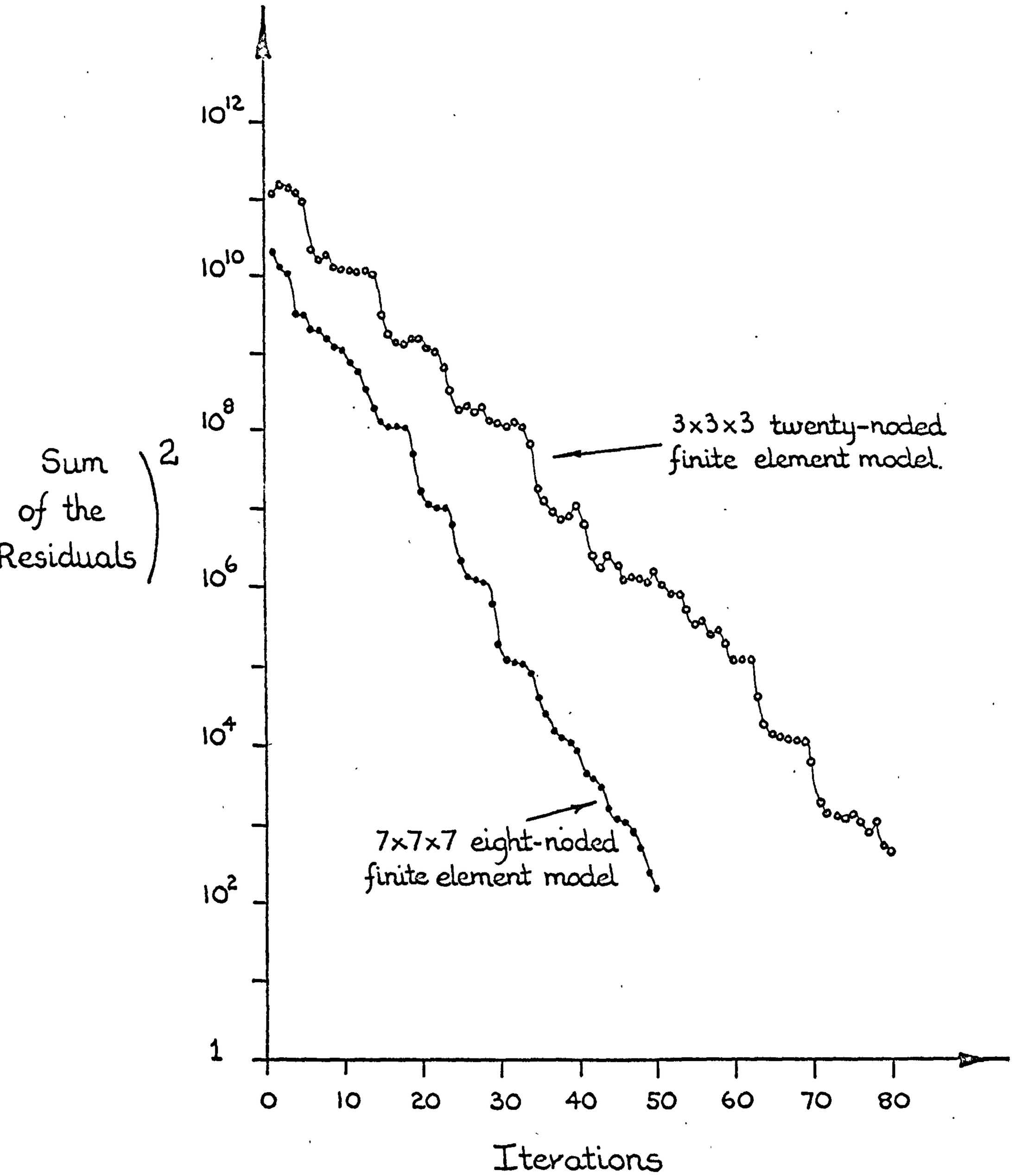


FIG. A2.4 Convergence of the solutions obtained for the Bousinesq models shown in FIG. A2.3.

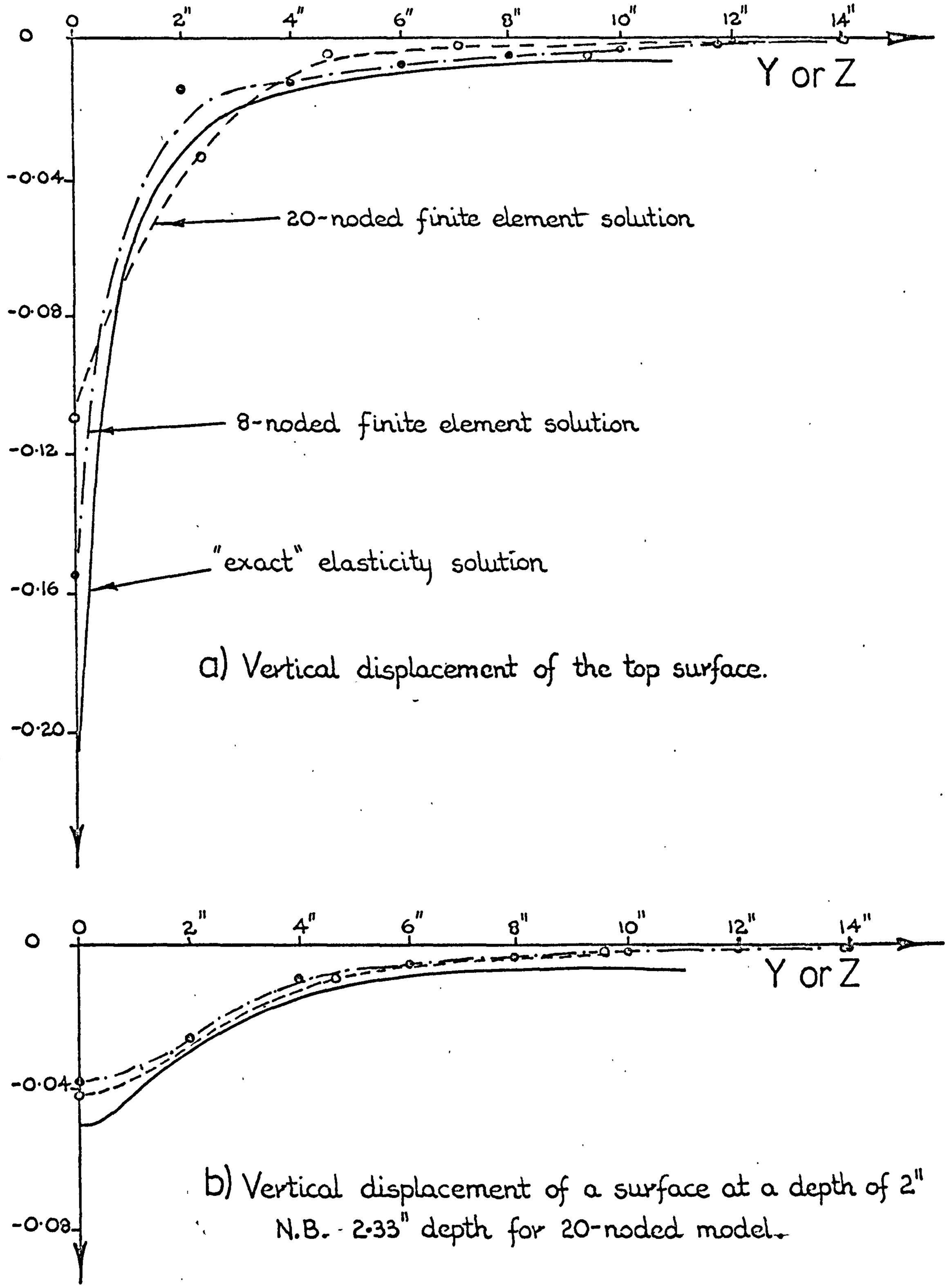


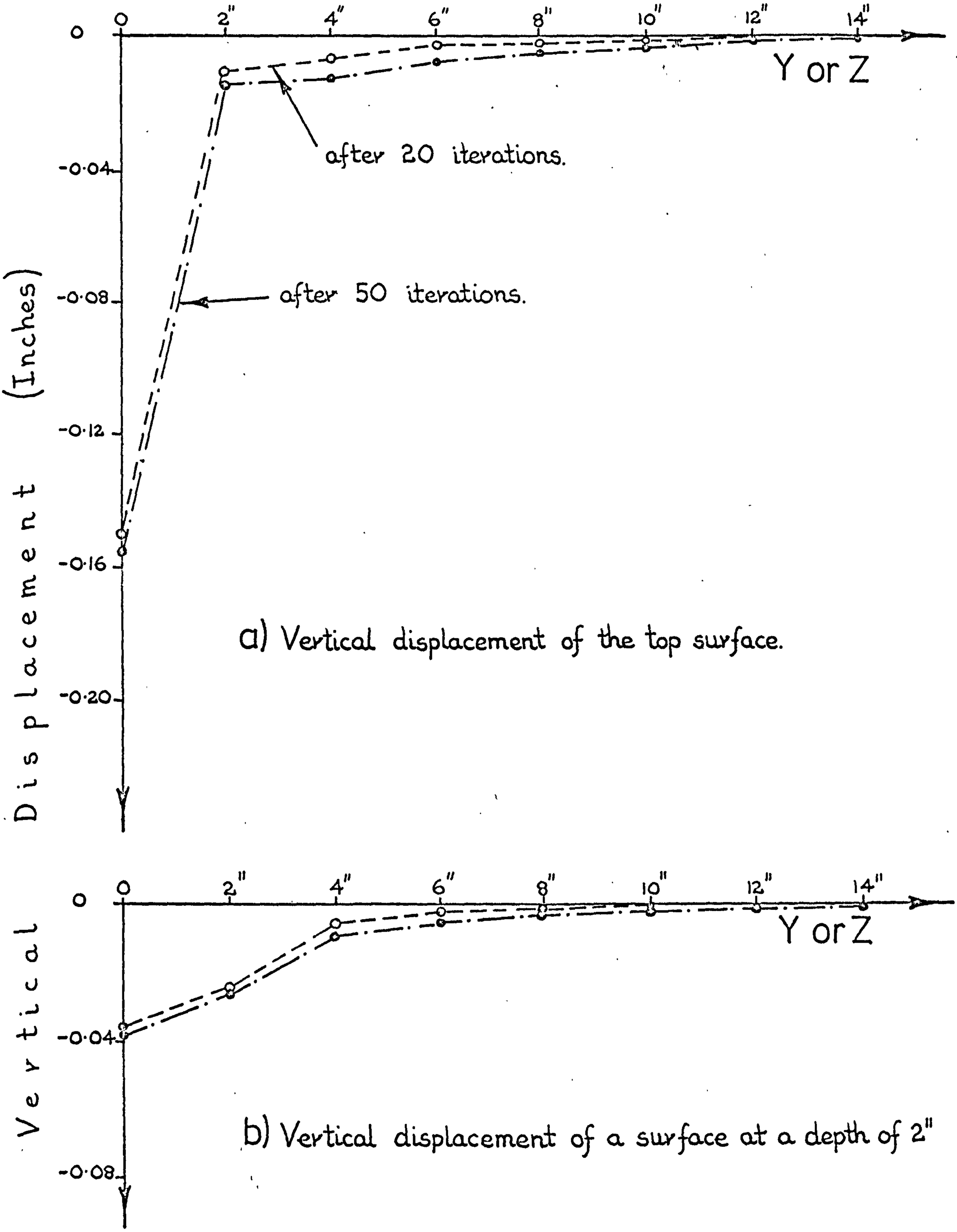
FIG. A2. 5 Comparison of the vertical displacement distributions obtained from the two 3-D finite element models shown in FIG. A2.3 and that of the "exact" elasticity solution. (138)

8-noded element idealisation followed slightly more closely the generally accepted "exact" solution. FIG.A2.6 shows how the displacement distribution obtained for the 8-noded element model converged to the final solution.

The equation solution time required (in seconds) for both finite element models using both solution methods is as follows:-

Model	Conjugate Gradients	Gauss Elimination (Solutions not Attempted)
7 x 7 x 7 Bousinesq 8-noded elements	5610	39252
3 x 3 x 3 Bousinesq 20-noded elements	2916	14269

In complete contrast to the thick plate problem, it is apparent from the above that the conjugate gradient solution method is far superior to that of the direct Gauss elimination method for the Bousinesq structure. Consequently, it is apparent that the number of iterations required for convergence varies greatly upon the conditioning of the structural equilibrium equations. In addition, it must be pointed out that the problems discussed above are not really a fair comparison between the conjugate gradient and elimination methods, since, the former represents essentially an 'in-core' solution while the latter is essentially an 'out-of-core' solution. For the



a) Vertical displacement of the top surface.

b) Vertical displacement of a surface at a depth of 2"

FIG. A2.6 Convergence of the displacement distributions for the 8-noded finite element model shown in FIG. A2.3.

conjugate gradient method applied to dental structures, in which all of the finite elements would probably be different, the element [k]s, for the size of the computer employed, would have to be stored during the solution process on disc and would therefore have to be retrieved during each iteration. It is in fact the reading and writing of data to the disc files which greatly escalates the computer time required to solve a particular problem. Indeed, with the Gauss elimination method an 'in-core' solution takes approximately only one third of the time it takes to solve the same system of equations stored on disc.

Although the direct Gauss elimination method was adopted for the work reported in this thesis, due to its greater predictability, it is evident that more effort should be devoted to developing the conjugate gradient method as this may well be a more profitable approach for certain types of problems. (For further discussions on the conjugate gradient method and a computer listing of the solution Algorithm, the interested reader is referred to Yettram and Hirst (132).).

APPENDIX THREE

THE AXISYMMETRIC FINITE ELEMENT ANALYSIS

PROGRAM LISTING

APPENDIX THREE

```
'LIST' (LP)
'PROGRAM' (AXITHERMDISC)
'INPUT' 1=CRO
'OUTPUT' 2=LPO
'EXTENDED'
'BEGIN'
'COMMENT'
A PROGRAM FOR AXISYMMETRIC STRESS ANALYSIS USING TRIANGULAR
SECTION ANNULAR-3-NODED 2-DEGREE OF FREEDOM PER NODE
LINEAR DISPLACEMENT FORMULATED FINITE ELEMENTS.
THERMAL EFFECTS ARE INCLUDED.  AVERAGE ELEMENT
CENTROIDAL TEMPERATURES ABOVE DATUM ARE +VE WHILE THOSE
BELOW ARE -VE.  THOSE AT DATUM TEMPERATURE (I.E. NO
THERMAL CONTRIBUTION) ARE ZERO.
THE MODIFIED FORM OF THE STRUCTURAL STIFFNESS MATRIX [MBK] IS
FORMED ON DISC BACKING FILE UDVKW2MBKAX1 AND THE NODAL FORCE
AND DISPLACEMENT VECTORS ARE HELD IN CORE. THE EQUATIONS ARE
SOLVED BY DIRECT GAUSS ELIMINATION IN BLOCK FORM.
NODAL DISPLACEMENTS AND ELEMENT STRESS AND STRAIN COMPONENTS
ARE OUTPUT.
THIS VERSION ALLOWS ZERO-VALUED DISPLACEMENTS ONLY.
MK. 5  5TH. JUNE 1972.
KEITH W. J. WRIGHT.
DEPT. MECH. ENG.
BRUNEL UNIVERSITY.
UXBRIDGE.;
'INTEGER'
NRB,NCB,
NCMBK,NRMBK,
          COUNTER,
     JOB,NELEM,NONOP,LO,CO,MANND,NOMAT;
'PROCEDURE' TIMENOW;
'EXTERNAL';
'PROCEDURE' COPYSTRING;
'BEGIN'
     'INTEGER' CS;
    L1:CS:=READCH;
     'IF' CS#CODE('('')') 'THEN' 'GOTO' L1;
    L2:CS:=READCH;
     'IF' CS#CODE('('')') 'THEN'
      'BEGIN'
          PRINTCH(CS);
          'GOTO' L2;
      'END';
  'END' OF COPYSTRING;
'PROCEDURE' USESTORE(N,S,T,G,L);
'VALUE' N,L,G;
'INTEGER' N,L,G;
'STRING' S,T;
'EXTERNAL';
'PROCEDURE' PUT PART(N,K,A,X,Y);
'VALUE' N;
```

```
'INTEGER' N,K;
'REAL' X,Y;
'ARRAY' A;
'EXTERNAL';
'PROCEDURE' GET PART(N,K,A,X,Y);
'VALUE' N;
'INTEGER' N,K;
'REAL' X,Y;
'ARRAY' A;
'EXTERNAL';
'PROCEDURE' MATMULT(A,B,C,X,Y,Z);
'VALUE' X,Y,Z;
'INTEGER' X,Y,Z;
'REAL' 'ARRAY'
A,B,C;
'COMMENT'
THIS PROCEDURE POST=MULTIPLIES THE MATRIX A(X*Y) BY THE
MATRIX B(Y*Z) AND PUTS THE PRODUCT IN C(X*Z);
'BEGIN'
'INTEGER'
I,J,K;
'FOR' I:=1 'STEP' 1 'UNTIL' X 'DO'
'FOR' K:=1 'STEP' 1 'UNTIL' Z 'DO'
'BEGIN'
C[I,K]:=0;
'FOR' J:=1 'STEP' 1 'UNTIL' Y 'DO'
C[I,K]:=C[I,K]+A[I,J]*B[J,K];
'END';
'END' OF MATMULT;
'PROCEDURE' MATRANMULT(G,H,F,X,Y,Z);
'VALUE' X,Y,Z;
'INTEGER' X,Y,Z;
'REAL' 'ARRAY' G,H,F;
'COMMENT'
THIS PROCEDURE PRE=MULTIPLIES THE MATRIX H(Y*Z) BY THE
TRANSPOSE OF A MATRIX G(Y*X)=N,B. THE TRANSPOSE
OF G IS CARRIED OUT DURING THE PROCEDURE SO THAT ONLY
THE MATRIX G IS REQUIRED.;
'BEGIN'
'INTEGER' I,J,K;
'FOR' I:=1 'STEP' 1 'UNTIL' X 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' Z 'DO'
'BEGIN'
F[I,J]:=0;
'FOR' K:=1 'STEP' 1 'UNTIL' Y 'DO'
F[I,J]:=F[I,J]+G[K,I]*H[K,J];
'END';
'END' OF MATRANMULT;
'PROCEDURE' MATINVERSE(A,N,INVERSEA);
'VALUE' N;
'REAL' 'ARRAY' A,INVERSEA;
'INTEGER' N;
'COMMENT'
THIS PROCEDURE INVERTS THE MATRIX [A] OF ORDER (N*N) AND
STORES THE RESULT IN INVERSEA. THE INVERSION PROCEDURE
UTILISES THE GAUSS ELIMINATION METHOD WHICH REDUCES THE MATRIX
```

```
[A] TO AN UPPER TRIANGULAR MATRIX, TO CONSERVE MAXIMUM
ACCURACY THE DIAGONAL PIVOTAL TERMS ARE RE-ARRANGED BY
INTERCHANGING THE ROWS - THIS BRINGS THE LARGER TERMS OF A
PARTICULAR COLUMN ONTO THE MAIN DIAGONAL AND HENCE ARE
SUBSEQUENTLY USED AS THE PIVOTAL TERMS.;
'BEGIN'
'REAL' 'ARRAY' B[1:N,1:2*N],X[1:N,1:N];
'REAL' PIVOT,TT;
'INTEGER' M,I,J,K;
M:=2*N;
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
'BEGIN'
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
B[I,J]:=A[I,J];
'FOR' J:=N+1 'STEP' 1 'UNTIL' M 'DO'
B[I,J]:='IF' I+N=J 'THEN' 1 'ELSE' 0
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
'BEGIN'
PIVOT:=B[I,I];
'FOR' J:=I+1 'STEP' 1 'UNTIL' N 'DO'
'IF' ABS(PIVOT)<ABS(B[I,J]) 'THEN'
'BEGIN'
'FOR' K:=1 'STEP' 1 'UNTIL' M 'DO'
'BEGIN'
TT:=B[I,K];
B[I,K]:=B[J,K];
B[J,K]:=TT;
'END';
PIVOT:=B[I,I];
'END';
'FOR' K:=M 'STEP' -1 'UNTIL' I 'DO'
B[I,K]:=B[I,K]/B[I,I];
'FOR' J:=I+1 'STEP' 1 'UNTIL' N 'DO'
'FOR' K:=M 'STEP' -1 'UNTIL' I 'DO'
B[J,K]:=B[J,K]-B[I,K]*B[J,I];
'END';
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
X[N,J]:=B[N,N+J];
'FOR' I:=N-1 'STEP' -1 'UNTIL' 1 'DO'
'BEGIN'
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
X[I,J]:=B[I,N+J];
'FOR' K:=N 'STEP' -1 'UNTIL' I+1 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
X[I,J]:=X[I,J]-B[I,K]*X[K,J];
'END';
'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
INVERSEA[I,J]:=X[I,J];
'END' OF PROCEDURE MATINVERSE;
'PROCEDURE' DBLOKGAUSS2(F,NCB,NRB);
'VALUE' NCB,NRB;
'INTEGER' NCB,NRB;
'REAL' 'ARRAY' F;
'COMMENT'
```

```
THIS PROCEDURE SOLVES THE SET OF LINEAR SIMULTANEOUS EQUATIONS
[MBK][D]=[F] USING THE GAUSS ELIMINATION METHOD IN BLOCK FORM,
THE MATRIX [MBK] IS PARTITIONED INTO 2 BY 2 SUB-MATRICES OR
BLOCKS AND IS A MODIFIED ARRANGEMENT OF A BANDED SYMMETRICAL
MATRIX OF ORDER 2*NRB ELEMENTS SQUARE,
THE SOLUTION [D] WHICH IS WRITTEN-OVER THE MATRIX
[F], IS OBTAINED BY EFFECTIVELY REDUCING THE ORIGINAL SYMMETRIC
MATRIX TO AN UPPER TRIANGULAR MATRIX AND THEN BACK SUBSTITUTING.
THESE PROCESSES ARE CARRIED OUT DIRECTLY ON THE MODIFIED FORM
OF THE SYMMETRIC MATRIX, WHICH IS STORED ON DISC BACKING STORE
IN BLOCK FORM IN THE FILE UDVKW2MBKAX1, THE BLOCKS ARE STORED
AS A STRING BEGINNING AT BLOCK NCB=1 OF NRB=1 TO
NCB=NCB OF NRB=1 ETC, RIGHT THROUGH TO NCB=NCB
OF NRB=NRB, THE FORCE MATRIX [F] IS STORED IN CORE.;
'BEGIN'
'INTEGER'
S,T,
P,PP,PW,
II,JJ,K,I,J;
'REAL'
BUG;
'INTEGER' 'ARRAY'
MAP[1:NCB];
'REAL' 'ARRAY'
PRMBK,WRMBK[1:2,1:2*NCB],FF1,FF2[1:2,1:1],
A,A2,A3,INVERSEA[1:2,1:2],DD[1:NCB,1:2,1:2];
'FOR' I:=1 'STEP' 1 'UNTIL' NRB 'DO'
'BEGIN'
P:=PP:=4*NCB*(I-1)+1;
GET PART(10,P,PRMBK,PRMBK[1,1],PRMBK[2,2*NCB]);
'FOR' II:=1 'STEP' 1 'UNTIL' 2 'DO'
'FOR' JJ:=1 'STEP' 1 'UNTIL' 2 'DO'
A[II,JJ]:=PRMBK[II,JJ];
MATINVERSE(A,2,INVERSEA);
      S:=2*I-2;
'FOR' II:= S+1 'STEP' 1 'UNTIL' S+2 'DO'
FF1[II-S,1]:=F[II,1];
MATMULT(INVERSEA,FF1,FF2,2,2,1);
'FOR' II:=S+1 'STEP' 1 'UNTIL' S+2 'DO'
F[II,1]:=FF2[II-S,1];
'FOR' K:=NCB 'STEP' -1 'UNTIL' 1 'DO'
'BEGIN'
      'IF' I-(NRB-NCB+1)+K>NCB 'THEN' 'GOTO' L1;
BUG:=0;
      S:=2*K-2;
'FOR' II:=1 'STEP' 1 'UNTIL' 2 'DO'
'FOR' JJ:=S+1 'STEP' 1 'UNTIL' S+2 'DO'
'BEGIN'
DD[K,II,JJ-S]:=A[II,JJ-S]:=PRMBK[II,JJ];
BUG:=BUG+PRMBK[II,JJ];
'END';
'IF' BUG=0 'THEN'
'BEGIN'
MAP[K]:=0;
'GOTO' L1;
'END';
```

```
MATMULT(INVERSEA,A,A2,2,2,2);
'FOR' II:=1 'STEP' 1 'UNTIL' 2 'DO'
'FOR' JJ:=S+1 'STEP' 1 'UNTIL' S+2 'DO'
PRMBK[II,JJ]:=A2[II,JJ-S];
MAP[K]:=1;
L1:'END';
'FOR' J:=1 'STEP' 1 'UNTIL' NCB-1 'DO'
'BEGIN'
'IF' I+J>NRB 'THEN' 'GOTO' V1;
'IF' MAP[J+1]=0 'THEN' 'GOTO' V1;
'FOR' II:=1 'STEP' 1 'UNTIL' 2 'DO'
'FOR' JJ:=1 'STEP' 1 'UNTIL' 2 'DO'
A3[JJ,II]:=DD[J+1,II,JJ];
P:=PW:=4*NCB*(I+J-1)+1;
GET PART(10,P,WRMBK,WRMBK[1,1],WRMBK[2,2*NCB]);
'FOR' K:=1 'STEP' 1 'UNTIL' NCB-J 'DO'
'BEGIN'
      S:=2*(J+K)-2;
'FOR' II:=1 'STEP' 1 'UNTIL' 2 'DO'
'FOR' JJ:=S+1 'STEP' 1 'UNTIL' S+2 'DO'
MATMULT(A3,A,A2,2,2,2);
      S:=2*K-2;
'FOR' II:=1 'STEP' 1 'UNTIL' 2 'DO'
'FOR' JJ:=S+1 'STEP' 1 'UNTIL' S+2 'DO'
A[II,JJ-S]:=PRMBK[II,JJ];
WRMBK[II,JJ]:=WRMBK[II,JJ]-A2[II,JJ-S];
'END';
P:=PW;
PUT PART(10,P,WRMBK,WRMBK[1,1],WRMBK[2,2*NCB]);
MATMULT(A3,FF2,FF1,2,2,1);
      S:=2*(1+J)-2;
'FOR' II:= S+1 'STEP' 1 'UNTIL' S+2 'DO'
F[II,1]:=F[II,1] - FF1[II-S,1];
V1:'END';
P:=PP;
PUT PART(10,P,PRMBK,PRMBK[1,1],PRMBK[2,2*NCB]);
'END' OF FORWARD ELIMINATION;
WRITE TEXT('('''('2C')'TIME%AT%END%OF%FORWARD%ELIMINATION')');
TIMENOW;
'FOR' I:=NRB-1 'STEP' -1 'UNTIL' 1 'DO'
'BEGIN'
P:=PW:=4*NCB*(I-1)+1;
GET PART(10,P,WRMBK,WRMBK[1,1],WRMBK[2,2*NCB]);
'FOR' J:=2 'STEP' 1 'UNTIL' NCB 'DO'
'BEGIN'
'IF' I+J-1>NRB 'THEN' 'GOTO' V2;
BUG:=0;
'FOR' II:=1 'STEP' 1 'UNTIL' 2 'DO'
'FOR' JJ:=2*J-1 'STEP' 1 'UNTIL' 2*J 'DO'
'BEGIN'
A[II,JJ-(2*J-2)]:=WRMBK[II,JJ];
BUG:=BUG+WRMBK[II,JJ]
'END';
'IF' BUG=0 'THEN' 'GOTO' V2;
      S:=2*(I+J-1)-2;
'FOR' II:=S+1 'STEP' 1 'UNTIL' S+2 'DO'
```

```
FF1[II-S,1]:=F[II,1];
MATMULT(A,FF1,FF2,2,2,1);
'FOR' II:=2*I-1 'STEP' 1 'UNTIL' 2*I 'DO'
F[II,1]:=F[II,1] - FF2[II-(2*I-2),1];
V2:'END';
'END' OF BACK SUBSTITUTION;
'END' OF PROCEDURE DBLOKGAUSS2.;
SELECT INPUT(1);
SELECT OUTPUT(2);
'COMMENT'
     THE JOB NO, AND GENERAL STRUCTURE DATA ARE READ IN.;
PAPERTHROW;
     JUB:=READ;
     WRITE TEXT('('''('2C')'JOB%NO.%%%')');
     PRINT(JOB,3,0);
     NEWLINE(1);
     COPYSTRING;
          NELEM:=READ;
          NONOP:=READ;
          LO:=READ;
          CO:=READ;
          MANND:=READ;
          NOMAT:=READ;
          NRMBK:=2*NONOP;
          NCMBK:=2*(MANND+1);
          NRB:=NONOP;
          NCB:=MANND+1;
     WRITE TEXT('('''('2C')'NUMBER%OF%ELEMENTS%%%')');
     PRINT(NELEM,3,0);
     WRITE TEXT('('''('2C')'NUMBER%OF%NODAL%POINTS%%%')');
     PRINT(NONOP,3,0);
     WRITE TEXT('('''('2C')'NUMBER%OF%APPLIED%LOADS%%%')');
     PRINT(LO,3,0);
     WRITE TEXT
     ('('''('2C')'NUMBER%OF%APPLIED%DISPLACEMENTS/RESTRAINTS%%')');
     PRINT(CO,3,0);
     WRITE TEXT
     ('('''('2C')'MAX.%NODE%NO.%DIFFERENCE%IN%ANY%ONE%ELEMENT%%')');
     PRINT(MANND,3,0);
     WRITE TEXT('('''('2C')'NO.%OF%MATERIAL%TYPES%%%')');
     PRINT(NOMAT,3,0);
     WRITE TEXT('('''('2C')'NO.%OF%ELEMENTS%REQIRED%FOR%[MBK]=%%%')');
     PRINT(NRMBK*NCMBK,0,6);
     PAPERTHROW;
     'BEGIN'
          'REAL' 'ARRAY'
          MAT[1:NOMAT,1:3],B[1:4,1:6],D[1:4,1:4],BD[1:6,1:4],
          ETF[1:6,1:1],ETS[1:4,1:1],
          STN1[1:4,1:1],
               TRI[1:6],COORD[1:NONOP,1:2],
          K[1:6,1:6],WMBK[1:2,1:2*NCB],
     F,X,N1,Q,STF[1:NRMBK,1:1],ELDIS[1:6,1:1],STN,STS[1:4,1:1];
          'REAL'
          RBAR,ZBAR,AI,BI,CI,AJ,BJ,CJ,AM,BM,CM,AREA,FACT,PREVELEM,
          EXCOF,TEMP1,TEMP2,EPSMAX,EPSMIN,GAMMAX,BETA1,BETA,
          SIGMAX,SIGMIN,TORMAX,ALPHA1,ALPHA,
```

```
            TERM);
      'INTEGER'
PREVMAT,
      K1,J1,II,JJ,KK1,KK2,KJ1,KJ2,
            I,J,NE,S,T,P,PMBK);
      'INTEGER' 'ARRAY'
      NON[1:NELEM,1:6];
'COMMENT'
      THE DATA REGARDING THE DIFFERENT MATERIALS OF THE ELEMENTS
      ARE NOW READ IN. THE SEQUENCE BEING E  MU  AND EXCOF, (THE
      COEFFICIENT OF THERMAL EXPANSION N. B. UNITS MUST BE
      COMPLEMENTARY TO DEGREES IN ELEMENT DATA.);
'FOR' I:=1 'STEP' 1 'UNTIL' NOMAT 'DO'
      'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
      MAT[I,J]:=READ;
WRITE TEXT('('''('2C')'MATERIAL%NO.'('11S')'E'('13S')'MU'('13S')'
EXCOF')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' NOMAT 'DO'
'BEGIN'
      NEWLINE(1);
      SPACE(5);
      PRINT(I,3,0);
      SPACE(8);
      PRINT(MAT[I,1],0,4);
      SPACE(2);
      PRINT(MAT[I,2],0,4);
      SPACE(2);
      PRINT(MAT[I,3],0,4);
'END';
'COMMENT'
      THE COORDINATES OF THE NODES OF THE STRUCTURE ARE READ IN,
            NODE NO. 1  R  (RADIAL)  Z  (AXIAL)
            NODE NO. 2  R  (RADIAL)  Z  (AXIAL) ETC.=;
'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
            COORD[I,J]:=READ;
PAPERTHROW;
WRITE TEXT
('('''('2C')''('6S')'NODE%NO.'('7S')'R%COORDINATE
'('3S')'Z%COORDINATE')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'BEGIN'
      NEWLINE(1);
      SPACE(8);
      PRINT(I,3,0);
      SPACE(6);
      'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
      PRINT(COORD[I,J],0,6);
'END';
'COMMENT'
      EACH ELEMENT NO. MATERIAL NO. TEMPERATURE NO. (NO. OF
      DEGREES ABOVE/BELOW DATUM AND NODE NOS. ARE READ IN.
      THE NODE NOS. MUST BE GIVEN IN AN ANTICLOCKWISE DIRECTION;
      'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
```

```
      'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
            NON[I,J]:=READ;
PAPERTHROW;
WRITE TEXT('('''('2C')'ELEMENT'('2S')'MATERIAL
%%TEMPERATURE%%%%ELEMENT%NODE%NUMBERS%(ANTICLOCKWISE%DIRECTION)
'('1C')'%%%%%NO.'('8S')'NO.'('8S')'NO.'('8S')'I
'('8S')'J'('8S')'K'('1C')''')');
'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
      NEWLINE(1);
      'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
      PRINT(NON[I,J],7,0);
'END';
'COMMENT'
THE DISC BACKING STORE FILE IS OPENED,UDVKW2MBKAX1.
THE WORKING AREA OF CORE USED TO FORM THE ROW-BLOCKS OF [MBK]
BEFORE TRANSFERRING TO BACKING FILE IS INITIALISED. THIS IS
THEN USED TO INITIALISE THE REQUIRED FILE AREA TO STORE [MBK];
USESTORE(10,'('ED')','('UDVKW2MBKAX1')',1,NRMBK*NCMBK);
'FOR' I:=1 'STEP' 1 'UNTIL' 2 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 2*NCB 'DO'
      WMBK[I,J]:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' NRB 'DO'
'BEGIN'
      P:=4*NCB*(I-1)+1;
      PUT PART(10,P,WMBK,WMBK[1,1],WMBK[2,2*NCB]);
'END' OF INITIALISING DISC FILE FOR [MBK].;
'FOR' I:=1 'STEP' 1 'UNTIL' NRMBK 'DO'
      STF[I,1]:=0;
'COMMENT'
      EACH ELEMENT STIFFNESS MATRIX IS NOW CALC.
      AND SUBSEQUENTLY DUMPED INTO [MBK].;
PREVMAT:=0;
WRITE TEXT('('''('2C')'ELEMENT%NO.')');
'FOR' NE:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
NEWLINE(1);
PRINT(NE,3,0);
'COMMENT'
      THE COORDINATES OF EACH ELEMENT ARE NOW SET UP;
            TRI[1]:=COORD[NON[NE,4],1];
            TRI[2]:=COORD[NON[NE,4],2];
           TRI[3]:=COORD[NON[NE,5],1];
            TRI[4]:=COORD[NON[NE,5],2];
            TRI[5]:=COORD[NON[NE,6],1];
            TRI[6]:=COORD[NON[NE,6],2];
'COMMENT'
      THE [B] MATRIX IS NOW SET UP.;
RBAR:=(TRI[1]+TRI[3]+TRI[5])/3;
ZBAR:=(TRI[2]+TRI[4]+TRI[6])/3;
AI:=TRI[3]*TRI[6]-TRI[5]*TRI[4];
BI:=TRI[4]-TRI[6];
CI:=TRI[5]-TRI[3];
AJ:=TRI[5]*TRI[2]-TRI[1]*TRI[6];
BJ:=TRI[6]-TRI[2];
CJ:=TRI[1]-TRI[5];
```

A3.1

```
AM:=TRI[1]*TRI[4]-TRI[3]*TRI[2];
BM:=TRI[2]-TRI[4];
CM:=TRI[3]-TRI[1];
'FOR' I:=1 'STEP' 1 'UNTIL' 4 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
            B[I,J]:=0;
B[1,2]:=B[4,1]:=CI;
B[1,4]:=B[4,3]:=CJ;
B[1,6]:=B[4,5]:=CM;
B[2,1]:=B[4,2]:=BI;
B[2,3]:=B[4,4]:=BJ;
B[2,5]:=B[4,6]:=BM;
B[3,1]:=AI/RBAR+BI+CI*ZBAR/RBAR;
B[3,3]:=AJ/RBAR+BJ+CJ*ZBAR/RBAR;
B[3,5]:=AM/RBAR+BM+CM*ZBAR/RBAR;
AREA:=ABS((TRI[1]*(TRI[4]-TRI[6])+TRI[3]*(TRI[6]-TRI[2])
+TRI[5]*(TRI[2]-TRI[4]))/2);
PRINT(AREA,0,6);
FACT:=1/(2*AREA);
'FOR' I:=1 'STEP' 1 'UNTIL' 4 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
            B[I,J]:=FACT*B[I,J];
'COMMENT'
      THE ELASTICITY MATRIX [D] IS FORMED FOR THE ELEMENT IF IT
      DIFFERS FROM THAT USED FOR THE PREVIOUS ELEMENT.;
      'IF' NON[NE,2] # PREVMAT 'THEN'
'BEGIN'
WRITE TEXT('('ELASTICITY%%%')');
      TERM:=MAT[NON[NE,2],1]/(1+MAT[NON[NE,2],2])
       *(1-MAT[NON[NE,2],2])/(1-2*MAT[NON[NE,2],2]);
     D[1,4]:=D[2,4]:=D[3,4]:=D[4,1]:=D[4,2]:=D[4,3]:=0;
      D[1,1]:=D[2,2]:=D[3,3]:=TERM;
      D[1,2]:=D[1,3]:=D[2,1]:=D[2,3]:=D[3,1]:=D[3,2]:=
      TERM*MAT[NON[NE,2],2]/(1-MAT[NON[NE,2],2]);
      D[4,4]:=TERM/2*(1-2*MAT[NON[NE,2],2])/(1-MAT[NON[NE,2],2]);
      PREVMAT:=NON[NE,2];
'END';
'COMMENT'
      THE PRODUCT [B]T[D] IS FORMED;
MATRANMULT(B,D,BD,6,4,4);
'COMMENT'
      THE PRODUCT [B]T[D][B] IS FORMED;
MATMULT(BD,B,K,6,4,6);
'COMMENT'
      FINALLY THE ELEMENT STIFFNESS MATRIX IS FORMED;
      FACT:=2*3.14159*RBAR*AREA;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
            K[I,J]:=K[I,J]*FACT;
'COMMENT'
      THE ELEMENTS OF [K] ARE NOW DUMPED INTO [MBK]
      IN THE DISC BACKING FILE UDVKW2MBKAX1.;
'FOR' I:=4 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN'
      P:=PMBK:=4*NCB*(NON[NE,I]-1)+1;
      GET PART(10,P,WMBK,WMBK[1,1],WMBK[2,2*NCB]);
```

A3.1

```
'FOR' J:=4 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN'
      'IF' NON[NE,I]>NON[NE,J] 'THEN' 'GOTO' NODUMP;
      K1:=2*(I-3)-1;
      J1:=2*(J-3)-1;
      KK1:=K1;
      KK2:=K1+1;
      KJ1:=J1+1;
      KJ2:=J1;
      T:=2*(NON[NE,J]-NON[NE,I]);
      WMBK[1,T+1]:=WMBK[1,T+1] + K[K1,J1];
      WMBK[1,T+2]:=WMBK[1,T+2] + K[KK1,KJ1];
      WMBK[2,T+1]:=WMBK[2,T+1] + K[KK2,KJ2];
      WMBK[2,T+2]:=WMBK[2,T+2] + K[KK2,KJ1];
   NODUMP:'END';
   P:=PMBK;
   PUT PART(10,P,WMBK,WMBK[1,1],WMBK[2,2*NCB]);
'END' OF DUMPING [K].;
WRITE TEXT('('DUMPED%0.K.')');
      'IF' NON[NE,3] # 0 'THEN'
'BEGIN'
      'COMMENT'
           THE ELEMENT THERMAL FORCE VECTOR IS NOW DERIVED AND
      SUBSEQUENTLY DUMPED INTO THE STRUCTURAL THERMAL FORCE
      VECTOR, FIRST THE INTEGRAL OF [B]T[D] MATRIX IS FORMED;
      'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
      'FOR' J:=1 'STEP' 1 'UNTIL' 4 'DO'
         BD[I,J]:=BD[I,J] * FACT;
      'COMMENT'
           THE ELEMENT THERMAL STRAIN VECTOR IS NOW FORMED;
     EXCOF:=MAT[NON[NE,2],3];
     ETS[1,1]:=EXCOF * NON[NE,3];
     ETS[2,1]:=ETS[3,1]:=ETS[1,1];
     ETS[4,1]:=0;
'COMMENT'
      THE ELEMENT THERMAL FORCE VECTOR IS OBTAINED;
MATMULT(BD,ETS,ETF,6,4,1);
'COMMENT'
      THE ELEMENT THERMAL FORCES ARE NOW DUMPED INTO
      THE STRUCTURAL THERMAL FORCE VECTOR.;
STF[2*NON[NE,4] - 1,1]:=STF[2*NON[NE,4] - 1,1] + ETF[1,1];
STF[2*NON[NE,4],1]:=STF[2*NON[NE,4],1] + ETF[2,1];
STF[2*NON[NE,5] - 1,1]:=STF[2*NON[NE,5] - 1,1] + ETF[3,1];
STF[2*NON[NE,5],1]:=STF[2*NON[NE,5],1] + ETF[4,1];
STF[2*NON[NE,6] - 1,1]:=STF[2*NON[NE,6] - 1,1] + ETF[5,1];
STF[2*NON[NE,6],1]:=STF[2*NON[NE,6],1] + ETF[6,1];
'END' OF CALC. ELEMENT THERMAL FORCE CONTRIBUTION;
'END' OF SETTING UP MOD. STRUCTURAL STIFF. MATRIX;
'COMMENT'
      THE FORCE AND DISPLACEMENT VECTORS ARE NOW FORMED.
THE VECTOR NUMBERING SEQUENCE IS--
      NODE NO. 1 R-DIRECTION (RADIAL) U-DISPLACEMENT.
      NODE NO. 1 Z-DIRECTION (AXIAL) V-DISPLACEMENT.
      NODE NO. 2 R- ETC.
THE NODAL FORCES HAVE THE VALUE OF-
      1) ITS PRESCRIBED VALUE
```

```
   OR 2) ZERO=(THIS IMPLIES EITHER NO APPLIED FORCE OR
            THAT OF A RESTRAINT).
NODAL DISPLACEMENTS CAN HAVE ONE OF THREE VALUES.
      1) ITS PRESCRIBED VALUE
   OR 2) ZERO=(IF IT IS FREE TO MOVE AND UNPRESCRIBED)
   OR 3) 0.000001=(IF IT IS RESTRAINED).
N.B. NO PRESCRIBED VALUES WITH THIS VERSION. 5/6/1972.
INITIALLY BOTH VECTORS ARE ZEROED AND THEN ONLY PRESCRIBED
FORCES AND THEN PRESCRIBED DISPLACEMENTS AND RESTRAINTS ARE
ACTUALLY READ IN,  THE SEQUENCE BEING IN BOTH CASES
      =EQUATION NO. FOLLOWED BY FORCE/DISPLACEMENT.
FINALLY THE COMPLETED DISPLACEMENT VECTOR IS SCANNED AND
IF THERE IS AN APPLIED DISPLACEMENT THE PROCEDURE BANDMULT
IS CALLED WHICH MODIFIES THE APPLIED FORCE VECTOR=AND
ZEROES THE APPROPRIATE TERMS OF [MBK] LEAVING A 1 ON THE
MAIN DIAGONAL TERM (I.E. IN THE FIRST COLUMN OF [MBK]).
IF THERE IS AN APPLIED RESTRAINT THEN AGAIN THE
NECESSARY ZEROEING IS CARRIED OUT TO [MBK].;
'FOR' I:=1 'STEP' 1 'UNTIL' NRMBK 'DO'
      F[I,1]:=X[I,1]:=N1[I,1]:=Q[I,1]:=0;
PAPERTHROW;
WRITE TEXT('('''('2C')'CO%%NO.'('6S')'EQUATION%NO.'('6S')'
APPLIED%DISPLACEMENTS%%(CONSTRAINT=0.000001)
'('1C')'''('37S')'(INCHES)')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' CO 'DO'
      'BEGIN'
            J:=READ;
            X[J,1]:=READ;
NEWLINE(1);
SPACE(3);
PRINT(I,3,0);
SPACE(6);
PRINT(J,4,0);
SPACE(13);
PRINT(X[J,1],0,6);
      'END' OF SETTING UP DISPLACEMENT VECTOR;
WRITE TEXT('('''('2C')'LO%%NO.'('6S')'EQUATION%NO.'('6S')'
APPLIED%FORCES'('1C')'''('34S')'LBF.')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' LO 'DO'
      'BEGIN'
            J:=READ;
            F[J,1]:=READ;
NEWLINE(1);
SPACE(3);
PRINT(I,3,0);
SPACE(6);
PRINT(J,4,0);
SPACE(13);
PRINT(F[J,1],0,6);
      'END' OF SETTING UP LOAD VECTOR;
'COMMENT'
      THE STRUCTURAL NODAL THERMAL FORCES ARE NOW ADDED TO
       THE ELASTIC FORCES.;
'FOR' I:=1 'STEP' 1 'UNTIL' NRMBK 'DO'
```

```
      F[I,1]:=F[I,1] + STF[I,1];
'FOR' I:=1 'STEP' 1 'UNTIL' NRB 'DO'
'BEGIN'
      P:=PMBK:=4*NCB*(I-1) + 1;
      GET PART(10,P,WMBK,WMBK[1,1],WMBK[2,2*NCB]);
      'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
      'BEGIN'
            'IF' X[2*(I-1)+J,1]#0 'THEN'
            'BEGIN'
                  'FOR' II:=1 'STEP' 1 'UNTIL' NCMBK 'DO'
                  WMBK[J,II]:=0;
                  'FOR' II:=1 'STEP' 1 'UNTIL' 2 'DO'
                  WMBK[II,J]:=0;
                  WMBK[J,J]:=1;
                  F[2*(I-1)+J,1]:=0;
      'END';
'END';
'FOR' J:=1 'STEP' 1 'UNTIL' NCB-1 'DO'
'BEGIN'
      'IF' I+J>NRB 'THEN' 'GOTO' NOROW;
      'FOR' II:=1 'STEP' 1 'UNTIL' 2 'DO'
      'BEGIN'
            'IF' X[2*(I+J-1)+II,1]#0 'THEN'
            'BEGIN'
                  'FOR' JJ:=1 'STEP' 1 'UNTIL' 2 'DO'
                  WMBK[JJ,2*J+II]:=0;
            'END';
      'END';
NOROW:'END';
P:=PMBK;
      PUT PART(10,P,WMBK,WMBK[1,1],WMBK[2,2*NCB]);
'END' OF MODS. TO [MBK] FOR CONSTRAINTS.;
WRITE TEXT
('('!('2C')'TIME%AT%COMMENCEMENT%OF%EQUATION%SOLUTION%%')');
TIMENOW;
DBLOKGAUSS2(F,NCB,NRB);
WRITE TEXT
('('!('2C')'TIME%AT%COMPLETION%OF%EQUATION%SOLUTION%%')');
TIMENOW;
'COMMENT'
      THE NODAL POINT DISPLACEMENTS AND FORCES ARE NOW OUTPUT;
PAPERTHROW;
 WRITE TEXT
('('!('2C')'%%NODE'('22S')'NODAL%%%DISPLACEMENTS
'('29S')'NODAL%%%FORCES'('1C')'%%%NO.'('27S')'(INCHES)
'('41S')'(LBF)'('1C')''('22S')'RADIAL'('20S')'AXIAL'('20S')'
RADIAL'('19S')'AXIAL'('1C')'
'('24S')'U'('24S')'V'('24S')'FR'('23S')'FZ'('1C')'')');
'FOR' I:=1 'STEP' 1'UNTIL' NONOP 'DO'
      'BEGIN'
      NEWLINE(1);
      SPACE(2);
      PRINT(I,3,0);
      SPACE(10);
      PRINT(F[2*I-1,1],0,6);
      SPACE(10);
```

```
      PRINT(F[2*I,1],0,6);
      SPACE(10);
      SPACE(10);
      'END';
'COMMENT'
      THE ELEMENT DISPLACEMENT VECTOR IS FORMED AND THE
      CENTROIDAL STRAIN AND STRESS VALUES CALC. BY REFORMING THE
      [B] AND [D] MATRICES.;
PAPERTHROW;
WRITE TEXT('('''('2C')'''('28S')'*****%%%ELEMENT%%%CENTROIDAL%%%
STRESS%%%COMPONENTS%%%%*****'('2C')'
ELEMENT%%%STRESS%Z%%%%%%STRESS%R%%%%STRESS%THETA%%%%STRESS%RZ
%%%%SIGMA%MAX%%%%%SIGMA%MIN%%%%%%%TOR%MAX%%%%%%%ALPHA
'('1C')'
%%NO,%%%%%STRAIN%Z%%%%%%STRAIN%R%%%%STRAIN%THETA%%%%STRAIN%RZ
%%%EPSILON%MAX%%%EPSILON%MIN%%%%%GAMMA%MAX%%%%%%%BETA'('1C')'')');
      COUNTER:=1;
PREVMAT:=0;
'FOR' NE:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
'IF' COUNTER=17 'THEN'
'BEGIN'
PAPERTHROW;
WRITE TEXT('('''('2C')'''('28S')'*****%%%ELEMENT%%%CENTROIDAL%%%
STRESS%%%COMPONENTS%%%%*****'('2C')'
ELEMENT%%%STRESS%Z%%%%%%STRESS%R%%%%STRESS%THETA%%%%STRESS%RZ
%%%%SIGMA%MAX%%%%%SIGMA%MIN%%%%%%%TOR%MAX%%%%%%%ALPHA
'('1C')'
%%NO,%%%%%STRAIN%Z%%%%%%STRAIN%R%%%%STRAIN%THETA%%%%STRAIN%RZ
%%%EPSILON%MAX%%%EPSILON%MIN%%%%%GAMMA%MAX%%%%%%%BETA'('1C')'')');
      COUNTER:=1;
'END';
'COMMENT'
EACH ELEMENTS COORDINATES AND DISPLACEMENT VECTORS ARE SET UP;
          TRI[1]:=COORD[NON[NE,4],1];
          TRI[2]:=COORD[NON[NE,4],2];
          TRI[3]:=COORD[NON[NE,5],1];
          TRI[4]:=COORD[NON[NE,5],2];
          TRI[5]:=COORD[NON[NE,6],1];
          TRI[6]:=COORD[NON[NE,6],2];
ELDIS[1,1]:=F[2*NON[NE,4] - 1,1];
ELDIS[2,1]:=F[2*NON[NE,4],1];
ELDIS[3,1]:=F[2*NON[NE,5] - 1,1];
ELDIS[4,1]:=F[2*NON[NE,5],1];
ELDIS[5,1]:=F[2*NON[NE,6] - 1,1];
ELDIS[6,1]:=F[2*NON[NE,6],1];
'COMMENT'
      THE [B] MATRIX IS NOW SET UP.;


      **INSERT A3.1 AS ABOVE**


'COMMENT'
```

```
      THE ELASTICITY MATRIX [D] IS FORMED FOR THE ELEMENT IF IT
      DIFFERS FROM THAT USED FOR THE PREVIOUS ELEMENT.;
      'IF' NON[NE,2] ≠ PREVMAT 'THEN'
'BEGIN'
      TERM:=MAT[NON[NE,2],1]/(1+MAT[NON[NE,2],2])
       *(1-MAT[NON[NE,2],2])/(1-2*MAT[NON[NE,2],2]);
     D[1,4]:=D[2,4]:=D[3,4]:=D[4,1]:=D[4,2]:=D[4,3]:=0;
      D[1,1]:=D[2,2]:=D[3,3]:=TERM;
      D[1,2]:=D[1,3]:=D[2,1]:=D[2,3]:=D[3,1]:=D[3,2]:=
      TERM*MAT[NON[NE,2],2]/(1-MAT[NON[NE,2],2]);
      D[4,4]:=TERM/2*(1-2*MAT[NON[NE,2],2])/(1-MAT[NON[NE,2],2]);
      PREVMAT:=NON[NE,2];
'END';
'COMMENT'
      THE ELEMENT STRAINS ARE CALC. BY POST MULTIPLYING [B] BY THE
      ELEMENT NODAL DISPLACEMENTS.
      FROM THESE THE PRINCIPAL STRAINS ARE EVALUATED.
       I.E. EPSMAX,EPSMIN, AND THE MAX SHEAR STRAIN GAMMAX.
        GAMMAX IS -VE IF STRAIN R > STRAIN Z.
        BETA IS THE ANGLE MEASURED FROM THE HORIZONTAL R AXIS TO
        THE PLANE OF EPSMAX. THE ANGLE IF +VE IS MEASURED
        ANTICLOCKWISE AND IF -VE CLOCKWISE.;
MATMULT(B,ELDIS,STN,4,6,1);
TEMP1:=SQRT((STN[2,1] - STN[1,1])↑2 + STN[4,1]↑2);
TEMP2:=(STN[1,1] + STN[2,1])/2;
EPSMAX:=TEMP2 + TEMP1/2;
EPSMIN:=TEMP2 - TEMP1/2;
GAMMAX:=TEMP1;
'IF' STN[2,1] > STN[1,1] 'THEN' GAMMAX:=-GAMMAX;
BETA1:=90*ARCTAN(STN[4,1]/(STN[2,1] - STN[1,1]))/3.14159265;
'IF' STN[4,1]/(STN[2,1] - STN[1,1]) > 0 'THEN'
      'BEGIN'
       'IF' STN[4,1] > 0 'THEN'
           BETA:=BETA1
        'ELSE'
           BETA:=BETA1 + 90;
      'END'
      'ELSE'
      'BEGIN'
        'IF' STN[4,1] > 0 'THEN'
           BETA:=BETA1 - 90
        'ELSE'
           BETA:=BETA1;
      'END';
'COMMENT'
      FINALLY THE ELEMENT STRESS COMPONENTS ARE CALC.
BY SUBTRACTING THE ELEMENT THERMAL STRAIN COMPONENTS
FROM THE ELEMENT TOTAL STRAIN COMPONENTS AND PRE-
MULTIPLYING THEM BY THE ELASTICITY [D] MATRIX.
FROM THESE THE PRINCIPAL STRESSES ARE EVALUATED
  I.E. SIGMAX,SIGMIN,AND THE MAX. SHEAR STRESS TORMAX.
  TORMAX IS -VE IF STRESS R > STRESS Z.
  ALPHA IS THE ANGLE MEASURED FROM THE HORIZONTAL R AXIS TO
  THE PLANE OF SIGMAX. THE ANGLE IF +VE IS MEASURED
  ANTICLOCKWISE AND IF -VE CLOCKWISE.;
  EXCOF:=MAT[NON[NE,2],3];
```

```
        ETS[1,1]:=EXCOF * NON[NE,3];
        ETS[2,1]:=ETS[3,1]:=ETS[1,1];
        ETS[4,1]:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' 4 'DO'
      STN1[I,1]:=STN[I,1] - ETS[I,1];
MATMULT(D,STN1,STS,4,4,1);
TEMP1:=SQRT((STS[2,1] - STS[1,1])↑2 + 4*STS[4,1]↑2);
TEMP2:=(STS[1,1] + STS[2,1])/2;
SIGMAX:=TEMP2 + TEMP1/2;
SIGMIN:=TEMP2 - TEMP1/2;
TORMAX:=TEMP1/2;
'IF' STS[2,1] > STS[1,1] 'THEN' TORMAX:=-TORMAX;
ALPHA1:=90*ARCTAN(2*STS[4,1]/(STS[2,1] - STS[1,1]))/3.14159265;
'IF' 2*STS[4,1]/(STS[2,1] - STS[1,1]) > 0 'THEN'
      'BEGIN'
        'IF' 2*STS[4,1] > 0 'THEN'
           ALPHA:=ALPHA1
        'ELSE'
           ALPHA:=ALPHA1 + 90;
      'END'
      'ELSE'
      'BEGIN'
        'IF' 2*STS[4,1] > 0 'THEN'
           ALPHA:= ALPHA1 - 90
        'ELSE'
           ALPHA:=ALPHA1;
      'END';
NEWLINE(2);
PRINT(NE,3,0);
SPACE(2);
'FOR' I:=1 'STEP' 1 'UNTIL' 4 'DO'
      PRINT(STS[I,1],0,5);
PRINT(SIGMAX,0,5);
PRINT(SIGMIN,0,5);
PRINT(TORMAX,0,5);
SPACE(1);
PRINT(ALPHA,3,1);
NEWLINE(1);
SPACE(8);
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'
PRINT(STN[I,1],0,5);
PRINT(STN[4,1]/2,0,5);
PRINT(EPSMAX,0,5);
PRINT(EPSMIN,0,5);
PRINT(GAMMAX,0,5);
SPACE(1);
PRINT(BETA,3,1);
COUNTER:=COUNTER + 1;
'END' OF STRESS AND STRAIN CALC. & OUTPUT.;
'END';
'END' OF (AXITHERMDISC) WK. BEG. 5/6/1972.;
```

APPENDIX FOUR

THE AXISYMMETRIC FINITE ELEMENT ANALYSIS DATA CHECK PLOT PROGRAM LISTING

APPENDIX FOUR

```
'LIST' (LP)
'LIBRARY' (ED,SUBGROUPSRA3)
'LIBRARY' (ED,SUBGROUPSRGP)
'LIBRARY' (ED,SUBGROUPS-RS)
'PROGRAM' (PLOTMESHAX3T)
'COMPACT DATA'
'MIXED SEGMENTS'
'INPUT' 1=CRO
'OUTPUT' 2=LPO
'TRACE' 2
'BEGIN'
      'COMMENT'
      A PROGRAM FOR PLOTTING INDIRECTLY THE MESH PATTERN OF A
      STRUCTURE TO BE ANALYSED USING K. WRIGHTS AXISYMMETRIC
      FINITE ELEMENT ANALYSIS PROGRAM. THIS PROGRAM USES
      3-NODED-2-DEGREE OF FREEDOM PER NODE LINEAR
      DISPLACEMENT FORMULATED TRIANGULARLY SHAPED
      ANNULI TYPE ELEMENTS.
      KEITH W.J. WRIGHT,
      DEPT. MECH. ENG.,
      BRUNEL UNIVERSITY,
      UXBRIDGE.;
      'INTEGER'
            NELEM,NONOP,I,J,NE;
      'REAL'
            SCALER,X1,Y1,X2,Y2,X3,Y3,X4,Y4;
      'REAL' 'ARRAY'
      DUMMY,MTFNAM,PICNAM[1:5],
      TITLE,XAXIS,YAXIS[1:5];
      'INTEGER' 'PROCEDURE' INSTRARR(S,A);
      'STRING' S;
      'REAL' 'ARRAY' A;
      'EXTERNAL';
'PROCEDURE' HGPTAPE(L,BCD,IS,IG,IR);
'INTEGER' L,IS,IG,IR;
'REAL' 'ARRAY' BCD;
'EXTERNAL';
      'PROCEDURE' HGPLOT(X,Y,IC,L);
      'REAL' X,Y;
      'INTEGER' IC,L;
      'EXTERNAL';
      'PROCEDURE' HGpAXISV(X,Y,BCD,NC,S,THETA,XMIN,DX,GAp,NH);
      'REAL' X,Y,S,THETA,XMIN,DX,GAP;
      'INTEGER' NC,NH;
      'REAL' 'ARRAY' BCD;
      'EXTERNAL';
      'PROCEDURE' HGPSYMBL(X,Y,HEIGHT,BCD,THETA,N);
      'REAL' X,Y,HEIGHT,THETA;
      'INTEGER' N;
      'REAL' 'ARRAY' BCD;
      'EXTERNAL';
      SELECT INPUT(1);
```

```
SELECT OUTPUT(2);
'COMMENT'
THE FOLLOWING ARE NOW READ IN,
     SPACES MAY BE USED TO MAKE UP THE NO, OF CHARACTERS,
     SPACES MAY BE USED TO MAKE UP THE NO, OF CHARACTERS,
ALL CHARACTER STRINGS SHOULD BE ON ONE CARD
N.B, XAXIS,YAXIS ETC, MUST BE TERMINATED BY*
MTFNAM=STRING OF 12 CHARACTERS NAMING MAG TAPE FILE
PICNAM=STRING OF 12 CHARACTERS NAMING THE PIC TO BE PLOTTED
XAXIS=STRING OF 20 CHARACTERS FORMING R-AXIS LABEL.
YAXIS=STRING OF 20 CHARACTERS FORMING Z-AXIS LABEL.
     SCALER=THE QUANTITY WHICH SCALES THE COORDINATES
     TO FIT THE PLOTTER PAPER,
          NELEM=THE NO. OF ELEMENTS IN THE STRUCTURE.
          NONOP=THE NO. OF NODES IN THE STRUCTURE.;
INSTRARR('('*')',MTFNAM);
INSTRARR('('*')',PICNAM);
     INSTRARR('('*')',XAXIS);
     INSTRARR('('*')',YAXIS);
     SCALER:=READ;
     NELEM:=READ;
     NONOP:=READ;
     'BEGIN'
          'REAL' 'ARRAY'
               COORD[1:NONOP,1:2];
          'INTEGER' 'ARRAY'
     NON[1:NELEM,1:6];
'COMMENT'
     THE COORDINATES OF THE NODES OF THE STRUCTURE ARE READ IN,
          NODE NO 1 R(RADIAL) COORDINATE   Z(AXIAL) COORDINATE
          NODE NO 2 R(RADIAL) COORDINATE   Z(AXIAL) COORDINATE--;
'FUR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
          COORD[I,J]:=READ;
PAPERTHROW;
WRITE TEXT
('('''('2C')''('6S')'NODE%NO.'('7S')'R%COORDINATE'('3S')'
Z%COORDINATE')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'BEGIN'
NEWLINE(1);
SPACE(8);
PRINT(I,3,0);
SPACE(8);
'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
'BEGIN'
PRINT(COORD[I,J],1,5);
SPACE(5);
'END';
'END';
'COMMENT'
     EACH ELEMENT NO., MATERIAL NO., TEMPERATURE NO.,(NO. OF
     DEGREES ABOVE/BELOW DATUM) AND NODE NOS. ARE NOW READ IN.
     THE NODE NOS. MUST BE GIVEN IN AN ANTICLOCKWISE DIRECTION;
     'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
```

```
      'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
            NON[I,J]:=READ;
PAPERTHROW;
WRITE TEXT('('''('2C')'ELEMENT'('2S')'MATERIAL
%%TEMPERATURE%%%%ELEMENT%NODE%NUMBERS%(ANTICLOCKWISE%DIRECTION)
'('1C')'%%%%%NO.'('8S')'NO.'('8S')'NO.'('8S')'I
'('8S')'J'('8S')'K'('1C')''')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
NEWLINE(1);
      'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
PRINT(NON[I,J],7,0);
'END';
            'COMMENT'
            THE COORDINATES ARE NOW SCALED READY FOR PLOTTING,
            BY MULTIPLYING THEM BY SCALER.;
            'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
            'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
            COORD[I,J]:=COORD[I,J]*SCALER;
'COMMENT'
      A MAG TAPE FILE IS PICKED UP AND NAMED;
      HGPTAPE(0,MTFNAM,0,0,0);
            'COMMENT'
            THE PLOTTER IS NOW INITIALISED.;
HGPLOT(0.0,0.0,0,15,1);
'COMMENT'
      THE SERIAL NO. AND PICTURE NAME ARE WRITTEN ON THE
      MAG. TAPE FILE.;
      HGPTAPE(1,PICNAM,0,0,0);
            'COMMENT'
            THE ORIGIN IS SET AT Y VALUE OF 26 CM.
            AND X VALUE OF 10 CM;
      HGPLOT(10.0,26.0,0,4);
            'COMMENT'
            THE PLOTTER X AND Y AXES,(CORRESPONDING TO THE R AND Z
            PROBLEM AXES), ARE NOW DRAWN AND LABELLED.;
HGPAXISV(0.0,0.0,XAXIS,-25,6,0.0,0.0,0.2,2.032,4);
HGPAXISV(0.0,0.0,YAXIS,25,17,90.0,0.0,0.2,2.032,4);
            'COMMENT'
            THE PLOT TITLE IS DRAWN ON THE GRAPH.;
            'COMMENT'
            EACH ELEMENTS NODAL COORDINATES ARE NOW SET
            UP AND THE ELEMENT SUBSEQUENTLY DRAWN.;
'FOR' NE:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
X1:=COORD[NON[NE,4],1];
Y1:=COORD[NON[NE,4],2];
X2:=COORD[NON[NE,5],1];
Y2:=COORD[NON[NE,5],2];
X3:=COORD[NON[NE,6],1];
Y3:=COORD[NON[NE,6],2];
      HGPLOT(X1,Y1,3,0);
HGPLOT(X2,Y2,2,0);
      HGPLOT(X3,Y3,2,0);
      HGPLOT(X1,Y1,2,0);
```

```
            'END';
            'COMMENT'
            THE PLOTTER BUFFER IS EMPTIED.;
            HGPLOT(0,0,0.0,0,2);
'COMMENT'
      THE MAG TAPE FILE IS NOW CLOSED;
      HGPTAPE(2,DUMMY,0,0,0);
      'END';
'END' OF (PLOTMESHAX3T) WK. BEG.29/1/73;
```

APPENDIX FIVE

THE PLANE STRESS FINITE ELEMENT ANALYSIS

PROGRAM LISTING

APPENDIX FIVE

```
'LIST' (LP)
'PROGRAM' (PLANESTRDISC)
'INPUT' 1=CR0
'OUTPUT' 2=LP0
'EXTENDED'
'BEGIN'
'COMMENT'
      A PROGRAM FOR PLANE-STRESS ANALYSIS USING
      RECTANGULAR 4-NODED-2-DEGREE OF FREEDOM PER NODE LINEAR
      DISPLACEMENT FORMULATED FINITE ELEMENTS.
      [3 BY 3 GAUSS INTEGRATING POINTS].
THE MODIFIED FORM OF THE STRUCTURAL STIFFNESS MATRIX [MBK] IS
FORMED ON DISC BACKING STORE AND THE NODAL FORCE AND DISPLACEMENT
VECTORS ARE HELD IN CORE. THE EQUATIONS ARE SOLVED BY DIRECT
GAUSS ELIMINATION. NODAL DISPLACEMENTS AND ELEMENT STRESS AND
STRAIN COMPONENTS ARE OUTPUT.
VERSION ALLOWS ZERO-VALUED DISPLACEMENTS(CONSTRAINTS) ONLY.
MK. 2  ALLOWS FOR ELEMENT NODAL FORCE OUTPUT REQUEST. 3/4/73.
KEITH W. J. WRIGHT,
DEPT. MECH. ENG.
BRUNEL UNIVERSITY,
UXBRIDGE.;
'INTEGER'
NELNF,
NOJOBS,NOJ,
NCB,NRB,
NOTHICK,
NCMBK,NRMBK,
      JOB,NELEM,NONOP,LO,CO,MANND,NOMAT;
'PROCEDURE' TIMENOW;
'EXTERNAL';
'PROCEDURE' COPYSTRING;

**SEE APPENDIX THREE FOR COMPLETE LISTING**

  'END' OF COPYSTRING;
'PROCEDURE' USESTORE(N,S,T,G,L);

**SEE APPENDIX THREE FOR COMPLETE LISTING**

'EXTERNAL';
'PROCEDURE' PUT PART(N,K,A,X,Y);

**SEE APPENDIX THREE FOR COMPLETE LISTING**

'EXTERNAL';
'PROCEDURE' GET PART(N,K,A,X,Y);

**SEE APPENDIX THREE FOR COMPLETE LISTING**

'EXTERNAL';
'PROCEDURE' MATINVERSE(A,N,INVERSEA);
```

```
**SEE APPENDIX THREE FOR COMPLETE LISTING**

'END' OF PROCEDURE MATINVERSE;
'PROCEDURE' MATMULT(A,B,C,X,Y,Z);

**SEE APPENDIX THREE FOR COMPLETE LISTING**

'END' OF MATMULT;
'PROCEDURE' MATRANMULT(G,H,F,X,Y,Z);

**SEE APPENDIX THREE FOR COMPLETE LISTING**

'END' OF MATRANMULT;
'PROCEDURE' DBLOKGAUSS2(F,NCB,NRB);

**SEE APPENDIx THREE FOR COMPLETE LISTING**

'END' OF PROCEDURE DBLOKGAUSS2,;
SELECT INPUT(1);
SELECT OUTPUT(2);
'COMMENT'
      THE JOB NO, AND GENERAL STRUCTURE DATA ARE READ IN.;
PAPERTHROW;
      JUB:=READ;
WRITE TEXT('('!('2C')'JOB%NO.%%%')');
PRINT(JOB,3,0);
      NEWLINE(1);
COPYSTRING;
      NELEM:=READ;
      NONOP:=READ;
      LO:=READ;
      CO:=READ;
      MANND:=READ;
      NOMAT:=READ;
      NOTHICK:=READ;
      NELNF:=READ;
      NRMBK:=2*NONOP;
      NCMBK:=2*(MANND+1);
      NRB:=NONOP;
NCB:=MANND+1;
WRITE TEXT('('!('2C')'NUMBER%OF%ELEMENTS%%%')');
PRINT(NELEM,3,0);
WRITE TEXT('('!('2C')'NUMBER%OF%NODAL%POINTS%%%')');
PRINT(NONOP,3,0);
WRITE TEXT('('!('2C')'NUMBER%OF%APPLIED%LOADS%%%')');
PRINT(LO,3,0);
WRITE TEXT
('('!('2C')'NUMBER%OF%APPLIED%DISPLACEMENTS/RESTRAINTS%%')');
PRINT(CO,3,0);
WRITE TEXT
('('!('2C')'MAX.%NODE%NO.%DIFFERENCE%IN%ANY%ONE%ELEMENT%%')');
PRINT(MANND,3,0);
WRITE TEXT('('!('2C')'NO.%OF%MATERIAL%TYPES%%%')');
PRINT(NOMAT,3,0);
WRITE TEXT('('!('2C')'NO.%OF%MATERIAL%THICKNESSES%%%')');
```

```
PRINT(NOTHICK,3,0);
WRITE TEXT('('''('2C')'NO.%OF%ELEMENTS%WHERE%ELEMENT%NODAL%FORCES%
ARE%REQUIRED%%%')');
PRINT(NELNF,3,0);
WRITE TEXT('('''('2C')'NO.%OF%ELEMENTS%REQIRED%FOR%[MBK]=%%%')');
PRINT(NRMBK*NCMBK,0,6);
PAPERTHROW;
'BEGIN'
      'REAL' 'ARRAY'
ELNF[1:8,1:1],
            D1[1:3,1:3],
TR,COF,INVT,DTR[1:3,1:3],
      ELDIS[1:8,1:1],STN,STS[1:3,1:1],
MAT[1:NOMAT,1:5],ELTHICK[1:NOTHICK],
COORD[1:NONOP,1:2],WMBK[1:2,1:2*NCB],
HH,XI1,ETA1[1:3],ELCO[1:4,1:2],
LAM[1:2,1:4],JAY,INVERSEJ[1:2,1:2],
D[1:3,1:3],INVJLAM[1:2,1:4],
B,DB[1:3,1:8],SUBK,K[1:8,1:8],
F,X,N1,Q[1:NRMBK,1:1];
'REAL'
      RAD,DETT,
XI,ETA,H1,H2,H,X1,X2,X3,X4,DETJ,TERM,
      EPSMAX,EPSMIN,GAMMAX,BETA1,BETA,
TEMP1,TEMP2,SIGMAX,SIGMIN,TORMAX,ALPHA,ALPHA1,
THICKNESS;
'INTEGER'
      NOF,
            COUNTER,
            K1,J1,II,JJ,KK1,KK2,KJ1,KJ2,
      I,J,PREVMAT,PREVORIN,NE,ET,XII,S,T,P,PMBK;
      'INTEGER' 'ARRAY'
NOELF[1:NELNF,1:1],
      NON[1:NELEM,1:8];
'COMMENT'
THE DATA REGARDING THE DIFFERENT MATERIALS OF THE ELEMENTS
ARE NOW READ IN, THE SEQUENCE IS EX MU-XY EY MU-YX AND G-XY.;
'FOR' I:=1 'STEP' 1 'UNTIL' NOMAT 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 5 'DO'
      MAT[I,J]:=READ;
PAPERTHROW;
WRITE TEXT
('('''('2C')'MATERIAL%NO.'('11S')'EX'('14S')'MU%XY'('12S')'
EY'('15S')'MU%YX'('12S')'G%XY')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' NOMAT 'DO'
'BEGIN'
      NEWLINE(1);
      SPACE(5);
      PRINT(I,3,0);
      SPACE(8);
      'FOR' J:=1 'STEP' 1 'UNTIL' 5 'DO'
'BEGIN'
      PRINT(MAT[I,J],0,4);
SPACE(4);
'END';
```

```
'END';
'COMMENT'
      THE ELEMENT MATERIAL THICKNESSES ARE READ IN.;
'FOR' I:=1 'STEP' 1 'UNTIL' NOTHICK 'DO'
      ELTHICK[I]:=READ;
WRITE TEXT('('''('2C')'MATERIAL%THICKNESS'('10S')'THICKNESS
'('1C')''('10S')'NO.'('15S')'(INCHES).')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' NOTHICK 'DO'
'BEGIN'
      NEWLINE(1);
SPACE(8);
      PRINT(I,3,0);
SPACE(13);
      PRINT(ELTHICK[I],0,4);
'END';
'COMMENT'
      THE COORDINATES OF THE NODES OF THE STRUCTURE ARE READ IN.
           NODE NO. 1   X COORDINATE---Y COORDINATE
            NODE NO. 2   X COORDINATE--ETC.;
'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
            COORD[I,J]:=READ;
PAPERTHROW;
WRITE TEXT
('('''('2C')''('6S')'NODE%NO.'('7S')'X%COORDINATE'('3S')'
Y%COORDINATE')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'BEGIN'
NEWLINE(1);
SPACE(8);
PRINT(I,3,0);
SPACE(8);
'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
'BEGIN'
PRINT(COORD[I,J],1,5);
SPACE(5);
'END';
'END';
'COMMENT'
      EACH ELEMENT NO. MATERIAL NO. ORIENTATION NO. AND NODE
      NOS. ARE NOW READ IN.
      THE NUDE NOS. MUST BE GIVEN IN A CLOCKWISE DIRECTION.;
      'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
      'FOR' J:=1 'STEP' 1 'UNTIL' 8 'DO'
            NON[I,J]:=READ;
PAPERTHROW;
WRITE TEXT('('''('2C')'ELEMENT'('2S')'MATERIAL'('2S')'ORIENTATION
%%THICKNESS'('8S')'ELEMENT%NODE%NUMBERS
%%%(CLUCKWISE%DIRECTION)
'('1C')'%%%%%NO'('8S')'NO'('8S')'NO'('9S')'NO'('9S')'
I'('9S')'J'('9S')'K'('9S')'M'('1C')'')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
```

```
NEWLINE(1);
'FOR' J:=1 'STEP' 1 'UNTIL' 8 'DO'
PRINT(NON[I,J],7,0);
'END';
'COMMENT'
THE DISC BACKING STORE FILE IS OPENED.
THE WORKING AREA OF CORE USED TO FORM THE ROW-BLOCKS
OF [MBK] BEFORE TRANSFERRING TO BACKING FILE IS INITIALISED.
THIS IS THEN  USED TO INITIALISE THE REQUIRED FILE AREA
TO STORE [MBK],;
USESTORE(10,'('ED')','('UDVKW2MBK2D1')',1,NRMBK*NCMBK);
'FOR' I:=1 'STEP' 1 'UNTIL' 2 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 2*NCB 'DO'
WMBK[I,J]:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' NRB 'DO'
'BEGIN'
      P:=4*NCB*(I-1)+1;
      PUT PART(10,P,WMBK,WMBK[1,1],WMBK[2,2*NCB]);
'END' OF INITIALISING DISC FILE AREA [MBK],;
PREVMAT:=0;
PREVORIN:=10000;
'COMMENT'
      EACH ELEMENT STIFFNESS MATRIX IS NOW CALC.
      AND SUBSEQUENTLY DUMPED INTO [MBK],;
HH[1]:=HH[3]:=0,5555556;
HH[2]:=0,8888889;
XI1[1]:=ETA1[1]:=-0,77459667;
XI1[2]:=ETA1[2]:=0,00000000;
XI1[3]:=ETA1[3]:=0,77459667;
WRITE TEXT('('!('1C')'ELEMENT%NO.')');
'FOR' NE:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
NEWLINE(1);
PRINT(NE,3,0);
'COMMENT'
      THE COORDINATES OF EACH ELEMENT ARE NOW SET UP;
            ELCO[1,1]:=COORD[NON[NE,5],1];
            ELCO[1,2]:=COORD[NON[NE,5],2];
            ELCO[2,1]:=COORD[NON[NE,6],1];
            ELCO[2,2]:=COORD[NON[NE,6],2];
            ELCO[3,1]:=COORD[NON[NE,7],1];
            ELCO[3,2]:=COORD[NON[NE,7],2];
            ELCO[4,1]:=COORD[NON[NE,8],1];
            ELCO[4,2]:=COORD[NON[NE,8],2];
'COMMENT'
      THE ELASTICITY MATRIX [D1] FOR EACH ELEMENT IS SET UP
      IF THE MATERIAL DIFFERS FROM THAT OF THE PREVIOUS ELEMENT,;
'IF' NON[NE,2]#PREVMAT 'THEN'
'BEGIN'
      TERM:=1-MAT[NON[NE,2],2]*MAT[NON[NE,2],4];
      D1[1,1]:=MAT[NON[NE,2],1]/TERM;
      D1[1,2]:=MAT[NON[NE,2],4]*D1[1,1];
      D1[2,2]:=MAT[NON[NE,2],3]/TERM;
      D1[2,1]:=MAT[NON[NE,2],2]*D1[2,2];
       D1[3,3]:=MAT[NON[NE,2],5]*2;
      D1[1,3]:=D1[2,3]:=D1[3,1]:=D1[3,2]:=0;
```

```
PREVMAT:=NON[NE,2];
      'GOTO' LAB1;
'END' OF SETTING UP [D1] FOR THIS MATERIAL;
'COMMENT'
      THE ELASTICITY MATRIX FORMED ABOVE IS FOR
      PLANESTRESS ANALYSIS WITH
      A MATERIAL WHOSE ORTHOTROPIC AXES COINCIDE WITH
      THOSE OF THE GLOBAL AXES OF THE STRUCTURE,
      IF THE AXES DIFFER THE [D1] MATRIX IS NOW TRANSFORMED
      FROM ITS LOCAL TO THE STRUCTURE GLOBAL AXES.;
'IF' NON[NE,3]#PREVORIN 'THEN'
'BEGIN'
      LAB1:
    'BEGIN'
'COMMENT'
            THE TRANSFORMATION MATRIX [TR] IS SET UP.;
      RAD:=NON[NE,3]*3.14159265359/180;
      TR[1,1]:=TR[2,2]:=(COS(RAD))↑2;
      TR[1,2]:=TR[2,1]:=(SIN(RAD))↑2;
      TR[3,3]:=TR[1,1]-TR[1,2];
      TR[3,2]:=SIN(RAD)*COS(RAD);
      TR[3,1]:=-TR[3,2];
      TR[1,3]:=2*TR[3,2];
      TR[2,3]:=-2*TR[3,2];
'COMMENT'
      [TR] IS NOW INVERTED;
           MATINVERSE(TR,3,INVT);
      'COMMENT'
            [D1][TR] IS NOW FORMED;
      MATMULT(D1,TR,DTR,3,3,3);
'COMMENT'
FINALLY [TR]INVERSE IS POST MULT. BY [D1][TR] & PUT IN [D];
      MATMULT(INVT,DTR,D,3,3,3);
      PREVORIN:=NON[NE,3];
'END';
      D[1,3]:=D[1,3]/2;
     D[2,3]:=D[2,3]/2;
      D[3,3]:=D[3,3]/2;
'END' OF TRANSFORMATION OF [D1] RESULT IN [D];
'FOR' I:=1 'STEP' 1 'UNTIL' 8 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 8 'DO'
      K[I,J]:=0;
THICKNESS:=ELTHICK[NON[NE,4]];
'FOR' ET:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' XII:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
      XI:=XI1[XII];
      ETA:=ETA1[ET];
      H1:=HH[XII];
      H2:=HH[ET];
      H:=H1*H2*THICKNESS;
'COMMENT'
THE TERMS MAKING UP [LAMBDA] ARE CALC.;
      X1:=1+XI;
      X2:=1-XI;
       X3:=1+ETA;
```

A5.1

```
      X4;=1-ETA;
'COMMENT'
     THE MATRIX [LAMBDA] IS FORMED;
           LAM[1,1];=-0.25*X4;
           LAM[1,2];=-0.25*X3;
           LAM[1,3];=0.25*X3;
           LAM[1,4];=0.25*X4;
           LAM[2,1];=-0.25*X2;
           LAM[2,2];=0.25*X2;
           LAM[2,3];=0.25*X1;
           LAM[2,4];=-0.25*X1;
'COMMENT'
      [JAY] IS FORMED BY POST MULTIPLYING [LAMBDA] BY
       THE ELEMENT NODAL COORDINATES.;
MATMULT(LAM,ELCO,JAY,2,4,2);
'COMMENT'
      THE DETERMINANT OF [JAY] IS CALC.;
DETJ;= JAY[1,1]*JAY[2,2]-JAY[2,1]*JAY[1,2];
      'IF' ABS(JAY[1,1]) < ABS(JAY[1,2]) 'THEN'
      'BEGIN'
           'IF' JAY[2,1]=0 'THEN'
           'BEGIN'
                NEWLINE(1);
                PRINT(XI,1,6);
                     PRINT(ETA,1,6);
                WRITE TEXT('('%%JACOBIAN%[JAY]%MATRIX%FOR%THIS%
                      GAUSS%POINT')');
                NEWLINE(1);
                PRINT(JAY[1,1],0,6);
                PRINT(JAY[1,2],0,6);
                NEWLINE(1);
                PRINT(JAY[2,1],0,6);
                PRINT(JAY[2,2],0,6);
           'END';
      'END';
'COMMENT'
      THE INVERSE OF [JAY] IS NOW FOUND.;
           MATINVERSE(JAY,2,INVERSEJ);
'CUMMENT'
      THE PRODUCT [INVERSEJ][LAM] IS CALC.;
MATMULT(INVERSEJ,LAM,INVJLAM,2,2,4);
'COMMENT'
      THE MATRIX [B] IS FORMED FROM THE TERMS OF [INVJLAM].;
'FOR' I;=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' J;=1 'STEP' 1 'UNTIL' 8 'DO'
B[I,J];=0;
B[1,1];=B[3,2];=INVJLAM[1,1];
B[1,3];=B[3,4];=INVJLAM[1,2];
B[1,5];=B[3,6];=INVJLAM[1,3];
B[1,7];=B[3,8];=INVJLAM[1,4];
B[2,2];=B[3,1];=INVJLAM[2,1];
B[2,4];=B[3,3];=INVJLAM[2,2];
B[2,6];=B[3,5];=INVJLAM[2,3];
B[2,8];=B[3,7];=INVJLAM[2,4];
'COMMENT'
THE PRODUCT OF [D][B] IS NOW FORMED.;
```

A5.1

```
MATMULT(D,B,DB,3,3,8);
'COMMENT'
THE MATRIX [D][B] IS PRE-MULTIPLIED BY [B] TRANSPOSED;
MATRANMULT(B,DB,SUBK,8,3,8);
'COMMENT'
FINALLY THE SUBK FOR THIS GAUSS POINT IS MULTIPLIED BY DETJ
THE ELEMENT THICKNESS AND THE WEIGHTING FACTOR H ARE ADDED
INTO THE ELEMENT STIFFNESS MATRIX [K].;
'FOR' I;=1 'STEP' 1 'UNTIL' 8 'DO'
'FOR' J;=1 'STEP' 1 'UNTIL' 8 'DO'
K[I,J];=K[I,J]+DETJ*H*SUBK[I,J];
'END' OF ELEMENT STIFFNESS FORMATION.;
'COMMENT'
      THE ELEMENTS OF [K] ARE NOW DUMPED INTO [MBK]
      IN THE DISC BACKING FILE UDVKW2MBK201.;
'FOR' I;=5 'STEP' 1 'UNTIL' 8 'DO'
'BEGIN'
P;=PMBK;=4*NCB*(NON[NE,I]-1)+1;
GET PART(10,P,WMBK,WMBK[1,1],WMBK[2,2*NCB]);
'FOR' J;=5 'STEP' 1 'UNTIL' 8 'DO'
'BEGIN'
      'IF' NON[NE,I]>NON[NE,J] 'THEN' 'GOTO' NODUMP;
      K1;=2*(I-4)-1;
      J1;=2*(J-4)-1;
            KK1;=K1;
            KK2;=K1+1;
            KJ1;=J1+1;
            KJ2;=J1;
      T;=2*(NON[NE,J]-NON[NE,I]);
      WMBK[1,T+1];=WMBK[1,T+1] + K[K1,J1];
      WMBK[1,T+2];=WMBK[1,T+2] + K[KK1,KJ1];
      WMBK[2,T+1];=WMBK[2,T+1] + K[KK2,KJ2];
      WMBK[2,T+2];=WMBK[2,T+2] + K[K1+1,J1+1];
NODUMP;'END';
P;=PMBK;
PUT PART(10,P,WMBK,WMBK[1,1],WMBK[2,2*NCB]);
      'END';
WRITE TEXT('('DUMPED%O.K.')');
'END' OF SETTING UP MOD. STRUCTURAL STIFF. MATRIX;
'COMMENT'
      THE FORCE AND DISPLACEMENT VECTORS ARE NOW FORMED.
THE VECTOR NUMBERING SEQUENCE IS--
            NODE NO. 1   X-DIRECTION (U-DISPLACEMENT)
            NODE NO. 1   Y-DIRECTION (V-DISPLACEMENT)
            NODE NO. 2   X-DIRECTION  -ETC.
THE NODAL FORCES HAVE THE VALUE OF-
      1) ITS PRESCRIBED VALUE
   OR 2) ZERO-(THIS IMPLIES EITHER NO APPLIED FORCE OR A REST.)
NODAL DISPLACEMENTS CAN HAVE ONE OF THREE VALUES.
      1) ITS PRESCRIBED VALUE
   OR 2) ZERO-(IF IT IS FREE TO MOVE AND UNPRESCRIBED)
   OR 3) 0.000001-(IF IT IS RESTRAINED).
N.B. NO PRESCRIBED VALUES WITH THIS VERSION 2/3/1972
INITIALLY BOTH VECTORS ARE ZEROED AND THEN ONLY PRESCRIBED
FORCES AND THEN PRESCRIBED DISPLACEMENTS AND RESTRAINTS
ARE ACTUALLY READ IN.  THE SEQUENCE IS IN BOTH CASES
```

A5.1

```
      =EQUATION NO. FOLLOWED BY FORCE/DISPLACEMENT.
FINALLY THE COMPLETED DISPLACEMENT VECTOR IS SCANNED AND
IF THERE IS AN APPLIED DISPLACEMENT THE PROCEDURE BANDMULT
IS CALLED WHICH MODIFIES THE APPLIED FORCE VECTOR-AND
ZEROES THE APPROPRIATE TERMS OF [MBK] LEAVING A 1 ON THE
MAIN DIAGONAL TERM (I.E. IN THE FIRST COLUMN OF [MBK]).
IF THERE IS AN APPLIED RESTRAINT THEN AGAIN THE
NECESSARY ZEROEING IS CARRIED OUT TO [MBK].;
'FOR' I:=1 'STEP' 1 'UNTIL' NRMBK 'DO'
      F[I,1]:=X[I,1]:=N1[I,1]:=Q[I,1]:=0;
PAPERTHROW;
WRITE TEXT('('''('2C')'CO%%NO.'('6S')'EQUATION%NO.'('6S')'
APPLIED%DISPLACEMENTS%%(CONSTRAINT=0.000001)
'('1C')''('37S')'(INCHES)')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' CO 'DO'
      'BEGIN'
            J:=READ;
            X[J,1]:=READ;
NEWLINE(1);
SPACE(3);
PRINT(I,3,0);
SPACE(6);
PRINT(J,4,0);
SPACE(13);
PRINT(X[J,1],0,6);
      'END' OF SETTING UP DISPLACEMENT VECTOR;
WRITE TEXT('('''('2C')'LO%%NO.'('6S')'EQUATION%NO.'('6S')'
APPLIED%FORCES'('1C')''('34S')'LBF.')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' LO 'DO'
      'BEGIN'
            J:=READ;
            F[J,1]:=READ;
NEWLINE(1);
SPACE(3);
PRINT(I,3,0);
SPACE(6);
PRINT(J,4,0);
SPACE(13);
PRINT(F[J,1],0,6);
      'END' OF SETTING UP LOAD VECTOR;
'FOR' I:=1 'STEP' 1 'UNTIL' NRB 'DO'
'BEGIN'
P:=PMBK:=4*NCB*(I-1)+1;
GET PART(10,P,WMBK,WMBK[1,1],WMBK[2,2*NCB]);
'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
'BEGIN'
'IF' X[2*(I-1)+J,1]#0 'THEN'
      'BEGIN'
      'FOR' II:=1 'STEP' 1 'UNTIL' NCMBK 'DO'
       WMBK[J,II]:=0;
      'FOR' II:=1 'STEP' 1 'UNTIL' 2 'DO'
        WMBK[II,J]:=0;
        WMBK[J,J]:=1;
      'END';
```

```
'END';
'FOR' J:=1 'STEP' 1 'UNTIL' NCB-1 'DO'
'BEGIN'
 'IF' I+J>NRB 'THEN' 'GOTO' NOROW;
 'FOR' II:=1 'STEP' 1 'UNTIL' 2 'DO'
      'BEGIN'
       'IF' X[2*(I+J-1)+II,1]#0 'THEN'
            'BEGIN'
            'FOR' JJ:=1 'STEP' 1 'UNTIL' 2 'DO'
            WMBK[JJ,2*J+II]:=0;
            'END';
      'END';
NOROW:'END';
P:=PMBK;
PUT PART(10,P,WMBK,WMBK[1,1],WMBK[2,2*NCB]);
'END' OF MODS. TO [MBK] FOR APPLIED CONSTRAINTS.;
WRITE TEXT
('('''('2C')'TIME%AT%COMMENCEMENT%OF%EQUATION%SOLUTION%%')');
TIMENOW;
DBLOKGAUSS2(F,NCB,NRB);
WRITE TEXT
('('''('2C')'TIME%AT%COMPLETION%OF%EQUATION%SOLUTION%%')');
TIMENOW;
'COMMENT'
     THE NODAL POINT DISPLACEMENTS AND FORCES ARE NOW OUTPUT;
PAPERTHROW;
WRITE TEXT('('''('2C')'%%NODE'('22S')'NUDAL%%%DISPLACEMENTS
'('29S')'NUDAL%%%FORCES'('1C')'%%%NO.'('27S')'(INCHES)
'('41S')'(LBF)'('1C')''('20S')'X%DIRECTION'('14S')'Y%
DIRECTION'('14S')'X%DIRECTION'('14S')'Y%DIRECTION'('1C')'
'('24S')'U'('24S')'V'('24S')'FX'('23S')'FY'('1C')'')');
'FOR' I:=1 'STEP' 1'UNTIL' NONOP 'DO'
      'BEGIN'
      NEWLINE(1);
      SPACE(2);
      PRINT(I,3,0);
      SPACE(12);
      PRINT(F[2*I-1,1],1,6);
      SPACE(15);
      PRINT(F[2*I,1],1,6);
      SPACE(10);
      SPACE(10);
      'END';
'COMMENT'
      EACH ELEMENT DISPLACEMENT VECTOR IS NOW FORMED
      AND THE [B] AND [D] MATRICES RE-CALC, THE CENTROIDAL
      STRAIN AND STRESS COMPONENTS ARE THEN EVALUATED AND OUTPUT
      I. E. AT XI=ETA=0;
PAPERTHROW;
WRITE TEXT('('''('2C')''('28S')'*****%%%ELEMENT%%%
CENTROIDAL%%%STRESS%%%COMPONENTS%%%%*****'('2C')'
'('1C')'ELEMENT%%%%STRESS%XX'('7S')'STRESS%YY'('7S')'
STRESS%XY'('7S')'SIGMA%MAX'('7S')'SIGMA%MIN'('8S')'TOR%MAX
'('8S')'ALPHA'('1C')'
%%%NO.'('5S')'STRAIN%XX'('7S')'STRAIN%YY'('7S')'
STRAIN%XY'('6S')'EPSILON%MAX'('5S')'EPSILON%MIN'('6S')'GAMMA%MAX
```

```
'('8S')'BETA'('1C')'')');
COUNTER:=1;
PREVMAT:=0;
PREVORIN:=1000;
'COMMENT'
      THE POSITIONS WHERE THE ELEMENT STRAIN AND STRESS
      COMPONENTS ARE EVALUATED ARE CALC. AND THE MATRIX
      [LAMBDA] REFORMED-THIS IS THE SAME FOR ALL ELEMENTS.;
            XI:=0;
            ETA:=0;
'COMMENT'
THE TERMS MAKING UP [LAMBDA] ARE CALC.;
      X1:=1+XI;
      X2:=1-XI;
       X3:=1+ETA;
      X4:=1-ETA;
'COMMENT'
     THE MATRIX [LAMBDA] IS FORMED;
            LAM[1,1]:=-0.25*X4;
            LAM[1,2]:=-0.25*X3;
            LAM[1,3]:=0.25*X3;
            LAM[1,4]:=0.25*X4;
            LAM[2,1]:=-0.25*X2;
            LAM[2,2]:=0.25*X2;
            LAM[2,3]:=0.25*X1;
            LAM[2,4]:=-0.25*X1;
'FOR' NE:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
'COMMENT'
      THE COORDINATES OF EACH ELEMENT ARE NOW SET UP;
            ELCO[1,1]:=COORD[NON[NE,5],1];
            ELCO[1,2]:=COORD[NON[NE,5],2];
            ELCO[2,1]:=COORD[NON[NE,6],1];
            ELCO[2,2]:=COORD[NON[NE,6],2];
            ELCO[3,1]:=COORD[NON[NE,7],1];
            ELCO[3,2]:=COORD[NON[NE,7],2];
            ELCO[4,1]:=COORD[NON[NE,8],1];
            ELCO[4,2]:=COORD[NON[NE,8],2];
'IF' COUNTER=17 'THEN'
'BEGIN'
PAPERTHROW;
WRITE TEXT('('''('2C')''('28S')'*****%%%%ELEMENT%%%
CENTROIDAL%%%STRESS%%%COMPONENTS%%%%*****'('2C')'
'('1C')'ELEMENT%%%%STRESS%XX'('7S')'STRESS%YY'('7S')'
STRESS%XY'('7S')'SIGMA%MAX'('7S')'SIGMA%MIN'('8S')'TOR%MAX
'('8S')'ALPHA'('1C')'
%%%NO.'('5S')'STRAIN%XX'('7S')'STRAIN%YY'('7S')'
STRAIN%XY'('6S')'EPSILON%MAX'('5S')'EPSILON%MIN'('6S')'GAMMA%MAX
'('8S')'BETA'('1C')''')');
            COUNTER:=1;
'END';
'COMMENT'
      THE DISPLACEMENT VECTOR OF EACH ELEMENT IS SET UP.;
      ELDIS[1,1]:=F[2*(NON[NE,5]-1)+1,1];
      ELDIS[2,1]:=F[2*(NON[NE,5]-1)+2,1];
      ELDIS[3,1]:=F[2*(NON[NE,6]-1)+1,1];
```

```
      ELDIS[4,1]:=F[2*(NON[NE,6]-1)+2,1];
      ELDIS[5,1]:=F[2*(NON[NE,7]-1)+1,1];
      ELDIS[6,1]:=F[2*(NON[NE,7]-1)+2,1];
      ELDIS[7,1]:=F[2*(NON[NE,8]-1)+1,1];
       ELDIS[8,1]:=F[2*(NON[NE,8]-1)+2,1];
'COMMENT'
      THE PLASTICITY MATRIX [D1] FOR EACH ELEMENT IS SET UP
      IF THE MATERIAL DIFFERS FROM THAT OF THE PREVIOUS ELEMENT.;
'IF' NUN[NE,2]#PREVMAT 'THEN'
'BEGIN'
      TERM:=1-MAT[NON[NE,2],2]*MAT[NON[NE,2],4];
      D1[1,1]:=MAT[NON[NE,2],1]/TERM;
      D1[1,2]:=MAT[NON[NE,2],4]*D1[1,1];
      D1[2,2]:=MAT[NON[NE,2],3]/TERM;
      D1[2,1]:=MAT[NON[NE,2],2]*D1[2,2];
       D1[3,3]:=MAT[NON[NE,2],5]*2;
      D1[1,3]:=D1[2,3]:=D1[3,1]:=D1[3,2]:=0;
PREVMAT:=NON[NE,2];
'GUTO' LAB2;
'END' OF SETTING UP [D1] FOR THIS MATERIAL;
'CUMMENT'
      THE ELASTICITY MATRIX FORMED ABOVE IS FOR
      PLANESTRESS ANALYSIS WITH
      A MATERIAL WHOSE ORTHOTROPIC AXES COINCIDE WITH
      THOSE OF THE GLOBAL AXES OF THE STRUCTURE.
      IF THE AXES DIFFER THE [D1] MATRIX IS NOW TRANSFORMED
      FROM ITS LOCAL TO THE STRUCTURE GLOBAL AXES.;
'IF' NON[NE,3]#PREVORIN 'THEN'
'BEGIN'
      LAB2;
    'BEGIN'
'COMMENT'
            THE TRANSFORMATION MATRIX [TR] IS SET UP.;
      RAD:=NON[NE,3]*3.14159265359/180;
      TR[1,1]:=TR[2,2]:=(COS(RAD))↑2;
      TR[1,2]:=TR[2,1]:=(SIN(RAD))↑2;
      TR[3,3]:=TR[1,1]-TR[1,2];
      TR[3,2]:=SIN(RAD)*COS(RAD);
      TR[3,1]:=-TR[3,2];
      TR[1,3]:=2*TR[3,2];
      TR[2,3]:=-2*TR[3,2];
'CUMMENT'
      [TR] IS NOW INVERTED;
           MATINVERSE(TR,3,INVT);
      'COMMENT'
            [D1][TR] IS NOW FORMED;
      MATMULT(D1,TR,DTR,3,3,3);
'COMMENT'
[TR]INVERSE IS POST-MULTIPLIED BY [D1][TR] AND PUT IN [D];
      MATMULT(INVT,DTR,D,3,3,3);
      PREVORIN:=NON[NE,3];
'END';
      D[1,3]:=D[1,3]/2;
     D[2,3]:=D[2,3]/2;
      D[3,3]:=D[3,3]/2;
'END' OF TRANSFORMATION OF [D1] RESULT IN [D];
```

```
'COMMENT'
      [JAY] IS FORMED BY POST MULTIPLYING [LAMBDA] BY
       THE ELEMENT NODAL COORDINATES.;
MATMULT(LAM,ELCO,JAY,2,4,2);
'COMMENT'
      THE DETERMINANT OF [JAY] IS CALC.;
DETJ:= JAY[1,1]*JAY[2,2]-JAY[2,1]*JAY[1,2];
'COMMENT'
      THE INVERSE OF [JAY] IS NOW FOUND.;
           MATINVERSE(JAY,2,INVERSEJ);
'COMMENT'
      THE PRODUCT [INVERSEJ][LAM] IS CALC.;
MATMULT(INVERSEJ,LAM,INVJLAM,2,2,4);
'COMMENT'
      THE MATRIX [B] IS FORMED FROM THE TERMS OF [INVJLAM].;
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 8 'DO'
B[I,J]:=0;
B[1,1]:=B[3,2]:=INVJLAM[1,1];
B[1,3]:=B[3,4]:=INVJLAM[1,2];
B[1,5]:=B[3,6]:=INVJLAM[1,3];
B[1,7]:=B[3,8]:=INVJLAM[1,4];
B[2,2]:=B[3,1]:=INVJLAM[2,1];
B[2,4]:=B[3,3]:=INVJLAM[2,2];
B[2,6]:=B[3,5]:=INVJLAM[2,3];
B[2,8]:=B[3,7]:=INVJLAM[2,4];
'COMMENT'
      THE ELEMENT STRAINS ARE CALC. BY POST-MULTIPLYING [B] BY
      THE ELEMENT NODAL DISPLACEMENTS.
      FROM THESE THE PRINCIPAL STRAINS ARE THEN EVALUATED
       I.E. EPSMAX,EPSMIN, AND THE MAX. SHEAR STAIN GAMMAX.
      GAMMAX IS -VE IF STRAIN XX > STRAIN YY.
      BETA IS THE ANGLE MEASURED FROM THE +VE HORIZONTAL X AXIS
      TO THE PLANE OF EPSMAX. THE ANGLE IF +VE IS MEASURED
       ANTICLOCKWISE AND IF -VE CLOCKWISE.;
MATMULT(B,ELDIS,STN,3,8,1);
      TEMP1:=SQRT((STN[1,1]-STN[2,1])↑2 + STN[3,1]↑2);
      TEMP2:=(STN[1,1] + STN[2,1])/2;
      EPSMAX:=TEMP2 + TEMP1/2;
      EPSMIN:=TEMP2 - TEMP1/2;
      GAMMAX:=TEMP1;
      BETA1:=90*ARCTAN(STN[3,1]/(STN[1,1] - STN[2,1]))/
      3.14159265359;
           'IF' STN[3,1]/(STN[1,1] - STN[2,1]) > 0 'THEN'
       'BEGIN'
            'IF' STN[3,1] > 0 'THEN'
                 BETA:=BETA1
            'ELSE'
                 BETA:=BETA1 +  90;
'END'
'ELSE'
 'BEGIN'
            'IF' STN[3,1] > 0 'THEN'
                 BETA:=BETA1 - 90
            'ELSE'
                 BETA:=BETA1;
```

```
          'END';
'COMMENT'
       FINALLY THE ELEMENT STRESS COMPONENTS ARE CALC.
      FROM THESE THE PRINCIPAL STRESSES ARE THEN EVALUATED
      I.E. SIGMAX,SIGMIN, AND THE MAX. SHEAR STRESS TORMAX
      TORMAX IS -VE IF STRESS XX > STRESS YY.
      ALPHA IS THE ANGLE MEASURED FROM THE +VE HORIZONTAL X
      AXIS TO THE PLANE OF SIGMAX. THE ANGLE IF +VE IS MEASURED
       ANTICLOCKWISE AND IF -VE CLOCKWISE.;
MATMULT(D,STN,STS,3,3,1);
          TEMP1:=SQRT((STS[1,1]-STS[2,1])↑2 + 4*STS[3,1]↑2);
      TEMP2:=(STS[1,1] + STS[2,1])/2;
      SIGMAX:=TEMP2 + TEMP1/2;
      SIGMIN:=TEMP2 - TEMP1/2;
      TORMAX:=TEMP1/2;
        'IF' STS[1,1] > STS[2,1] 'THEN' TORMAX:=-TORMAX;
      ALPHA1:=90*ARCTAN(2*STS[3,1]/(STS[1,1] - STS[2,1]))/
      3.14159265359;
       'IF' 2*STS[3,1]/(STS[1,1]-STS[2,1]) > 0 'THEN'
            'BEGIN'
                 'IF' 2*STS[3,1] > 0 'THEN'
                      ALPHA:=ALPHA1
                 'ELSE'
                      ALPHA:=ALPHA1+90;
           'END'
           'ELSE'
            'BEGIN'
                 'IF' 2*STS[3,1] > 0 'THEN'
                      ALPHA:=ALPHA1-90
                 'ELSE'
                                ALPHA:=ALPHA1;
                'END';
      NEWLINE(2);
      PRINT(NE,3,0);
      SPACE(5);
      'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'
      'BEGIN'
      PRINT(STS[I,1],6,1);
      SPACE(5);
      'END';
      PRINT(SIGMAX,6,1);
      SPACE(5);
      PRINT(SIGMIN,6,1);
      SPACE(5);
      PRINT(TORMAX,6,1);
      SPACE(5);
      PRINT(ALPHA,3,3);
      NEWLINE(1);
      SPACE(9);
      SPACE(2);
      PRINT(STN[1,1],1,6);
      SPACE(5);
      PRINT(STN[2,1],1,6);
      SPACE(5);
      PRINT(STN[3,1]/2,1,6);
      SPACE(5);
```

```
PRINT(EPSMAX,1,6);
SPACE(5);
PRINT(EPSMIN,1,6);
SPACE(5);
PRINT(GAMMAX,1,6);
SPACE(5);
PRINT(BETA,3,3);
COUNTER:=COUNTER+1;
'END' OF STAIN AND STRESS CALC.;
'IF' NELNF=0 'THEN' 'GOTO' HOMEANDRY;
'FOR' I:=1 'STEP' 1 'UNTIL' NELNF 'DO'
NOELF[I,1]:=READ;
PREVMAT:=0;
PREVORIN:=10000;
COUNTER:=1;
PAPERTHROW;
WRITE TEXT
('('!'('2C')''('19S')'****%ELEMENT%%NODAL%%FORCES%****'('2C')'
ELEMENT'('28S')'ELEMENT%%%NODE%%%NUMBERS'('1C')'
%%NO'('26S')'FX%%%AND%%%FY%%%FORCE%%%CUMPONENTS'('1C')'
'('20S')'I'('17S')'J'('17S')'K'('17S')'M'('1C')'
'('19S')'FX'('16S')'FX'('16S')'FX'('16S')'FX'('1C')'
'('19S')'FY'('16S')'FY'('16S')'FY'('16S')'FY'('1C')'')');
'FOR' NOF:=1 'STEP' 1 'UNTIL' NELNF 'DO'
'BEGIN'
      NE:=NOELF[NOF,1];
'IF' COUNTER=13 'THEN'
'BEGIN'
PAPERTHROW;
WRITE TEXT
('('!'('2C')''('19S')'****%ELEMENT%%NODAL%%FORCES%****'('2C')'
ELEMENT'('28S')'ELEMENT%%%NODE%%%NUMBERS'('1C')'
%%NO'('26S')'FX%%%AND%%%FY%%%FORCE%%%COMPONENTS'('1C')'
'('20S')'I'('17S')'J'('17S')'K'('17S')'M'('1C')'
'('19S')'FX'('16S')'FX'('16S')'FX'('16S')'FX'('1C')'
'('19S')'FY'('16S')'FY'('16S')'FY'('16S')'FY'('1C')'')');
COUNTER:=1;
'END';
'COMMENT'
      THE COORDINATES OF EACH ELEMENT ARE NOW SET UP;
            ELCO[1,1]:=COORD[NON[NE,5],1];
            ELCO[1,2]:=COORD[NON[NE,5],2];
            ELCO[2,1]:=COORD[NON[NE,6],1];
            ELCO[2,2]:=COORD[NON[NE,6],2];
            ELCO[3,1]:=COORD[NON[NE,7],1];
            ELCO[3,2]:=COORD[NON[NE,7],2];
            ELCO[4,1]:=COORD[NON[NE,8],1];
            ELCO[4,2]:=COORD[NON[NE,8],2];
'COMMENT'
      THE ELASTICITY MATRIX [D1] FOR EACH ELEMENT IS SET UP
      IF THE MATERIAL DIFFERS FROM THAT OF THE PREVIOUS ELEMENT.;
'IF' NON[NE,2]#PREVMAT 'THEN'
'BEGIN'
      TERM:=1-MAT[NON[NE,2],2]*MAT[NON[NE,2],4];
      D1[1,1]:=MAT[NON[NE,2],1]/TERM;
      D1[1,2]:=MAT[NON[NE,2],4]*D1[1,1];
```

```
      D1[2,2]:=MAT[NON[NE,2],3]/TERM;
      D1[2,1]:=MAT[NON[NE,2],2]*D1[2,2];
       D1[3,3]:=MAT[NON[NE,2],5]*2;
      D1[1,3]:=D1[2,3]:=D1[3,1]:=D1[3,2]:=0;
PREVMAT:=NON[NE,2];
            'GOTO' LAB3;
'END' OF SETTING Up [D1] FOR THIS MATERIAL;
'COMMENT'
      THE ELASTICITY MATRIX FORMED ABOVE IS FOR
      PLANESTRESS ANALYSIS WITH
      A MATERIAL WHOSE ORTHOTROPIC AXES COINCIDE WITH
      THOSE OF THE GLOBAL AXES OF THE STRUCTURE.
      IF THE AXES DIFFER THE [D1] MATRIX IS NOW TRANSFORMED
      FROM ITS LOCAL TO THE STRUCTURE GLOBAL AXES.;
'IF' NON[NE,3]#PREVORIN 'THEN'
'BEGIN'
            LAB3;
     'BEGIN'
'COMMENT'
             THE TRANSFORMATION MATRIX [TR] IS SET UP.;
      RAD:=NON[NE,3]*3.14159265359/180;
      TR[1,1]:=TR[2,2]:=(COS(RAD))↑2;
      TR[1,2]:=TR[2,1]:=(SIN(RAD))↑2;
      TR[3,3]:=TR[1,1]-TR[1,2];
      TR[3,2]:=SIN(RAD)*COS(RAD);
      TR[3,1]:=-TR[3,2];
      TR[1,3]:=2*TR[3,2];
      TR[2,3]:=-2*TR[3,2];
'COMMENT'
      [TR] IS NOW INVERTED;
          MATINVERSE(TR,3,INVT);
      'COMMENT'
            [D1][TR] IS NOW FORMED;
      MATMULT(D1,TR,DTR,3,3,3);
      'COMMENT'
      FINALLY [TR]INVERSE IS POST MULTIPLIED BY
      [D1][TR] AND PUT IN [D];
      MATMULT(INVT,DTR,D,3,3,3);
      PREVORIN:=NON[NE,3];
'END';
      D[1,3]:=D[1,3]/2;
     D[2,3]:=D[2,3]/2;
      D[3,3]:=D[3,3]/2;
'END' OF TRANSFORMATION OF [D1] RESULT IN [D];
'FOR' I:=1 'STEP' 1 'UNTIL' 8 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 8 'DO'
      K[I,J]:=0;
THICKNESS:=ELTHICK[NON[NE,4]];
'FOR' ET:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' XII:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'


      **INSERT A5.1 AS ABOVE**
```

```
'END' OF ELEMENT STIFFNESS FORMATION.;
'COMMENT'
      THE DISPLACEMENT VECTOR OF EACH ELEMENT IS SET UP.;
      ELDIS[1,1]:=F[2*(NON[NE,5]-1)+1,1];
      ELDIS[2,1]:=F[2*(NON[NE,5]-1)+2,1];
      ELDIS[3,1]:=F[2*(NON[NE,6]-1)+1,1];
      ELDIS[4,1]:=F[2*(NON[NE,6]-1)+2,1];
      ELDIS[5,1]:=F[2*(NON[NE,7]-1)+1,1];
      ELDIS[6,1]:=F[2*(NON[NE,7]-1)+2,1];
      ELDIS[7,1]:=F[2*(NON[NE,8]-1)+1,1];
       ELDIS[8,1]:=F[2*(NON[NE,8]-1)+2,1];
'COMMENT'
      THE ELEMENT NODAL FORCES ARE OBTAINED BY POST-MULTIPLYING
      THE ELEMENT STIFFNESS MATRIX [K] BY THE ELEMENT
      DISPLACEMENT VECTOR [ELDIS].;
MATMULT(K,ELDIS,ELNF,8,8,1);
NEWLINE(2);
PRINT(NE,3,0);
SPACE(10);
'FOR' I:=5 'STEP' 1 'UNTIL' 8 'DO'
'BEGIN'
      PRINT(NON[NE,I],4,0);
      SPACE(11);
'END';
NEWLINE(1);
SPACE(14);
'FOR' I:=1,3,5,7 'DO'
'BEGIN'
      PRINT(ELNF[I,1],3,6);
      SPACE(5);
'END';
NEWLINE(1);
SPACE(14);
'FOR' I:=2,4,6,8 'DO'
'BEGIN'
      PRINT(ELNF[I,1],3,6);
      SPACE(5);
'END';
COUNTER:=COUNTER + 1;
'END' OF ELEMENT NODAL FORCES CALCS.;
HOMEANDRY;
'END';
'END' OF (PLANESTRDISC) WK. BEG. 2/4/73;
```

APPENDIX SIX

THE PLANE STRESS/STRAIN FINITE ELEMENT ANALYSIS DATA CHECK PLOT PROGRAM LISTING

APPENDIX SIX

```
'LIST' (LP)
'LIBRARY' (ED,SUBGROUPSRA3)
'LIBRARY' (ED,SUBGROUPSRGP)
'LIBRARY' (ED,SUBGROUPS-RS)
'PROGRAM' (PLOTMESH4N2D)
'COMPACT DATA'
'MIXED SEGMENTS'
'INPUT' 1=CRO
'OUTPUT' 2=LPO
'TRACE'2
'BEGIN'
      'COMMENT'
      A PROGRAM FOR PLOTTING INDIRECTLY THE MESH PATTERN OF A
      STRUCTURE TO BE ANALYSED USING K. WRIGHTS PLANE
      STRESS/STRAIN FINITE ELEMENT ANALYSIS PROGRAMS.
      THESE PROGRAMS USE RECTANGULAR 4-NODED-2-DEGREE OF FREEDOM
      PER NODE LINEAR DISPLACEMENT FORMULATED FINITE ELEMENTS.
MK, 2  PLOTS CORTICAL BONE GRAIN DIRECTION MATERIAL NO 1. 1/1/73.
      KEITH W.J. WRIGHT,
      DEPT. MECH. ENG.,
      BRUNEL UNIVERSITY,
      UXBRIDGE.;
      'INTEGER'
            NELEM,NONOP,I,J,NE;
      'REAL'
      XAV,YAV,ANGRAD,XMIN,XMAX,YMIN,YMAX,XBAR,YBAR,
            SCALER,X1,Y1,X2,Y2,X3,Y3,X4,Y4;
      'REAL' 'ARRAY'
DUMMY,MTFNAM,PICNAM[1:15],
      TITLE,XAXIS,YAXIS[1:5];
      'INTEGER' 'PROCEDURE' INSTRARR(S,A);
      'STRING' S;
      'REAL' 'ARRAY' A;
      'EXTERNAL';
      'PROCEDURE' HGPLOT(X,Y,IC,L);
      'REAL' X,Y;
      'INTEGER' IC,L;
      'EXTERNAL';
'PROCEDURE' HGPTAPE(L,BCD,IS,IG,IR);
'INTEGER' L,IS,IG,IR;
'REAL' 'ARRAY' BCD;
'EXTERNAL';
      'PROCEDURE' HGPAXISV(X,Y,BCD,NC,S,THETA,XMIN,DX,GAP,NH);
      'REAL' X,Y,S,THETA,XMIN,DX,GAP;
      'INTEGER' NC,NH;
      'REAL' 'ARRAY' BCD;
      'EXTERNAL';
      'PROCEDURE' HGPSYMBL(X,Y,HEIGHT,BCD,THETA,N);
      'REAL' X,Y,HEIGHT,THETA;
      'INTEGER' N;
      'REAL' 'ARRAY' BCD;
      'EXTERNAL';
```

```
      SELECT INPUT(1);
      SELECT OUTPUT(2);
      'COMMENT'
      THE FULLOWING ARE NOW READ IN.
      MTFNAM=STRING OF 12 CHARACTERS NAMING MAG TAPE FILE
      PICNAM=STRING OF 12 CHARACTERS NAMING PIC TO BE PLOTTED
            TITLE=STRING OF 30 CHARACTERS FORMING JOB TITLE.
            XAXIS=STRING OF 20 CHARACTERS FORMING X-AXIS LABEL.
            YAXIS=STRING OF 20 CHARACTERS FORMING Y-AXIS LABEL.
N.B.  EACH STRING SHOULD BE ON A SEPARAIE CARD WITH * PUNCHED
      IN COLUMN 80
            SCALER=THE QUANTITY WHICH SCALES THE COURDINATES
            TO FIT THE PLOTTER PAPER.
            NELEM=THE NO. OF ELEMENTS IN THE STRUCTURE.
            NONOP=THE NO. OF NODES IN THE STRUCTURE.;
INSTRARR('('*')',MTFNAM);
INSTRARR('('*')',PICNAM);
      INSTRARR('('*')',TITLE);
      INSTRARR('('*')',XAXIS);
      INSTRARR('('*')',YAXIS);
      SCALER:=READ;
      NELEM:=READ;
      NONOP:=READ;
      'BEGIN'
            'REAL' 'ARRAY'
                  COORD[1:NONOP,1:2];
            'INTEGER' 'ARRAY'
                  NON[1:NELEM,1:8];
'COMMENT'
      THE COORDINATES OF THE NODES OF THE STRUCTURE ARE READ IN.
          NODE NO. 1  X COORDINATE--Y COORDINATE
           NODE NO. 2  X COORDINATE--ETC.;
'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
            COORD[I,J]:=READ;
PAPERTHROW;
WRITE TEXT
('('''('2C')''('6S')'NODE%NO.'('7S')'X%COORDINATE'('3S')'
Y%COORDINATE')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'BEGIN'
NEWLINE(1);
SPACE(8);
PRINT(I,3,0);
SPACE(8);
'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
'BEGIN'
PRINT(COORD[I,J],1,5);
SPACE(5);
'END';
'END';
'COMMENT'
      EACH ELEMENT NO. MATERIAL NO. ORIENTATION NO. AND NODE
      NOS. ARE NOW READ IN.
      THE NODE NOS. MUST BE GIVEN IN A CLOCKWISE DIRECTION.;
```

```
      'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
      'FOR' J:=1 'STEP' 1 'UNTIL' 8 'DO'
           NON[I,J]:=READ;
PAPERTHROW;
WRITE TEXT('('''('2C')'ELEMENT'('2S')'MATERIAL'('2S')'ORIENTATION
%%THICKNESS'('8S')'ELEMENT%NODE%NUMBERS
%%%(CLOCKWISE%DIRECTION)
'('1C')'%%%%%NO'('8S')'NO'('8S')'NO'('9S')'NO'('9S')'
I'('9S')'J'('9S')'K'('9S')'M'('1C')''')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
NEWLINE(1);
'FOR' J:=1 'STEP' 1 'UNTIL' 8 'DO'
PRINT(NON[I,J],7,0);
'END';
            'COMMENT'
            THE COORDINATES ARE NOW SCALED READY FOR PLOTTING.
            BY MULTIPLYING THEM BY SCALER.;
            'FOR' I:=1 'STEP' 1 'UNTIL' NONOP '40'
            'FOR' J:=1 'STEP' 1 'UNTIL' 2 'DO'
            COORD[I,J]:=COORD[I,J]*SCALER;
'COMMENT'
      A MAG TAPE FILE IS PICKED UP AND NAMED;
HGPTAPE(0,MTFNAM,0,0,0);
            'COMMENT'
            THE PLOTTER IS NOW INITIALISED.;
HGPLOT(0,0,0,0,15,1);
'COMMENT'
      THE SERIAL NO. AND PICTURE NAME ARE WRITTEN ON
      THE MAG. TAPE FILE.;
HGPTAPE(1,PICNAM,0,0,0);
            'COMMENT'
            THE ORIGIN IS SET AT Y VALUE OF 26 CM.
            AND X VALUE OF 0.5 CM.;
            HGPLOT(0.5,26.0,0,4);
            'COMMENT'
            THE X AND Y AXIS ARE NOW DRAWN AND LABELED.;
HGPAXISV(0.0,0.0,XAXIS,-25,20.0,0.0,0.0,1.0,5.08,4);
HGPAXISV(0.0,0.0,YAXIS,25,20.0,90.0,0.0,1.0,5.08,4);
            'COMMENT'
            THE PLOT TITLE IS DRAWN ON THE GRAPH.;
      HGPSYMBL(5.0,25.0,0.5,TITLE,0.0,35);
            'COMMENT'
EACH ELEMENTS NODAL COORDINATES ARE NOW SET
UP AND THE ELEMENT SUBSEQUENTLY DRAWN.;
'FOR' NE:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
      X1:=COORD[NON[NE,5],1];
      Y1:=COORD[NON[NE,5],2];
      X2:=COORD[NON[NE,6],1];
      Y2:=COORD[NON[NE,6],2];
      X3:=COORD[NON[NE,7],1];
      Y3:=COORD[NON[NE,7],2];
      X4:=COORD[NON[NE,8],1];
      Y4:=COORD[NON[NE,8],2];
```

```
                HGPLOT(X1,Y1,3,0);
            HGPLOT(X2,Y2,2,0);
                HGPLOT(X3,Y3,2,0);
                HGPLOT(X4,Y4,2,0);
                HGPLOT(X1,Y1,2,0);
            'END';
'COMMENT'
     THE NODE NOS. ARE NOW ADDED TO THE PICTURE;
'FOR' NE:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'BEGIN'
     INSTRARR('('*')',DUMMY);
     'IF' NE>99 'THEN' I:=4;
     'IF' NE<100 'THEN' I:=3;
     'IF' NE<10 'THEN' I:=2;
     HGPSYMBL(COORD[NE,1],COORD[NE,2],0.2,DUMMY,0,I);
     J:=READCH;
     'IF' J=CODE('(';')') 'THEN'
     'BEGIN'
          L1:  J:=READCH;
          'IF' J=CODE('('EL')') 'THEN'
          'GOTO' L2 'ELSE' 'GOTO' L1;
     'END';
     L2;
'END';
'COMMENT'
     THE ELEMENT NOS. ARE NOW ADDED TO HTS PICTURE;
'FOR' NE:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
     INSTRARR('('*')',DUMMY);
     'IF' NE>99 'THEN' I:=4;
     'IF' NE<100 'THEN' I:=3;
     'IF' NE<10 'THEN' I:=2;
XAV:= COORD[NON[NE,5],1] + COORD[NON[NE,6],1]
   + COORD[NON[NE,7],1] + COORD[NON[NE,8],1];
YAV:= COORD[NON[NE,5],2] + COORD[NON[NE,6],2]
   + COORD[NON[NE,7],2] + COORD[NON[NE,8],2];
XAV:=XAV/4;
YAV:=YAV/4;
HGPSYMBL(XAV,YAV,0,3,DUMMY,0,I);
     J:=READCH;
     'IF' J=CODE('(';')') 'THEN'
     'BEGIN'
     L3:  J:=READCH;
          'IF' J=CODE('('EL')') 'THEN'
     'GOTO' L4 'ELSE' 'GOTO' L3;
     'END';
L4;
'END';
          'COMMENT'
          THE PLOTTER BUFFER IS EMPTIED.;
          HGPLOT(0,0,0,0,0,2);
'COMMENT'
     THE MAG TAPE FILE IS NOW CLOSED;
HGPTAPE(2,DUMMY,0,0,0);
     'END';
'END' OF (PLOTMESH4N2D) WK. BEG. 24/7/1972.;
```

APPENDIX SEVEN

THE THREE-DIMENSIONAL FINITE ELEMENT ANALYSIS

PROGRAM LISTING (20-NODED ELEMENTS)

APPENDIX SEVEN

```
'LIST' (LP)
'PROGRAM' (ANISO3D20N)
'INPUT' 1=CRO
'OUTPUT' 2=LPO
'EXTENDED'
'BEGIN'
'COMMENT'
A PROGRAM FOR 3-DIMENSIONAL STRESS ANALYSIS USING 20-NODED,
3-DEGREES-OF-FREEDOM PER NODE,LINEARLY VARYING STRAIN
BRICK-SHAPED FINITE ELEMENTS.
THE MATERIALS CAN BE ORTHOTROPIC-THE PROPERTIES BEING FED IN
WITH RESPECT TO THE ELEMENTS LOCAL XBAR,YBAR,ZBAR,CARTESIAN
COORDINATE DIRECTIONS IN THE SEQUENCE
      EX,EY,EZ,MU-YX,MU-ZX,MU-ZY,G-XY,G-YZ,G-ZX.
THE ELEMENTS ORIENTATION MUST BE GIVEN IN DEGREES MEASURED FROM
THE +VE STRUCTURAL X,Y,Z AXES TO THE ELEMENTS +VE LOCAL
XBAR,YBAR,ZBAR AXES IN THE SEQUENCE
X;XBAR,Y;XBAR,Z;XBAR,X;YBAR,Y;YBAR,Z;YBAR,X;ZBAR,Y;ZBAR,Z;ZBAR.
THE STRUCTURE IS ARRANGED IN A RIGHT-HAND COORD. AXIS SYSTEM AND
THE NODES MUST BE NUMBERED IN THE +VE Z, Y AND X COORDINATE AXIS
DIRECTION RESPECTIVELY. THIS ENSURES THE ELEMENT NODE NOS.
(WHICH ARE NUMBERED IN THE SAME SYSTEM), INCREASE PROGRESSIVELY.
ELEMENT [K] FORMED USING NUMERICAL INTEGRATION.
14 POINT INTEGRATION. REF. IRONS.
THE MODIFIED FORM OF THE STRUCTURAL STIFFNESS MATRIX [MBK] IS
FORMED ON DISC BACKING STORE AND THE NODAL FORCE AND DISPLACEMENT
VECTORS ARE HELD IN CORE. THE EQUATIONS ARE SOLVED BY DIRECT
GAUSS ELIMINATION. NODAL DISPLACEMENTS ARE OUTPUT AND STRESS
AND STRAIN COMPONENTS ARE ALSO OUTPUT AT THE FOLLOWING POSITIONS
      POST. 1 CENTROID OF FACE AT XI = +1.
      POST. 2 CENTROID OF FACE AT XI = -1.
      POST. 3 CENTROID OF FACE AT ETA = +1.
      POST. 4 CENTROID OF FACE AT ETA = -1.
      POST. 5 CENTROID OF FACE AT ZETA = +1.
      POST. 6 CENTROID OF FACE AT ZETA = -1.
      POST. 7 CENTROID OF ELEMENT AT XI = ETA = ZETA = 0.
      POST. 8 ELEMENT SEQUENCE NODE NO.1.
      POST. 9 ELEMENT SEQUENCE NODE NO.3.
      POST.10 ELEMENT SEQUENCE NODE NO.6.
      POST.11 ELEMENT SEQUENCE NODE NO.8.
      POST.12 ELEMENT SEQUENCE NODE NO.13.
      POST.13 ELEMENT SEQUENCE NODE NO.15.
      POST.14 ELEMENT SEQUENCE NODE NO.18.
      POST.15 ELEMENT SEQUENCE NODE NO.20.
VERSION ALLOWS FOR ZERO-RESTRAINTS ONLY.
MK. 2 14 SEPT. 1972.
KEITH W. J. WRIGHT,
DEPT. MECH. ENG.
BRUNEL UNIVERSITY.
UXBRIDGE.!
'INTEGER'
      NRMBK,NCMBK,NRB,NCB,
```

```
      JOB,NELEM,NONOP,LO,CO,MANND,NOMAT;
'PROCEDURE' TIMENOW;
'EXTERNAL';
'PROCEDURE' COPYSTRING;

**SEE APPENDIX THREE FOR COMPLETE LISTING**

  'END' OF COPYSTRING;
'PROCEDURE' USESTORE(N,S,T,G,L);

**SEE APPENDIX THREE FOR COMPLETE LISTING**

'EXTERNAL';
'PROCEDURE' PUT PART(N,K,A,X,Y);

**SEE APPENDIX THREE FOR COMPLETE LISTING**

'EXTERNAL';
'PROCEDURE' GET PART(N,K,A,X,Y);

**SEE APPENDIX THREE FOR COMPLETE LISTING**

'EXTERNAL';
'PROCEDURE' MATINVERSE(A,N,INVERSEA);

**SEE APPENDIX THREE FOR COMPLETE LISTING**

'END' OF PROCEDURE MATINVERSE;
'PROCEDURE' MATMULT(A,B,C,X,Y,Z);

**SEE APPENDIX THREE FOR COMPLETE LISTING**

'END' OF MATMULT;
'PROCEDURE' BTDB(B,D,K,X,Y);
'VALUE' X,Y;
'INTEGER' X,Y;
'REAL' 'ARRAY' B,D,K;
'COMMENT'
      THIS PROCEDURE FORMS THE DIAGONAL TERMS AND THOSE ABOVE
      OF A SYMMETRIC FINITE ELEMENT STIFFNESS MATRIX [K].
      THE MATRIX [K] IS FORMED FROM THE TRIPLE PRODUCT [B]T[D][B],
      WHERE [B] IS OF ORDER (X BY Y),[D] OF ORDER (X BY X)
      AND [K] OF ORDER (Y BY Y),;
'BEGIN'
      'REAL' 'ARRAY' BTD[1:Y,1:X];
      'INTEGER' I,J,L;
  'FOR' I:=1'STEP' 1 'UNTIL' Y 'DO'
  'FOR' J:=1 'STEP' 1 'UNTIL' X 'DO'
'BEGIN'
      BTD[I,J]:=0;
        'FOR' L:=1 'STEP' 1 'UNTIL' X 'DO'
      BTD[I,J]:=BTD[I,J] + B[L,I]*D[L,J];
'END';
'FOR' I:=Y 'STEP' -1 'UNTIL' 1 'DO'
'BEGIN'
        'FOR' J:=1 'STEP' 1 'UNTIL' I 'DO'
```

```
            'FOR' L:=1 'STEP' 1 'UNTIL' X 'DO'
          K[J,I]:=K[J,I] + BTD[J,L]*B[L,I];
      'END';
'END' OF PROCEDURE RTDB.;
'PROCEDURE' DBLOKGAUSS(F,NCB,NRB);
'VALUE' NCB,NRB;
'INTEGER' NCB,NRB;
'REAL' 'ARRAY' F;
'COMMENT'
THIS PROCEDURE SOLVES THE SET OF LINEAR SIMULTANEOUS EQUATIONS
[MBK][D]=[F] USING THE GAUSSIAN ELIMINATION METHOD IN BLOCK FORM.
THE MATRIX [MBK] IS PARTITIONED INTO 3 BY 3 SUB-MATRICES
OR BLOCKS AND IS A MODIFIED ARRANGEMENT OF A BANDED SYMMETRICAL
MATRIX OF ORDER 3*NRB ELEMENTS SQUARE.
THE SOLUTION [D] WHICH IS WRITTEN-OVER THE MATRIX
[F], IS OBTAINED BY EFFECTIVELY REDUCING THE ORIGINAL SYMMETRIC
MATRIX TO AN UPPER TRIANGULAR MATRIX AND THEN BACK SUBSTITUTING.
THESE PROCESSES ARE CARRIED OUT DIRECTLY ON THE MODIFIED FORM
OF THE SYMMETRIC MATRIX, WHICH IS STORED ON DISC BACKING STORE
IN BLOCK FORM IN FILE KW-WORKFILE2. THE BLOCKS ARE STORED
AS A STRING BEGINNING AT BLOCK NCB=1 OF NRB=1 TO
NCB=NCB OF NRB=1 ETC. RIGHT THROUGH TO NCB=NCB
OF NRB=NRR. THE FORCE MATRIX [F] IS STORED IN CORE.;
'BEGIN'
'INTEGER'
S,T,
P,PP,PW,
II,JJ,K,I,J;
'REAL'
BUG;
'INTEGER' 'ARRAY'
MAP[1:NCB];
'REAL' 'ARRAY'
PRMBK,WRMBK[1:3,1:3*NCB],
FF1,FF2[1:3,1:1],A,A2,A3,INVERSEA[1:3,1:3],DD[1:NCB,1:3,1:3];
'FOR' I:=1 'STEP' 1 'UNTIL' NRB 'DO'
'BEGIN'
P:=PP:=9*NCB*(I-1)+1;
GET PART(10,P,PRMBK,PRMBK[1,1],PRMBK[3,3*NCB]);
'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' JJ:=1 'STEP' 1 'UNTIL' 3 'DO'
A[II,JJ]:=PRMBK[II,JJ];
MATINVERSE(A,3,INVERSEA);
'FOR' II:=3*I-2 'STEP' 1 'UNTIL' 3*I 'DO'
FF1[II-(3*I-3),1]:=F[II,1];
MATMULT(INVERSEA,FF1,FF2,3,3,1);
'FOR' II:=3*I-2 'STEP' 1 'UNTIL' 3*I 'DO'
F[II,1]:=FF2[II-(3*I-3),1];
'FOR' K:=NCB 'STEP' -1 'UNTIL' 1 'DO'
'BEGIN'
'IF' I-(NRB-MANND)+K>NCB 'THEN' 'GOTO' L1;
BUG:=0;
'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' JJ:=3*K-2 'STEP' 1 'UNTIL' 3*K 'DO'
'BEGIN'
DD[K,II,JJ-(3*K-3)]:=A[II,JJ-(3*K-3)]:=PRMBK[II,JJ];
```

```
BUG:=BUG+PRMBK[II,JJ];
'END';
'IF' BUG=0 'THEN'
'BEGIN'
MAP[K]:=0;
'GOTO' L1;
'END';
MATMULT(INVERSEA,A,A2,3,3,3);
'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' JJ:=3*K-2 'STEp' 1 'UNTIL' 3*K 'DO'
PRMBK[II,JJ]:=A2[II,JJ-(3*K-3)];
MAP[K]:=1;
L1:'END';
'FOR' J:=1 'STEP' 1 'UNTIL' NCB-1 'DO'
'BEGIN'
'IF' I+J>NRB 'THEN' 'GOTO' V1;
'IF' MAP[J+1]=0 'THEN' 'GOTO' V1;
'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' JJ:=1 'STEP' 1 'UNTIL' 3 'DO'
A3[JJ,II]:=DD[J*1,II,JJ];
P:=PW:=9*NCB*(I+J-1)+1;
GET PART(10,P,WRMBK,WRMBK[1,1],WRMBK[3,3*NCB]);
'FOR' K:=1 'STEP' 1 'UNTIL' NCB-J 'DO'
'BEGIN'
'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' JJ:=3*(J+K)-2 'STEP' 1 'UNTIL' 3*(J+K) 'DO'
A[II,JJ=(3*(J+K)-3)]:=PRMBK[II,JJ];
MATMULT(A3,A,A2,3,3,3);
S:=3*K-3;
'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' JJ:=S+1 'STEP' 1 'UNTIL' S+3 'DO'
WRMBK[II,JJ]:=WRMBK[II,JJ]-A2[II,JJ-S];
'END';
P:=PW;
PUT PART(10,P,WRMBK,WRMBK[1,1],WRMBK[3,3*NCB]);
MATMULT(A3,FF2,FF1,3,3,1);
'FOR' II:=3*(I+J)-2 'STEP' 1 'UNTIL' 3*(I+J) 'DO'
F[II,1]:=F[II,1]-FF1[II-(3*(I+J)-3),1];
V1:'END';
P:=PP;
PUT PART(10,P,PRMBK,PRMBK[1,1],PRMBK[3,3*NCB]);
'END' OF FORWARD ELIMINATION;
WRITE TEXT('('''('2C')'TIME%AT%END%OF%FORWARD%ELIMINATION')');
TIMENOW;
'FOR' I:=NRB-1 'STEP' -1 'UNTIL' 1 'DO'
'BEGIN'
P:=PW:=9*NCB*(I-1)+1;
GET PART(10,P,WRMBK,WRMBK[1,1],WRMBK[3,3*NCB]);
'FOR' J:=2 'STEP' 1 'UNTIL' NCB 'DO'
'BEGIN'
'IF' I+J-1>NRB 'THEN' 'GOTO' V2;
BUG:=0;
'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' JJ:=3*J-2 'STEP' 1 'UNTIL' 3*J 'DO'
'BEGIN'
A[II,JJ-(3*J-3)]:=WRMBK[II,JJ];
```

```
BUG:=BUG+WRMBK[II,JJ];
'END';
'IF' BUG=0 'THEN' 'GOTO' V2;
'FOR' II:=3*(I+J-1)-2 'STEP' 1 'UNTIL' 3*(I+J-1) 'DO'
FF1[II-(3*(I+J-1)-3),1]:=F[II,1];
MATMULT(A,FF1,FF2,3,3,1);
'FOR' II:=3*I-2 'STEP' 1 'UNTIL' 3*I 'DO'
F[II,1]:=F[II,1]-FF2[II-(3*I-3),1];
V2:'END';
'END' OF BACK SUBSTITUTION;
'END' OF PROCEDURE DBLOKGAUSS.;
SELECT INPUT(1);
SELECT OUTPUT(2);
'COMMENT'
     THE GENERAL STRUCTURE DATA ARE READ IN. THE SEQUENCE IS..
          THE JOB NO.
          'JOB TITLE' (IN QUOTES).
          NUMBER OF ELEMENTS.
          NUMBER OF NODES.
          NO. OF NODES WHERE LOADS ARE APPLIED.
          NO. OF NODES WHERE CONSTRAINTS ARE APPLIED.
          MAX. NODE NO. DIFFERENCE IN ANY ELEMENT.
          NO. OF MATERIALS IN STRUCTURE.;
PAPERTHROW;
     JOB:=READ;
WRITE TEXT('('!('2C')'JOB%NO.%%%')');
PRINT(JOB,3,0);
     NEWLINE(1);
COPYSTRING;
     NELEM:=READ;
     NONOP:=READ;
     LO:=READ;
     CO:=READ;
     MANND:=READ;
     NOMAT:=READ;
NRB:=NONOP;
NCB:=MANND+1;
NRMBK:=NONOP*3;
NCMBK:=3*(MANND+1);
WRITE TEXT('('!('2C')'NUMBER%OF%ELEMENTS%%%')');
PRINT(NELEM,3,0);
WRITE TEXT('('!('2C')'NUMBER%OF%NODAL%POINTS%%%')');
PRINT(NONOP,3,0);
WRITE TEXT
('('!('2C')'NO.%OF%NODES%WHERE%LOADS%ARE%APPLIED%%%')');
PRINT(LO,3,0);
WRITE TEXT
('('!('2C')'NO.%OF%NODES%WHERE%CONSTRAINTS%ARE%APPLIED%%%')');
PRINT(CO,3,0);
WRITE TEXT
('('!('2C')'MAX.%NODE%NO.%DIFFERENCE%IN%ANY%ONE%ELEMENT%%')');
PRINT(MANND,3,0);
WRITE TEXT('('!('2C')'NO.%OF%MATERIAL%TYPES%%%')');
PRINT(NOMAT,3,0);
WRITE TEXT('('!('2C')'NO.%OF%ELEMENTS%REQIRED%FOR%[MBK]=%%%')');
PRINT(NRMBK*NCMBK,0,6);
```

```
PAPERTHROW;
'BEGIN'
'REAL' 'ARRAY'
      HH,XI1,ETA1,ZETA1[1:27],
      MAT[1:NOMAT,1:9],TR,D2,M[1:6,1:6],A[1:3,1:3],
      COORD[1:NONOP,1:3],WMBK[1:3,1:3*NCB],XI2,ETA2,ZETA2[1:6],
      ELCO[1:20,1:3],K[1:60,1:60],LAM[1:3,1:20],
      JAY,INVERSEJ[1:3,1:3],
      INVJLAM[1:3,1:20],B[1:6,1:60],D,D1[1:6,1:6],
      DIS,F[1:NRMBK,1:1],ELDIS[1:60,1:1],
      STN,STS[1:6,1:1];
'REAL'
      H1,H2,H3,
      TERM,
      H,XI,ETA,ZETA,X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,X12,X13,
      X14,DETJ,TEMP;
'INTEGER'
   ZE,ET,XII,
      I,J,PREVTYPE,LOOP,P,NE,P1,P2,II,JJ,COUNTER,PREVMAT,POST;
'INTEGER' 'ARRAY'
      NON[1:NELEM,1:33];
'COMMENT'
      THE DATA REGARDING THE DIFFERENT MATERIALS OF THE ELEMENTS
      ARE NOW READ IN, THE SEQUENCE BEING
            EX,EY,EZ,MU-YX,MU-ZX,MU-ZY,G-XY,G-YZ,G-ZX.
      THESE PROPERTIES ARE IN THE DIRECTIONS OF THE ELEMENTS
      LOCAL COORDINATES, THE ANGLES BETWEEN THESE LOCAL ELEMENT
      AXES AND THE STRUCTURE GLOBAL AXES ARE FED IN WITH THE
      ELEMENT DATA AND USED AS DIRECTION COSINES FOR THE
      ELASTICITY TRANSFORMATION MATRIX [TR].;
'FOR' I:=1 'STEP' 1 'UNTIL' NOMAT 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 9 'DO'
      MAT[I,J]:=READ;
PAPERTHROW;
WRITE TEXT('('('2C')'MATERIAL'('24S')'M%A%T%E%R%I%A%L'('10S')'
P%R%O%P%E%R%T%I%E%S'('1C')'%%NO.'('7S')'EX'('10S')'EY'('10S')'
EZ'('10S')'MU-YX'('7S')'MU-ZX'('7S')'MU-ZY'('7S')'G-XY'('8S')'
G-YZ'('8S')'G-ZX')');
'FOR' I:=1 'STEP' 1 'UNTIL' NOMAT 'DO'
'BEGIN'
NEWLINE(1);
PRINT(I,3,0);
SPACE(3);
'FOR' J:=1 'STEP' 1 'UNTIL' 9 'DO'
'BEGIN'
      PRINT(MAT[I,J],0,2);
      SPACE(1);
'END';
'END';
'COMMENT'
      THE COORDINATES OF THE NODES OF THE STRUCTURE ARE READ IN,
      THE SEQUENCE IS...
            NODE NO.1 X-COORDINATE Y-COORDINATE Z-COORDINATE
            NODE NO. 2 X-COORDINATE Y-COURDINATE ETC.;
'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
```

```
      CUORD[I,J]:=READ;
PAPERTHROW;
WRITE TEXT('('''('1C')''('25S')'*****%S%T%R%U%C%T%U%R%E%%%%%
N%O%D%A%L%%%%%C%O%O%R%D%I%N%A%T%E%S%*****')');
WRITE TEXT
('('''('2C')''('6S')'NODE%NO.'('7S')'X%COORDINATE'('3S')'
Y%COORDINATE'('3S')'Z%COORDINATE'('1C')''')');
'FOR' I:=1 'STEP' 1'UNTIL' NONOP 'DO'
'BEGIN'
      NEWLINE(1);
SPACE(8);
PRINT(I,3,0);
SPACE(8);
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
PRINT(COORD[I,J],2,4);
SPACE(5);
'END';
'END';
'COMMENT'
      EACH ELEMENT DATA ARE NOW READ IN. THE SEQUENCE IS-
      ELEMENT NO.
      MATERIAL NO.
      ELEMENT TYPE NO.  THIS ALLOWS THE PROGRAM TO SKIP FORMING
            STIFFNESS MATRICES FOR ELEMENTS HAVING THE SAME
            DIMENSIONS AND MATERIAL PROPERTIES OF ITS PREDECESSOR.
      ANISOTROPIC NO.   IF THE ELEMENT MATERIAL AXES ARE
            COINCIDENT WITH THE STRUCTURE AXES A ZERO IS PUNCHED,
            IF NOT A 1.
      THE NINE ANISOTROPIC ANGLES-THE ANGLES BETWEEN THE STRUCTURE
      AXES X,Y,Z, AND THE ELEMENT MATERIAL AXES XBAR,YBAR,ZBAR.
      THE SEQUENCE IS
      Y:ZBAR,Z:ZBAR.  IF THE ANISOTROPIC  NO. =ZERO,
      X:XBAR,Y:XBAR,Z:XBAR,X:YBAR,Y:YBAR,Z:YBAR,X:ZBAR,
      THEN THE NINE ANGLES MUST BE ZERO.
      AND FINALLY THE 20 NODE NOS. GIVEN IN THE Z, Y AND X
      COORDINATE DIRECTIONS RESPECTIVELY, COMMENCING AT THE NODE
      NEAREST TO THE ORIGIN, I.E. THE SMALLEST NODE NO.
      THE STRUCTURE MUST BE NO. SUCH THAT THE ELEMENT NODE NOS.
      ARE IN AN ASCENDING ORDER WHEN NUMBERED IN THE Z,Y AND X
      CARTESIAN COORDINATE DIRECTIONS RESPECTIVELY.
      THIS SIMPLIFIES THE ELEMENT STIFFNESS DUMPING PROCESS;
'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 33 'DO'
      NON[I,J]:=READ;
PAPERTHROW;
WRITE TEXT
('('''('1C')''('25S')'*****%%%%%E%%%L%%%E%%%M%%%E%%%N%%%T
'('12S')'D%%%A%%%T%%%A%%%%%*****')');
WRITE TEXT
('('''('2C')'ELEMENT%MATER.%TYPE%ANISO.%ANISOTROPIC%%ANGLES
%%ELEMENT%%NODE%%NUMBERS%%(SEQUENCE%Z-Y-X%%COORDINATE%%DIRECTIONS)
'('1C')'%%%%%NO.%%%NO.%%%NO.%%%NO.%%%%X%%%%%Y%%%%%Z
%%%%%%1%%%%%%2%%%%%%3%%%%%%4%%%%%%5%%%%%%6%%%%%%7%%%%%%8%%%%%%9
%%%%%10'('1C')''('48S')'11%%%%%12%%%%13%%%%14%%%%15%%%%16
%%%%%17%%%%18%%%%19%%%%20'('1C')''')');
```

```
'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
NEWLINE(2);
SPACE(3);
      'FOR' J:=1 'STEP' 1 'UNTIL' 7 'DO'
      PRINT(NON[I,J],3,0);
      'FOR' J:=14 'STEP' 1 'UNTIL' 23 'DO'
      PRINT(NON[I,J],4,0);
      NEWLINE(1);
      SPACE(27);
      'FOR' J:=8 'STEP' 1 'UNTIL' 10 'DO'
      PRINT(NON[I,J],3,0);
      'FOR' J:=24 'STEP' 1 'UNTIL' 33 'DO'
      PRINT(NON[I,J],4,0);
      NEWLINE(1);
      SPACE(27);
      'FOR' J:=11 'STEP' 1 'UNTIL' 13 'DO'
      PRINT(NON[I,J],3,0);
'END';
'COMMENT'
      THE DISC BACKING STORE FILE IS OPENED,
      THE WORKING AREA OF CORE USED TO FORM THE
      ROW-BLOCKS OF [MBK] BEFORE TRANSFERRING TO BACKING
      FILE IS INITIALISED.  THIS IS THEN
      USED TO INITIALISE THE REQUIRED FILE AREA TO STORE [MBK].;
USESTORE(10,'('ED')','('KW-WORKFILE2')',1,NRMBK*NCMBK);
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 3*NCB 'DO'
WMBK[I,J]:=0;
'FOR' I:=1 'STEP' 1 'UNTIL' NRB 'DO'
'BEGIN'
P:=9*NCB*(I-1)+1;
PUT PART(10,P,WMBK,WMBK[1,1],WMBK[3,3*NCB]);
'END' OF INITIALISING DISC FILE AREA.;
'COMMENT'
      EACH ELEMENT STIFFNESS MATRIX IS NOW CALC. IF THE ELEMENT
      TYPE DIFFERS FROM ITS PREDECESSOR AND IS SUBSEQUENTLY
      DUMPED INTO [MBK] IN THE DISC BACKING FILE KW-WORKFILE2.
      I4 POINT INTEGRATION.;
HH[1]:=HH[2]:=HH[3]:=HH[4]:=HH[5]:=
HH[6]:=HH[7]:=HH[8]:=0.335180055;
HH[9]:=HH[10]:=HH[11]:=HH[12]:=HH[13]:=HH[14]:=0.886426593;
XI1[1]:=XI1[2]:=XI1[5]:=XI1[6]:=-0.758786911;
XI1[3]:=XI1[4]:=XI1[7]:=XI1[8]:=0.758786911;
ETA1[1]:=ETA1[2]:=ETA1[3]:=ETA1[4]:=0.758786911;
ETA1[5]:=ETA1[6]:=ETA1[7]:=ETA1[8]:=-0.758786911;
ZETA1[1]:=ZETA1[3]:=ZETA1[5]:=ZETA1[7]:=-0.758786911;
ZETA1[2]:=ZETA1[4]:=ZETA1[6]:=ZETA1[8]:=0.758786911;
XI1[9]:=XI1[11]:=XI1[12]:=XI1[14]:=0.0000;
ETA1[10]:=ETA1[11]:=ETA1[12]:=ETA1[13]:=0.0000;
ZETA1[9]:=ZETA1[10]:=ZETA1[13]:=ZETA1[14]:=0.0000;
XI1[13]:=ETA1[9]:=ZETA1[12]:=0.795822426;
XI1[10]:=ETA1[14]:=ZETA1[11]:=-0.795822426;
PREVTYPE:=0;
PAPERTHROW;
'FOR' NE:=1 'STEP' 1 'UNTIL' NELEM 'DO'
```

```
'BEGIN'
NEWLINE(1);
WRITE TEXT('('''('1C')'ELEMENT%NO,')');
PRINT(NON[NE,1],3,0);
'IF' NON[NE,3]=PREVTYPE 'THEN'
'BEGIN'
WRITE TEXT('('SAME%OVERALL%DIMENSIONS%&%MATERIAL%
PROPS,%AS%PREDECESSOR')');
'GOTO' SAMEK;
'END';
'COMMENT'
     THE ELEMENT STIFFNESS MATRIX [K] IS INITIALISED.;
'FOR' I:=1 'STEP' 1 'UNTIL' 60 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 60 'DO'
K[I,J]:=0;
'COMMENT'
     THE COORDINATES OF EACH ELEMENT ARE NOW SET UP.;
'FOR' I:=14 'STEP' 1 'UNTIL' 33'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
ELCO[I-13,J]:=COORD[NON[NE,I],J];
'COMMENT'
     THE [M] MATRIX FOR THIS MATERIAL IS SET UP. THIS IS THEN
     INVERTED TO FIND THE ELASTICITY MATRIX WHOSE
     PROPERTY AXES COINCIDES WITH THE STRUCTURE AXES.;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
     M[I,J]:=0;
J:=NON[NE,2];
M[1,1]:=1/MAT[J,1];
M[2,2]:=1/MAT[J,2];
M[3,3]:=1/MAT[J,3];
M[4,4]:=1/MAT[J,7];
M[5,5]:=1/MAT[J,8];
M[6,6]:=1/MAT[J,9];
M[1,2]:=M[2,1]:=-MAT[J,4]/MAT[J,2];
M[1,3]:=M[3,1]:=-MAT[J,5]/MAT[J,3];
M[2,3]:=M[3,2]:=-MAT[J,6]/MAT[J,3];
MATINVERSE(M,6,D1);
WRITE TEXT('('MATERIAL%NO')');
PRINT(NON[NE,2],3,0);
'IF' NON[NE,4]=1 'THEN'
'BEGIN'
'COMMENT'
     THE ARRAY OF THE NINE DIRECTION COSINES ARE SET UP.;
A[1,1]:=NON[NE,5];
A[2,1]:=NON[NE,6];
A[3,1]:=NON[NE,7];
A[1,2]:=NON[NE,8];
A[2,2]:=NON[NE,9];
A[3,2]:=NON[NE,10];
A[1,3]:=NON[NE,11];
A[2,3]:=NON[NE,12];
A[3,3]:=NON[NE,13];
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
A[I,J]:=COS(A[I,J]*3.14159265359/180);
```

A7.1

```
'COMMENT'
      THE ELASTICITY TRANSFORMATION MATRIX [TR] IS NOW SET UP.;
TR[1,1]:=A[1,1]*A[1,1];
TR[1,2]:=A[2,1]*A[2,1];
TR[1,3]:=A[3,1]*A[3,1];
TR[1,4]:=A[1,1]*A[2,1];
TR[1,5]:=A[2,1]*A[3,1];
TR[1,6]:=A[1,1]*A[3,1];
TR[2,1]:=A[1,2]*A[1,2];
TR[2,2]:=A[2,2]*A[2,2];
TR[2,3]:=A[3,2]*A[3,2];
TR[2,4]:=A[1,2]*A[2,2];
TR[2,5]:=A[2,2]*A[3,2];
TR[2,6]:=A[1,2]*A[3,2];
TR[3,1]:=A[1,3]*A[1,3];
TR[3,2]:=A[2,3]*A[2,3];
TR[3,3]:=A[3,3]*A[3,3];
TR[3,4]:=A[1,3]*A[2,3];
TR[3,5]:=A[2,3]*A[3,3];
TR[3,6]:=A[1,3]*A[3,3];                                          A7.1
TR[4,1]:=2*A[1,1]*A[1,2];
TR[4,2]:=2*A[2,1]*A[2,2];
TR[4,3]:=2*A[3,1]*A[3,2];
TR[4,4]:=A[1,1]*A[2,2]+A[2,1]*A[1,2];
TR[4,5]:=A[2,1]*A[3,2]+A[3,1]*A[2,2];
TR[4,6]:=A[1,1]*A[3,2]+A[3,1]*A[1,2];
TR[5,1]:=2*A[1,2]*A[1,3];
TR[5,2]:=2*A[2,2]*A[2,3];
TR[5,3]:=2*A[3,2]*A[3,3];
TR[5,4]:=A[1,2]*A[2,3]+A[2,2]*A[1,3];
TR[5,5]:=A[2,2]*A[3,3]+A[3,2]*A[2,3];
TR[5,6]:=A[1,2]*A[3,3]+A[3,2]*A[1,3];
TR[6,1]:=2*A[1,3]*A[1,1];
TR[6,2]:=2*A[2,3]*A[2,1];
TR[6,3]:=2*A[3,3]*A[3,1];
TR[6,4]:=A[1,3]*A[2,1]+A[2,3]*A[1,1];
TR[6,5]:=A[2,3]*A[3,1]+A[3,3]*A[2,1];
TR[6,6]:=A[1,3]*A[3,1]+A[3,3]*A[1,1];
'COMMENT'
[D2] IS NOW FORMED BY THE TRIPLE PRODUCT [TR]T[D1][TR] AND
      THE RESULT PUT IN [D1]. NECESSARY TERMS ARE REFLECTED.;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
      D2[I,J]:=0;
BTDB(TR,D1,D2,6,6);
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' J:=I 'STEP' 1 'UNTIL' 6 'DO'
D1[I,J]:=D2[I,J];
'FOR' I:=1 'STEP' 1 'UNTIL' 5 'DO'
'FOR' J:=I+1 'STEP' 1 'UNTIL' 6 'DO'
D1[J,I]:=D1[I,J];
WRITE TEXT('('ELASTICITY%MATRIX%TRANSFURMED%O.K.'('1C')''')');
'END' OF TRANSFORMING ELASTICITY MATRIX;
'FOR' LOOP:=1 'STEP' 1 'UNTIL' 14 'DO'
'BEGIN'
WRITE TEXT('('INT.%PT.%%')');
```

```
PRINT(LOOP,2,0);
      XI:=XI1[LOOP];
      ETA:=ETA1[LOOP];
      ZETA:=ZETA1[LOOP];
      H:=HH[LOOP];
'COMMENT'
      THE TERMS MAKING UP [LAMBDA] ARE NOW CALC.;
X1:=1+XI;
X2:=1-XI;
X3:=1+ETA;
X4:=1-ETA;
X5:=1+ZETA;
X6:=1-ZETA;
X7:=1-XI↑2;
X8:=1-ETA↑2;
X9:=1-ZETA↑2;
X10:=2*XI;
X11:=2*ETA;
X12:=2*ZETA;
X13:=0.125;
X14:=0.25;
'COMMENT'
THE MATRIX [LAMBDA] IS NOW SET UP.;
LAM[1,1]:=X13*X4*X6*(X10+ETA*ZETA+1);
LAM[1,2]:=-X14*X9*X4;
LAM[1,3]:=X13*X4*X5*(X10+ETA-ZETA+1);
LAM[1,4]:=-X14*X8*X6;
LAM[1,5]:=-X14*X8*X5;
LAM[1,6]:=X13*X3*X6*(X10-ETA+ZETA+1);
LAM[1,7]:=-X14*X9*X3;
LAM[1,8]:=X13*X3*X5*(X10-ETA-ZETA+1);
LAM[1,9]:=-0.5*XI*X4*X6;
LAM[1,10]:=-0.5*XI*X4*X5;
LAM[1,11]:=-0.5*XI*X3*X6;
LAM[1,12]:=-0.5*XI*X3*X5;
LAM[1,13]:=X13*X4*X6*(X10-ETA-ZETA-1);
LAM[1,14]:=X14*X9*X4;
LAM[1,15]:=X13*X4*X5*(X10-ETA+ZETA-1);
LAM[1,16]:=X14*X8*X6;
LAM[1,17]:=X14*X8*X5;
LAM[1,18]:=X13*X3*X6*(X10+ETA-ZETA-1);
LAM[1,19]:=X14*X9*X3;
LAM[1,20]:=X13*X3*X5*(X10+ETA+ZETA-1);
LAM[2,1]:=X13*X2*X6*(X11+XI+ZETA+1);
LAM[2,2]:=-X14*X9*X2;
LAM[2,3]:=X13*X2*X5*(X11+XI-ZETA+1);
LAM[2,4]:=-0.5*ETA*X2*X6;
LAM[2,5]:=-0.5*ETA*X2*X5;
LAM[2,6]:=X13*X2*X6*(X11-XI-ZETA-1);
LAM[2,7]:=X14*X9*X2;
LAM[2,8]:=X13*X2*X5*(X11-XI+ZETA-1);
LAM[2,9]:=-X14*X7*X6;
LAM[2,10]:=-X14*X7*X5;
LAM[2,11]:=X14*X7*X6;
LAM[2,12]:=X14*X7*X5;
LAM[2,13]:=X13*X1*X6*(X11-XI+ZETA+1);
```

A7.2

```
LAM[2,14]:=-X14*X9*X1;
LAM[2,15]:=X13*X1*X5*(X11-XI-ZETA+1);
LAM[2,16]:=-0.5*ETA*X1*X6;
LAM[2,17]:=-0.5*ETA*X1*X5;
LAM[2,18]:=X13*X1*X6*(X11+XI-ZETA-1);
LAM[2,19]:=X14*X9*X1;
LAM[2,20]:=X13*X1*X5*(X11+XI+ZETA-1);
LAM[3,1]:=X13*X2*X4*(X12*XI+ETA+1);
LAM[3,2]:=-0.5*ZETA*X2*X4;
LAM[3,3]:=X13*X2*X4*(X12-XI-ETA-1);
LAM[3,4]:=-X14*X8*X2;
LAM[3,5]:=X14*X8*X2;
LAM[3,6]:=X13*X2*X3*(X12*XI-ETA+1);
LAM[3,7]:=-0.5*ZETA*X2*X3;
LAM[3,8]:=X13*X2*X3*(X12-XI+ETA-1);
LAM[3,9]:=-X14*X7*X4;
LAM[3,10]:=X14*X7*X4;
LAM[3,11]:=-X14*X7*X3;
LAM[3,12]:=X14*X7*X3;
LAM[3,13]:=X13*X1*X4*(X12-XI*ETA+1);
LAM[3,14]:=-0.5*ZETA*X1*X4;
LAM[3,15]:=X13*X1*X4*(X12+XI-ETA-1);
LAM[3,16]:=-X14*X8*X1;
LAM[3,17]:=X14*X8*X1;
LAM[3,18]:=X13*X1*X3*(X12-XI-ETA+1);
LAM[3,19]:=-0.5*ZETA*X1*X3;
LAM[3,20]:=X13*X1*X3*(X12+XI+ETA-1);
'COMMENT'
     [JAY] IS NOW FORMED BY POST-MULTIPLYING [LAMBDA] BY
THIS ELEMENTS NODAL COORDINATES.;
MATMULT(LAM,ELCO,JAY,3,20,3);
'COMMENT'
THE DETERMINANT OF [J] IS NOW CALC.;
DETJ:=JAY[1,1]*(JAY[2,2]*JAY[3,3]-JAY[3,2]*JAY[2,3])
-JAY[2,1]*(JAY[1,2]*JAY[3,3]-JAY[3,2]*JAY[1,3])
+JAY[3,1]*(JAY[1,2]*JAY[2,3]-JAY[2,2]*JAY[1,3]);
'COMMENT'
THE INVERSE OF [J] IS NOW FOUND.;
MATINVERSE(JAY,3,INVERSEJ);
WRITE TEXT('('JAY%INVT%O,K')');
'COMMENT'
THE PRODUCT OF [INVERSEJ] AND [LAMBDA] IS NOW FOUND.;
MATMULT(INVERSEJ,LAM,INVJLAM,3,3,20);
'COMMENT'
THE MATRIX [B] IS NOW FORMED FROM THE TERMS OF [INVJLAM].;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 60 'DO'
B[I,J]:=0;
     'FOR' I:=1 'STEP' 1 'UNTIL' 20 'DU'
      'BEGIN'
           J:=3*I;
         B[1,J-2]:=B[4,J-1]:=B[6,J]:=INVJLAM[1,I];
      'END';
     'FOR' I:=1 'STEP' 1 'UNTIL' 20 'DO'
           'BEGIN'
           J:=3*I;
```

A7.2

```
            B[2,J-1]:=B[4,J-2]:=B[5,J]:=INVJLAM[2,I];
        'END';
      'FOR' I:=1 'STEP' 1 'UNTIL' 20 'DO'
        'BEGIN'
            J:=3*I;
          B[3,J]:=B[5,J-1]:=B[6,J-2]:=INVJLAM[3,I];
        'END';
'COMMENT'
      THE ELASTICITY MATRIX [D] INCORPORATING DETJ AND THE
      WEIGHTING FACTOR H IS SET UP FOR THIS INTEGRATING POINT;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
D[I,J]:=D1[I,J]*DETJ*H;
'COMMENT'
      FINALLY THE ELEMENT STIFFNESS MATRIX [K] FOR THIS GAUSS PT.
      IS OBTAINED BY FORMING THE TRIPLE PRODUCT [B]T[D][B]
      (INCLUDING WEIGHTING FACTOR AND DETJ).;
BTDB(B,D,K,6,60);
'END' ELEM. STIFF. FORM. FOR THIS GAUSS PT.;
'COMMENT'
      THE ELEMENT STIFFNESS MATRIX FORMED CONTAINS THE MAIN
      DIAGONAL TERMS AND THOSE ABOVE IT.
      THE OTHER TERMS REQUIRED BELOW THE DIAGONAL TERMS
      ARE OBTAINED BY REFLECTION.;
'FOR' I:=1 'STEP' 1 'UNTIL' 20 'DO'
'BEGIN'
J:=3*I-2;
K[J+1,J]:=K[J,J+1];
K[J+2,J]:=K[J,J+2];
K[J+2,J+1]:=K[J+1,J+2];
'END' OF [K] TERM REFLECTION.;
WRITE TEXT('('[K]%FORMED%O.K.')');
SAMEK;
'COMMENT'
      THE ELEMENT STIFFNESS MATRIX IS NOW DUMPED INTO [MBK].
      THIS IS DONE BY READING THE APPROPRIATE ROW BLOCKS FROM THE
      BACKING STORE FILE ON DISC, DUMPING THE CONTRIBUTIONS
      AND RE-WRITING THE RESULT INTO THE
      FILE INTO ITS ORIGINAL POSITION;
'FOR' I:=14 'STEP' 1 'UNTIL' 33 'DO'
'BEGIN'
P:=P1:=9*NCB*(NON[NE,I]-1)+1;
GET PART(10,P,WMBK,WMBK[1,1],WMBK[3,3*NCB]);
'FOR' J:=I 'STEP' 1 'UNTIL' 33 'DO'
'BEGIN'
'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' JJ:=1 'STEP' 1 'UNTIL' 3 'DO'
WMBK[II,3*(NON[NE,J]-NON[NE,I])+JJ]:=
WMBK[II,3*(NON[NE,J]-NON[NE,I])+JJ]+
K[3*(I-14) + II,3*(J-14) + JJ];
'END';
P:=P1;
PUT PART(10,P,WMBK,WMBK[1,1],WMBK[3,3*NCB]);
'END';
PREVTYPE:=NON[NE,3];
WRITE TEXT('('%%%DUMPED%O.K.')');
```

```
'END' OF MOD. STRUCT. STIFFN. FORMULATION.;
'COMMENT'
      THE DISPLACEMENT AND FORCE VECTORS ARE NOW FORMED.
      THE VECTOR NUMBERING SEQUENCE IS --
            NODE NO. 1 X-DIRECTION  --  DISPLACEMENT/FORCE
           NODE NO. 1 Y-DIRECTION  --  DISPLACEMENT/FORCE
            NODE NO. 1 Z-DIRECTION  --  DISPLACEMENT/FORCE
            NODE NO.2 X-DIRECTION  --  DISPLACEMENT/ ETC.
      THE NODAL DISPLACEMENTS CAN HAVE THE VALUE OF --
            1) ZERO -(IF IT IS FREE TO MOVE AND UNPRESCRIBED).
      OR    2) 0.000001 -(IF IT IS CONSTRAINED).
      THE NODAL FORCES CAN HAVE THE VALUE OF --
            1) ZERO -(THIS IMPLIES EITHER NO APPLIED FORCE OR THAT
                                    OF A CONSTRAINT).
         OR   2)ITS PRESCRIBED VALUE.
INITIALLY BOTH VECTORS ARE ZEROED AND THEN ONLY THE DATA OF
THE NODES WHERE THE CONSTRAINTS/LOADS ARE APPLIED ARE READ IN
FOR APPLIED CONSTRAINTS THE SEQUENCE IS..
      NODE NO.       X DIRECT.       Y DIRECT.       Z DIRECT.
IF THE DEGREE OF FREEDOM IS CONSTRAINED A ONE IS PUNCHED IF FREE
THEN A ZERO. (N.B. THE PROG. SETS THE CONSTRAINT TO 0.000001).
FOR APPLIED FORCES THE SEQUENCE IS...
      NODE NO.       X DIRECT.       Y DIRECT.       Z DIRECT.
IF THE DEGREE OF FREEDOM HAS AN APPLIED LOAD THE VALUE
OF THE LOAD IS PUNCHED IF NOT ZERO.
      FINALLY THE COMPLETED DISPLACEMENT VECTOR IS SCANNED AND IF
      THERE IS AN APPLIED CONSTRAINT THE APPROPRIATE TERMS IN [MBK]
      ARE ZEROED LEAVING A 1 IN THE MAIN DIAGONAL POSITION IN THE
      FIRST COLUMN BLOCK IN [MBK].;
'FOR' I:=1 'STEP' 1 'UNTIL' NRMBK 'DO'
DIS[I,1]:=F[I,1]:=0;
PAPERTHROW;
WRITE TEXT('('''('2C')'CO%%NO.'('8S')'NODE%NO.'('7S')'
APPLIED%DISPLACEMENTS%%(CONSTRAINT=0.000001)
%%INCHES'('1C')''('37S')'X'('21S')'Y'('21S')'Z')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' CO 'DO'
      'BEGIN'
            J:=READ;
NEWLINE(1);
SPACE(3);
PRINT(I,3,0);
SPACE(6);
PRINT(J,4,0);
JJ:=3*(J-1);
'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
      TEMP:=READ;
      'IF' TEMP#0 'THEN'
            'BEGIN'
            TEMP:=0.000001;
            DIS[JJ+II,1]:=TEMP;
            SPACE(10);
            PRINT(TEMP,1,6);
            SPACE(1);
            'END'
```

```
            'ELSE'
            'BEGIN'
            SPACE(14);
            PRINT(TEMP,1,0);
            SPACE(4);
            'END';
      'END';
      'END' OF SETTING UP DISPLACEMENT VECTOR;
PAPERTHROW;
WRITE TEXT('('''('2C')'LO%%NO.'('8S')'NODE%NO.'('17S')'
APPLIED%NODAL%FORCES%(LBF)
'('1C')''('37S')'X'('21S')'Y'('21S')'Z')');
NEWLINE(1);
'FOR' I:=1 'STEP' 1 'UNTIL' LO 'DO'
      'BEGIN'
            J:=READ;
NEWLINE(1);
SPACE(3);
PRINT(I,3,0);
SPACE(6);
PRINT(J,4,0);
JJ:=3*(J-1);
'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
      TEMP:=READ;
      'IF' TEMP#0 'THEN'
            'BEGIN'
            F[JJ+II,1]:=TEMP;
            SPACE(8);
            PRINT(TEMP,0,6);
            'END'
            'ELSE'
            'BEGIN'
            SPACE(13);
            PRINT(TEMP,1,0);
            SPACE(4);
            'END';
      'END';
      'END' OF SETTING UP LOAD VECTOR;
'FOR' I:=1 'STEP' 1 'UNTIL' NRB 'DO'
'BEGIN'
    P:=P1:=9*NCB*(I-1)+1;
    GET PART(10,P,WMBK,WMBK[1,1],WMBK[3,3*NCB]);
    'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
    'BEGIN'
          'IF' DIS[3*(I-1)+J,1]#0 'THEN'
          'BEGIN'
'FOR' II:=1 'STEP' 1 'UNTIL' NCMBK 'DO'
WMBK[J,II]:=0;
                'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
                 WMBK[II,J]:=0;
                 WMBK[J,J]:=1;
          'END';
     'END';
     'FOR' J:=1 'STEP' 1 'UNTIL' NCB-1 'DO'
     'BEGIN'
```

```
'IF' I+J>NRB 'THEN' 'GOTO' K1;
          'FOR' II:=1 'STEP' 1 'UNTIL' 3 'DO'
         'BEGIN'
          'IF' DIS[3*(I+J-1)+II,1]≠0 'THEN'
              'BEGIN'
               'FOR' JJ:=1 'STEP' 1 'UNTIL' 3 'DO'
              WMBK[JJ,3*J+II]:=0;
         'END';
    'END';
    K1:'END';
P:=P1;
PUT PART(10,P,WMBK,WMBK[1,1],WMBK[3,3*NCB]);
'END' OF MODS. TO [MBK] FOR APPLIED CONSTRAINTS.;
WRITE TEXT('('''('2C')'TIME%AT%COMMENCEMENT%OF%
EQUATION%SOLUTION')');
TIMENOW;
DBLOKGAUSS(F,NCB,NRB);
WRITE TEXT('('''('2C')'TIME%AT%COMPLETION%OF%
EQUATION%SOLUTION')');
TIMENOW;
PAPERTHROW;
'COMMENT'
THE NODAL POINT DISPLACEMENTS ARE NOW PRINTED OUT AS OUTPUT.;
WRITE TEXT('('''('2C')''('40S')'NODAL%%%DISPLACEMENTS%%(INCHES).
'('2C')''('8S')'NODE%NO.'('16S')'DX'('28S')'
DY'('28S')'DZ'('2C')'')');
'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'BEGIN'
NEWLINE(1);
SPACE(8);
PRINT(I,3,0);
SPACE(15);
PRINT(F[3*I-2,1],1,7);
SPACE(18);
PRINT(F[3*I-1,1],1,7);
SPACE(18);
PRINT(F[3*I,1],1,7);
'END';
'COMMENT'
     THE POSITIONS WHERE THE ELEMENT STRAIN AND STRESS
     COMPONENTS ARE EVALUATED ARE SET.;
'FOR' I:=1 'STEP' 1 'UNTIL' 15 'DO'
     XI1[I]:=ETA1[I]:=ZETA1[I]:=0;
XI1[2]:=ETA1[4]:=ZETA1[6]:=1;
XI1[1]:=ETA1[3]:=ZETA1[5]:=-1;
XI1[8]:=XI1[9]:=XI1[10]:=XI1[11]:=-1;
XI1[12]:=XI1[13]:=XI1[14]:=XI1[15]:=1;
ETA1[8]:=ETA1[9]:=ETA1[12]:=ETA1[13]:=-1;
ETA1[10]:=ETA1[11]:=ETA1[14]:=ETA1[15]:=1;
ZETA1[8]:=ZETA1[10]:=ZETA1[12]:=ZETA1[14]:=-1;
ZETA1[9]:=ZETA1[11]:=ZETA1[13]:=ZETA1[15]:=1;
COUNTER:=3;
PREVTYPE:=0;
'FOR' NE:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'BEGIN'
     PAPERTHROW;
```

```
WRITE TEXT('('''('2C')''('30S')'****ELEMENT%%CENTROIDAL%%STRESS%%
      AND%%STRAIN%%COMPONENTS****')');
WRITE TEXT('('''('4C')'ELEMENT%POST%%%%
STRESS%XX'('8S')'STRESS%YY'('8S')'STRESS%ZZ'('8S')'
STRESS%XY'('8S')'STRESS%YZ'('8S')'STRESS%ZX'('1C')'
%%NO,%%%%NO,%%%%STRAIN%XX'('8S')'STRAIN%YY'('8S')'STRAIN%ZZ
'('8S')'STRAIN%XY'('8S')'STRAIN%YZ'('8S')'STRAIN%ZX'('1C')''')');
'COMMENT'
      THE NODAL DISPLACEMENT VECTOR OF EACH ELEMENT IS SET UP.;
'FOR' I:=14 'STEP' 1 'UNTIL' 33 'DO'
'BEGIN'
      II:=3*(I-14);
      JJ:=3*(NON[NE,I]-1);
        'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
      ELDIS[II+J,1]:=F[JJ+J,1];
'END' OF SETTING UP ELMT. DISPL. VECTOR.;
'COMMENT'
      THE COORDINATES OF EACH ELEMENT ARE NOW SET UP.;
'FOR' I:=14 'STEP' 1 'UNTIL' 33 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
      ELCO[I-13,J]:=COORD[NON[NE,I],J];
'COMMENT'
      THE ELASTICITY MATRIX IS FORMED FOR THIS ELEMENT.
      N.B. NO WEIGHTING FACTOR OR DETJ REQD. FOR STRESS MATRIX;
'IF' NON[NE,3]#PREVTYPE 'THEN'
'BEGIN'
'COMMENT'
      THE [M] MATRIX FOR THIS MATERIAL IS SET UP, THIS IS THEN
      INVERTED TO FIND THE ELASTICITY MATRIX WHOSE
      PROPERTY AXES COINCIDES WITH THE STRUCTURE AXES.;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
      M[I,J]:=0;
J:=NON[NE,2];
M[1,1]:=1/MAT[J,1];
M[2,2]:=1/MAT[J,2];
M[3,3]:=1/MAT[J,3];
M[4,4]:=1/MAT[J,7];
M[5,5]:=1/MAT[J,8];
M[6,6]:=1/MAT[J,9];
M[1,2]:=M[2,1]:=-MAT[J,4]/MAT[J,2];
M[1,3]:=M[3,1]:=-MAT[J,5]/MAT[J,3];
M[2,3]:=M[3,2]:=-MAT[J,6]/MAT[J,3];
MATINVERSE(M,6,D1);
'IF' NON[NE,4]=1 'THEN'
'BEGIN'
'COMMENT'
      THE ARRAY OF THE NINE DIRECTION COSINES ARE SET UP.;


      **INSERT A7,1 AS ABOVE**


'END' OF TRANSFORMING ELASTICITY MATRIX;
```

```
      'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
      'FOR' J:=1 'STEP' 1 'UNTIL' 6 'DO'
           D[I,J]:=D1[I,J];
      PREVTYPE:=NON[NE,3];
'END' OF [D] FORMATION FOR THIS ELEMENT;
      'COMMENT'
      THE STRAIN AND STRESS COMPONENTS ARE NOW EVALUATED
      AT THE 15 POSITIONS AS GIVEN IN THE PROG. DESCRIPTION;
'FOR' POST:=1 'STEP' 1 'UNTIL' 15 'DO'
'BEGIN'
      XI:=XI1[POST];
      ETA:=ETA1[POST];
      ZETA:=ZETA1[POST];
'COMMENT'
      THE TERMS MAKING UP [LAMBDA] ARE NOW CALC.;



      **INSERT A7,2 AS ABOVE**



'COMMENT'
      [JAY] IS NOW FORMED BY POST-MULTIPLYING [LAMBDA] BY
THIS ELEMENTS NODAL COURDINATES.;
MATMULT(LAM,ELCO,JAY,3,20,3);
'COMMENT'
THE INVERSE OF [J] IS NOW FOUND.;
MATINVERSE(JAY,3,INVERSEJ);
'COMMENT'
THE PRODUCT OF [INVERSEJ] AND [LAMBDA] IS NOW FOUND.;
      MATMULT(INVERSEJ,LAM,INVJLAM,3,3,20);
'COMMENT'
THE MATRIX [B] IS NOW FORMED FROM THE TERMS OF [INVJLAM].;
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 60 'DO'
B[I,J]:=0;
      'FOR' I:=1 'STEP' 1 'UNTIL' 20 'DU'
       'BEGIN'
           J:=3*I;
          B[1,J-2]:=B[4,J-1]:=B[6,J]:=INVJLAM[1,I];
       'END';
      'FOR' I:=1 'STEP' 1 'UNTIL' 20 'DO'
           'BEGIN'
           J:=3*I;
            B[2,J-1]:=B[4,J-2]:=B[5,J]:=INVJLAM[2,I];
       'END';
      'FOR' I:=1 'STEP' 1 'UNTIL' 20 'DO'
       'BEGIN'
           J:=3*I;
          B[3,J]:=B[5,J-1]:=B[6,J-2]:=INVJLAM[3,I];
       'END';
'COMMENT'
      THE STRAINS ARE CALC. AT THIS POSITION IN THE ELEMENT BY
      POST-MULTIPLYING [B] BY THE ELEMENT NODAL DISPLACEMENTS;
MATMULT(B,ELDIS,STN,6,60,1);
```

```
'COMMENT'
      FINALLY THE STRESS COMPONENTS ARE CALC. AT THIS
      POSITION IN THE ELEMENT BY POST-MULTIPLYING [D] BY
      THE CORRESPONDING STRAINS [STN].;
MATMULT(D,STN,STS,6,6,1);
'COMMENT'
      BEFORE THE STRESS AND STRAIN COMPONENTS ARE OUTPUT
        THE THREE SHEAR STRAINS HAVE TO BE HALVED
          I.E. STRAIN XY=GAMMA XY/2.;
'FOR' I:=4 'STEP' 1 'UNTIL' 6 'DO'
STN[I,1]:=STN[I,1]/2;
NEWLINE(2);
PRINT(NON[NE,1],3,0);
SPACE(2);
PRINT(POST,2,0);
SPACE(3);
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN'
PRINT(STS[I,1],6,1);
SPACE(6);
'END';
NEWLINE(1);
SPACE(16);
'FOR' I:=1 'STEP' 1 'UNTIL' 6 'DO'
'BEGIN'
PRINT(STN[I,1],1,6);
SPACE(6);
'END';
'END' OF STS. & STN. CALC. FOR THIS ELEM. POSITION.;
COUNTER:=COUNTER+1;
'END' OF STS AND STN CALCS. FOR THIS ELEM.;
'END';
'END' OF ANISO3D2ON WK. BEG. 11 SEPT 1972;
```

APPENDIX EIGHT

THE THREE-DIMENSIONAL FINITE ELEMENT ANALYSIS

DATA CHECK PROGRAM LISTING (20-NODED ELEMENTS)

APPENDIX EIGHT

```
'LIST' (LP)
'LIBRARY' (ED,SUBGROUPSRA3)
'LIRRARY' (ED,SUBGROUPSRGP)
'LIBRARY' (ED,SUBGROUPS-RS)
'PROGRAM' (PLOTMESH20N3)
'COMPACT DATA'
'MIXED SEGMENTS'
'INPUT' 1=CR0
'OUTPUT' 2=LP0
'TRACE' 2
'BEGIN'
      'COMMENT'
      A PROGRAM FOR PLOTTING INDIRECTLY THE MESH PATTERN OF A
      STRUCTURE  TO BE ANALYSED USING K. WRIGHTS 3-DIMENSIONAL
      ANISOTROPIC FINITE ELEMENT ANALYSIS PROGRAM WHICH USES
      20-NODED-3-DEGREE OF FREEDOM PER NODE LINEAR STRAIN,
      (QUADRATIC DISPLACEMENT) FORMULATED FINITE ELEMENTS.
      KEITH W.J. WRIGHT,
      DEPT. MECH. ENG.,
      BRUNEL UNIVERSITY,
      UXBRIDGE.;
      'INTEGER'
            NELEM,NONOP,I,J,NE;
      'REAL'
            SCALER,X1,Y1,X2,Y2,X3,Y3,X4,Y4;
      'REAL' 'ARRAY'
      DUMMY[1:5],
MTFNAM,PICNAM[1:5],
      AN,TRAN[1:3,1:3],
      TITLE,XAXIS,YAXIS[1:5];
      'INTEGER' 'PROCEDURE' INSTRARR(S,A);
      'STRING' S;
      'REAL' 'ARRAY' A;
      'EXTERNAL';
      'PROCEDURE' HGPLOT(X,Y,IC,L);
      'REAL' X,Y;
      'INTEGER' IC,L;
      'EXTERNAL';
      'PROCEDURE' HGPAXISV(X,Y,BCD,NC,S,THETA,XMIN,DX,GAP,NH);
      'REAL' X,Y,S,THETA,XMIN,DX,GAP;
      'INTEGER' NC,NH;
      'REAL' 'ARRAY' BCD;
      'EXTERNAL';
      'PROCEDURE' HGPSYMBL(X,Y,HEIGHT,BCD,THETA,N);
      'REAL' X,Y,HEIGHT,THETA;
      'INTEGER' N;
      'REAL' 'ARRAY' BCD;
      'EXTERNAL';
'PROCEDURE' HGPTAPE(L,BCD,IS,IG,IR);
'INTEGER' L,IS,IG,IR;
'REAL' 'ARRAY' BCD;
'EXTERNAL';
```

```
'PROCEDURE' MATMULT(A,B,C,X,Y,Z);

**SEE APPENDIX THREE FOR COMPLETE LISTING**

'END' OF MATMULT;
      SELECT INPUT(1);
      SELECT OUTPUT(2);
      'COMMENT'
      THE FOLLOWING ARE NOW READ IN.
            MTFNAM=STRING OF 12 CHARACTERS NAMING MAG TAPE FILE
            PICNAM=STRING OF 12 CHARACTERS NAMING THE PIC TO BE
            PLOTTED. BOTH MTFNAM & PICNAM TO BE TERMINATED BY *
            SCALER=THE QUANTITY WHICH SCALES THE COORDINATES
            TO FIT THE PLOTTER PAPER.
            NELEM=THE NO. OF ELEMENTS IN THE STRUCTURE.
      NUNOP=THE NO. OF NODES IN THE STRUCTURE.
      THE NINE AXES TRANSFORMATION ANGLES BETWEEN THE
      PLOTTER AXES X,Y,Z, AND THE ROTATED STRUCTURAL AXES
XBAR,YBAR,ZBAR, THE SEQUENCE IS.
      X:XBAR,Y:XBAR,ZXBAR,X:YBAR,Y:YBAR,Z:YBAR,X:ZBAR,
      Y:ZBAR,Z:ZBAR;
      INSTRARR('('*')',MTFNAM);
      INSTRARR('('*')',PICNAM);
      SCALER:=READ;
      NELEM:=READ;
      NUNOP:=READ;
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
AN[I,J]:=READ;
WRITE TEXT('('''('2C')'TRANSFORMATION%ANGLES')');
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
      NEWLINE(1);
      'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
      PRINT(AN[I,J],3,2);
'END';
'COMMENT'
      THE ANGLES NOW HAVE THEIR COSINES TAKEN AND ARE
      ARRANGED INTO THE AXES TRANSFORMATION MATRIX [TRAN];
'FOR' I:=1 'STEP' 1 'UNTIL' 3 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
      TRAN[J,I]:=COS(AN[I,J]*3.14159265359/180);
      'BEGIN'
            'REAL' 'ARRAY'
      COORD,N[1:NONOP,1:3],
      CORD,ST[1:3,1:1],C[1:20,1:3];
            'INTEGER' 'ARRAY'
      NON[1:NELEM,1:33];
'COMMENT'
      THE COORDINATES OF THE NODES OF THE STRUCTURE ARE READ IN.
      THE SEQUENCE IS...
            NODE NO.1 X-COORDINATE Y-COORDINATE Z-COORDINATE
            NODE NO. 2 X-COORDINATE Y-COORDINATE ETC.;
'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
      COORD[I,J]:=READ;
```

```
PAPERTHROW;
WRITE TEXT('('''('1C')''('25S')'*****%S%T%R%U%C%T%U%R%E%%%%%
N%O%D%A%L%%%%%C%O%O%R%D%I%N%A%T%E%S%*****')');
WRITE TEXT
('('''('2C')''('6S')'NODE%NO.'('7S')'X%COORDINATE'('3S')'
Y%COORDINATE'('3S')'Z%COORDINATE'('1C')''')');
'FOR' I:=1 'STEP' 1'UNTIL' NONOP 'DO'
'BEGIN'
      NEWLINE(1);
SPACE(8);
PRINT(I,3,0);
SPACE(8);
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
PRINT(COORD[I,J],2,4);
SPACE(5);
'END';
'END';
'COMMENT'
      EACH ELEMENT DATA ARE NOW READ IN. THE SEQUENCE IS-
      ELEMENT NO,
      MATERIAL NO,
      ELEMENT TYPE NO,  THIS ALLOWS THE PROGRAM TO SKIP FORMING
            STIFFNESS MATRICES FOR ELEMENTS HAVING THE SAME
            DIMENSIONS AND MATERIAL PROPERTIES OF ITS PREDECESSOR.
      ANISOTROPIC NO,   IF THE ELEMENT MATERIAL AXES ARE
            COINCIDENT WITH THE STRUCTURE AXES A ZERO IS PUNCHED,
            IF NOT A 1,
      THE NINE ANISOTROPIC ANGLES-THE ANGLES BETWEEN THE STRUCTURE
      AXES X,Y,Z, AND THE ELEMENT MATERIAL AXES XBAR,YBAR,ZBAR,
      THE SEQUENCE IS
      X:XBAR,Y:XBAR,Z:XBAR,X:YBAR,Y:YBAR,Z:YBAR,X:ZBAR,
      Y:ZBAR,Z:ZBAR,
      IF THE ANISOTROPIC NO=ZERO THEN THE NINE ANGLES MUST BE
      ZERO, AND FINALLY THE 20 NODE NO, GIVEN IN THE Z,Y AND X
      COORDINATE DIRECTIONS RESPECTIVELY, COMMENCING AT THE NODE
      NEAREST TO THE ORIGIN, I.E. THE SMALLEST NODE NO,
      THE STRUCTURE MUST BE NUMBERED SUCH THAT THE ELEMENT NODE
      NUS, ARE IN AN ASCENDING ORDER WHEN NUMBERED IN THE Z,Y & X
      CARTESIAN COORDINATE DIRECTIONS RESPECTIVELY,
      THIS SIMPLIFIES THE ELEMENT STIFFNESS DUMPING PROCESS;
'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 33 'DO'
      NUN[I,J]:=READ;
PAPERTHROW;
WRITE TEXT
('('''('1C')''('25S')'*****%%%%%E%%%L%%%E%%%M%%%E%%%N%%%T
'('12S')'D%%%A%%%T%%%A%%%%%*****')');
WRITE TEXT
('('''('2C')'ELEMENT%MATER.%TYPE%ANISO.%ANISOTROPIC%%
ANGLES%%ELEMENT%%NODE%%NUMBERS%%(SEQUENCE%Z-Y-X%%COORDINATE%%
DIRECTIONS)'('1C')'%%%%%NO.%%%NO.%%%NO.%%%NO.%%%%%%%%%Y%%%%%Z
%%%%%%1%%%%%%2%%%%%%3%%%%%%4%%%%%%5%%%%%%6%%%%%%7%%%%%%8%%%%%%9
%%%%%10'('1C')''('48S')'11%%%%%12%%%%%13%%%%%14%%%%%15%%%%%16
%%%%%17%%%%%18%%%%%19%%%%%20'('1C')''')');
'FOR' I:=1 'STEP' 1 'UNTIL' NELEM 'DO'
```

```
'BEGIN'
NEWLINE(2);
SPACE(5);
      'FOR' J:=1 'STEP' 1 'UNTIL' 7 'DO'
      PRINT(NON[I,J],3,0);
      'FOR' J:=14 'STEP' 1 'UNTIL' 23 'DO'
      PRINT(NON[I,J],4,0);
      NEWLINE(1);
      SPACE(27);
      'FOR' J:=8 'STEP' 1 'UNTIL' 10 'DO'
      PRINT(NON[I,J],3,0);
      'FOR' J:=24 'STEP' 1 'UNTIL' 33 'DO'
      PRINT(NON[I,J],4,0);
      NEWLINE(1);
      SPACE(27);
      'FOR' J:=11 'STEP' 1 'UNTIL' 13 'DO'
      PRINT(NON[I,J],3,0);
'END';
            'COMMENT'
            THE COORDINATES ARE NOW SCALED READY FOR PLOTTING,
            BY MULTIPLYING THEM BY SCALER.;
            'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
      'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
            COORD[I,J]:=COORD[I,J]*SCALER;
'COMMENT'
      THE COORDINATES ARE NOW TRANSFORMED AND PUT IN [N];
'FOR' I:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'BEGIN'
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
CORD[J,1]:=COORD[I,J];
MATMULT(TRAN,CORD,ST,3,3,1);
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
N[I,J]:=ST[J,1];
'END' OF TRANS. COORDS;
PAPERTHROW;
WRITE TEXT('('''('2C')'TRANSFORMED%AND%SCALED%COORDINATES')');
WRITE TEXT('('''('1C')''('25S')'*****%S%T%R%U%C%T%U%R%E%%%%%
N%U%D%A%L%%%%%C%O%O%R%D%I%N%A%T%E%S%*****')');
WRITE TEXT
('('''('2C')''('6S')'NODE%NO.'('7S')'X%COORDINATE'('3S')'
Y%COORDINATE'('3S')'Z%COORDINATE'('1C')''')');
'FOR' I:=1 'STEP' 1'UNTIL' NONOP 'DO'
'BEGIN'
      NEWLINE(1);
SPACE(8);
PRINT(I,3,0);
SPACE(8);
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
'BEGIN'
PRINT(N[I,J],2,4);
SPACE(5);
'END';
'END';
'COMMENT'
      A MAG TAPE FILE IS PICKED UP AND NAMED;
      HGPTAPE(0,MTFNAM,0,0,0);
```

```
            'COMMENT'
            THE PLOTTER IS NOW INITIALISED.;
      HGPLOT(0,0,0.0,15,1);
'COMMENT'
      THE SERIAL NO. AND PICTURE NAME ARE WRITTEN ON THE
      MAG. TAPE FILE.;
      HGPTAPE(1,PICNAM,0,0,0);
            'COMMENT'
      THE ORIGIN IS SET AT Y XALUE OF 24 CM AND X VALUE OF 10 CM;
HGPLOT(10,24,0,4);
            'COMMENT'
            EACH ELEMENTS NODAL COORDINATES ARE NOW SET
            UP AND THE ELEMENT SUBSEQUENTLY DRAWN.;
            'FOR' NE:=1 'STEP' 1 'UNTIL' NELEM 'DO'
            'BEGIN'
'FOR' I:=14 'STEP' 1 'UNTIL' 33 'DO'
'FOR' J:=1 'STEP' 1 'UNTIL' 3 'DO'
C[I-13,J]:=N[NON[NE,I],J];
      HGPLOT(C[1,1],C[1,3],3,0);
HGPLOT(C[2,1],C[2,3],2,0);
HGPLOT(C[3,1],C[3,3],2,0);
HGPLOT(C[5,1],C[5,3],2,0);
HGPLOT(C[7,1],C[7,3],2,0);
HGPLOT(C[8,1],C[8,3],2,0);
HGPLOT(C[6,1],C[6,3],2,0);
HGPLOT(C[4,1],C[4,3],2,0);
HGPLOT(C[1,1],C[1,3],2,0);
HGPLOT(C[9,1],C[9,3],2,0);
HGPLOT(C[13,1],C[13,3],2,0);
HGPLOT(C[14,1],C[14,3],2,0);
HGPLOT(C[15,1],C[15,3],2,0);
HGPLOT(C[17,1],C[17,3],2,0);
HGPLOT(C[20,1],C[20,3],2,0);
HGPLOT(C[19,1],C[19,3],2,0);
HGPLOT(C[18,1],C[18,3],2,0);
HGPLOT(C[16,1],C[16,3],2,0);
HGPLOT(C[13,1],C[13,3],2,0);
      HGPLOT(C[6,1],C[6,3],3,0);
HGPLOT(C[11,1],C[11,3],2,0);
HGPLOT(C[18,1],C[18,3],2,0);
      HGPLOT(C[8,1],C[8,3],3,0);
HGPLOT(C[12,1],C[12,3],2,0);
HGPLOT(C[20,1],C[20,3],2,0);
      HGPLOT(C[3,1],C[3,3],3,0);
HGPLOT(C[10,1],C[10,3],2,0);
HGPLOT(C[15,1],C[15,3],2,0);
            'END';
'COMMENT'
      THE NODE NOS. ARE NOW ADDED TO THE PICTURE;
'FOR' NE:=1 'STEP' 1 'UNTIL' NONOP 'DO'
'BEGIN'
INSTRARR('('*')',DUMMY);
'IF' NE>99 'THEN' I:=4;
'IF' NE<100 'THEN' I:=3;
'IF' NE<10 'THEN' I:=2;
HGPSYMBL(N[NE,1],N[NE,3],0,2,DUMMY,0,I);
```

```
J:=READCH;
'IF' J=CODE('('!')') 'THEN'
'BEGIN'
      L1: J:=READCH;
      'IF' J=CODE('('EL')') 'THEN'
      'GOTO' L2 'ELSE' 'GOTO' L1;
'END';
L2:
'END';
            'COMMENT'
            THE PLOTTER BUFFER IS EMPTIED.;
            MGPLOT(0.0,0.0,0.0,2);
'COMMENT'
      THE MAG TAPE FILE IS NOW CLOSED;
      MGPTAPE(2,DUMMY,0,0,0);
      'END';
'END' OF (PLOTMESH2ON3D) WK BEG 22/1/75;
```