A PROGRAMMING APPROACH TO THE NUMERICAL ANALYSIS OF ELASTO-PLASTIC CONTINUA

by

C.T. Dittmer M.Sc. (Eng.)

A thesis submitted in fulfilment of the requirements
for the degree Doctor of Philosophy

Department of Civil Engineering
University of Cape Town

June 1978

ABSTRACT

The application of a kinematic minimum principle involving a continuous functional subject to inequality constraints is described for the incremental analysis of elasto-plastic continua. A simple algorithm is used for solution of the resulting mathematical programming problem. The formulation is presented for problems in plane stress, plane strain or axial symmetry, using triangular constant strain finite elements, and is extended to the use of cubic quadrilateral isoparametric elements for which a numerical integration technique is employed to account for elasto-plastic interfaces within elements. The material is assumed to obey the von Mises yield condition, and be either elastic-perfectly plastic or linear kinematic hardening. Computational details and solution techniques are described, and numerical examples compared with experimental and numerical results in the literature. Some assessment is made of the relative computational efficiency of the method.

DECLARATION OF CANDIDATE

I hereby declare that this thesis is my own work and that it has not been submitted for a degree at any other university.

Signed by candidate Signature Removed

C.T. Dittmer

ACKNOWLEDGEMENTS

I wish to express my thanks to the following:

Professor J.B. Martin, who supervised the research

AECI Ltd for generous financial support

Lee Behm for typing the manuscript

the operating staff of the Computer Centre at the University of Cape Town

CONTENTS

APPENDIX Computer Programs and User Manuals

NOTATION

α_{ij}	Weighting factor for Gauss quadrature
[B]	Deformation matrix
$C_{ijk\ell}$	Isotropic tensor relating elastic strain to stress
δ_{ij}	Weighted Cartesian strain tensor, $\delta_{ij} = V'\varepsilon_{ij}$
Δ	Area of triangular finite element
$D_{ijk\ell}$	Inverse of $C_{ijk\ell}$
[D]	Elasticity matrix
ε_{ij}	Cartesian strain tensor
e	(Superscript or subscript) denotes elastic part
E	Young's modulus
E_p	Plastic modulus
F_i	Body force
G	Positive definite scalar
η	Ratio of equivalent stress to yield stress for a point to be treated as plastic
h_i	Shape function for isoparametric finite element
J	Jacobian operator
[K*]	System matrix
λ	Non-negative scalar
Λ	Non-negative scalar
ν	Poisson's ratio
n	Number of displacement degrees of freedom
[N]	Gradient matrix
p	(Superscript or subscript) denotes plastic part
p	Number of plastic multipliers
P	Nodal load

r,s	Natural coordinate system for isoparametric finite element
r,z,θ	Polar coordinate system
σ_{ij}	Cartesian stress tensor
$\bar{\sigma}_{ij}$	$\sigma_{ij} - E_p \varepsilon^p_{ij}$
σ_{eq}	von Mises equivalent stress
σ_o	Uniaxial yield stress
S	Surface of Body
T_i	Surface traction
u_i	Displacement
u,v	Displacement components
U^o_p, W^o	Functionals used in kinematic minimum principle
$\bar{U}^o_p$, $\bar{W}^o$	Functionals used in extended kinematic minimum principle
ϕ	Yield function
V	Volume of body
w	Band width of elastic system stiffness matrix
x_i	Cartesian coordinate
x,y,z	Cartesian coordinate system
'	(Prime) denotes quantity associated with a finite element

C H A P T E R 1

INTRODUCTION

The simplicity of the laws governing the mechanical behaviour of elastic solids in many cases permits an analytical solution. In elastic problems of greater complexity with regard to geometry, boundary conditions, loading or nonhomogeneous material properties, recourse is generally made to numerical solutions which exploit the minimum principles governing the mechanical behaviour of elastic solids. In particular the finite element method has been extensively used to determine approximate solutions by discretizing the description of the spatial field to a finite number of parameters.

For elasto-plastic solids the complexity of mechanical behaviour generally prohibits an analytical solution. Although the minimum principles governing the mechanical behaviour of elasto-plastic solids have been established for some time (Prager [1], [2]; Hodge and Prager [3]; Greenberg [4], [5] and Koiter [6]), it was only with developments in numerical methods that their exploitation become feasible. More recently Ceradini [7] and Maier [8] have given alternative forms for the minimum principles using elastic solutions for residual fields. Maier's kinematic minimum principle was derived from quadratic programming arguments.

Direct methods of elasto-plastic analysis which use the minimum principles and the finite element method include initial strain/initial stress and tangent modulus approaches. In the initial strain method devloped by Galagher, Padlog and Bijlaard [9], and Argyris [10], plastic strains during a load increment are treated as initial strains, the system stiffness matrix remaining elastic and unchanged. This iterative procedure fails for an elastic-perfectly plastic material as plastic strain increments are not uniquely described. The initial stress approach of Zienkiewicz, Valliapan and King [11] entails the iterative elastic distribution of 'initial stresses' until the requirements of equilibrium, the kinematic relations and constitutive laws are satisfied, again the system matrix remaining unchanged. Since the stress distribution is uniquely described by increments of strain, ideal plasticity can be accommodated. For the initial strain/initial stress methods the system matrix need be inverted once only, however the number of iterations required for convergence at each load increment may increase as plastic strain increments become larger. In these methods elastic unloading is automatically accommodated as the system matrix always reflects elastic stress-strain relations.

The tangent modulus or variable stiffness method of Pope [12], Swedlow [13], Marcal and King [14], and Yamada, Yoshimura and Sakurai [15] requires reformulation of the system matrix at each stage in the incremental loading procedure, taking account of adjustment to stress-strain relations due to plastic strains. Further, iteration is required for any load increment in which elastic unloading occurs as the system matrix must reflect the true stress-strain relations. This method can be used for perfectly plastic materials. Marcal [16], in comparing the two methods, derived the initial strain formulation from the tangent modulus approach.

More recently efforts have been directed towards formulating elasto-plastic problems as formal mathematical programming problems and to use standard programming techniques to determine a solution. Some examples are the work of Hodge, Belytscho and Herakovich [17], Sayegh and Rubenstein [18], Giacomini, Maier and Paterlini [19], de Donato and Maier [20], and Anand and Garg [21]. As shown by de Donato and Franchi [22] a linear complementary problem emerges which is fully equivalent to two dual quadratic programming problems. The solution can be determined from any one of the five formulations. However, in applying nonlinear programming algorithms, the size of matrices to be calculated for any of the formulations is prohibitive even for small numbers of finite elements. The most efficient formal programming technique for incremental elasto-plastic analysis appears to be the 'multistage loading' and 'reduced problem technique' used by de Donato and Franchi [22] and de Donato and Maier [20]. For this the yield surface is piecewise linearized, transforming the domain of permissible stress states into a series of linear inequalities. Since a plastic multiplier and linear inequality are associated with each yield plane the numbers of variables and constraints increases rapidly. 'Multistage loading' consists of the initial division of the load into a given number of 'sub-loads' each of which is increased from zero to its final value. Elastic unloading can only be considered at the beginning of each loading stage. To decrease the large number of variables the 'reduced problem technique' is employed in which yielding modes that appear unlikely to be activated are omitted from that stage of the problem, and any standard nonlinear programming technique used to determine the solution. Violation of ignored constraints necessitates iteration. To the writer's knowledge there is an absence in the literature of numerical applications of the formal mathematical programming approach to three-dimensional continuum problems, which though conceptually quite feasible, must present a formidable computational task.

In investigating Maier's theorem [8] derived from quadratic programming arguments, Martin [23] has given a simpler result in a kinematic minimum principle for the rate problem in elasto-plasticity, involving a continuous functional subject to inequality constraints. This was applied in incremental form to the plane truss problem by Martin and Reddy [24], resulting in a quadratic programming problem. A simple algorithm was suggested in which the programming problem reduced to solution of simultaneous linear equations subject to checks on constraints, violation of which necessitates iteration. In this thesis application of this minimum principle is extended to the incremental analysis of elasto-plastic continua. For the resulting programming problem it is not necessary to piecewise linearize the yield surface as in the formal quadratic programming approach, and thus the continuously differentiable von Mises yield function is assumed. For simplicity discussion will be limited to two-dimensional problems (plane stress, plane strain and axial symmetry); however the extension to general three-dimensional continua is directly obtained by inclusion of field variable components ignored in the two-dimensional case.

CHAPTER 2

SOME FUNDAMENTAL CONCEPTS

2.1 Introduction

In this thesis discussion is limited to bodies composed of an isotropic, homogeneous material. In the plastic range the material is assumed to obey the von Mises yield condition, and be either elastic-perfectly plastic, or linear kinematic hardening. Deformations are assumed to be isothermal and small in the sense that kinematic relations are linear in strain and displacement and equilibrium equations linear in stress and force. Loading of the body is assumed to be quasi-static so that inertia terms can be ignored.

In developing the general relationships governing the deformation of an elasto-plastic continuum, consider a body of volume V and surface S in a Cartesian coordinate system $x_i (i = 1,2,3)$. The body is subjected to body forces $F_i(x_k)$ on V, and surface tractions $T_i(x_k)$ on part of the surface S_T. On the remainder of the surface S_u displacements $u_i(x_k)$ are prescribed. The governing relations comprise equilibrium equations, kinematic relations and constitutive relations.

2.2 Equilibrium

The equilibrium equations are characterized by

$$\frac{\partial \sigma_{ij}}{\partial x_j} + F_i = 0 \quad \text{on} \quad V, \tag{2.1}$$

$$\sigma_{ij} = \sigma_{ji}, \tag{2.2}$$

and $$\sigma_{ij}\nu_j = T_i \quad \text{on} \quad S, \tag{2.3}$$

where $\sigma_{ij}(i,j = 1,2,3)$ is the stress tensor, and ν_i is the outward normal vector at a point on the surface. A statically admissible set of body forces F_i, surface tractions T_i, and stresses σ_{ij} must satisfy equations (2.1), (2.2) and (2.3).

2.3 Kinematic Relations

The strain field ε_{ij} is obtained from the displacement field u_i by means of the strain-displacement relations

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{2.4}$$

Further, the compatibility condition ensures the integrability of the strain field to within a rigid body motion, and may be expressed as

$$\frac{\partial^2 \varepsilon_{ij}}{\partial x_k\, \partial x_\ell} + \frac{\partial^2 \varepsilon_{k\ell}}{\partial x_i\, \partial x_j} = \frac{\partial^2 \varepsilon_{ik}}{\partial x_j\, \partial x_\ell} + \frac{\partial^2 \varepsilon_{j\ell}}{\partial x_i\, \partial x_k}. \tag{2.5}$$

A kinematically admissible set of strains and displacements must satisfy equations (2.4) and (2.5).

2.4 Constitutive Relations

The constitutive relations are written after breaking the strain tensor ε_{ij} into an elastic part ε^e_{ij} and a plastic part ε^p_{ij} such that

$$\varepsilon_{ij} = \varepsilon^e_{ij} + \varepsilon^p_{ij} . \tag{2.6}$$

The elastic strain and stress are linearly related:

$$\varepsilon^e_{ij} = C_{ijk\ell}\, \sigma_{k\ell} , \tag{2.7}$$

where $C_{ijk\ell}$ is an isotoropic fourth order tensor.

A yield function ϕ is introduced to describe plastic behaviour. In this thesis it is assumed that the yield function is a convex continuously differentiable scalar function and may be written as $\phi = \phi(\sigma_{ij})$ for elastic-perfectly plastic materials. For elastic-plastic or hardening behaviour we shall limit our discussion to a linear kinematic hardening model for the material. In this case the subsequent yield function may be written as $\phi = \phi(\sigma_{ij}, \varepsilon^p_{ij})$, where the plastic strains ε^p_{ij} represent the history of plastic strain from the virgin unstressed state, [25].

Yielding occurs when $\phi = 0$ and stress states such that $\phi > 0$ are inadmissible. The plastic strains remain unchanged for any stress increment imposed on a stress state for which $\phi < 0$ (elastic behaviour), or for which $\phi = 0$ and $\frac{\partial \phi}{\partial \sigma_{ij}} d\sigma_{ij} < 0$ (unloading). Thus changes in plastic strains can only occur for stress increments imposed on stress states for which $\phi = 0$ (yielding) and $\frac{\partial \phi}{\partial \sigma_{ij}} d\sigma_{ij} \geq 0$ (loading or neutral loading).

Figure 2.1 shows directions of stress increments from points on the yield surface $\phi = 0$ in stress space. In the case of hardening $\phi = \phi(\sigma_{ij}, \varepsilon^p_{ij})$ and thus the curve $\phi = 0$ in stress subspace represents the current yield surface corresponding to current plastic strains.

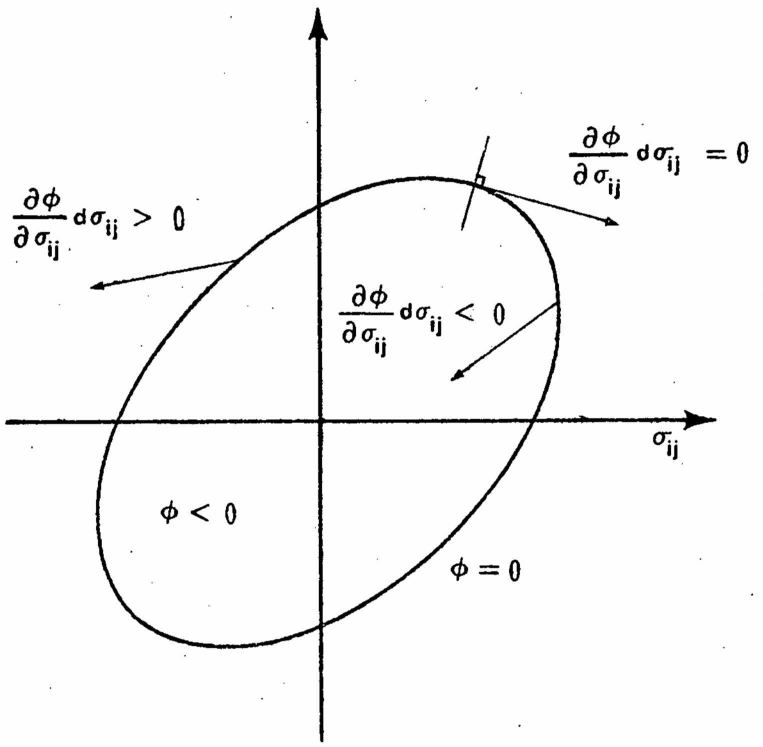


Figure 2.1 Directions of stress increments in stress space

When plastic strain changes do occur the plastic strain increment is proportional to the gradient of the yield function. Defining a non-negative scalar field $\lambda(x_i)$, we write the plastic strain increment as

$$d\varepsilon_{ij}^{p} = 0 \quad \text{if } \phi < 0,$$

$$\text{or } \phi = 0 \text{ and } \frac{\partial \phi}{\partial \sigma_{ij}} d\sigma_{ij} < 0; \qquad (2.8)$$

and $$d\varepsilon_{ij}^{p} = \lambda \frac{\partial \phi}{\partial \sigma_{ij}} \quad \text{if } \phi = 0 \text{ and } \frac{\partial \phi}{\partial \sigma_{ij}} d\sigma_{ij} \geq 0. \qquad (2.9)$$

In the case of an elastic-plastic or hardening material

$$\lambda = G \frac{\partial \phi}{\partial \sigma_{k\ell}} d\sigma_{k\ell}, \tag{2.10}$$

where the scalar G is positive definite. We shall consider the more general problem of a hardening solid as the elastic-perfectly plastic problem can be recovered as a special case.

For perfect plasticity stress changes such that $\frac{\partial \phi}{\partial \sigma_{ij}} d\sigma_{ij} > 0$ are inadmissible since they lead to stress states for which $\phi(\sigma_{ij}) > 0$. From equation (2.9) it follows that λ can then only be non-zero when $\frac{\partial \phi}{\partial \sigma_{ij}} d\sigma_{ij} = 0$, and hence in equation (2.10) λ can only be finite and non-zero if $G \to \infty$. Thus the elastic-perfectly plastic case is recovered from the elastic plastic case as the limit $G \to \infty$, in which case λ is non-negative but otherwise undetermined.

2.5 Von Mises Yield Condition and Linear Kinematic Hardening

The von Mises initial yield condition [26] assumes that plastic deformation becomes possible when the shear stress on a particular plane, the octahedral plane which is equally inclined to the three principal axes, reaches a limiting magnitude k. This is conventionally written in quadratic form so that the von Mises initial yield function is

$$\phi = \sigma_{so}^2 - k^2, \tag{2.11}$$

where σ_{so} is the octahedral shear stress. Expanding σ_{so} in terms of stress components in arbitrary Cartesian coordinate directions this becomes

$$\phi = \frac{1}{3}\{\sigma_{ij}\,\sigma_{ij} - \frac{1}{3}(\sigma_{kk})^2\} - k^2. \tag{2.12}$$

For a hardening material the yield function is $\phi = \phi(\sigma_{ij},\varepsilon^p_{ij})$. Adopting a kinematic hardening model such that subsequent yield surfaces are translations of the initial yield surface in stress space, retaining a constant shape, size and orientation, we may write the von Mises subsequent yield function as

$$\phi = \frac{1}{3}\{(\sigma_{ij} - c\varepsilon^p_{ij})(\sigma_{ij} - c\varepsilon^p_{ij}) - \frac{1}{3}(\sigma_{kk} - c\varepsilon^p_{kk})^2\} - k^2, \tag{2.13}$$

where c is a constant. (The term $c\varepsilon^p_{kk}$ vanishes since there is zero volume change associated with plastic deformation).

The most convenient idealization for the scalar hardening coefficient G in equation (2.10) is

$$\frac{1}{G} = c\left(\frac{\partial\phi}{\partial\sigma_{k\ell}}\frac{\partial\phi}{\partial\sigma_{k\ell}}\right). \tag{2.14}$$

This leads to a bilinear stress-strain curve in a monotonic loading test in simple tension, from which $c = E_p$, the plastic modulus, [25].

Thus writing

$$\bar{\sigma}_{ij} = (\sigma_{ij} - E_p\varepsilon^p_{ij}), \tag{2.15}$$

the von Mises yield function becomes

$$\phi = \frac{1}{3}\{\bar{\sigma}_{ij}\,\bar{\sigma}_{ij} - \frac{1}{3}(\bar{\sigma}_{kk})^2\} - k^2, \tag{2.16}$$

where $\bar{\sigma}_{kk} = \sigma_{kk}$. The limiting value of octahedral shear stress k is conventionally related to the uniaxial yield stress σ_o as

$$k^2 = \frac{2}{9}\sigma_o^2, \tag{2.17}$$

giving finally

$$\phi = \frac{1}{3}\{\bar{\sigma}_{ij}\,\bar{\sigma}_{ij} - \frac{1}{3}(\bar{\sigma}_{kk})^2 - \frac{2}{3}\sigma_o^2\}. \tag{2.18}$$

The elastic-perfectly plastic case is recovered by setting $E_p = 0$, in which case $\bar{\sigma}_{ij} = \sigma_{ij}$.

2.6 Plane Stress

In formulating particular continuum analysis problems it is often possible to reduce the complexity of a general three-dimensional formulation to one of two dimensions. Such idealizations include problems in plane stress, plane strain and axial symmetry. In this thesis we shall limit our discussion to this class of problems, although the formulation presented may be readily extended to general three-dimensional continua.

In the case of plane stress we consider thin sheets (plates) of material subjected to loads and imposed displacements at the boundary of the sheet and in the plane of the sheet. Let such a sheet lie in the x,y plane of a Cartesian coordinate system x,y,z. The non-zero stress components are σ_{xx}, σ_{yy} and σ_{xy}, which are assumed to be constant through the thickness of the sheet (z-direction), while the components σ_{zz}, σ_{yz} and σ_{zx} are taken to be zero throughout the body.

In consequence the associated strain components ε_{zz}, ε_{yz} and $\varepsilon_{zx}^{\dagger}$ are ignored in the analysis, which leads to minor violations of the compatibility condition (equation 2.5). However, the assumptions of plane stress provide very good approximations for thin plates under in-plane loading.

If u and v are respectively the displacement components in the x and y directions, then the strain-displacement relations (equation 2.4) reduce to

$$\varepsilon_{xx} = \frac{\partial u}{\partial x}, \quad \varepsilon_{yy} = \frac{\partial v}{\partial y}, \quad \varepsilon_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{2.19}$$

For an isotropic material the elastic constitutive relations (equation 2.7) are

$$\varepsilon^e_{xx} = \frac{1}{E}(\sigma_{xx} - \nu\sigma_{yy}),$$

$$\varepsilon^e_{yy} = \frac{1}{E}(\sigma_{yy} - \nu\sigma_{xx}), \tag{2.20}$$

and $$\varepsilon^e_{xy} = \frac{2(1+\nu)}{E}\sigma_{xy},$$

where E is the elastic modulus and ν is Poisson's ratio. Further, the von Mises yield condition for a kinematic hardening material (equation 2.18) in plane stress reduces to

---

† shear strains with Cartesian subscripts x,y,z denote engineering (and not tensorial) shear strains,

e.g. $\varepsilon_{xy} = \varepsilon_{12} + \varepsilon_{21} = 2\varepsilon_{12} = 2\varepsilon_{21}$.

$$\phi = \frac{2}{9}\{\bar{\sigma}_{xx}^2 + \bar{\sigma}_{yy}^2 - \bar{\sigma}_{xx}\bar{\sigma}_{yy} + 3\bar{\sigma}_{xy}^2 - \sigma_o^2\}, \qquad (2.21)$$

where $\bar{\sigma}_{xx} = \sigma_{xx} - E_p\varepsilon_{xx}^p$, etc.

2.7 Plane Strain

A body is considered to be in a state of plane strain if it extends a large (theoretically infinite) distance in, say, the z-direction, and has boundary conditions independent of z. In this case a representative sheet of unit thickness is considered in the analysis, as the displacement components u,v are functions of x and y only, and displacement in the z-direction is zero. It is evident that the strain components ε_{zz}, ε_{yz} and ε_{zx} are zero.

The stress components σ_{xx}, σ_{yy} and σ_{xy} can be non-zero, but although the shear components σ_{yz} and σ_{zx} are taken to be zero throughout the body, in general σ_{zz} does not vanish. Thus if plastic deformation occurs ($\phi = 0$) then

$$d\varepsilon_{zz}^p = \lambda \frac{\partial\phi}{\partial\sigma_{zz}} . \qquad (2.22)$$

However, plane strain assumptions give the total strain in the z-direction to be zero. Therefore, from the incremental form of equation (2.6) we have

$$d\varepsilon_{zz} = d\varepsilon_{zz}^e + d\varepsilon_{zz}^p = 0,$$

or

$$d\varepsilon_{zz}^e = - d\varepsilon_{zz}^p, \qquad (2.23)$$

a condition which must be imposed in the analysis.

Thus the strain-displacement relations (equation 2.4) reduce to those of the plane stress problem (equation 2.19) with the additional relation $\varepsilon_{zz} = 0$. For an isotropic material the elastic constitutive relations of equation (2.7) become

$$\varepsilon^e_{xx} = \frac{1}{E}\{\sigma_{xx} - \nu(\sigma_{yy} + \sigma_{zz})\},$$

$$\varepsilon^e_{yy} = \frac{1}{E}\{\sigma_{yy} - \nu(\sigma_{xx} + \sigma_{zz})\}, \qquad (2.24)$$

$$\varepsilon^e_{zz} = \frac{1}{E}\{\sigma_{zz} - \nu(\sigma_{xx} + \sigma_{yy})\},$$

and $$\varepsilon^e_{xy} = \frac{2(1+\nu)}{E}\sigma_{xy}.$$

The von Mises yield function for a body in plane strain and composed of a linear kinematic hardening material is

$$\phi = \frac{2}{9}\{\bar{\sigma}^2_{xx} + \bar{\sigma}^2_{yy} + \bar{\sigma}^2_{zz} - \bar{\sigma}_{xx}\bar{\sigma}_{yy} - \bar{\sigma}_{yy}\bar{\sigma}_{zz} - \bar{\sigma}_{zz}\bar{\sigma}_{xx} + 3\bar{\sigma}^2_{xy} - \sigma^2_o\}. \qquad (2.25)$$

2.8 Axial Symmetry

A problem frequently encountered in the analysis of continua is that of a body of revolution (axisymmetric solid) under axisymmetric loading. As in the cases of plane stress and plane strain the geometric representation can be reduced to one of two dimensions. From considerations of symmetry the state of strain at a point in the body is completely described by two displacement components lying in the plane containing the point and the axis of symmetry.

A polar coordinate system is conventionally employed. Let r and z denote respectively the radial and axial directions, and let θ denote the circumferential or tangential coordinate direction.

Any displacement in the radial direction will cause a circumferential strain $\varepsilon_{\theta\theta}$, but since stresses and strains are symmetrical with respect to the z-axis and are therefore independent of θ, it follows that the stress components $\sigma_{r\theta}$, $\sigma_{z\theta}$ and strain components $\varepsilon_{r\theta}$, $\varepsilon_{z\theta}$ must vanish throughout the body.

If u and v are respectively the displacement components in the r and z directions then the strain-displacement relations may be written as [27]

$$\varepsilon_{rr} = \frac{\partial u}{\partial r}, \quad \varepsilon_{zz} = \frac{\partial v}{\partial z}, \quad \varepsilon_{\theta\theta} = \frac{u}{r}, \quad \varepsilon_{rz} = \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r}. \tag{2.26}$$

For an isotropic material the elastic constitutive relations are

$$\begin{aligned} \varepsilon^e_{rr} &= \frac{1}{E}\{\sigma_{rr} - \nu(\sigma_{zz} + \sigma_{\theta\theta})\}, \\ \varepsilon^e_{zz} &= \frac{1}{E}\{\sigma_{zz} - \nu(\sigma_{rr} + \sigma_{\theta\theta})\}, \\ \varepsilon^e_{\theta\theta} &= \frac{1}{E}\{\sigma_{\theta\theta} - \nu(\sigma_{rr} + \sigma_{zz})\}, \end{aligned} \tag{2.27}$$

and

$$\varepsilon^e_{rz} = \frac{2(1+\nu)}{E}\sigma_{rz}.$$

For the axisymmetric case the von Mises yield function of equation (2.18) reduces to

$$\phi = \frac{2}{9}\{\bar{\sigma}_{rr}^2 + \bar{\sigma}_{zz}^2 + \bar{\sigma}_{\theta\theta}^2 - \bar{\sigma}_{rr}\bar{\sigma}_{zz} - \bar{\sigma}_{zz}\bar{\sigma}_{\theta\theta} - \bar{\sigma}_{\theta\theta}\bar{\sigma}_{rr} + 3\bar{\sigma}_{rz}^2 - \sigma_o^2\}. \tag{2.28}$$

CHAPTER 3

THE KINEMATIC MINIMUM PRINCIPLE FOR THE RATE PROBLEM IN ELASTO-PLASTICITY

3.1 Introduction

As a fundamental problem in elasto-plasticity we consider the response of a body to successive increments of load throughout its entire stress history. This incremental analysis is formulated initially in terms of rates (time derivatives) of the field variables.

3.2 The Classical Rate Problem in Elasto-Plasticity

The classical rate problem in elasto-plasticity may be stated as follows. Consider a body of volume V and surface S in a Cartesian coordinate system x_i, subjected to known body force rates $\dot{F}_i(x_j)$ on V, known traction rates $\dot{T}_i(x_j)$ on part of the surface S_T, and known displacement rates $\dot{u}_i(x_j)$ on the remainder of the surface S_u. As solution to the rate problem we seek displacement rates $\dot{u}_i(x_j)$ on S_T and V, reaction rates $\dot{T}_i(x_j)$ on S_u, a stress rate field $\dot{\sigma}_{ij}(x_k)$, and a strain rate field $\dot{\varepsilon}_{ij}(x_k)$. The governing equations comprise the rate forms for infinitesimal displacement of the equilibrium equations, kinematic relations and constitutive relations.

Since the equilibrium equations and kinematic relations (equation 2.1 through 2.5) are linear in force, stress, displacement and strain, it follows that the rate forms of these equations will be linear in force rates, stress rates, displacement rates and strain rates, and thus

$$\frac{\partial \dot{\sigma}_{ij}}{\partial x_j} + \dot{F}_i = 0 \quad \text{on } V, \tag{3.1}$$

$$\dot{\sigma}_{ij}\, \nu_j = \dot{T}_i \quad \text{on } S, \tag{3.2}$$

and

$$\dot{\varepsilon}_{ij} = \frac{1}{2}\left(\frac{\partial \dot{u}_i}{\partial x_j} + \frac{\partial \dot{u}_j}{\partial x_i} \right) \quad \text{on } V. \tag{3.3}$$

Rewriting the constitutive equations of section 2.4 in rate form we have

$$\dot{\varepsilon}_{ij} = \dot{\varepsilon}^e_{ij} + \dot{\varepsilon}^p_{ij}, \tag{3.4}$$

$$\dot{\varepsilon}^e_{ij} = C_{ijk\ell}\, \dot{\sigma}_{k\ell}, \tag{3.5}$$

$$\dot{\varepsilon}^p_{ij} = 0 \qquad \text{if } \phi < 0$$

$$\text{or } \phi = 0 \text{ and } \frac{\partial \phi}{\partial \sigma_{ij}} \dot{\sigma}_{ij} < 0,$$

$$\dot{\varepsilon}^p_{ij} = \lambda \frac{\partial \phi}{\partial \sigma_{ij}} \qquad \text{if } \phi = 0 \text{ and } \frac{\partial \phi}{\partial \sigma_{ij}} \dot{\sigma}_{ij} \geq 0 \tag{3.7}$$

and where for hardening materials

$$\lambda = G \frac{\partial \phi}{\partial \sigma_{k\ell}} \dot{\sigma}_{k\ell} \tag{3.8}$$

with G a positive definite scalar. For elastic-perfectly plastic behaviour λ is non-negative but otherwise undefined.

Although all the governing equations (3.1 through 3.8) are linear in rates of the field variables, it is not known a priori whether loading, neutral loading or unloading will occur in plastic regions of the body ($\phi = 0$), and thus the rate problem is not truly linear.

The rate problem at time t may be considered to be preceded by a succession of rate problems over the time interval $0 \leq \tau \leq t$. The response of the body over this interval is characterized by body forces $F_i(x_j,\tau)$, surface tractions $T_i(x_j,\tau)$, displacements $u_i(x_k,\tau)$, stresses $\sigma_{ij}(x_k,\tau)$ and strains $\varepsilon_{ij}(x_k,\tau)$. It is assumed that at time $t = 0$ the body is unstressed and the material in its virgin state.

Knowing the complete solution at time t we consider body force rates $\dot{F}_i$ on V, surface traction rates $\dot{T}_i$ on S_T, and displacement rates $\dot{u}_i$ on S_u. The rate forms of the governing equations permit a unique solution for the traction rates $\dot{T}_i$ on S_u, displacement rates $\dot{u}_i$ on S_T and V, the stress rate field $\dot{\sigma}_{ij}$ and the strain rate field $\dot{\varepsilon}_{ij}$.

Before formulating the kinematic minimum principle for the solution of the rate problem, it is necessary to discuss inversion of the constitutive relations to give the stress rate $\dot{\sigma}_{ij}$ in terms of the total strain rate $\dot{\varepsilon}_{ij}$.

3.3 Inversion of Constitutive Equations

Denoting the inverse of $C_{ijk\ell}$ by $D_{ijk\ell}$, equation (3.5) is inverted as

$$\dot{\sigma}_{ij} = D_{ijk\ell}\, \dot{\varepsilon}^e_{k\ell}. \tag{3.9}$$

Hence, for $\dot{\varepsilon}^{p}_{ij} = 0$ we have

$$\dot{\sigma}_{ij} = D_{ijk\ell} \, \dot{\varepsilon}_{k\ell} \text{ for } \phi < 0$$

$$\text{or } \phi = 0 \text{ and } D_{ijk\ell} \frac{\partial\phi}{\partial\sigma_{ij}} \dot{\varepsilon}_{k\ell} \leq 0. \qquad (3.10)$$

If for hardening behaviour $\dot{\varepsilon}^{p}_{ij} \neq 0$, then substituting for the elastic and plastic strain rates, equations (3.5), (3.7) and (3.8), in the expression for the total strain rate, equation (3.4), we have

$$\dot{\varepsilon}_{ij} = C_{ijk\ell} \, \dot{\sigma}_{k\ell} + G \frac{\partial\phi}{\partial\sigma_{ij}} \frac{\partial\phi}{\partial\sigma_{k\ell}} \dot{\sigma}_{k\ell}. \qquad (3.11)$$

Rearranging as

$$C_{ijk\ell} \, \dot{\sigma}_{k\ell} = \dot{\varepsilon}_{ij} - G \frac{\partial\phi}{\partial\sigma_{ij}} \frac{\partial\phi}{\partial\sigma_{k\ell}} \dot{\sigma}_{k\ell} \qquad (3.12)$$

and premultiplying by $D_{ijk\ell}$ gives

$$\dot{\sigma}_{ij} = D_{ijk\ell} \left(\dot{\varepsilon}_{k\ell} - G \frac{\partial\phi}{\partial\sigma_{k\ell}} \frac{\partial\phi}{\partial\sigma_{mn}} \dot{\sigma}_{mn} \right) . \qquad (3.13)$$

Further, multiplying by $\frac{\partial\phi}{\partial\sigma_{ij}}$ leads to

$$\frac{\partial\phi}{\partial\sigma_{ij}} \dot{\sigma}_{ij} = D_{ijk\ell} \frac{\partial\phi}{\partial\sigma_{ij}} \dot{\varepsilon}_{k\ell} - G \, D_{mnpq} \frac{\partial\phi}{\partial\sigma_{mn}} \frac{\partial\phi}{\partial\sigma_{pq}} \frac{\partial\phi}{\partial\sigma_{ij}} \dot{\sigma}_{ij}$$

from which it follows that

$$\frac{\partial \phi}{\partial \sigma_{ij}} \dot{\sigma}_{ij} = \frac{D_{ijk\ell} \frac{\partial \phi}{\partial \sigma_{ij}} \dot{\varepsilon}_{k\ell}}{\left(1 + G\, D_{mnpq} \frac{\partial \phi}{\partial \sigma_{mn}} \frac{\partial \phi}{\partial \sigma_{pq}}\right)} . \tag{3.14}$$

Substituting this expression in equation (3.13) and multiplying out gives finally

$$\dot{\sigma}_{ij} = D_{ijk\ell}\, \dot{\varepsilon}_{k\ell} - D_{ijk\ell} \frac{\partial \phi}{\partial \sigma_{k\ell}} \left[\frac{D_{pqrs} \frac{\partial \phi}{\partial \sigma_{pq}} \dot{\varepsilon}_{rs}}{\frac{1}{G} + D_{mnhg} \frac{\partial \phi}{\partial \sigma_{mn}} \frac{\partial \phi}{\partial \sigma_{hg}}} \right] \tag{3.15}$$

for $\phi = 0$ and $D_{ijk\ell} \frac{\partial \phi}{\partial \sigma_{ij}} \dot{\varepsilon}_{k\ell} \geq 0$.

Equations (3.10) and (3.15) are thus the inverted constitutive rate equations for hardening behaviour, the elastic-perfectly plastic case being recovered as the limit $G \to \infty$ in equation (3.15). The inversion of the constitutive equations may be shown to be unique, [25].

3.4 The Kinematic Minimum Principle for the Rate Problem

Martin [23] has proposed an extended kinematic minimum principle for the rate problem in elasto-plasticity. Consider the inverted constitutive equations (3.10) and (3.15). These may be derived from a discontinuous potential functional W^o defined by

$$W^o = W^o(\dot{\varepsilon}_{ij}), \qquad \dot{\sigma}_{ij} = \frac{\partial W^o}{\partial \varepsilon_{ij}} \tag{3.16}$$

where

$$W^{o} = \frac{1}{2} D_{ijk\ell} \dot{\varepsilon}_{ij}\dot{\varepsilon}_{k\ell} \tag{3.17}$$

for $\phi < 0$

or $\phi = 0$ and $D_{ijk\ell} \frac{\partial\phi}{\partial\sigma_{ij}} \dot{\varepsilon}_{k\ell} \leq 0$,

and

$$W^{o} = \frac{1}{2} D_{ijk\ell} \dot{\varepsilon}_{ij}\dot{\varepsilon}_{k\ell} - \frac{1}{2} \frac{\left(D_{ijk\ell} \frac{\partial\phi}{\partial\sigma_{ij}} \dot{\varepsilon}_{k\ell} \right)^2}{\left(\frac{1}{G} + D_{pqrs} \frac{\partial\phi}{\partial\sigma_{pq}} \frac{\partial\phi}{\partial\sigma_{rs}}\right)} \tag{3.18}$$

for $\phi = 0$ and $D_{ijk\ell} \frac{\partial\phi}{\partial\sigma_{ij}} \dot{\varepsilon}_{k\ell} \geq 0$.

So as to construct a kinematic minimum principle for the rate problem let us suppose that $\dot{\varepsilon}^*_{ij}$, $\dot{u}^*_i$ defined on V satisfy the rate form of the strain-displacement relations (equation 3.3) and the kinematic boundary conditions $\dot{u}^*_i = \dot{u}_i$ on S_u. The solution of the rate problem is that member of the class $\dot{\varepsilon}^*_{ij}$, $\dot{u}^*_i$ which renders an absolute minimum the functional

$$U^{o}_{p} (\dot{\varepsilon}^*_{ij}, \dot{u}^*_i) = \int_V W^{o}(\dot{\varepsilon}^*_{ij})dV - \int_V \dot{F}_i\dot{u}^*_i dV - \int_{S_T} \dot{T}_i\dot{u}^*_i dS . \tag{3.19}$$

Because the functional W^{o} is discontinuous, Martin broadens the class of variables and replaces W^{o} with a continuous potential functional $\overline{W}^{o}$ subject to inequality constraints. Dividing the body into two regions, let the plastic part of the body (where $\phi = 0$) be denoted collectively by V_p, and the elastic part of the body (where $\phi < 0$) be denoted collectively by V_e. Defining a non-negative scalar field $\lambda^*(x_k)$ over V_p, Martin introduces the functional

$$\overline{W}^o(\dot{\varepsilon}^*_{ij}, \lambda^*) = \frac{1}{2} D_{ijk\ell} \left(\dot{\varepsilon}^*_{ij} - \lambda^* \frac{\partial\phi}{\partial\sigma_{ij}}\right)\left(\dot{\varepsilon}^*_{k\ell} - \lambda^* \frac{\partial\phi}{\partial\sigma_{k\ell}}\right) + \frac{(\lambda^*)^2}{2G} \tag{3.20}$$

and the constraints $\lambda^* = 0$ in V_e (3.21)

$$\lambda^* \geq 0 \text{ in } V_p .$$

For an arbitrary choice of λ^*

$$\overline{W}^o(\dot{\varepsilon}^*_{ij}, \lambda^*) \geq W^o(\dot{\varepsilon}_{ij}) \tag{3.22}$$

with equality occurring when λ^* takes the value which gives the minimum value of $\overline{W}^o$ subject to the constraints of equations (3.21). In this case the actual plastic strain rate is given by

$$\dot{\varepsilon}^p_{ij} = \lambda^* \frac{\partial\phi}{\partial\sigma_{ij}} . \tag{3.23}$$

The proof may be obtained by differentiating the quadratic $\overline{W}^o$ with respect to λ^*. A stationary value occurs for

$$\lambda^* = \frac{D_{ijk\ell} \frac{\partial \phi}{\partial \sigma_{ij}} \dot{\varepsilon}^*_{k\ell}}{\frac{1}{G} + D_{pqrs} \frac{\partial \phi}{\partial \sigma_{pq}} \frac{\partial \phi}{\partial \sigma_{rs}}} . \tag{3.24}$$

The second derivative of $\overline{W}^o$ with respect to λ^* is

$$\frac{1}{G} + D_{ijk\ell} \frac{\partial \phi}{\partial \sigma_{ij}} \frac{\partial \phi}{\partial \sigma_{k\ell}} ,$$

and since $G \geq 0$ this expression is positive definite and indicates the stationary point is a minimum.

From equation (3.24) we see that the sign of λ^* is governed by the sign of $D_{ijk\ell} \frac{\partial \phi}{\partial \sigma_{ij}} \dot{\varepsilon}_{k\ell}$. Substituting the expression for λ^* in equation (3.20), the least value of $\overline{W}^o$ is given by equation (3.18) if

$$D_{ijk\ell} \frac{\partial \phi}{\partial \sigma_{ij}} \dot{\varepsilon}_{k\ell} \geq 0. \tag{3.25}$$

If the expression (3.25) is less than zero, the least value of $\overline{W}^o$ is given by $\lambda^* = 0$, in which case $\overline{W}^o$ reduces to equation (3.17).

The minimum principle (3.19) is thus extended: the solution of the rate problem is given by that member of the class $\dot{u}^*(x_k)$, $\dot{\varepsilon}^*_{ij}(x_k)$, $\lambda^*(x_k)$ which renders an absolute minimum the functional

$$\overline{U}^o_p(\dot{u}^*_i, \dot{\varepsilon}^*_{ij}, \lambda^*) = \int_V \overline{W}^o(\dot{\varepsilon}^*_{ij}, \lambda^*) dV - \int_V \dot{F}_i \dot{u}^*_i dV - \int_{S_T} \dot{T}_i \dot{u}^*_i dS \tag{3.26}$$

subject to the constraints $\lambda^* = 0$ in V_e (where $\phi < 0$)

and $\lambda^* \geq 0$ in V_p (where $\phi = 0$) . (3.27)

CHAPTER 4

APPLICATION OF THE KINEMATIC MINIMUM PRINCIPLE TO THE CONTINUUM PROBLEM

4.1 Introduction

The complexity of the governing equations for the extended kinematic minimum principle excludes the possibility of an analytical solution to any realistic problem. However, by employing numerical analysis techniques the minimum principle can be exploited directly. To this end we discretize the spatial field into an assemblage of finite elements. The degree of accuracy with which the numerical solution corresponding to the assemblage of elements approximates the true solution of the original continuum problem, depends on the fineness of the finite element subdivision and the sophistication or complexity of the individual elements.

Limiting discussion to the two-dimensional case we initially consider subdivision of the body into simple triangular constant strain finite elements, as the constant strain (and therefore constant stress) condition within each element enforces elastic-plastic interfaces to occur only at inter-element boundaries. Later the formulation will be extended to higher-order finite elements where numerical integration techniques permit elastic-plastic interfaces to occur within individual elements.

4.2 Application to Constant Strain Finite Elements

We discretize a continuum into an assemblage of m triangular constant strain finite elements. These give continuity of velocity along common boundaries of adjacent elements. Velocities of the assemblage are described by velocity components at each node.

Consider a generic element lying in the x-y plane of an x,y,z coordinate system, with apices 1,2,3 numbered anti-clockwise and having coordinates (x_1,y_1), (x_2,y_2), (x_3,y_3). Since discussion is limited to problems in plane stress, plane strain or axial symmetry, the velocity field of the element is described by two components of velocity $\dot{u}$, $\dot{v}$ respectively in the x,y directions, at each apex.

Adopting the approach of Zienkiewicz [34], we choose a linear velocity function over the element

$$\dot{u}(x,y) = \alpha_1 x + \alpha_2 y + \alpha_3,$$

$$\dot{v}(x,y) = \alpha_4 x + \alpha_5 y + \alpha_6. \qquad (4.1)$$

The six constants $\alpha_1, \ldots, \alpha_6$ are obtained by solving six simultaneous equations produced by inserting nodal coordinates and corresponding velocities.

Thus

$$\dot{u}(x,y) = \frac{1}{2\Delta}\{\chi_1\dot{u}_1 + \chi_2\dot{u}_2 + \chi_3\dot{u}_3\},$$

$$\dot{v}(x,y) = \frac{1}{2\Delta}\{\chi_1\dot{v}_1 + \chi_2\dot{v}_2 + \chi_3\dot{v}_3\}, \qquad (4.2)$$

where

$$\chi_1 = a_1 + b_1 x + c_1 y$$

$$\chi_2 = a_2 + b_2 x + c_2 y, \qquad (4.3)$$

$$\chi_3 = a_3 + b_3 x + c_3 y;$$

and where

$$a_1 = x_2 y_3 - x_3 y_2,$$

$$b_1 = y_2 - y_3, \qquad (4.4)$$

$$c_1 = x_3 - x_2,$$

with remaining coefficients obtained by cyclic permutation of subscripts in the order 1,2,3. Further

$$2\Delta = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = 2 \text{ x area of triangle } 1,2,3. \qquad (4.5)$$

For the case of plane stress the strain-displacement relations of equations (2.19) give in rate form

$$\begin{Bmatrix} \dot{\varepsilon}_{xx} \\ \dot{\varepsilon}_{yy} \\ \dot{\varepsilon}_{xy} \end{Bmatrix} = \frac{1}{2\Delta} \begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{bmatrix} \begin{Bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \\ \dot{u}_3 \\ \dot{v}_3 \end{Bmatrix},$$

(4.6)

and for plane strain

$$\begin{Bmatrix} \dot{\varepsilon}_{xx} \\ \dot{\varepsilon}_{yy} \\ \dot{\varepsilon}_{zz} \\ \dot{\varepsilon}_{xy} \end{Bmatrix} = \frac{1}{2\Delta} \begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{bmatrix} \begin{Bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \\ \dot{u}_3 \\ \dot{v}_3 \end{Bmatrix} . \tag{4.7}$$

Writing equations (4.6) and (4.7) in matrix form

$$\{\dot{\varepsilon}'\} = [B']\{\dot{u}'\}, \tag{4.8}$$

where the prime denotes an element matrix or vector. Components of $\{\dot{u}'\}$ are element node velocities, and $[B']$ is a linear matrix which depends on element nodal coordinates only. The third row of $[B']$ is zero in equation (4.7) enforcing the plane strain requirement of zero total strain in the z-direction.

So as to employ a consistant notation, for the axisymmetric case we replace r, z and θ with coordinate subscripts x, y and z respectively, to give the rate form of the strain-displacement relations (2.26) as

$$\begin{Bmatrix} \dot{\varepsilon}_{xx} \\ \dot{\varepsilon}_{yy} \\ \dot{\varepsilon}_{zz} \\ \dot{\varepsilon}_{xy} \end{Bmatrix} = \frac{1}{2\Delta} \begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ d_1 & 0 & d_2 & 0 & d_3 & 0 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{bmatrix} \{\dot{u}'\}, \tag{4.9}$$

where $$d_1 = \frac{a_1}{x} + b_1 + c_1 \frac{y}{x}, \tag{4.10}$$

with similar expressions for d_2 and d_3 obtained by interchange of subscripts.

In view of equation (4.10) coefficients corresponding to the circumferential strain rate $\dot{\varepsilon}_{zz}$ are dependent on position within the element. To facilitate integration of the strain rate field by enforcing constant strain conditions, it is convenient to treat the centroidal value of circumferential strain rate as constant across the element. Thus the x,y variables in equation (4.10) are replaced by centroidal values $\bar{x},\bar{y}$, where

$$\bar{x} = \frac{1}{3}(x_1 + x_2 + x_3), \qquad \bar{y} = \frac{1}{3}(y_1 + y_2 + y_3) . \tag{4.11}$$

Hence equation (4.9) may also be written as $\{\dot{\varepsilon}'\} = [B']\{\dot{u}'\}$, where $[B']$ is a linear matrix depending only on element geometry.

Breaking the strain rate tensor into elastic and plastic parts as before, it is convenient to introduce weighted strain rates defined by

$$\dot{\delta}^e_{ij} = V'\dot{\varepsilon}^e_{ij} ,$$

$$\dot{\delta}^p_{ij} = V'\dot{\varepsilon}^p_{ij} ,$$

$$\dot{\delta}_{ij} = \dot{\delta}^e_{ij} + \dot{\delta}^p_{ij} , \tag{4.12}$$

where V' is the element volume. In the case of plane stress the element is assumed to be of uniform thickness and in plane strain unit thickness, so that the element volume is given by the product of element area and thickness. For the axisymmetric case the element volume is that of a body of revolution and is therefore given by $2\pi\bar{x}\Delta$, where $\bar{x}$ is the radius to the element centroid and Δ is the triangular sectional area.

Consider now the assemblage of elements. It is assumed that certain nodal velocity components are constrained to be zero throughout the loading history. Ordering remaining velocity components we define the velocity vector $\{\dot{u}\}$ of say, n components. Ordering the elements of the assemblage and taking weighted strain rate components in turn gives a weighted strain rate vector $\{\dot{\delta}\}$ of say, k components. Using the strain rate-velocity relationship of equation (4.8) for each element of the assemblage and taking account of the element volume leads to a k by n weighted deformation matrix [B] for the system:

$$\{\dot{\delta}\} = [B]\{\dot{u}\}. \tag{4.13}$$

The inverted elastic constitutive equations for an isotropic material are written in matrix form for an element as

$$\{\sigma'\} = [D']\{\varepsilon^e\}', \tag{4.14}$$

where for plane stress

$$[D'] = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \tag{4.15a}$$

and for plane strain or axial symmetry

$$[D'] = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}. \tag{4.15b}$$

Writing the constitutive relations (4.14) in terms of weighted strains and in rate form gives

$$\{\dot{\sigma}'\} = \frac{1}{V'} [D']\{\dot{\delta}^e\}'. \tag{4.16}$$

Taking stress rate components of the ordered elements in the same manner as the weighted strain rate vector $\{\dot{\delta}\}$ we define a stress rate vector $\{\dot{\sigma}\}$, also of k components. Using equation (4.16) for each of the ordered elements in turn, we may assemble a weighted elasticity matrix [D] such that

$$\{\dot{\sigma}\} = [D]\{\dot{\delta}^e\} . \tag{4.17}$$

The matrix [D] is k by k, symmetric and block-diagonal.

Weighted plastic strain rates are given by

$$\dot{\delta}^p_{ij} = V'\lambda \frac{\partial\phi}{\partial\sigma_{ij}} . \tag{4.18}$$

Differentiating the von Mises yield function for plane stress (equation 2.21) gives

$$\begin{Bmatrix} \dot{\delta}^p_{xx} \\ \dot{\delta}^p_{yy} \\ \dot{\delta}^p_{xy} \end{Bmatrix} = V'\lambda \begin{Bmatrix} 2\bar{\sigma}_{xx} - \bar{\sigma}_{yy} \\ 2\bar{\sigma}_{yy} - \bar{\sigma}_{xx} \\ 6\bar{\sigma}_{xy} \end{Bmatrix} , \tag{4.19}$$

where $\bar{\sigma}_{xx} = \sigma_{xx} - E_p \varepsilon^p_{xx}$, etc .

For plane strain or axial symmetry the yield functions of equations (2.25) or (2.28) give plastic strain rates as

$$\begin{Bmatrix} \dot{\delta}^p_{xx} \\ \dot{\delta}^p_{yy} \\ \dot{\delta}^p_{zz} \\ \dot{\delta}^p_{xy} \end{Bmatrix} = V'\lambda \begin{Bmatrix} 2\bar{\sigma}_{xx} - \bar{\sigma}_{yy} - \bar{\sigma}_{zz} \\ 2\bar{\sigma}_{yy} - \bar{\sigma}_{xx} - \bar{\sigma}_{zz} \\ 2\bar{\sigma}_{zz} - \bar{\sigma}_{xx} - \bar{\sigma}_{yy} \\ 6\bar{\sigma}_{xy} \end{Bmatrix} . \tag{4.20}$$

It is convenient to normalize the gradient vector $\frac{\partial\phi}{\partial\sigma_{ij}}$ by setting

$$\left\{ \frac{\partial\phi}{\partial\sigma_{ij}} \right\}' = \frac{\partial\phi}{\partial\sigma_{ij}} \left(\frac{\partial\phi}{\partial\sigma_{k\ell}} \frac{\partial\phi}{\partial\sigma_{k\ell}} \right)^{-\frac{1}{2}}, \tag{4.21}$$

and introduce a different scalar Λ such that the weighted plastic strain rates of equation (4.18) become

$$\dot{\delta}^p_{ij} = \Lambda \left\{\frac{\partial\phi}{\partial\sigma_{ij}}\right\}'. \tag{4.22}$$

Defining a vector $\{\Lambda\}$ as being the ordered multipliers for the assemblage of m elements, and using equation (4.22) for the ordered elements in turn, a gradient matrix [N] is assembled for the system:

$$\{\dot{\delta}^p\} = [N]\{\Lambda\}. \tag{4.23}$$

The matrix [N] is k by m and has in its first column the gradient vector $\{\frac{\partial\phi}{\partial\sigma}\}'$ for the first element appearing in the first few rows; in its second column the vector $\{\frac{\partial\phi}{\partial\sigma}\}'$ for the second element appearing in the next few rows, and so on. All other elements of [N] are zero.

In section (2.5) the idealization chosen for the scalar hardening coefficient G is

$$\frac{1}{G} = E_p \left(\frac{\partial\phi}{\partial\sigma_{k\ell}} \frac{\partial\phi}{\partial\sigma_{k\ell}} \right), \qquad (4.24)$$

but since the gradient vector has been normalized, this reduces to

$$\frac{1}{G} = E_p. \qquad (4.25)$$

Thus we introduce a k by k diagonal matrix $[D_p]$ whose non-zero entries consist of the terms $\frac{E_p}{V'}$ for the ordered elements. The elastic-perfectly plastic case is recovered as the limit $E_p \to 0$.

Consider now the functional $\overline{U}_p^o$ of equation (3.26) and assume for the present that all elements of the assemblage undergo plastic deformation. Expressed in matrix form the volume integral of the functional $\overline{W}^o$ of equation (3.20) is

$$\int_V \overline{W}^o dV = \frac{1}{2}(\{\dot{\delta}\} - [N]\{\Lambda\})^T[D](\{\dot{\delta}\} - [N]\{\Lambda\})$$

$$+ \frac{1}{2}\{\Lambda\}^T[D_p]\{\Lambda\}. \qquad (4.26)$$

Velocities on the boundary of the assemblage can be expressed as linear functions of velocities at boundary nodes. Hence the rate form of potential energy of the boundary tractions T_i can be written in terms of a load rate vector $\{\dot{P}\}$ and boundary node velocities. The elements of $\{\dot{P}\}$ are calculated from

$$\{\dot{u}\}^T\{\dot{P}\} = \int_{S_T} \dot{T}_i \dot{u}_i^* dS. \qquad (4.27)$$

In the absence of body forces, substitution of equations (4.13), (4.26) and (4.27) in the expression for the functional $\overline{U}_p^o$ (equation 3.26) yields

$$\overline{U}_p^o = \frac{1}{2}\{\dot{u}\}^T[B^TDB]\{\dot{u}\} - \frac{1}{2}\{\Lambda\}^T[B^TDN]\{\dot{u}\} \tag{4.28}$$

$$- \frac{1}{2}\{\dot{u}\}^T[N^TDB]\{\Lambda\} + \frac{1}{2}\{\Lambda\}^T[N^TDN + D_p]\{\Lambda\} - \{\dot{u}\}^T\{\dot{P}\}.$$

Introducing a column vector $\{\dot{u}:\Lambda\}$ comprising of n elements of $\{\dot{u}\}$ and *m* elements of $\{\Lambda\}$, this becomes

$$\overline{U}_p^o = \frac{1}{2}\{\dot{u}:\Lambda\}^T[K^*]\{\dot{u}:\Lambda\} - \{\dot{u}:\Lambda\}^T\{\dot{P}:0\}, \tag{4.29}$$

where the system matrix [K*] is symmetric and given by

$$[K^*] = \left[\begin{array}{c|c} [B^TDB] & [-\, B^TDN] \\ \hline [-\, N^TDB] & [N^TDN + D_p] \end{array}\right] . \tag{4.30}$$

The upper left submatrix $[B^TDB]$ is the usual displacement method elastic system stiffness matrix. Indeed, if $\{\Lambda\} \equiv 0$, then $\overline{U}_p^o$ reduces to the rate form of potential energy for the system.

The solution to a particular rate problem is thus given by that value of $\{\dot{u}:\Lambda\}$ which minimizes $\overline{U}_p^o$ in equation (4.29) subject to the constraints that for each element of $\{\Lambda\}$

$$\Lambda = 0 \quad \text{if} \quad \phi < 0 \quad \text{(elastic)}, \tag{4.31a}$$

and

$$\Lambda \geq 0 \quad \text{if} \quad \phi = 0 \quad \text{(plastic)}. \tag{4.31b}$$

It is implicit that $\{\dot{P}\}$ is given, and that $\{\sigma\}$ and $\{\varepsilon^p\}$ are known for the assemblage so that the constraints (4.31) can be explicitly determined.

Since $\overline{U}_p^o$ is homogeneous in the rates we may determine the solution to the incremental problem: minimize $\overline{U}_p^o$ with respect to $\{\Delta u:\Lambda\}$, where

$$\overline{U}_p^o = \frac{1}{2}\{\Delta u:\Lambda\}^T[K^*]\{\Delta u:\Lambda\} - \{\Delta u:\Lambda\}^T\{\Delta P:0\}, \tag{4.32}$$

subject to the constraints of equations (4.31).

This constrained minimization problem can be stated as a formal mathematical programming problem and a general algorithm employed to determine the solution. Defining a vector $\{\phi\}$ corresponding to magnitudes of the yield function for the ordered elements, the quadratic programming problem is

$$\min \left\{\overline{U}_p^o(\Delta u,\Lambda) \;\middle|\; \{\Lambda\} \geq 0, \quad \{\phi\}^T\{\Lambda\} = 0\right\}. \tag{4.33}$$

A quadratic functional is to be minimized subject to bounds (non-negativity constraints on $\{\Lambda\}$) and linear constraints to ensure Λ is zero if $\phi < 0$.

The state-of-the-art in general quadratic programming algorithms is such that computer running times increase significantly with increase in numbers of variables and constraints. Moreover this problem must be solved for each load increment and hence computation time for any realistic continuum problem becomes prohibitive. Fortunately because of the nature of the minimization problem and its constraints, it is possible to develop an efficient intuitive solution algorithm which does not rely on formal mathematical programming techniques. This will be presented in the following section.

4.3 An Algorithm for the Minimization of $\overline{U}^{o}_{p}$

In Martin and Reddy's [24] application of the minimum principle to the truss problem, an algorithm was suggested for solution of the programming problem. This algorithm may be extended to the continuum finite element formulation.

Consider the assemblage of elements at an arbitrary stage in the loading program when the load on the structure, which was initially in the virgin state, is {P}. Further, let {P} be such that at least part of the structure is plastic (i.e. some elements in V_p). The displacements, elastic strains, plastic strains and stresses of the assemblage corresponding to the load {P} are known.

For the solution to the next load increment {ΔP} we are required to minimize $\overline{U}^{o}_{p}$ (equation 4.32) subject to the constraints of equations (4.31). Those elements governed by constraint (4.31a) are elastic (ie in V_e) and are easily identified. However for those elements which are plastic and are therefore governed by constraints (4.31b) we do not know a priori which will load or which will unload.

If all elements of the vector $\{\Delta u:\Lambda\}$ are non-zero the least value of the quadratic functional $\overline{U}^{o}_{p}$ is given by solution of the set of simultaneous linear equations

$$[K^*]\{\Delta u:\Lambda\} = \{P:0\}. \qquad (4.34)$$

This suggests an algorithm for the minimization of $\overline{U}^{o}_{p}$ based on an initial guess that elements in V_p are such that $\Lambda = 0$ or $\Lambda \neq 0$.

We proceed as follows. Identifying all elements for which $\phi < 0$, and those elements for which $\phi = 0$ and in which unloading is guessed to occur, we eliminate corresponding rows and columns of $[K^*]$.

The remaining simultaneous linear equations (4.34) are then solved for $\{\Delta u\}$ and proposed non-zero elements of $\{\Lambda\}$. This represents a trial solution for $\{\Delta u\}$ and $\{\Lambda\}$.

Necessary and sufficient conditions for the solution to be correct are: elements of $\{\Lambda\}$ are non-negative if they were assumed to be non-zero, and $\frac{\partial\phi}{\partial\sigma_{ij}} \Delta\sigma_{ij} < 0$ for those elements for which $\phi = 0$ but were guessed to be unloading. If these checks are not satisfied the solution has not been found and we must revise the choice of elements in V_p for which we guess Λ is zero, and re-solve equations (4.34).

In revising the choice of loading or unloading finite elements in V_p we consider those elements of $\{\Lambda\}$ which did not satisfy the constraints of equations (4.31b). If on one hand Λ was assumed to be zero and $\frac{\partial\phi}{\partial\sigma_{ij}} \Delta\sigma_{ij} > 0$, this element of $\{\Lambda\}$ is now assumed to be non-zero. On the other hand if Λ was assumed to be non-zero and Λ is negative, it is now assumed to be zero.

Convergence of the iterative procedure has not been conclusively proved, but experience indicates that it is rapid and fails only as the limit load is approached in the elastic-perfectly plastic case. Further, it will be shown in the following section that the formulation and minimization algorithm can be reduced to the conventional tangent modulus approach. If loads are increased monotonically unloading of elements very seldom occurs, and thus the best initial guess in the iterative procedure is that Λ is non-zero for all elements in V_p. This initial assumption was also used for non-monotonic loading, in which case the algorithm was found to converge within one or two iterations.

4.4 Reduction to Tangent Modulus Approach

Consider an arbitrary stage in the loading program at which some finite elements are plastic. For the next load increment the system matrix [K*] is evaluated and, following the minimization algorithm, particular rows and columns are eliminated corresponding to elastic elements, or plastic elements guessed to unload. To perform this elimination process it is convenient to consider these rows and columns as set to zero and unity inserted on the leading diagonal, thus retaining the original dimension of [K*].

Let the system matrix corresponding to the correct solution be $[\overline{K}^*]$. That is, the solution of the equations

$$[\overline{K}^*]\{\Delta u:\Lambda\} = \{\Delta P:0\} \tag{4.35}$$

is $\{\Delta u:\Lambda\} = \{\Delta\bar{u}:\bar{\Lambda}\}$ where $\{\Delta\bar{u}:\bar{\Lambda}\}$ satisfy the constraints of equations (4.31). Expanding $[\overline{K}^*]$ in equation (4.35) this becomes

$$\left[\begin{array}{c|c} B^TDB & -B^TDN \\ \hline -N^TDB & N^TDN + D_p \end{array}\right]\begin{Bmatrix}\Delta u \\ \Lambda\end{Bmatrix} = \begin{Bmatrix}\Delta P \\ 0\end{Bmatrix}, \tag{4.36}$$

with certain rows and columns set to zero.

Partitioning equations (4.36) gives

$$[-N^TDB]\{\Delta u\} + [N^TDN + D_p]\{\Lambda\} = \{0\}. \tag{4.37}$$

Since [D] and $[D_p]$ are positive definite it follows that

$$\{\Lambda\} = [N^T DN + D_p]^{-1}[N^T DB]\{\Delta u\}. \tag{4.38}$$

Substituting for $\{\Lambda\}$ in the remainder of the partitioned equations (4.36) gives

$$[B]^T[\bar{D}][B]\{\Delta u\} = \{\Delta P\}, \tag{4.39}$$

where $$[\bar{D}] = [D - DN(N^T DN + D_p)^{-1}N^T D]. \tag{4.40}$$

This stress-strain matrix $[\bar{D}]$ reflects the current constitutive relations for each element of the assemblage. Solving equation (4.39) will yield $\{\Delta u\} = \{\Delta\bar{u}\}$.

Investigating the form of $[\bar{D}]$ we see that like the elastic system elasticity matrix $[D]$, $[\bar{D}]$ is also symmetric and composed of submatrices along a diagonal band. Each matrix corresponds to an element of the assemblage. For the ith element, these submatrices are $\frac{1}{V'}[D']_i$ and $\frac{1}{V'}[\bar{D}']_i$, and the normalized gradient vector is $\{\frac{\partial\phi}{\partial\sigma}\}_i'$.

Consider this ith element of the assemblage. If it was elastic, or plastic and unloading, Λ_i was assumed zero and the corresponding row and column of $[K^*]$ set to zero. Evaluating equation (4.40) will result in

$$[\bar{D}']_i = [D']_i. \tag{4.41}$$

If Λ_i was assumed non zero the ith submatrix of $[\bar{D}]$ in equation (4.40) is equivalent to

$$[\bar{D}']_i = \left[[D'] - \frac{[D']\{\frac{\partial\phi}{\partial\sigma}\}'\{\frac{\partial\sigma}{\partial\sigma}\}'^T[D']}{E_p + \{\frac{\partial\phi}{\partial\sigma}\}'^T[D']\{\frac{\partial\phi}{\partial\sigma}\}'} \right]_i , \qquad (4.42)$$

which is in an identical form to the element elasto-plastic matrix $[D]^*_{ep}$ derived by Zienkiewicz et al [11] for the conventional tangent modulus approach.

In following the steps of the tangent modulus approach a block-diagonal matrix $[D_T]$ is assembled for the system by considering each finite element in turn and using equation (4.41) if the element is elastic, or plastic and guessed to unload; and equation (4.42) if the element is plastic and guessed to load. Having assembled $[D_T]$, a solution for $\{\Delta u\}$ is obtained from the equation

$$[B]^T[D_T][B]\{\Delta u\} = \{\Delta P\}, \qquad (4.43)$$

and $\Delta\sigma_{ij}$ evaluated for each plastic element. The solution is correct if for each element in V_p: $\frac{\partial\phi}{\partial\sigma_{ij}}\Delta\sigma_{ij} < 0$ for elements guessed to unload, and $\frac{\partial\phi}{\partial\sigma_{ij}}\Delta\sigma_{ij} \geq 0$ for elements guessed to load. If these constraints are not satisfied new assumptions are made as to loading and unloading plastic elements and a new tangent modulus matrix $[D_T]$ assembled. The process is then repeated.

In the programming approach a solution is correct if: $\Lambda = 0$ for elastic elements; $\frac{\partial\phi}{\partial\sigma_{ij}}\Delta\sigma_{ij} < 0$ for plastic elements assumed to unload; and $\Lambda \geq 0$ for plastic elements assumed to load. This non-negativity constraint on Λ is identical to the constraint on the sign $\frac{\partial\phi}{\partial\sigma_{ij}}\Delta\sigma_{ij}$ for loading plastic elements in the tangent modulus approach.

Recalling equation (2.10)

$$\lambda = G \frac{\partial \phi}{\partial \sigma_{k\ell}} d \sigma_{k\ell},$$

and since G is positive definite,

$$\Lambda \geq 0 \longleftrightarrow \frac{\partial \phi}{\partial \sigma_{ij}} \Delta\sigma_{ij} \geq 0.$$

We have thus shown that the solution algorithm suggested for the kinematic minimum principle reduces to the conventional tangent modulus approach for incremental elasto-plastic analysis. The matrix $[\bar{D}]$ is identical to $[D_T]$, and is a tangent modulus stress-strain matrix to account for modified stiffness under loading.

To the writer's knowledge convergence of the conventional tangent modulus algorithm has not been proved, although it is widely accepted and used, [14], [15], [28]. Hence reduction of the solution algorithm suggested for the quadratic programming problem to the tangent modulus approach certainly provides a measure of confidence in the iterative procedure suggested in the preceding section.

4.5 Solution Procedure for the Incremental Problem

In presenting the application of the minimum principle to constant strain finite elements we have discussed the evaluation of the system matrix [K*] on a global basis. In so doing all m elements of the assemblage are represented in each submatrix of [K*], resulting in an (n+m) square system matrix. Then, following the algorithm for minimizing $\bar{U}_p^o$, rows and columns corresponding to elastic elements and unloading plastic elements are deleted and a solution obtained.

In the computer implementation of the minimum principle both computational effort and computer storage requirements can be substantially reduced by evaluating only relevant element submatrices of [K*].

Further, for each load increment only part of [K*] need be reformulated. As noted previously the $[B^TDB]$ submatrix is the usual displacement method elastic system stiffness matrix and remains unchanged for each determination of [K*]. However, the remaining submatrices of [K*] must be calculated for each load increment since the gradient matrix [N] depends on current stress, and in the kinematic hardening case, current plastic strain.

Instead of the global approach presented above we take each element in turn and evaluate the element deformation matrix $[B']_i$, $(i = 1,2,...,m)$. These matrices are stored for later use. Each six by six elastic element stiffness matrix $[B^TDB]_i'$ $(i = 1,2,...,m)$ is explicitly evaluated and assembled in appropriate positions corresponding to element node displacements, to form the n by n elastic system stiffness matrix $[B^TDB]$. This is stored so that it may be retrieved for each reformulation of [K*].

The body is assumed to be initially in the virgin state and therefore all elements of the assemblage are elastic. Thus, since $\{\Lambda\} \equiv 0$, the solution of the first load increment corresponds to a linear elastic analysis

$$[B^TDB]\{\Delta u\} = \{\Delta P\}, \tag{4.44}$$

where $\{\Delta P\}$ need give only relative magnitudes of components of the load vector. On the basis of this solution relative magnitudes of stress components are calculated for each element, and the smallest load factor determined to cause at least one element to enter V_p.

(Fortuitously or through symmetry more than one element may correspond to the smallest load factor). For the assemblage, increments of stress, strain, load and displacement are scaled by this load factor and become current totals of each respective quantity. All plastic strains are zero but for the following load increment there is at least one plastic element.

Consider this as representing an arbitrary stage in the loading program when part of the structure is plastic (i.e. some elements in V_p). The system matrix [K*] is required for the following load increment. Firstly, the elastic system matrix $[B^TDB]$ is retrieved from storage and entered into [K*]. To determine the remainder of the symmetric system matrix only terms on and above the leading diagonal are evaluated. Recall that the best initial guess for the solution algorithm is that $\Lambda \neq 0$ for all elements in V_p. Beginning with [K*] corresponding to the elastic matrix $[B^TDB]$, each element is considered in turn, and if plastic the next row and column of [K*] assigned to it. For each of these elements in V_p a six by one element matrix $\{ - [B']^T[D']\{\frac{\partial\phi}{\partial\sigma}\}'\}$ and scalar $\frac{1}{V'}\left(\{\frac{\partial\phi}{\partial\sigma}\}'^T[D'] \{\frac{\partial\phi}{\partial\sigma}\}' + E_p \right)$ is evaluated and entered into the appropriate positions of its assigned column to form the upper triangle of [K*].

Thus a condensed system matrix is produced wherein the number of columns in the $[- B^TDN]$ submatrix is equal to the number of elements currently in V_p. The plastic submatrices of [K*] are also stored in case of iteration within a load increment, in which case rows and columns are deleted corresponding to unloading plastic elements.
Having determined a satisfactory solution to the current load increment, totals of stress, strain, load and displacement are updated and [K*] determined for the succeeding load increment.

In this way the magnitude of each load increment is determined by the lowest load multiplier which causes an elastic element to yield. Without significant loss of accuracy the total number of load increments is decreased by including in V_p elements close to yielding, and correcting stresses after the load increment. Further, symmetry and sparseness of the system matrix may be exploited and an efficient storage scheme and associated solution routine employed. These computational details will be described in Chapter 5.

4.6 Application to Higher Order Finite Elements

The use of constant strain finite elements in the application of the minimum principle to continuum problems requires discretization of the body into a large number of elements (especially in regions where plastic deformations occur), if results of an analysis are to be meaningful. This in turn implies a large number of displacement degrees of freedom, and since an additional degree of freedom corresponds to each loading plastic element, computer storage requirements and computational effort increase significantly when large regions of the body are plastic.

A saving of storage requirements and computational effort can be achieved by use of higher order finite elements. Again limiting discussion to the cases of plane stress, plane strain and axial symmetry, we shall in particular consider use of cubic quadrilateral isoparametric finite elements, [29].

Since each of these elements is connected to twelve nodes (one at each corner and two on each side), a cubic variation of the displacement field is defined across each element. This implies quadratic variation of strain within an element and thus the strain displacement relations [B'] are no longer constant. In the case of linear elastic analysis, integration of the strain field cannot be performed explicitly, necessitating use of a numerical integration technique such as Gauss quadrature. This technique is further exploited in the elasto-plastic case.

Discretizing a continuum into an assemblage of cubic quadrilateral isoparametric finite elements consider a generic element lying in the x, y plane of a global Cartesian coordinate system x, y, z, as shown in Figure 4.1 A natural coordinate system r, s is defined on the element as shown, where $-1 \leq r \leq 1$, $-1 \leq s \leq 1$. Coordinates x, y of any point within the element are

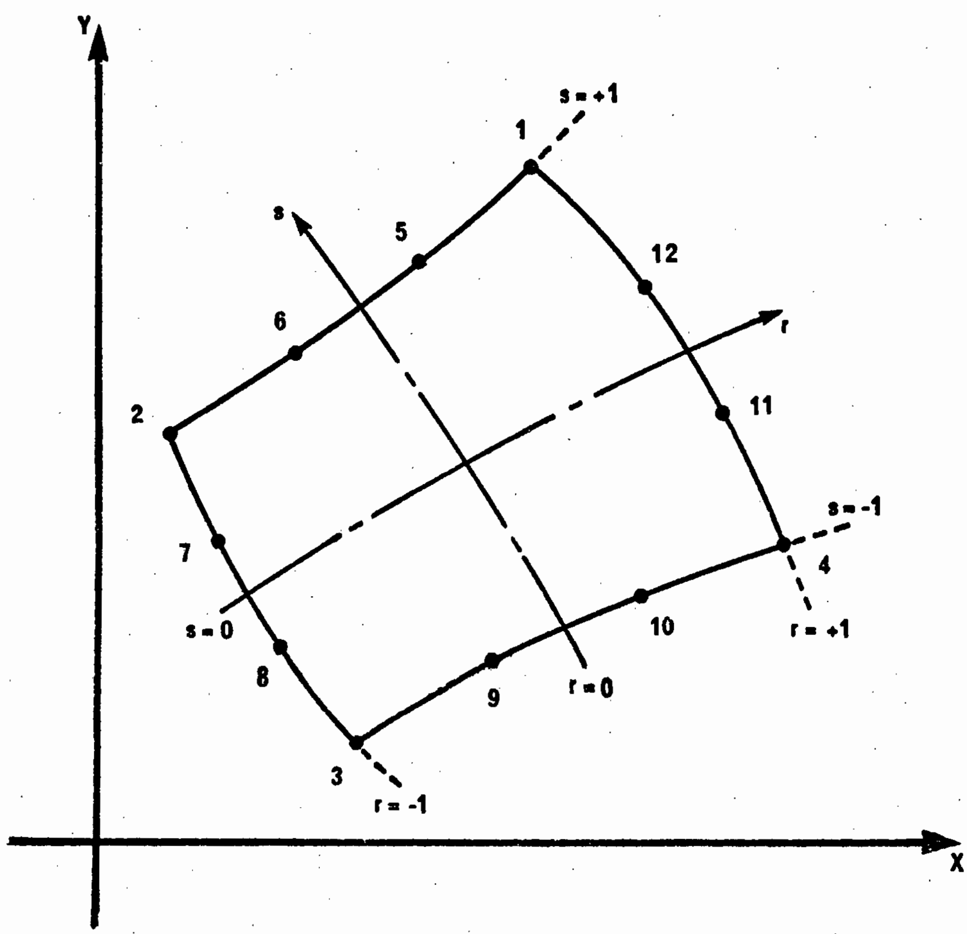


Figure 4.1 A cubic quadrilateral isoparametric finite element

described in terms of element node coordinates by means of shape functions. A shape function h_i is associated with each node of the element and has unit value at that node and zero value at all other element nodes. The coordinate interpolations are

$$x = \sum_{i=1}^{12} h_i x_i, \qquad y = \sum_{i=1}^{12} h_i y_i, \tag{4.45}$$

where $x_i, y_i (i = 1,2,\ldots,12)$ are the global coordinates of the twelve nodes of the element.

The shape functions h_i are defined in terms of the natural coordinate system of the element, which has variables r, s that vary from -1 to +1. Defining the quantities $r_o = rr_i$ and $s_o = ss_i$ for the ith node of the element, then for the corner nodes $(i = 1,2,3,4)$ the shape functions are

$$h_i = \frac{1}{32}(1 + r_o)(1 + s_o)\{-10 + 9(r^2 + s^2)\}. \tag{4.46a}$$

For the remaining element nodes $(i = 5,6,\ldots,12)$

$$h_i = \frac{9}{32}(1 + r_o)(1 - s^2)(1 + 9s_o) \tag{4.46b}$$

$$\text{for } (r_i, s_i) = (-1,-\tfrac{1}{3}),(1,-\tfrac{1}{3}),(-1,\tfrac{1}{3}) \text{ and } (1,\tfrac{1}{3});$$

and

$$h_i = \frac{9}{32}(1 + s_o)(1 - r^2)(1 + 9r_o) \tag{4.46c}$$

$$\text{for } (r_i, s_i) = (-\tfrac{1}{3},-1),(\tfrac{1}{3},-1),(-\tfrac{1}{3},1) \text{ and } (\tfrac{1}{3},1).$$

In view of equations (4.45) and (4.46) an advantage of the isoparametric formulation is apparent in that elements can have curved boundaries.

The basis of the isoparametric formulation is that the shape functions h_i used to describe geometry are also used to describe displacements at any point in the element in terms of displacements at the element nodes. Thus

$$u = \sum_{i=1}^{12} h_i u_i , \qquad v = \sum_{i=1}^{12} h_i v_i , \tag{4.47}$$

where u,v are displacements respectively in the x, y directions at a point in the element, and u_i, v_i are the element node displacement components. As a consequence of (4.45) and (4.47) the cubic quadrilateral isoparametric element is both compatible and complete, [29].

Element strains are obtained in terms of derivatives of displacements with respect to global coordinate directions. But, because element displacements are defined in the natural coordinate system (equations 4.47), the inverse of the Jacobian operator is used to relate global coordinate derivatives to natural coordinate derivatives:

$$\begin{Bmatrix} \frac{\partial}{\partial x} \\ \\ \frac{\partial}{\partial y} \end{Bmatrix} = J^{-1} \begin{Bmatrix} \frac{\partial}{\partial r} \\ \\ \frac{\partial}{\partial s} \end{Bmatrix} , \tag{4.48}$$

$$\text{where} \quad J = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} . \tag{4.49}$$

The inverse of J exists provided there is a unique correspondence between natural and global coordinates for the element. Using equations (2.4), (4.47) and (4.48) element strains ε_{ij} are obtained from element node displacements u_k for any point within the element:

$$\varepsilon_{ij}(r,s) = B_{ijk}(r,s)\, u_k, \tag{4.50}$$

where the subscripts i,j refer to the relevant strain components for plane stress, plane strain or axial symmetry. As before in the case of plane strain elements of B_{ijk} corresponding to ε_{zz} are zero.

The element elastic stiffness matrix is

$$[k'] = \int_{V'} B_{ij} D_{ijk\ell} B_{k\ell}\, dV, \tag{4.51}$$

where the integration extends over the element volume. Writing this in matrix notation

$$[k'] = \int_{V'} [B^T DB]'\, dV. \tag{4.52}$$

Since the elements of [B'] are functions of the natural coordinates r,s the volume integration is performed over the natural coordinate volume by writing the element volume differential dV in terms of r,s. This is

$$dV = t \det J\, dr\, ds \tag{4.53}$$

where det J is the determinant of the Jacobian operator (equation 4.49). In the case of plane stress t is the element thickness, in plane strain t is unity, and in the axisymmetric case t is 2π times the radius from the axis of symmetry to the point (r,s).

In general the inverse of J in equation (4.48) and the integral in equation (4.52) cannot be explicitly evaluated and thus numerical integration must be used. The twenty-four by twenty-four element elastic stiffness matrix is evaluated as anywhere in the assemblage. In a manner which is analogous to the treatment of constant strain finite elements discussed previously, the elastic-plastic interface is determined by evaluating the yield function at each Gauss point of the assemblage and on this basis dividing the body into an elastic domain V_e where $\phi < 0$, and a plastic domain V_p where $\phi = 0$.

To formulate the minimum principle the functional $\bar{W}^o$ of equation (3.20) is required, integrated over the body. Defining a functional $\bar{W}^o_q$ for the qth element of the assemblage as

$$\bar{W}^o_q = \frac{1}{2} D_{ijk\ell} \left(\dot{\varepsilon}_{ij} - \lambda\frac{\partial\phi}{\partial\sigma_{ij}}\right)\left(\dot{\varepsilon}_{k\ell} - \lambda\frac{\partial\phi}{\partial\sigma_{k\ell}}\right) + \frac{\lambda^2}{2G}, \tag{4.55}$$

it follows from the discretization of the body into m elements that

$$\int_V \bar{W}^o \, dV = \sum_{q=1}^{m} \int_{V'_q} \bar{W}^o_q \, dV, \tag{4.56}$$

where V is the volume of the body and V'_q is the volume of the qth element of the assemblage.

Substituting the rate form of the strain-displacement relations (4.50) into the expression for $\bar{W}^o_q$ gives in matrix form for the qth element

$$\bar{W}^o_q = \frac{1}{2} \left([B']\{\dot{u}'\} - \lambda\{\frac{\partial\phi}{\partial\sigma}\}'\right)^T [D'] \left([B']\{\dot{u}'\} - \lambda\{\frac{\partial\phi}{\partial\sigma}\}'\right) + \frac{\lambda^2}{2G} \tag{4.57}$$

where $\{\dot{u}'\}$ are element node velocites, and $[B']$, λ and $\{\frac{\partial\phi}{\partial\sigma}\}'$ are functions of the element natural coordinates r, s.

Exploiting the numerical integration technique, a plastic multiplier λ is associated with each Gauss point of the body, so that the plastic strain rate for the ℓth Gauss point is given by

$$\{\varepsilon^p\}'_\ell = \lambda_\ell \{\frac{\partial\phi}{\partial\sigma}\}'_\ell . \tag{4.58}$$

Hence, using Gauss quadrature to integrate equation (4.57) yields

$$\int_{V'_q} \bar{W}^o_q \, dV = \sum_{i,j} \alpha_{ij} \bar{W}^o_q(r_i,s_j)t \det J, \tag{4.59}$$

and substituting this in equation (4.56) we have finally

$$\int_V \bar{W}^o \, dV = \sum_{q=1}^{m} \sum_{i,j} \alpha_{ij} \bar{W}^o_q(r_i,s_j)t \det J. \tag{4.60}$$

To formulate $\bar{U}^o_p$ the rate form of the potential energy of the boundary tractions is required. In the isoparametric formulation velocities along the boundaries of an element are described directly by the shape functions and element node velocities. Thus the rate form of the potential energy of the traction rates $\dot{T}$ (r,s) can be written in terms of an element nodal load rate vector $\{\dot{P}'\}$ and the element node velocities $\{\dot{u}'\}$ as

$$\{\dot{u}'\}^T\{\dot{P}'\} = \int_{S'_T} \dot{u}(r,s)\dot{T}(r,s)dS. \tag{4.61}$$

In general numerical integration must be used to evaluate the integral. Using Gauss quadrature the expression for $\{\dot{P}'\}$ becomes

$$\{\dot{P}'\} = \sum_{i,j} \alpha_{ij} \Big(\sum_{\rho=1}^{12} H_\rho(r_i,s_j)\dot{T}(r_i,s_j)\Big)t \det J, \tag{4.62}$$

where H is a vector of shape functions. The global load rate vector $\{\dot{P}'\}$ is obtained by summing contributions from each element nodal load rate vector in appropriate positions corresponding to the ordered velocity vector $\{\dot{u}\}$ for the assemblage.

In the absence of body forces the minimum principle for the assemblage of isoparametric elements becomes:

$$\text{minimize } \bar{U}_p^o = \int_V \bar{W}^o \, dV - \{\dot{u}\}^T\{\dot{P}\} \tag{4.63}$$

subject to the constraints $\lambda = 0$ in V_e and $\lambda \geq 0$ in V_p. In a manner similar to the formulation for constant strain elements, a system matrix [K*] corresponding to all possible degrees of freedom may be assembled from equation (4.63). Since a plastic multiplier λ is associated with each Gauss point of the body, the size of this system matrix will be $(n + 9m)$ square. As before the governing equations are homogeneous in the rates and hence a solution may be determined for a load increment $\{\Delta P\}$ by employing the minimization algorithm to delete appropriate rows and columns. In so doing each Gauss point of the body is considered in an analogous manner to the treatment of each constant strain element, as described in the minimization algorithm given in section (4.3).

The global formulation for the isoparametric problem has been presented for the sake of completeness and for comparison with the global constant strain formulation. In practice the full $(n + 9m)$ square system matrix [K*] is not evaluated. We proceed as follows: at the start of the analysis problem the elastic system stiffness matrix $[B^TDB]$ is evaluated and stored. Since all integration points are in V_e the first load increment corresponds to a linear elastic analysis. These results are scaled by a load factor just large enough to ensure $\phi = 0$ for at least one integration point.

At an arbitrary stage in the loading when part of the body is plastic we wish to assemble the system matrix. As before the elastic system matrix is retrieved from storage and entered into [K*]. Considering each Gauss point of the assemblage in turn we identify those points in V_p and, beginning from $[K^*] \equiv [B^T DB]$, assign an additional column and row of [K*] to each plastic integration point. For the ℓth plastic point the terms to be entered in appropriate positions of the ℓth additional column are evaluated as the vector $\{\alpha_{ij}[B']^T[D']\{\frac{\partial\phi}{\partial\sigma}\}' t \det J\}$ and scalar $\alpha_{ij}\left(\{\frac{\partial\phi}{\partial\sigma}\}'^T[D']\{\frac{\partial\phi}{\partial\sigma}\}' + E_p\right)t \det J$, where $[B']$, $\{\frac{\partial\phi}{\partial\sigma}\}'$, α_{ij}, t and det J are evaluated at the point ℓ. Having assembled the condensed system matrix the solution algorithm may proceed as described for the constant strain finite elements but analogously we consider each integration point instead of each element.

Provided the isoparametric element discretization is not too coarse, it is found that a further saving in computer storage and computation can be achieved if only one column of the $[-B^T DN]$ and $[N^T DN + D_p]$ submatrices is assigned to each plastic or partially plastic finite element, and contributions from all plastic integration points within that element added into this single column. Thus whether an element is totally plastic or an elastic-plastic interface occurs within the element a single 'average' value of plastic multiplier λ is associated with all non-zero plastic strains within the element. In plastic regions stress gradients are generally small, especially in the case of no strain hardening. It follows that variation in yield function gradient (and thus plastic strain direction) across the plastic domain will also be of the same order. The effect therefore, of associating only one plastic multiplier with each plastic or partially plastic element is to average out plastic strain magnitudes across the element. However employing this device results in large savings in computation and computer storage, while still providing a reasonable approximate solution to the problem.

Illustrative numerical examples will be presented in Chapter 6 for both constant strain finite element analyses and cubic quadrilateral isoparametric finite element analyses. In the following Chapter solution techniques and computational details will first be given.

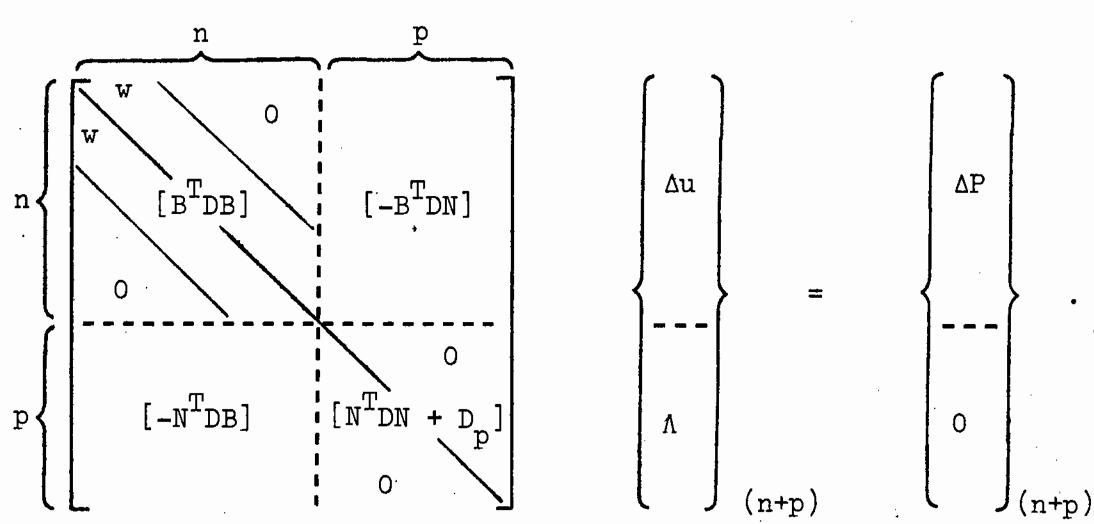
C H A P T E R 5

SOLUTION TECHNIQUES, UNIQUENESS AND SOME NOTES ON THE COMPUTATIONAL PROCEDURE

5.1 Introduction

The structure of the system matrix [K*] may be exploited to minimize computer storage requirements and reduce computational effort.

Consider solution of the system of simultaneous linear equations (4.34). Let there be n displacement increment variables and at an arbitrary stage in the loading history, p plastic multiplier variables. Since only relevant element submatrices of [K*] are evaluated the system matrix is (n + p) square. [K*] is symmetric, and the n by n submatrix $[B^TDB]$ is banded, the band-width depending on the nodal numbering of the assemblage. Let the half band-width be w. In general, employing a judicious nodal numbering scheme, w << n. The p by p submatrix $[N^TDN + D_p]$ is diagonal, while the n by p submatrix $[-B^TDN]$ is sparse but of no fixed form. We may depict the set of (n + p) simultaneous linear equations as follows:



(5.1)

The symmetry and sparseness of [K*] permit an efficient computer storage scheme:

(i) only half the band of the elastic system stiffness matrix $[B^T DB]$ is evaluated. As this submatrix is identical for all system matrices [K*] throughout the loading program, it is stored on a peripheral storage device (magnetic-drum, -disc or -tape) and retrieved at the start of each load increment.

(ii) the submatrix $[-B^T DN]$ must be recalculated for each load increment as entries depend on current yield function gradients for those stress points on the yield surface. For each load increment this submatrix is stored on a peripheral device as unloading of stress points may occur necessitating iteration within a load increment.

(iii) the submatrix $[N^T DN + D_p]$ is also dependent on current stresses and is therefore evaluated for each increment, but since it is diagonal only the diagonal elements are stored.

The minimum storage required to evaluate the (n + p) unknowns 'in core' is that required in (i), (ii) and (iii) above, plus an additional work area of size p by p. This gives a total core requirement of $(nw + (n+1)p + p^2)$ for [K*]. In general $w \ll n$ and p varies from zero according to development of plastic regions in the body. However, in the analysis of realistic problems the maximum value of p is usually of order $p \ll n$. In comparison, core storage required for the complete system matrix is $(n^2 + 2np + p^2)$ storage locations.

Use of minimum storage requirements and variation in p throughout the loading history suggest consideration of alternate solution techniques for the set of simultaneous linear equations.

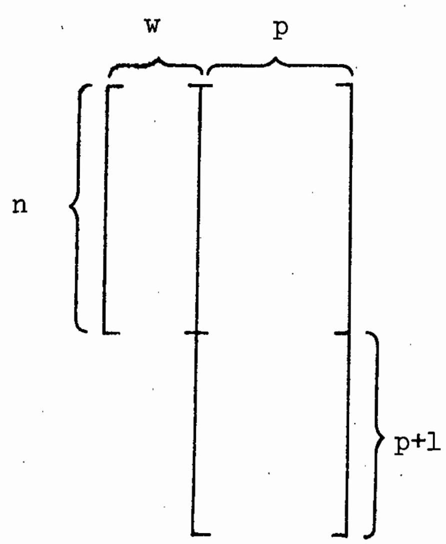


Figure 5.1 Minimum core storage requirements for system matrix [K*]

5.2 Partitioning and Triangular Decomposition

Rewriting equation (5.1) concisely as

$$\left[\begin{array}{c|c} K_1 & K_2 \\ \hline K_2^T & K_3 \end{array}\right] \left\{\begin{array}{c} \Delta u \\ \hline \Lambda \end{array}\right\} = \left\{\begin{array}{c} \Delta P \\ \hline 0 \end{array}\right\} , \qquad (5.2)$$

and eliminating displacement increments from the second set of the partitioned equations gives

$$[K_3 - K_2^T K_1^{-1} K_2]\{\Lambda\} = \{-K_2^T K_1^{-1} \Delta P\}, \qquad (5.3)$$

where we note that the inverse of the elastic system matrix $[K_1]$ is required. This appears attractive as $[K_1]$ is unchanged for each load increment and therefore need be inverted once only for each analysis problem. However the inverse of a symmetric banded matrix is symmetric but not banded and therefore approximately $\frac{1}{2}n^2$ storage locations would be required to store a lower or upper triangle. Instead, using elementary row operations $[K_1]$ is decomposed into the product of two triangular matrices, each retaining the half band-width of $[K_1]$. Further, since $[K_1]$ is symmetric these triangular matrices are transposes of one another, and thus we may write

$$[K_1] = [U^T][U], \tag{5.4}$$

where [U] is an n x n upper triangular matrix with band-width w. Entries in the band of [U] are determined from $[K_1]$ in situ, occupying the storage space originally allocated to the half band of $[K_1]$.

Substituting for $[K_1]$ in equation (5.3) gives

$$[K_3 - K_2^T(U^TU)^{-1}K_2]\{\Lambda\} = \{-K_2^T(U^TU)^{-1}\Delta P\}, \tag{5.5}$$

and writing $[V] = [(U^T)^{-1}K_2]$ this becomes

$$[K_3 - V^TV]\{\Lambda\} = \{-V^T(U^T)^{-1}\Delta P\}. \tag{5.6}$$

In evaluating [V] the ith column is calculated from

$$\{V\}_i = [U^T]^{-1}\{K_2\}_i, \tag{5.7}$$

or more conveniently

$$[U^T]\{V\}_i = \{K_2\}_i, \tag{5.8}$$

where the triangularity of $[U^T]$ reduces evaluation of $\{V\}_i$ to a forward substitution process only. The columns of [V] are calculated in situ in $[K_2]$ and thus no additional storage is required. Once [V] has been determined the p by p matrix $[\bar{K}_3]$ can be formed, where

$$[\bar{K}_3] = [K_3 - V^T V], \qquad (5.9)$$

and is stored in the p by p work area described previously.

Using forward substitution on $[U^T]$ and pre-multiplying by $[V^T]$, the right-hand side of equation (5.3) is evaluated as

$$\{\bar{P}\} = \{-V^T (U^T)^{-1} \Delta P\}, \qquad (5.10)$$

and hence equation (5.3) reduces to

$$[\bar{K}_3]\{\Lambda\} = \{\bar{P}\}, \qquad (5.11)$$

where the matrix $[\bar{K}_3]$ is symmetric. Hence the p plastic multipliers are obtained using Gauss elimination and back substitution.

The plastic submatrix $[K_2]$ which was destroyed in the formation of [V] is now retrieved from peripheral storage (where it was retained in the event of iteration within a load increment). Writing the first n equations of (5.2) as

$$[U^T U]\{\Delta u\} = \{\Delta P - K_2 \Lambda\}, \qquad (5.12)$$

the right hand side is evaluated explicitly as {P*}. Introducing a new vector {v} such that

$$\{v\} = [U]\{\Delta u\}, \tag{5.13}$$

equation (5.12) becomes

$$[U^T]\{v\} = \{P^*\}, \tag{5.14}$$

from which {v} is determined by forward substitution. Displacement increments {Δu} are then obtained through back substitution in equation (5.13).

The $[U^T][U]$ decomposition of $[K_1]$ is performed only once in the analysis procedure. Since the subsequent (p + 3) substitutions on [U] (or $[U^T]$) do not require much computational effort, the solution technique is efficient when p is small - that is when loading is such that only a small part of the body is plastic. Computational effort increases significantly as p increases because in addition to forward and back substitutions on [U] or $[U^T]$, the plastic multipliers are determined from equation (5.11). This entails evaluation of $[\bar{K}_3]$ and {P*}, and Gauss reduction of p simultaneous linear equations.

5.3 Gauss Elimination

The second technique considered involves Gauss elimination operating on both elastic and plastic submatrices of [K*] for each solution. Again referring to the matrices of equation (5.2) only the half band of $[K_1]$ and elements of $[K_2]$ and $[K_3]$ on and above the leading diagonal are stored as depicted in Figure (5.1). A modified Gauss elimination procedure is employed to operate on the distorted matrix and load increment vector, reducing [K*] to an upper triangular form. Plastic multipliers and displacement increments are then obtained by back substitution.

Note that this solution procedure entails Gauss elimination of the constant submatrix $[K_1]$ for each solution, and is therefore inefficient when p is small.

5.4 Relative Computational Efficiency

In the preceding sections two different solution techniques have been proposed for solution of the set of simultaneous linear equations (4.34). The computational effort required by each method can be approximately assessed.

Since zero entries in the system matrix permit a reduction in the arithmetic required to produce a solution, the density, or number of non-zero entries in a matrix, affects the computation time. Employing a judicious nodal numbering scheme, for most assemblages of finite elements the elastic stiffness matrix $[K_1]$ is tightly banded. For convenience let us here assume that the band-width is fully populated with non-zero entries. Density of the $[K_2]$ submatrix is known: in the case of triangular constant strain finite elements there are only six non-zero entries per column, while for cubic quadrilateral isoparametric elements there are twenty-four non-zero entries per column. However, the number of arithmetic operations performed on $[K_2]$ during either solution procedure is dependent on the position of the first non-zero entry encountered in each column of $[K_2]$. This, in turn, is dependent on the nodal numbering scheme and node numbers corresponding to plastic elements. Although overall computational effort expended in producing a solution by either method is therefore problem dependent, an 'average' assessment can be made.

Again consider the solution of a system of equations (5.2) in n displacement increment variables and at a stage in the loading when there are p plastic multiplier variables. As before let the half band-width of $[K_1]$ be w. Assume that the $[U^T][U]$ decomposition of $[K_1]$ has already been performed. Treating a computer subtraction operation as equivalent to one of addition, and considering only significant third-order terms in n, w and p, we find that the technique of partitioning and triangular decomposition requires approximately $(np^2 + 2nwp)$ additions and $(np^2 + nwp)$ multiplications to produce a solution. The second technique (Gauss elimination of [K*]) consists of approximately $(\frac{1}{4}np^2 + nwp + 2nw^2)$ additions and $(\frac{1}{4}np^2 + nwp + nw^2)$ multiplications.

Since an addition operation takes approximately three quarters of the execution time required to perform a multiplication operation, the number of computations for each technique may be written in terms of addition operations only as $(2{,}3np^2 + 3{,}3nwp)$ and $(0{,}6np^2 + 2{,}3nwp + 3{,}3nw^2)$ respectively. Equating these numbers of addition operations, the two methods have approximately the same efficiency when

$$1{,}7p^2 + wp - 3{,}3w^2 = 0, \tag{5.15}$$

i.e. when $p = 1{,}1w$. (From equation (5.15) the comparison of computational effort for each of the two methods is independent of n).

In summary, we deduce that the technique of partitioning and triangular decomposition is more efficient than Gauss elimination of [K*] until the number of plastic multiplier variables is approximately equal to the half band-width of $[K_1]$. If p increases beyond this we expect the latter technique to become more efficient.

Figures (5.2a) - (5.2d) show the execution time† required per solution plotted against p, for four analyses of continua, each with monotonic increase of load. The nodal numbering of each problem was such that the band of $[K_1]$ was densely populated, having few zero terms. In Figures (5.2)(a) and (b) the analyses employed isoparametric finite elements, while those of Figures (5.2)(c) and (d) are results of constant strain finite element analyses. Note in (c) and (d) the curves intersect at a value of p slightly greater than p = 1,1w.

As noted previously the number of arithmetic operations performed on the $[K_2]$ submatrix of equation (5.2) is dependent on two factors. Firstly, since a string of zero entries in a column of the submatrix permits a reduction in computation, the position of the first non-zero entry encountered during solution affects the total computation time. The second factor (which directly influences the first) is the overall density of $[K_2]$. In the quantitative assessment of numbers of arithmetic operations resulting in the computational equivalence of the two solution methods at p = 1,1w, average density and random locations of non-zero entries in $[K_2]$ were assumed. In (c) and (d) the additional sparseness of $[K_2]$ for constant strain element analyses (six non-zero entries per column compared with twenty-four) causes the partitioning technique to remain more efficient for values of p slightly greater than p = 1,1w.

In the computer programs developed by the writer to implement the application of the minimum principle to elasto-plastic analysis of continua, both solution techniques are available. Since relative computational efficiencies of the two techniques are problem dependent with variation related to node numbers of plastic elements, the computer

† Computation times refer to a UNIVAC 1106 computer

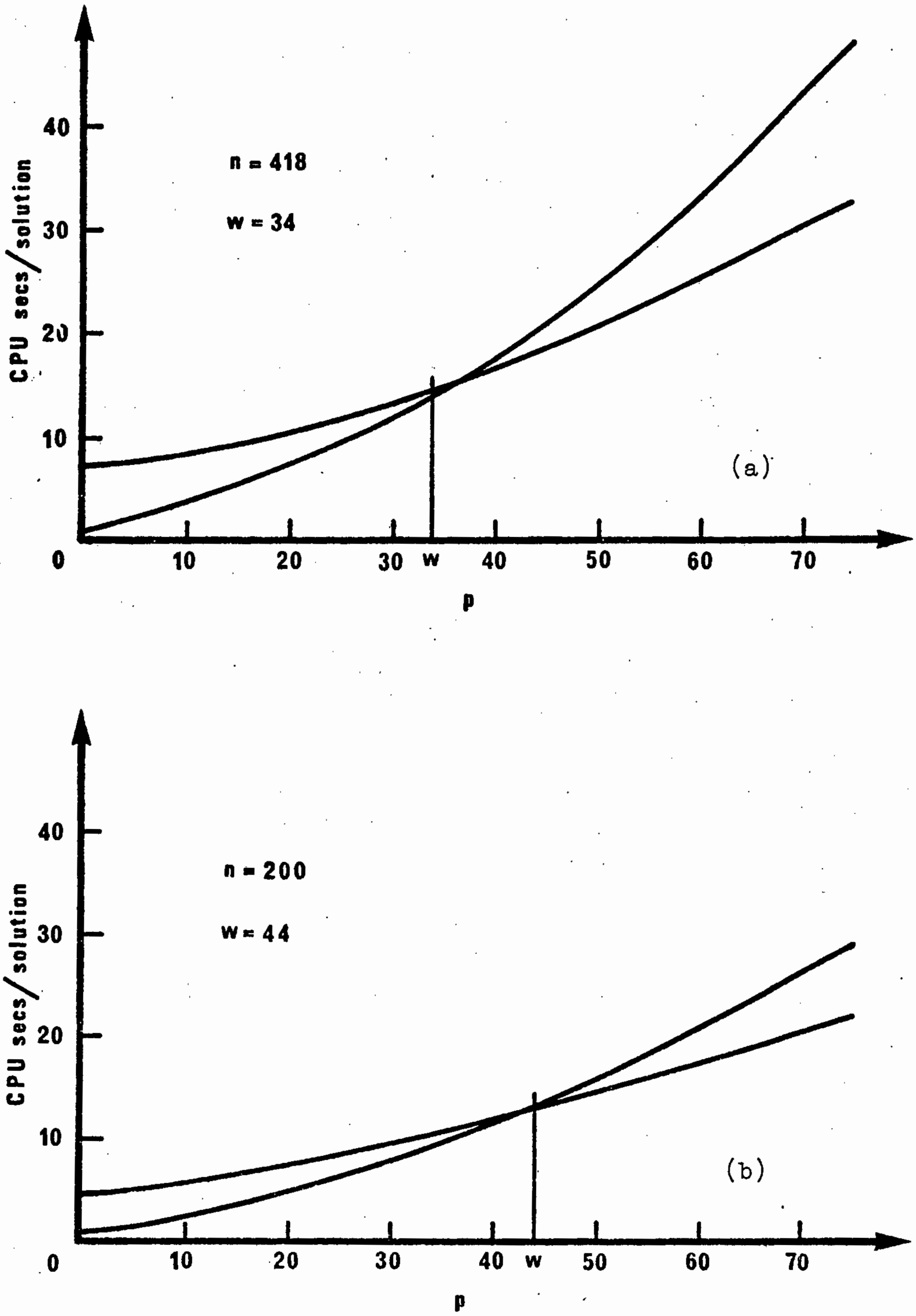


Figure 5.2 CPU time/solution versus number of plastic multipliers p

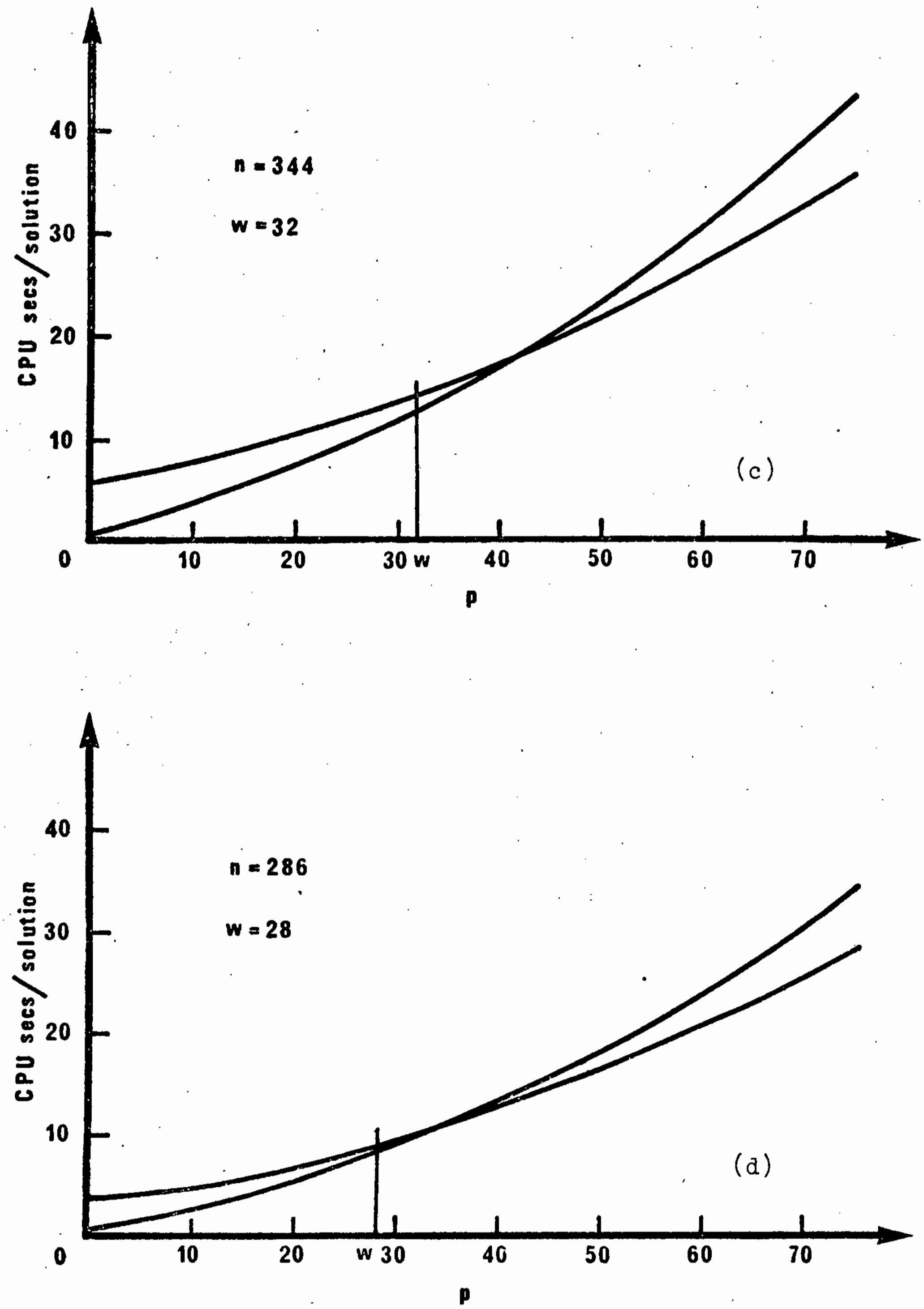


Figure 5.2 CPU time/solution versus number of plastic multipliers p

programs select the solution method arbitrarily on the basis of $p < w$ or $p \geq w$: if $p < w$ partitioning and triangular decompsition is used, and if $p \geq w$ Gauss reduction of [K*] is used.

5.5 Uniqueness of Solution in Elastic-Perfectly Plastic Case

The functional $\bar{W}_p^o$ of equation (3.20) subject to constraints (3.21) represents in rate form, the strain energy of the body. It follows therefore, that in the numerical application of the minimum principle the system matrix [K*] is positive definite unless flow occurs in the elastic-perfectly plastic case.

Consider an assemblage of elastic-perfectly plastic constant strain finite elements at the point of flow. Let there be say, p plastic elements at this stage. Ordering the plastic elements and taking the weighted plastic strain rate components in turn leads to a vector $\{\dot{\delta}^p\}$. At flow the plastic strain rate is the total strain rate, hence

$$\{\dot{\delta}^p\} = [\bar{B}]\{\dot{u}\}, \qquad (5.16)$$

where $\{\dot{u}\}$ is the vector of nodal velocities and $[\bar{B}]$ is a condensed system deformation matrix containing only rows of [B] corresponding to elements of $\{\dot{\delta}^p\}$. From equation (4.23)

$$\{\dot{\delta}^p\} = [\bar{N}]\{\Lambda\}, \qquad (5.17)$$

where $[\bar{N}]$ is a condensed gradient matrix containing only columns of [N] corresponding to non-zero elements of $\{\Lambda\}$. Combining equations (5.16) and (5.17) gives

$$[\bar{B}]\{\dot{u}\} = [\bar{N}]\{\Lambda\}. \qquad (5.18)$$

Now defining $[\bar{D}]$ as a block diagonal matrix containing element elasticity matrices $\frac{1}{V'}[D']$ for the ordered plastic elements, the system of equations at flow is

$$[B^T DB]\{\dot{u}\} - [\bar{B}^T \bar{D}\bar{N}]\{\Lambda\} = \{\dot{P}\} \qquad (5.19)$$

and $$-[\bar{N}^T \bar{D}\bar{B}]\{\dot{u}\} + [\bar{N}^T \bar{D}\bar{N}]\{\Lambda\} = \{0\}. \qquad (5.20)$$

Substituting for $[\bar{B}]\{\dot{u}\}$ from equation (5.18) in the partitioned equations (5.20) gives

$$[-\bar{N}^T \bar{D}\bar{N} + \bar{N}^T \bar{D}\bar{N}]\{\Lambda\} = \{0\},$$

that is

$$[0]\{\Lambda\} = \{0\}, \qquad (5.21)$$

indicating that the system matrix [K*] is singular at the load corresponding to incipient plastic flow. The results (5.21) was obtained by considering an assemblage of constant strain finite elements. Following a similar argument for elastic-perfectly plastic isoparametric finite elements leads to an identical result.

In the numerical analysis of test problems the value of the determinant was found to be unreliable in establishing the limit load. This is attributed to two causes. Firstly, the analysis procedure is incremental and piecewise-linear in the sense that elastic and plastic regions of the body are treated as unchanged during a load increment of finite magnitude. New elastic-plastic boundaries are then established and the next load increment applied. In this way the limit load of the assemblage is not approached asymptotically, but in increments of finite magnitude. The second cause is attributed to numerical error accumulation during successive incremental solutions. Since the determinant is evaluated as the product of terms on the leading diagonal of the triangulated system matrix, possible ill-conditioning of the system of equations as the limit load is approached makes the value of the determinant unreliable. At this stage the minimization algorithm does not always converge, but may iterate repetitively interchanging the same groups of loading and unloading plastic elements in successive cycles.

In elastic-perfectly plastic problems a good estimate of limit load is obtained by determining the load at which load increment magnitudes become negligibly small for several successive load increments.

5.6 Load Increment Magnitude

In the finite element application of the minimum principle, the maximum magnitude of any load increment is determined by the smallest increment of load which causes an element of V_e (where $\phi < 0$), to enter V_p (where $\phi = 0$).

Consider an arbitrary stage in the loading program of an assemblage of elements. Current totals of stress, elastic strain, plastic strain, load and displacement are known. For the next load increment the system matrix [K*] is assembled. In determining a solution for this increment the load increment vector $\{\Delta P\}$ need only reflect relative magnitudes of load components. After a solution is determined satisfying the constraints, we have increments of stress, elastic strain, plastic strain and displacement corresponding to the arbitrary magnitude of applied load increment $\{\Delta P\}$.

To determine the correct load increment magnitude we consider each elastic element (for the isoparametric case each elastic integration point) and determine a factor ρ such that $\phi(\sigma_{ij} + \rho\Delta\sigma_{ij}, \varepsilon^p_{ij}) = 0$, where σ_{ij} and ε^p_{ij} are current totals of stress and plastic strain at completion of the previous load increment, and $\Delta\sigma_{ij}$ is the stress increment corresponding to the current load increment $\{\Delta P\}$.

From the expression for the von Mises yield function (equation 2.18) we have

$$\frac{1}{3}\left\{(\bar{\sigma}_{ij} + \rho\Delta\sigma_{ij})(\bar{\sigma}_{ij} + \rho\Delta\sigma_{ij}) - \frac{1}{3}(\bar{\sigma}_{kk} + \rho\Delta\sigma_{hh})^2\right\} - \frac{2}{9}\sigma_o^2 = 0 \tag{5.22}$$

To solve for ρ define a quantity ϕ^* as

$$\phi^*(\alpha,\beta) = \frac{1}{3}\left\{\alpha_{ij}\beta_{ij} - \frac{1}{3}\alpha_{kk}\beta_{\ell\ell}\right\}, \tag{5.23}$$

and expand equation (5.22) to give

$$\rho^2\phi^*(\Delta\sigma,\Delta\sigma) + 2\rho\phi^*(\bar{\sigma},\Delta\sigma) + \phi^*(\bar{\sigma},\bar{\sigma}) - \frac{2}{9}\sigma_o^2 = 0 \tag{5.24}$$

Since $\phi^*(\bar{\sigma},\Delta\sigma)$ and $\phi^*(\bar{\sigma},\bar{\sigma})$ are non-negative and $\phi^*(\Delta\sigma,\Delta\sigma)$ is always greater than zero, the positive root of equation (5.24) is given by

$$\rho = \frac{-\phi^*(\bar{\sigma},\Delta\sigma) + \sqrt{\{\phi^*(\bar{\sigma},\Delta\sigma)\}^2 - \phi^*(\Delta\sigma,\Delta\sigma)\{\phi^*(\bar{\sigma},\bar{\sigma}) - \frac{2}{9}\sigma_o^2\}}}{\phi^*(\Delta\sigma,\Delta\sigma)} \tag{5.25}$$

A value of ρ is calculated for each element of V_e and the smallest, ρ_{min}, identified. All increment quantities for the current solution are scaled by this factor ρ_{min} and added to current totals to give the new state of the assemblage. The element of V_e associated with ρ_{min} is included in V_p for the next load increment, as its stress point now lies in the yield surface. As noted previously, fortuitously there may be more than one element of V_e associated with ρ_{min}, within the bounds of computational accuracy.

5.7 Determination of V_p

Following the procedure described in the previous section for determining load increment magnitudes leads to a large number of load increments, with actual magnitudes of increments becoming very small as large regions of the body become plastic. So as to decrease overall computation time elements of V_e for which ϕ is close to zero are included in V_p.

At an arbitrary stage we wish to determine elements of V_p for evaluation of the current system matrix [K*]. For the previous load increment ρ_{min} was determined and all increment quantities scaled by this factor. Hence, current totals of stress, strain, load and displacement were established. Considering each element of the assemblage in turn, (for the isoparametric case each integration point), the von Mises equivalent stress σ_{eq} is evaluated, where

$$\sigma_{eq} = \sqrt{\phi(\bar{\sigma}_{ij}) + \sigma_o^2}. \tag{5.26}$$

If $\sigma_{eq} \geq \eta\sigma_o$, where η is a preassigned constant, the element (integration point) is included in V_p for the start of the current load increment. After this increment stress corrections are made to ensure that $\phi = 0$ for each element in V_p.

Yamada et al [15] follow a similar procedure and arbitrarily set $\eta = 0,995$. However, for all the numerical examples investigated by the writer a value of $\eta = 0,99$ was found to introduce negligible errors while decreasing computation time significantly. Unless otherwise stated or for the purpose of comparison of computation time, this value of η is adopted for the numerical examples of Chapter 6.

5.8 Correction of Stresses in V_p for Elastic-Perfectly Plastic Case

In the elastic-perfectly plastic case stress points which do not unload undergo neutral loading, and for an infinitesimal stress increment the stress point moves in the yield surface. For a stress increment of finite magnitude the stress point moves along the tangent to the yield surface at that point, resulting in an inadmissible stress state for which $\phi > 0$. The stress point is returned to the yield surface along a radial path by scaling stress components by a suitable factor.

In so doing equilibrium of the assemblage is violated. This could be accounted for by determining the nodal loads corresponding to the equilibrium violation and adding to the current total of load, continuing in an iterative manner until equilibrium is satisfied. However, for the numerical examples investigated by the writer this was deemed unnecessary as correction factors smaller than 0,998 did not occur.

CHAPTER 6

NUMERICAL EXAMPLES

6.1 Introduction

To illustrate application of the extended minimum principle representative results for analyses of some numerical examples are given below. Where possible comparisons are made with published results, thereby indicating orders of accuracy and efficiency. In particular results are given for a Vee-notched tension specimen in plane stress, plane strain or axial symmetry for either perfect plasticity or strain hardening. Comparisons are made with experimental and numerical results from the literature. The second numerical example is a deep cantilever in plane stress subjected to a parabolic shear distribution over the free end. No strain hardening is assumed, and upper bound values of the limit load are obtained for monotonically increasing load. Elasto-plastic loading and unloading are shown for cyclic loading. Results are compared with those of other numerical analyses. Finally, an axisymmetric pressure vessel-flush nozzle junction is analyzed under increasing internal pressure. Comparisons are made with experimental and numerical results.

6.2 Vee-Notched Tension Specimen

One of the earliest papers giving results of a numerical analysis of an elastic-plastic continuum is that of Marcal and King, (1967) [14]. A notched tension specimen was analyzed in plane stress, plane strain and axial symmetry, assuming an elastic-perfectly plastic von Mises material. This problem has subsequently been investigated by others such as Yamada, Yoshimura and Sakurai, (1968) [15], Zienkiewicz, Valliapan and King (1969) [11], and Anand and Shaw (1977) [30], each giving results for plane stress analyses only. Experimental results for such a specimen have been reported by Theokaris and Marketos (1963) [31], who used a technique of birefringent coating and polarized light to determine principal strain distributions for the elastic-perfectly plastic plane stress problem.

The geometry of the specimen is shown in Figure 6.2.1. The notch-depth to half-width ratio is 1 to 2 and notch angle 90°. For the present analysis mechanical properties of the material are $E = 20000$ kg/mm^2; $\nu = 0{,}3$; $\sigma_o = 30$ kg/mm^2, and for the case of hardening $E_p = 650$ kg/mm^2.

From Figure 6.2.1 it may be seen that due to two way symmetry only one quadrant need be considered in the analysis. Loading consists of a uniformly distributed tensile load applied to the ends of the specimen.

Using the computer programs listed in the appendix various analyses of the notched specimen were performed for monotonically increasing tensile loading. The first set comprises constant strain finite element analyses using 244 elements and 143 nodes. The element discretization of a quadrant is shown in Figure 6.2.2.

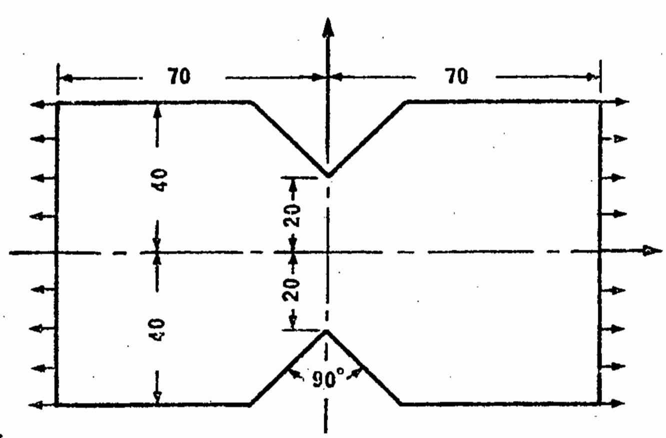


Figure 6.2.1 Vee-notched tension specimen

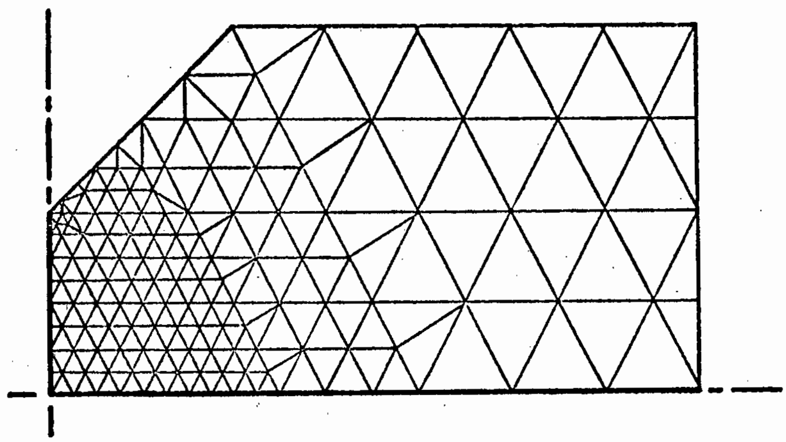


Figure 6.2.2 Constant strain finite element mesh for quadrant of vee-notched specimen

With no strain hardening, plastic elements at some representative stages of calculation are shown in Figure 6.2.3, 4 and 5 respectively for the cases of plane stress, plane strain and axial symmetry. In these Figures load levels are given in dimensionless form as σ_m/σ_o where σ_m is the mean stress at the minimum section. The effect of orientation of triangular constant strain elements, previously observed by Yamada et al [15] and Anand and Shaw [30], is clearly evident.

For the plane stress analysis the plastic enclaves of Figures 6.2.6b and 6.2.6c were drawn by smoothing the jagged elasto-plastic boundaries of finite element analyses due to Yamada et al and the present method, respectively. Experimental results reported by Theokaris and Marketos [31] are shown in Figure 6.2.6a. Although the results of the present analysis are in excellent agreement with those of Yamada et al, the experimental results show development of a continuous plastic region across the width of the specimen at a lower load than that indicated by the numerical analyses. This is attributed to the fact that Theokaris and Marketos give only a minimum value for the uniaxial yield stress σ_o,

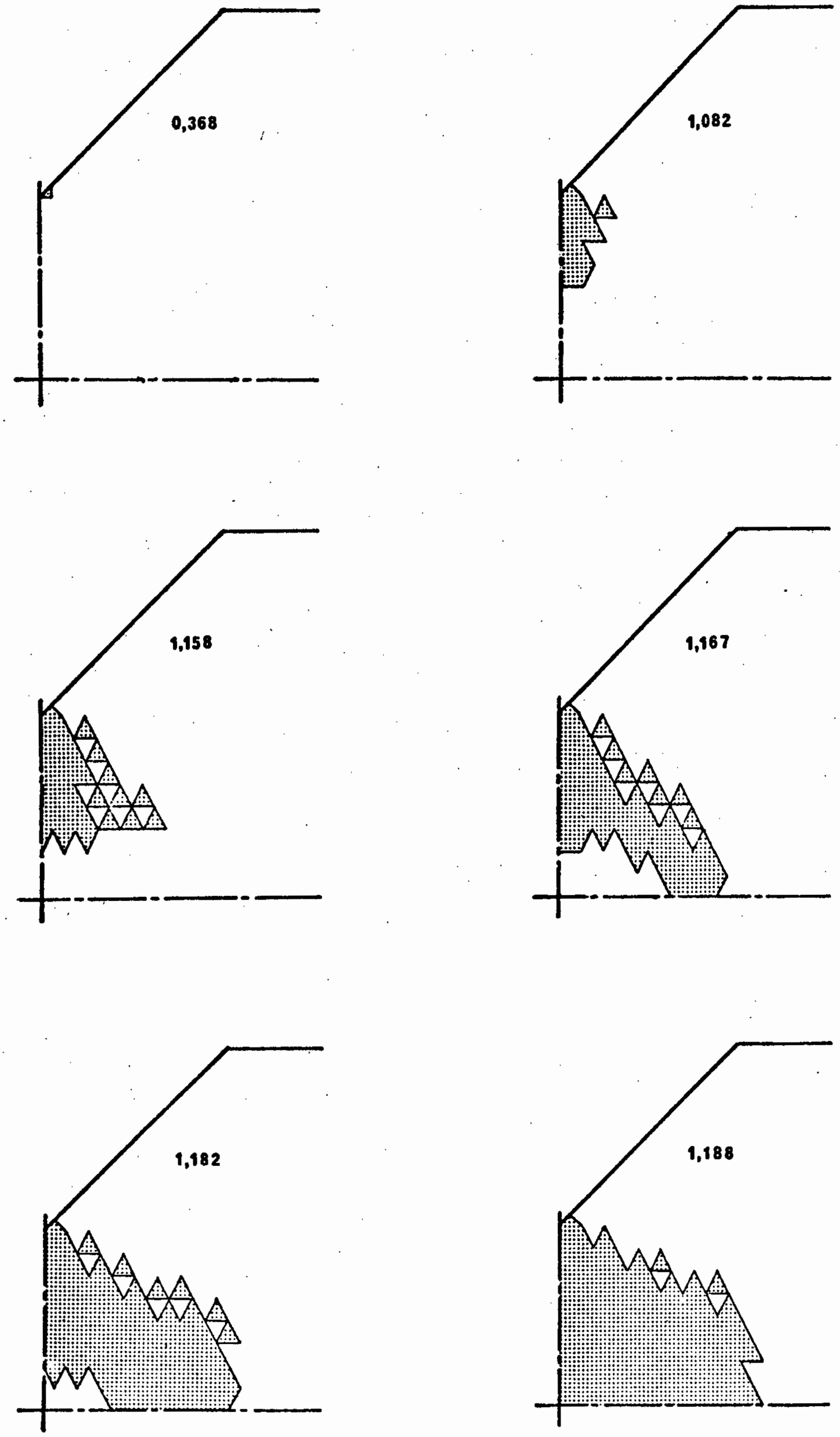


Figure 6.2.3 V-notched specimen, plane stress: constant strain finite element analysis, plastic elements at values of σ_m/σ_o

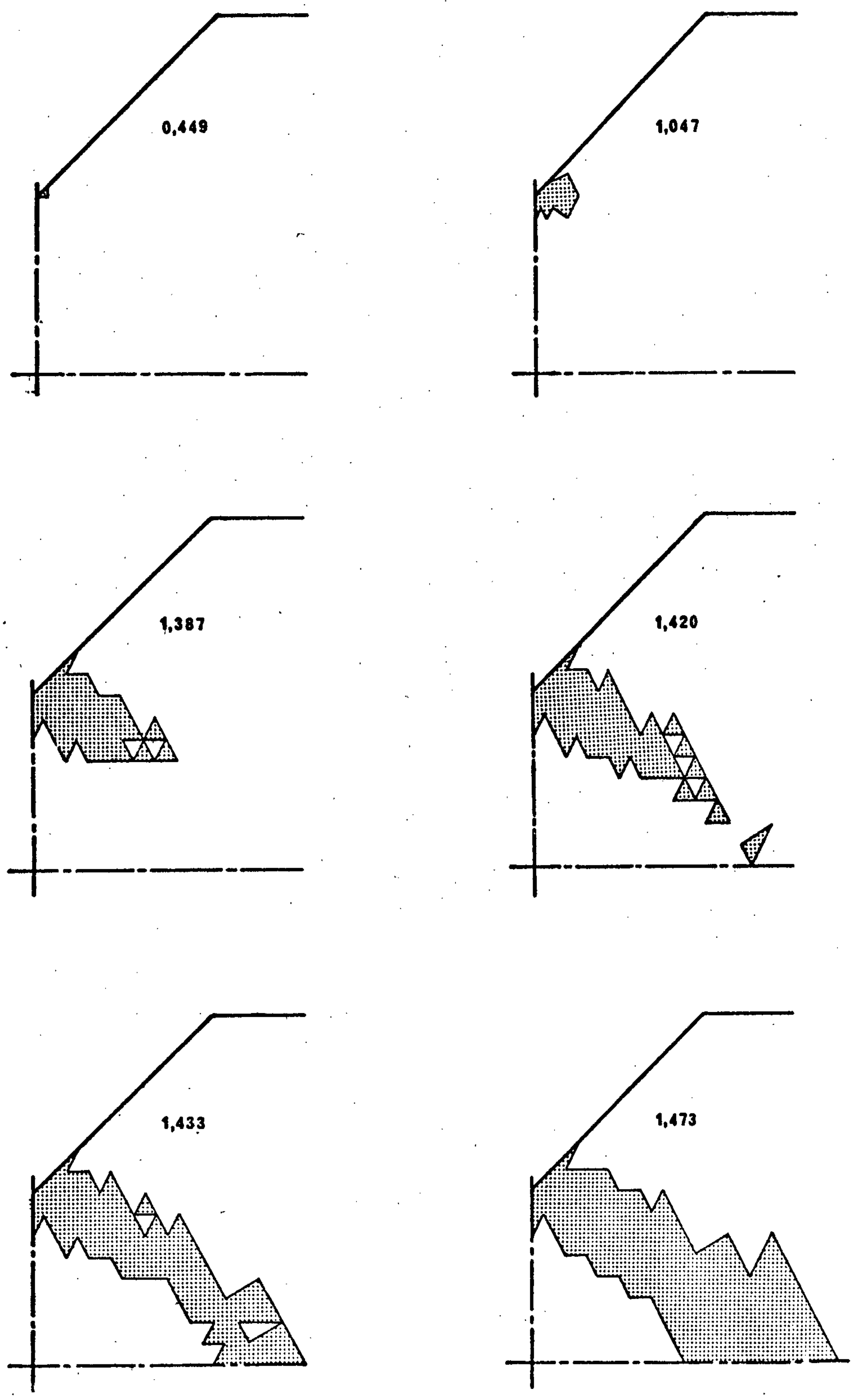


Figure 6.2.4 V-notched specimen, plane strain: constant strain finite element analysis, plastic elements at values of σ_m/σ_o

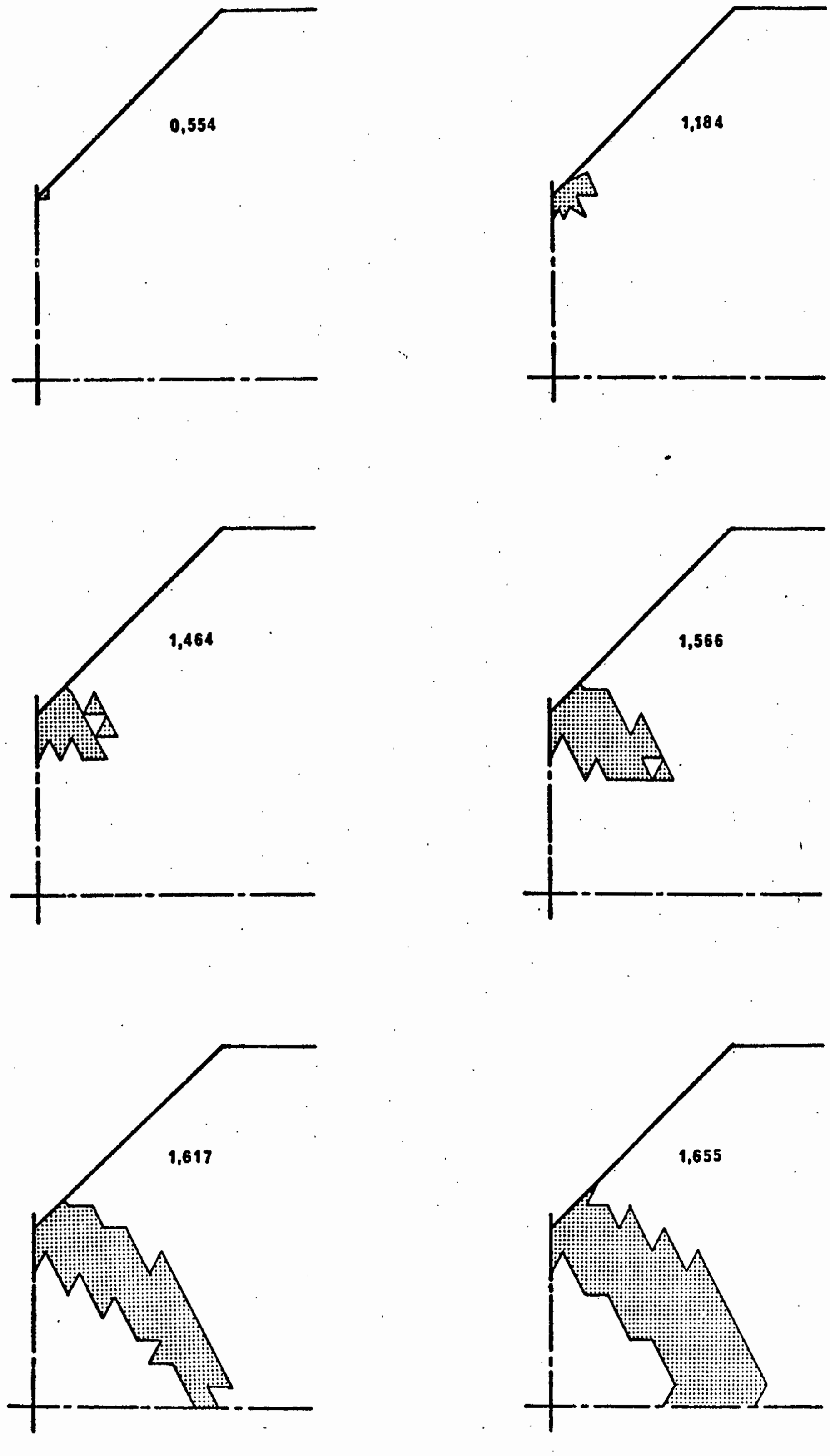


Figure 6.2.5 V-notched specimen, axial symmetry: constant strain finite element analysis, plastic elements at values of σ_m/σ_o

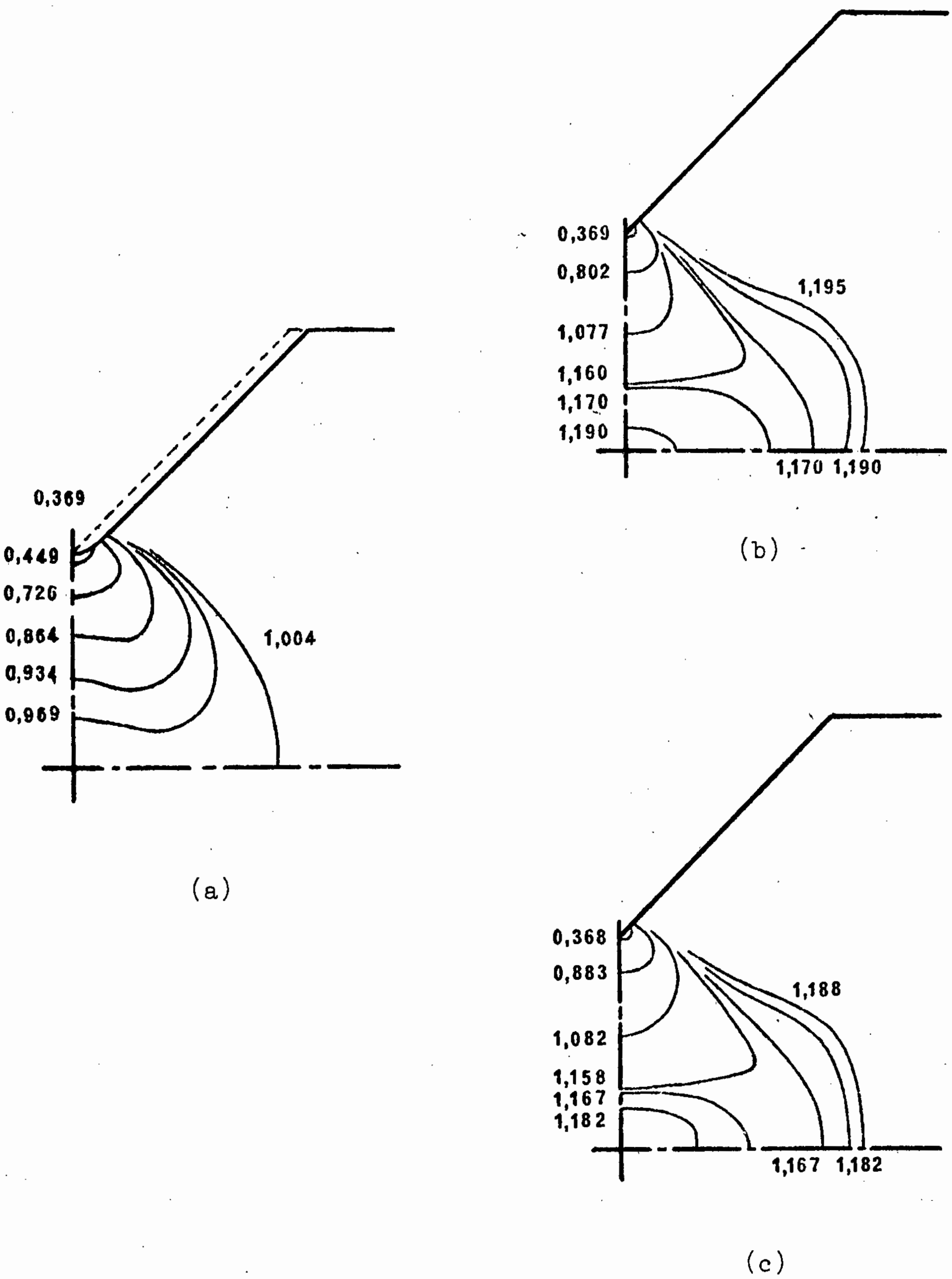


Figure 6.2.6 Vee-notched specimen, plane stress: plastic enclaves at values of σ_m/σ_o; (a) Theokaris and Marketos, experimental; (b) Yamada et al; (c) Present analysis

from which the enclaves at dimensionaless load levels in Figure 6.2.6a were drawn. If the true yield stress was slightly higher than this minimum value, then all plastic regions of the numerical analyses would be in reasonable agreement with those of the experimental results, with the exception of the value of initial yield load. For the numerical analyses the initial yield load is dependent on the finite element mesh in the region of the notch root, while for the experimental investigation the notch root radius was 0,18 of the notch depth. Thus comparisons of initial yield load are not meaningful.

Figure 6.2.7 shows in dimensionless form the maximum strain versus applied load for plane stress, plane strain and axisymmetric analyses using constant strain elements. The dashed curves in the Figure represent results of analyses assuming a kinematic hardening material with plastic modulus $E_p = 650$ kg/mm^2, ($E_p = 0{,}0325E$). For all analyses the maximum longitudinal strain occurred in the element which had become plastic at the initial yield load. Again strain magnitudes are dependent on the finite element mesh at the notch root. The influence of work hardening is only significant at high loads.

Zienkiewicz, Valliapan and King (1969) [11] give results for the elasto-plastic analysis of a similar V-notched specimen under conditions of plane strain. No strain hardening was assumed. Mechanical properties are given as $E = 7000$ kg/mm^2; $\nu = 0{,}2$ and $\sigma_o = 24{,}3$ kg/mm^2. Plastic enclaves due to Ziekiewicz et al are shown in Figure 6.2.8a for representative values of σ_m/σ_o. Using the same constant strain finite element mesh as used by Ziekiewicz et al (149 elements, 94 nodes), plastic enclaves resulting from an analysis using the present method are indicated in Figure 6.2.8b for similar load levels. Good agreement is obtained.

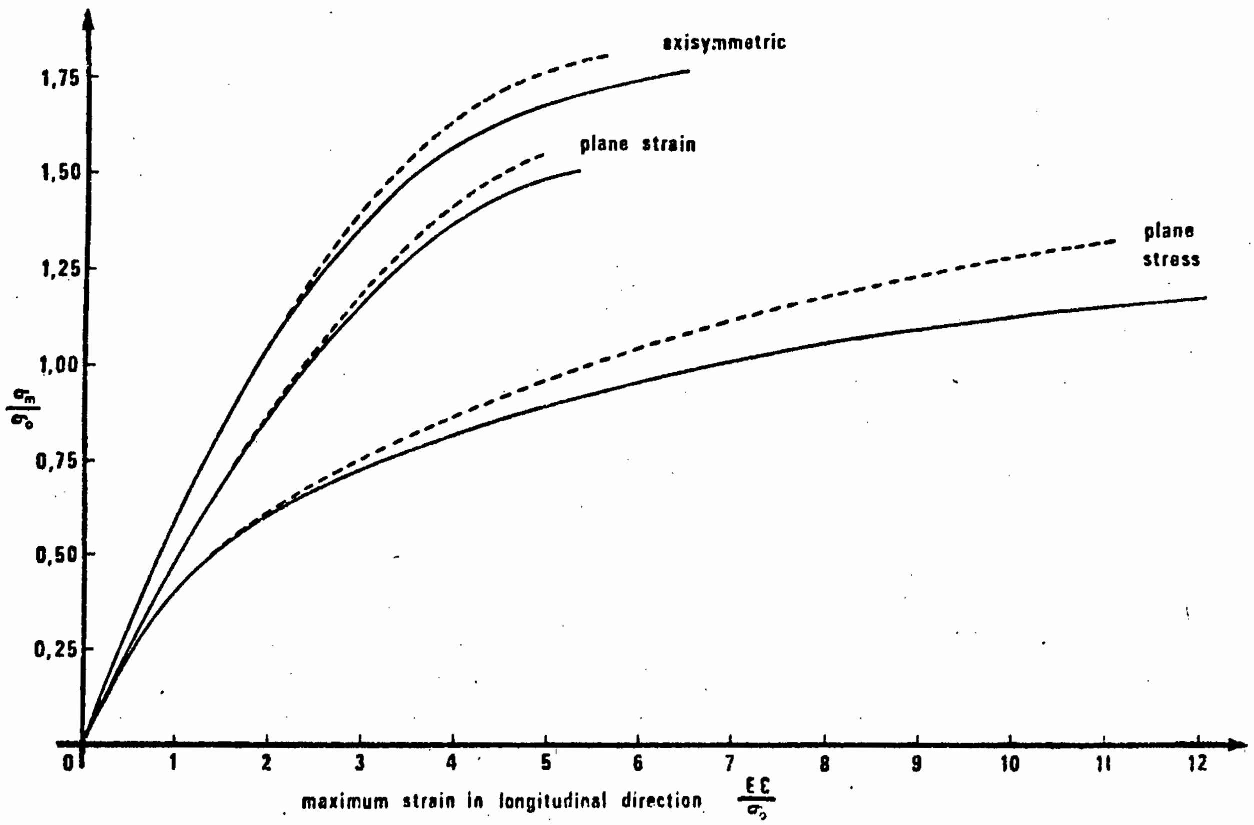


Figure 6.2.7 Load-strain curves for V-notched specimen, constant strain finite element analysis

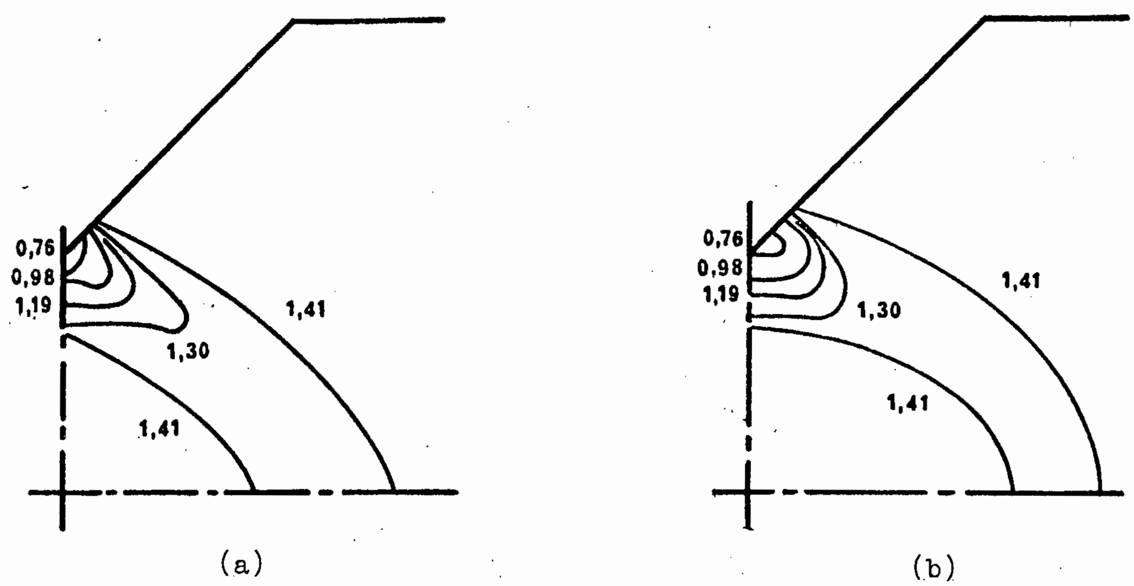


Figure 6.2.8 V-notched specimen, plane strain: plastic enclaves at values of σ_m/σ_o ; (a) Zienkiewicz et al; (b) Present analysis

The next set of results represent cubic quadrilateral isoparametric analyses of a similar notched tension specimen. A mesh of 15 elements and 100 nodes was used for the quadrant as shown in Figure 6.2.9. Nine integration points are indicated for each element, corresponding to third order Gauss integration. Because of the coarseness of the mesh, a plastic multiplier λ was associated with each integration point (and not one 'average' multiplier for each element). With no strain hardening plastic integration points are shown in Figure 6.2.10, 11, and 12 respectively for cases of plane stress, plane strain and axial symmetry. Even though a coarse finite element mesh was used these plastic regions are similar to those of constant strain finite element analysis shown in Figure 6.2.3, 4 and 5. Further, the curves of maximum longitudinal strain versus applied load shown in Figure 6.2.13 for the isoparametric results are in excellent agreement with those of Figure 6.2.7. In both these Figures solid lines correspond to elastic-perfectly plastic analyses while dashed curves result from a hardening material E_p = 0,0325E. In the case of Figure 6.2.13 the maximum longitudinal strain occurs at the Gauss point closest to the notch root.

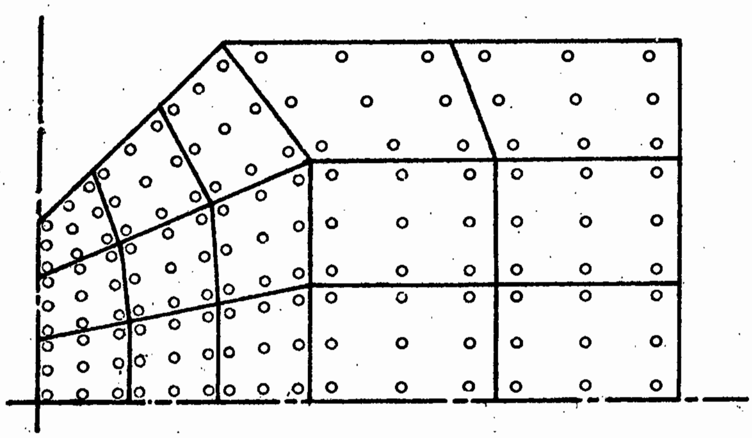


Figure 6.2.9 Cubic quadrilateral isoparametric finite element mesh for quadrant of V-notched specimen

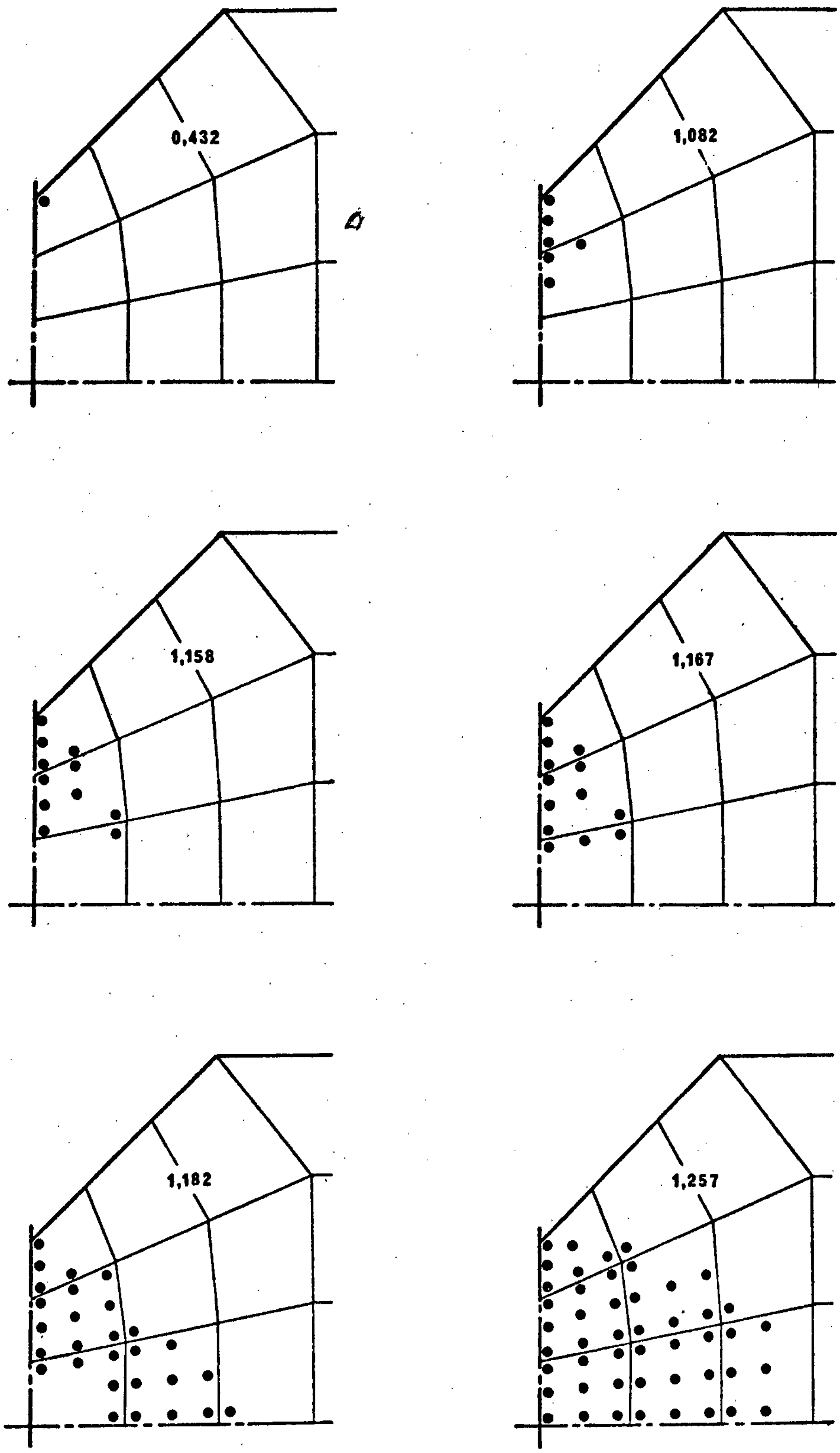


Figure 6.2.10 V-notched specimen, plane stress: isoparametric finite element analysis, plastic integration points at values of σ_m/σ_o

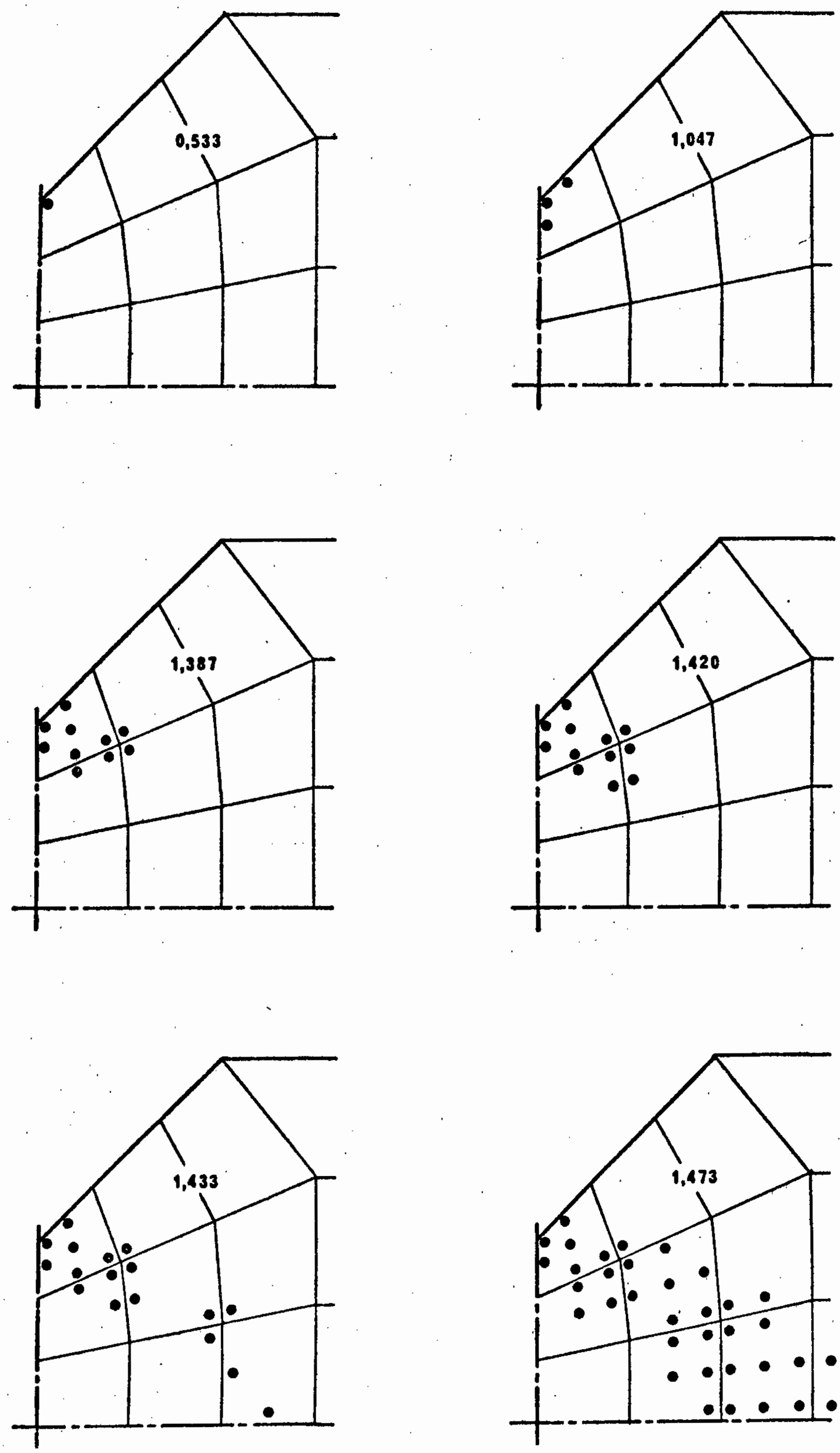


Figure 6.2.11 V-notched specimen, plane strain: isoparametric finite element analysis, plastic integration points at values σ_m/σ_o

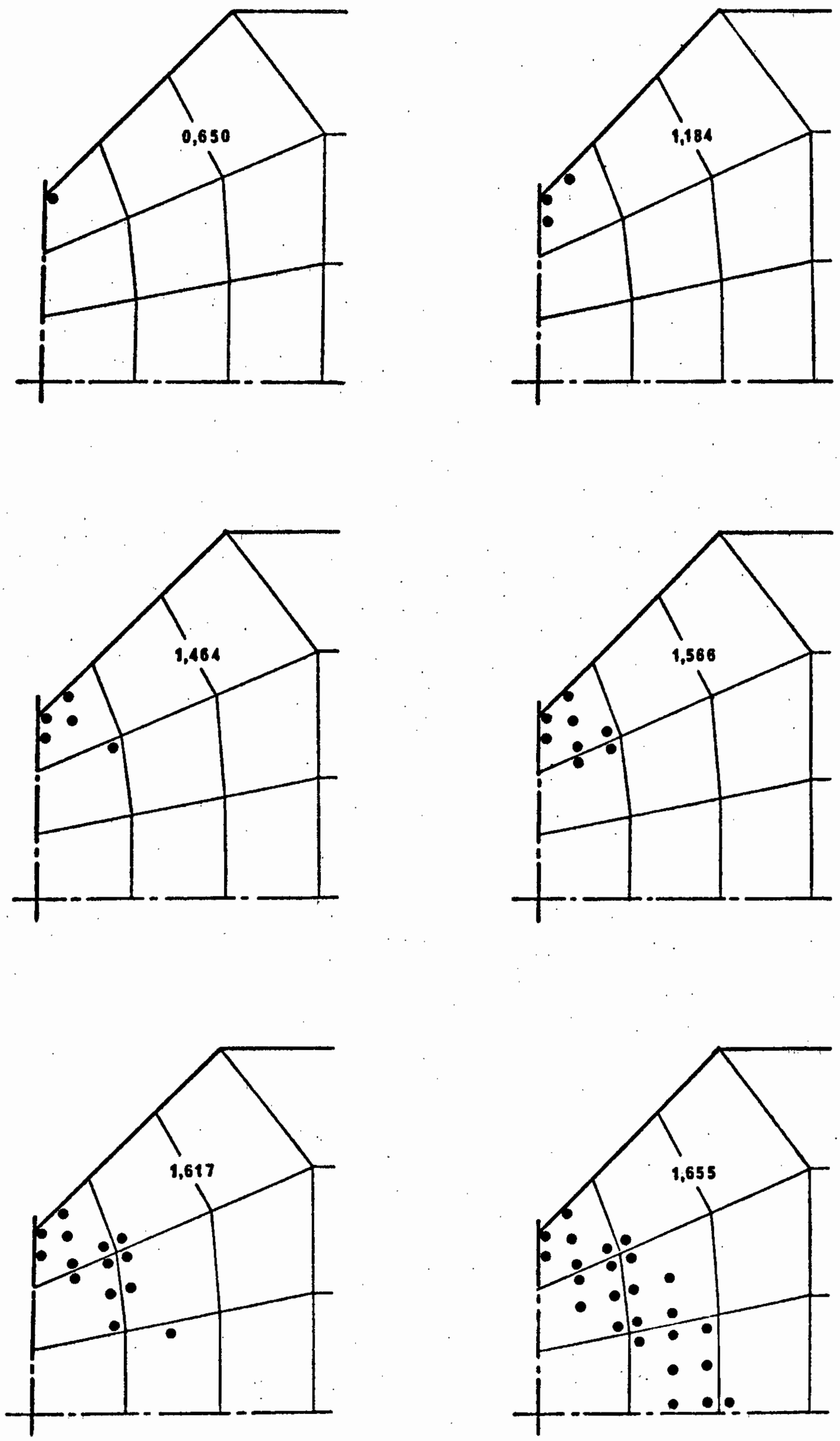


Figure 6.2.12 V-notched specimen, axial symmetry: isoparametric finite element analysis, plastic integration points at values of σ_m/σ_o

The influence of the factor η (see section 5.7), the Gauss integration order, and the number of plastic multipliers associated with each element, are summarized in Table 6.2.7. Sixteen analyses of the V-notched specimen were performed using the element configuration of Figure 6.2.9, to a load of σ_m/σ_o = 1,473. Conditions of plane strain and no strain hardening were assumed. The columns of Table 6.2.1 represent the following: analysis number; η, ratio of von Mises equivalent stress to yield stress for an elastic integration point to be treated as plastic; integration order; number of load increments to a final load of σ_m/σ_o = 1,473; number of plastic integration points at final load; number of plastic integration points at final load as percentage of total number of integration points; u, longitudinal displacement (in mm) at end of specimen on axis of symmetry; and CPU time (mins:secs) to produce result on a UNIVAC 1106 computer.

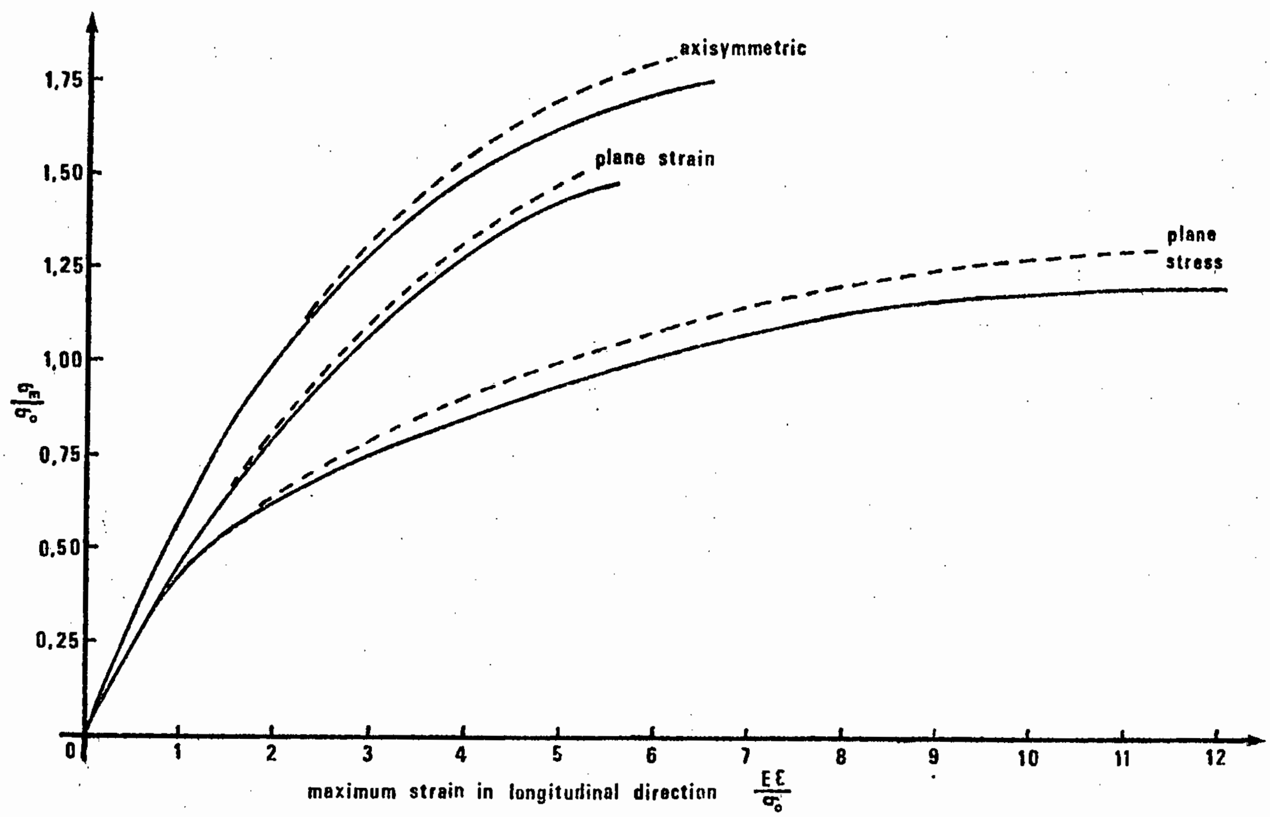


Figure 6.2.13 Load-strain curves for V-notched specimen, isoparametric finite element analysis

Analysis	η	Int. order	λ/elem.	Load incrmts.	Plastic int. pts.	% Plastic	u (mm)	CPU time (m:s)
1	1,000	4	16	58	57	24	0,0927	10:30
2	1,000	3	9	32	31	23	0,0929	3:06
3	0,995	4	16	24	59	25	0,0929	3:23
4	0,995	3	9	19	32	24	0,0930	1:38
5	0,990	4	16	19	63	26	0,0931	3:00
6	0,990	3	9	15	34	25	0,0933	1:20
7	0,975	4	16	14	70	29	0,0942	2:11
8	0,975	3	9	11	37	27	0,0942	1:04
9	1,000	4	1	26	25	10	0,0905	1:45
10	1,000	3	1	17	16	12	0,0906	0:57
11	0,995	4	1	21	33	14	0,0904	1:32
12	0,995	3	1	12	20	15	0,0906	0:43
13	0,990	4	1	17	40	17	0,0905	1:16
14	0,990	3	1	10	22	16	0,0907	0:38
15	0,975	4	1	13	55	23	0,0907	1:02
16	0,975	3	1	9	28	21	0,0908	0:38

Table 6.2.1 V-notched specimen, plane strain: analyses to σ_m/σ_o = 1,473

The plastic integration points of Figure 6.2.11 correspond to analysis 6 in the Table. It may be assumed that analysis 1 produced the most accurate results since fourth order integration, sixteen plastic multipliers per element, and fifty-eight load increments were required to reach the final load, taking a total of 10 min 30 secs CPU time. With a small sacrifice in accuracy computation time can be drastically reduced. Consider, for example, analysis 6. This consists of third order integration, nine plastic multipliers per element, and a value of 0,990 for η. This reduced the number of load increments to fifteen, resulting in an 87% reduction in computation time. The sacrifice in accuracy amounted to a 1% difference in the percentage of plastic integration points, and 0,65% difference in deflection u. In analyses 9 to 16 it is apparent that the finite element mesh is too coarse to consider only one 'average' plastic multiplier per element. Although in these cases the deflection u is reasonably close to the value of analysis 1, the extent of the plastic region (shown by the percentage of plastic integration points) is not reliable. In comparison, the deflection u corresponding to a triangular constant strain element analysis to the same load was found to be 0,0938 mm.

Comparisons of computational efficiencies for different elasto-plastic analysis techniques are not always meaningful. By considering an analysis method alone (and not the associated computer program) it is not possible to assess even approximate numbers of arithmetic operations to be performed, and thus the only measure of relative efficiency is in comparison of computation times required for the same numerical example. However, not only are program running times machine dependent, but certain quantities are dependent on the method of analysis - for example, number of stages in the analysis to a particular load. Thus strict comparisons of execution times for the present method and published results are not

Reference	Method	Element	Nodes	Elements	Load $\frac{\sigma_m}{\sigma_o}$	Stages	Computer	Time(min)
Marcal & King	Tangent Modulus	Constant Strain Triangle	150	250	1,232	8	IBM 7090	15
Yamada et al			149	259	1,192	31		70
Yamada et al			144	245	1,224	51	HITAC 5020E	29
Present analysis			143	244	1,188	34	UNIVAC 1106	11
Present analysis	1 λ/int. pt.	Isopara-metric	100	15	1,257	23		3,5
Present analysis	1 λ/element					15		1
Anand & Shaw	Tangent Modulus	Linear Strain Triangle	172	75	1,236	–	IBM 370/158	46,5
Anand & Shaw		Constant Strain Triangle		300	1,210	–		21,5
Present analysis						20	UNIVAC 1106	2,5

Table 6.2.2 V-notched tension specimen in plane stress, no strain hardening

meaningful. Further, numerical results in the literature generally do not state whether 'computation time' is taken to be total machine time or central processing unit (CPU) time, often significantly different. In this thesis all computation times given for results of the present analyses are CPU times.

In view of the above, computation times listed in Table 6.2.2 for the vee-notched specimen in plane stress should be assessed as only indicative of the order of magnitude of times required by the various analysis procedures. With this in mind the present approach appears attractive in terms of relative computational efficiency. More meaningful comparisons with published results for which similar computers were used, and to the same load level, will be given for the following numerical example.

6.3 Deep Cantilever in Plane Stress

The elasto-plastic analysis of a cantilever of rectangular cross-section in plane stress and with length to depth ratio equal to unity has been reported in the literature. Zienkiewicz and Valliapan (1971) [32] used a constant elasticity matrix initial stress method, whereas de Donato and Franchi (1973) [22] and de Donato and Maier (1973) [20] employed quadratic programming techniques to determine a solution. Using a finite difference technique Neal (1968) [33] established a lower bound for the limit load by determining loads corresponding to safe and statically admissible stress fields.

The cantilever depth d is equal to the length L as shown in Figure 6.3.1. The built-in condition is represented by an L by 2d plane of the same material, also of unit thickness, and fully constrained at the three internal boundaries. Loading consists of a vertical shear force distributed parabolically over the free end BF as shown in the Figure.

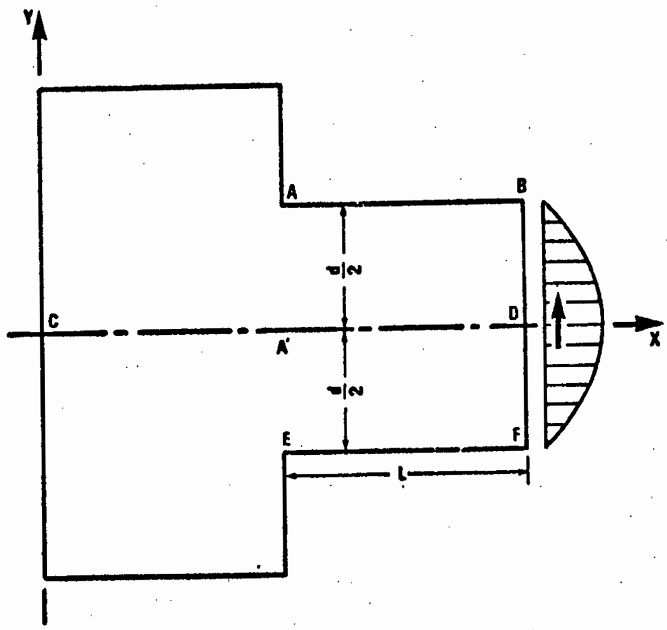
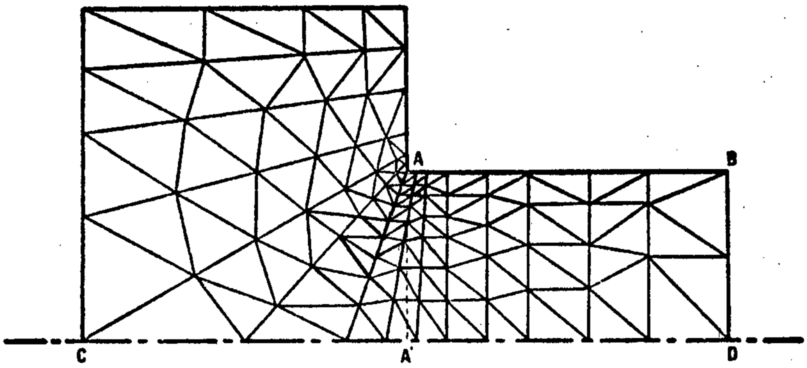


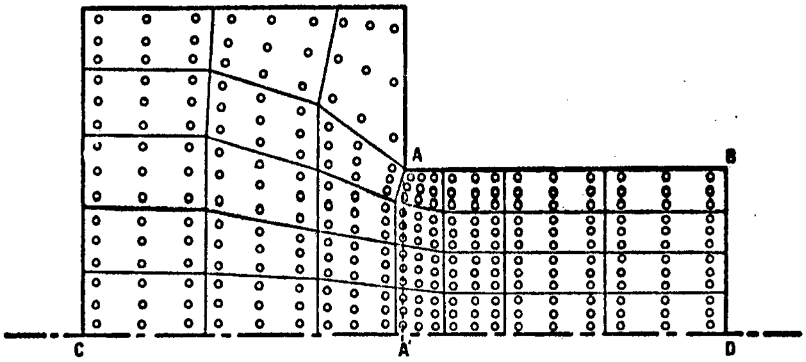
Figure 6.3.1   Deep cantilever in plane stress

If the magnitude of the applied shear stress at point D is $\bar{s}$ then the total load is $W = \frac{2d}{3}\bar{s}$, and the shear stress on BF given by $\sigma_{xy} = \frac{\bar{s}}{d^2}(d^2 - 4y^2)$. Constraining horizontal displacement components along the axis of symmetry CD, only one half of the structure need be considered.

In each of the papers cited above a mesh of 158 triangular constant strain finite elements and 97 nodes was used to represent the half structure. Mechanical properties for the elastic-perfectly plastic von Mises material were given as $E = 1000$ kg/mm^2; $\nu = 0,0$; and $\sigma_o = 32$ kg/mm^2. These values were used in the present analysis. Figures 6.3.2(a) and (b) respectively show configuration of the 158 constant strain elements, and an isoparametric element mesh consisting of 31 elements and 192 nodes. In both cases the element size is decreased in the region of point A, the extreme fibre on the line of maximum moment AA'. The parabolic end load is represented by equivalent nodal loads on BD.



(a)



(b)

Figure 6.3.2 Finite element meshes

The lower bound on the limit load established by Neal is $M/M_o = 0{,}921$; where $M = WL$ and M_o is the fully plastic moment for the section given by $M_o = \frac{1}{4}\, bd^2\sigma_o$. Figure 6.3.3 shows plastic regions of the cantilever resulting from various numerical analyses to this load level. Referring to this Figure, (a) results from the present analysis using constant strain elements. The effect of element orientation is again apparent. The plastic integration points of (b) and (c) are from iso-parametric finite element analyses using the present method - in (b) one plastic multiplier is associated with each integration point, while in (c) one average multiplier is associated with each element. The plastic enclave shown in (d) was obtained using the nonlinear computer

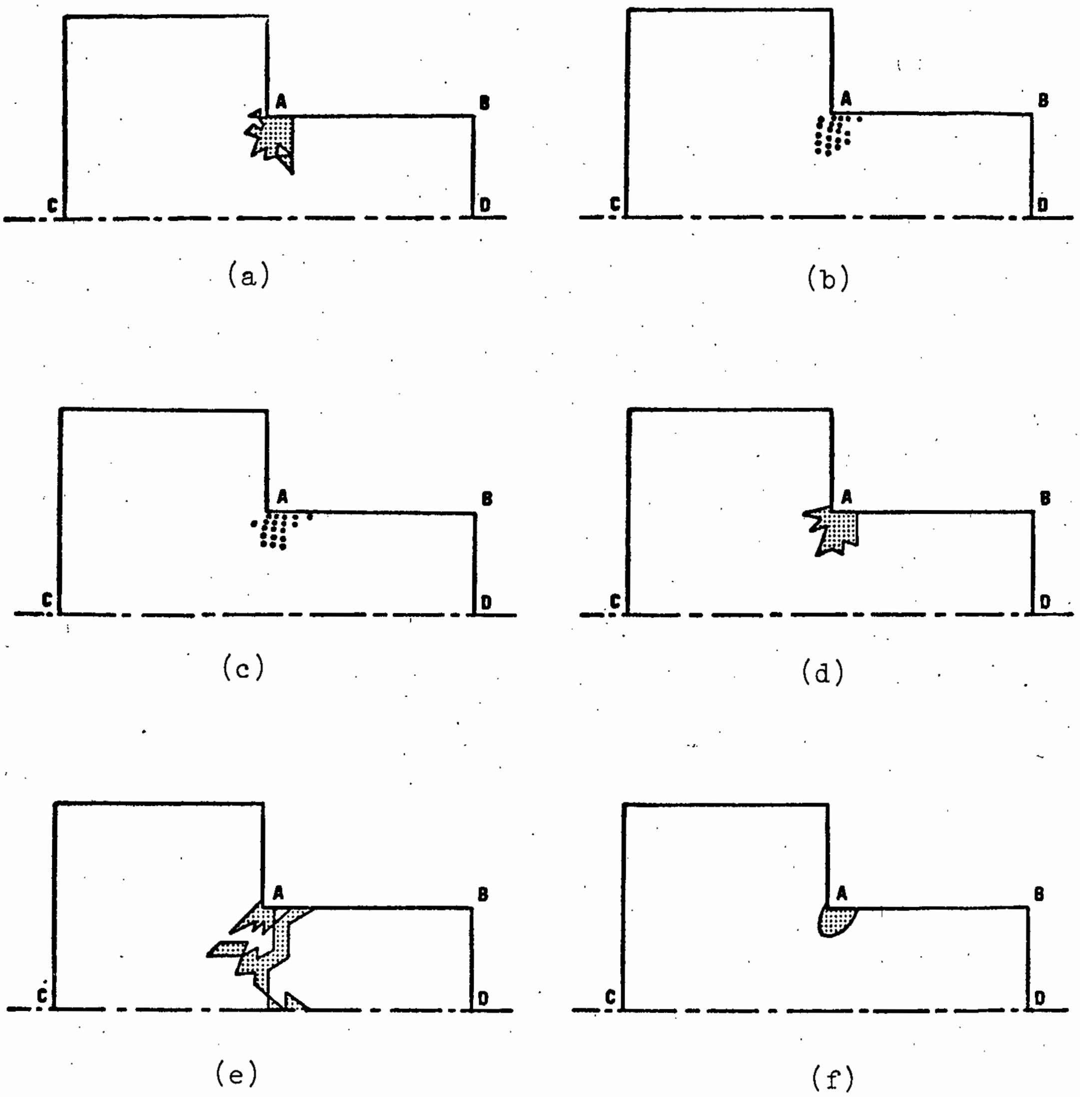


Figure 6.3.3 Plastic enclaves at M/M_o = 0,921

(a) Present analysis, constant strain

(b) Present analysis, isoparametric, 1λ/integration point

(c) Present analysis, isoparametric, 1λ/element

(d) Program NONSAP

(e) de Donato and Franchi

(f) Zienkiewicz and Valliapan, M/M_o = 0,929 program NONSAP developed by Bathe [28], which uses a tangent modulus approach for material nonlinearity. A mesh of 158 triangular constant strain elements and 97 nodes was also used. The extent of the plastic region is in good agreement with those shown in (a), (b) and (c). At the same load level plastic elements from de Donato and Franchi are indicated in (e). This significantly larger and irregular plastic region is not in agreement with other results. The plastic enclave of (f) was given by Zienkiewicz and Valliapan and corresponds to a slightly higher load $M/M_o = 0{,}929$.

The horizontal stress σ_x on the line AA' (see Figure 6.3.2) is plotted in dimensionless form in Figure 6.3.4 for the load $M/M_o = 0{,}921$. Stresses from the present analyses are in good agreement with those resulting from the program NONSAP, but do not agree with those of the de Donato and Franchi or Neal, especially in the region which is not undergoing plastic deformation. However Neal's stress distribution results from limit analysis in which stresses in the elastic region are not known - any safe and statically admissible stress field is permissible for the lower bound solution.

Figure 6.3.5 shows displacement in the y-direction along the centre line CD at the load $M/M_o = 0{,}921$. Again results of the present analyses are in good agreement with Bathe's program NONSAP, but do not coincide with de Donato and Franchi.

To illustrate elastic unloading after plastic deformation the hysteresis loops A and B in Figure 6.3.6 correspond to load cycles A and B shown diagrammatically in Figure 6.3.7. In both Figures the vertical axis is the applied load M/M_o. The horizontal axis in Figure 6.3.6 gives in dimensionless form vertical deflection at point D on the free end of the cantilever (see Figure 6.3.1). These results are from a constant strain finite element analysis using 158 elements and 97 nodes. The total

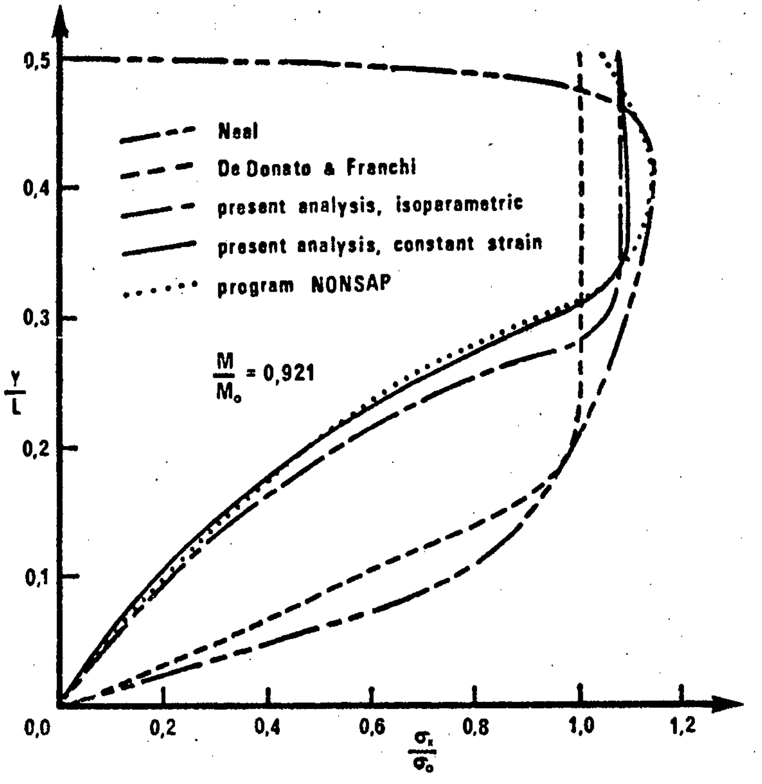


Figure 6.3.4 Horizontal stress on AA'

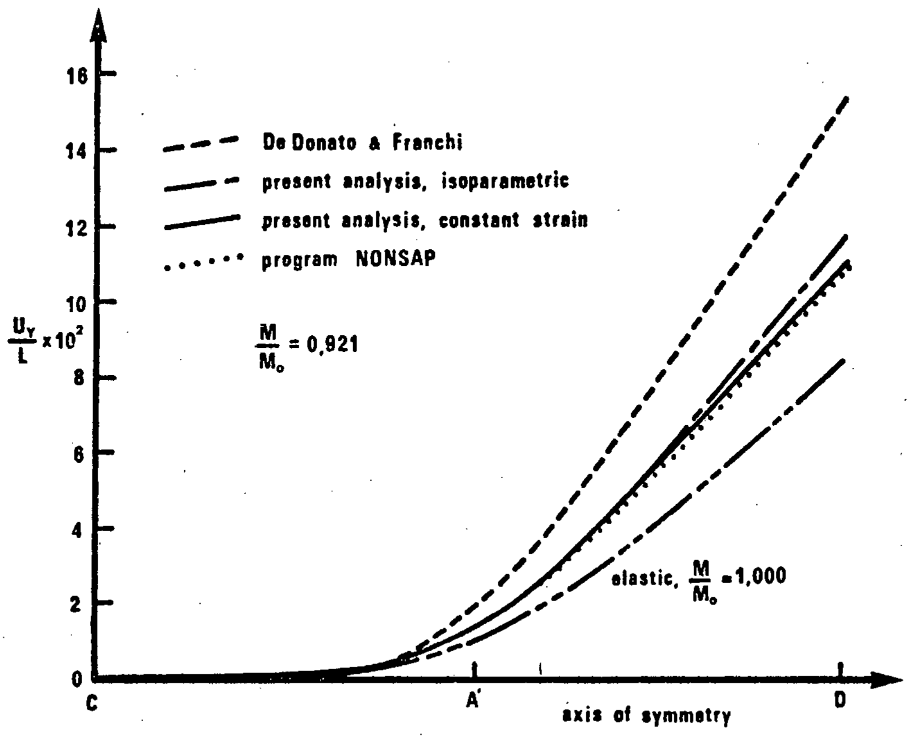


Figure 6.3.5 Displacement in y-direction along centre line CD; $M/M_o = 0{,}921$

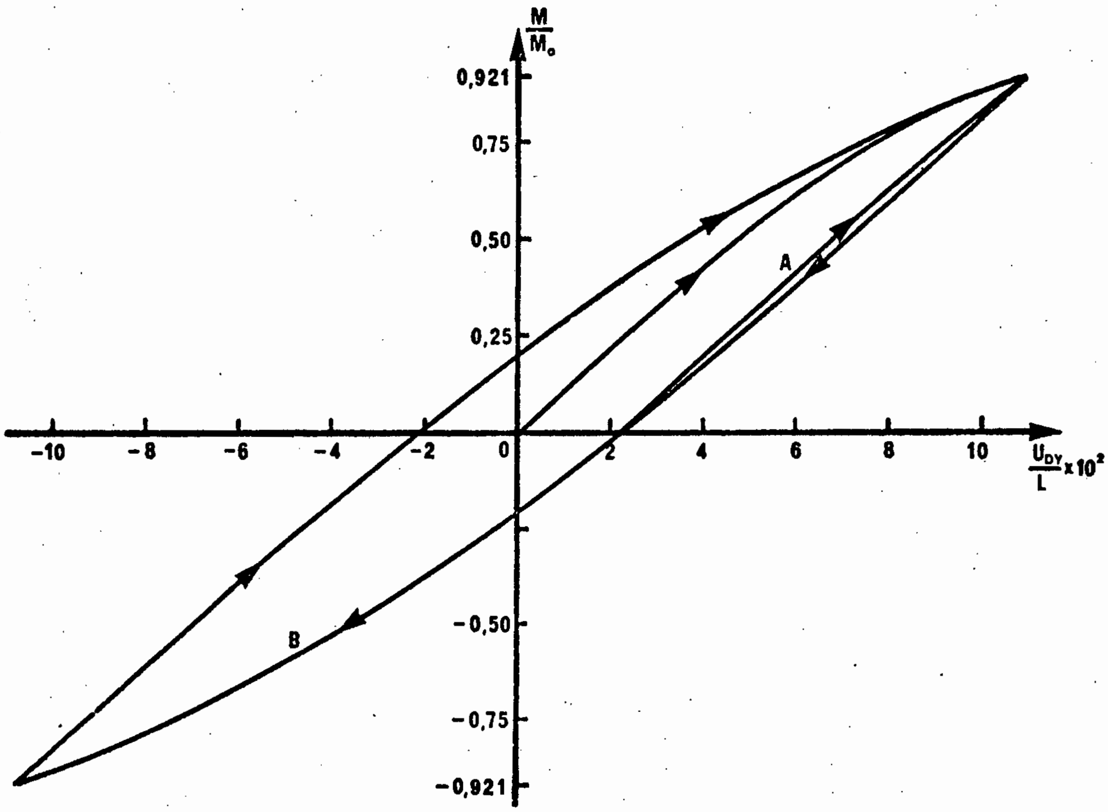


Figure 6.3.6 Load-deflection characteristics for cyclic loading

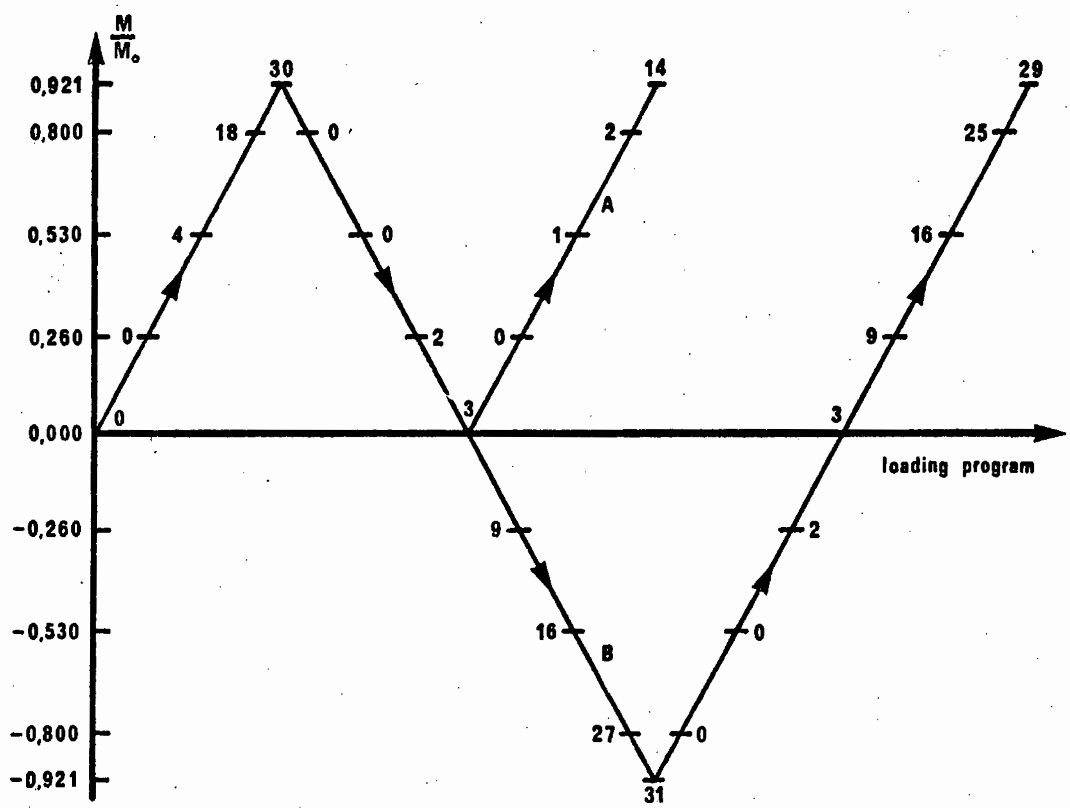


Figure 6.3.7 Cyclic loading programs A and B

numbers of elements deforming plastically at representative values of load are given in the diagram of Figure 6.3.7.

Load cycle A begins from the initial unstressed and undeformed state, incrementing the load until M/M_o = 0,921. At this stage thirty elements are plastic. The direction of loading is then reversed which causes all plastic elements to unload elastically. However, before the total load has become zero three elements again deform plastically. After reaching zero, load is then re-applied until M/M_o = 0,921. At this stage only fourteen elements become plastic again, while the vertical deflection at D is within 0,2% of its value at termination of the first load direction of cycle A.

For load cycle B the direction of loading is also reversed at M/M_o = 0,921, but is then maintained in this direction until M/M_o = -0,921. Load is then re-applied in the initial direction up to the value M/M_o = 0,921. Cycles between the two extreme load values show similar numbers of plastic elements at corresponding load levels of loading or unloading.

Figures 6.3.8 and 6.3.9 give results obtained for the cantilever from isoparametric finite element analyses either associating a plastic multiplier λ with each integration point, or associating one average multiplier with each element. From the load-deflection curve it is seen that using a single average multiplier for each element leads to an apparent increase in stiffness. Since even small regions undergoing large plastic strains can cause large displacements, decreasing these peak strain values by averaging the plastic multiplier over the element would cause the assemblage to exhibit stiffer load-deflection characteristics.

Horizontal stresses σ_x on the line AA' are plotted for two load levels in Figure 6.3.9 and indicate the disparity in stress distribution for the two analyses. Since the elastic part of any strain increment is evaluated as the difference between increments in plastic strain and

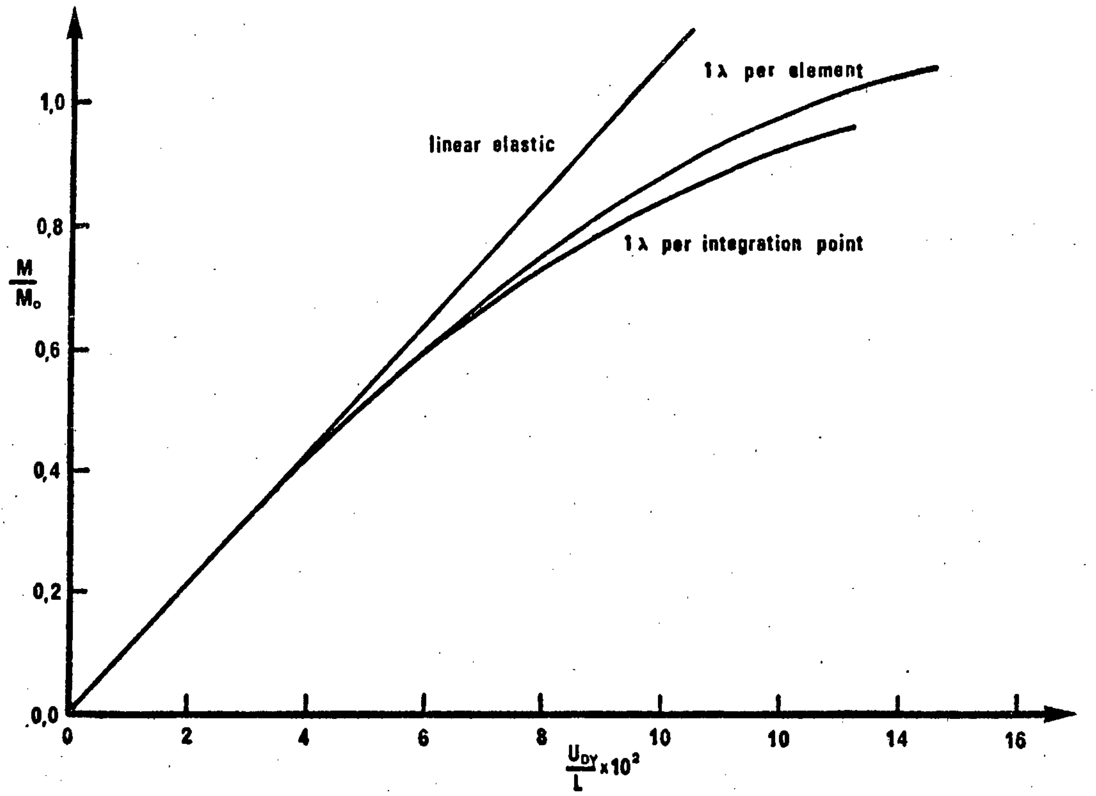


Figure 6.3.8 Load-deflection curves from isoparametric element analysis

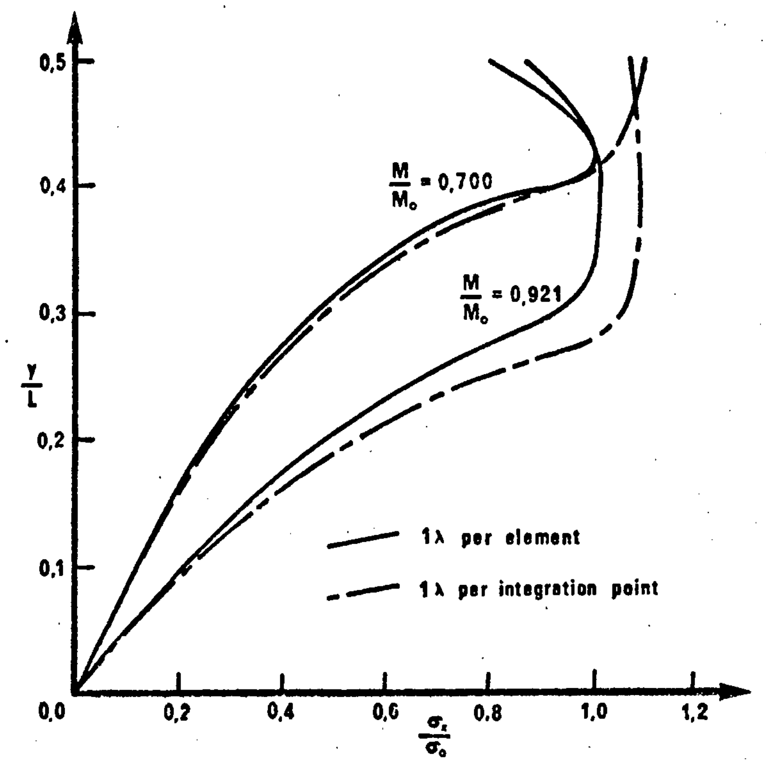


Figure 6.3.9 Horizontal stresses on AA'

total strain, any error in plastic strain due to averaging plastic multipliers within an element introduces errors in stress increments and hence in total stresses. However this approach does provide reasonable estimates of such quantities as the extent of plastic regions and limit load for the elastic-perfectly plastic case, while requiring much less computer storage and computation time.

Plastic regions indicated in Figure 6.3.10 correspond to the stages at which the present analyses were terminated when successive load increment magnitudes become less than 0,5% of the total load applied. Load levels indicated are thus estimates of the limit load according to the analysis performed. In this Figure (a), (b) and (c) correspond respecitvely to the analysis of Figure 6.3.3 (a), (b) and (c). Since

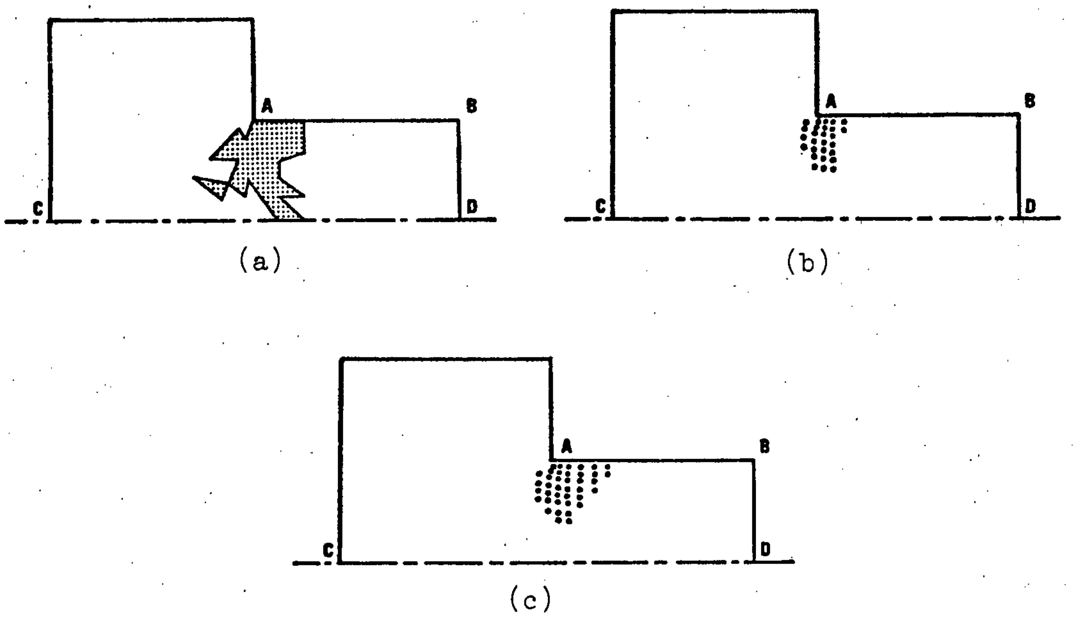


Figure 6.3.10 Plastic enclaves at termination of present analyses

(a) Constant strain elements, $M/M_o = 1,083$

(b) Isoparametric elements, 1λ/integration point, $M/M_o = 0,955$

(c) Isoparametric elements, 1λ/element, $M/M_o = 1,063$

Reference	Method	Element	Nodes	Elements	Load $\frac{M}{M_o}$	Stages	Computer	Time (min:sec)
de Donato & Franchi	Q.P.	Constant Strain Triangle	97	158	0,921	6	UNIVAC 1106	11:43
NONSAP †	Tangent Modulus					26		6:48
Present analysis						26		1:37
Present analysis	1 λ/int.pt.	Isopara-metric	192	31		18		2:28
Present analysis	1λ/element					17		1:38

† 8 node quadrilateral isoparametric degenerated to constant strain triangle

Table 6.3.1 Deep beam in plane stress

the present formulation is derived from a kinematic minimum principle, these estimates of ultimate load correspond to upper bounds on the limit load, while Neal's lower bound value is $M/M_o = 0{,}921$. The constant strain analysis required 5 min 58 sec CPU time, and consisted of 43 load increments. The isoparametric analyses consisted of 24 load increments for the case of one plastic multiplier per integration point, taking 3 min 45 sec CPU time; while using one average multiplier per element required 2 min 45 sec for a total of 29 load increments.

As noted previously categorical comparisons of computation times for the present analyses and published results cannot be made. However, the times listed in Table 6.3.1 are more meaningful than those given in Table 6.2.2 for the previous example, as in this case all analyses were performed to the same load ($M/M_o = 0{,}921$), and on similar computers. Constant strain triangular elements were not directly available in the program NONSAP, and therefore eight noded quadrilateral isoparametric elements were degenerated to form three noded constant strain triangles. In this way the same number of displacement degrees of freedom as the present constant strain element analysis was ensured.

It is evident from the Table that in this case the present approach is significantly more efficient in terms of computation performed.

6.4 Pressure Vessel - Flush Nozzle Junction

An experimental investigation into the elasto-plastic behaviour of flush nozzles in spherical pressure vessels has been reported by Dinno and Gill [35]. A numerical analysis of one of these specimens has been performed by Nayak and Zienkiewicz [36], using eight noded parabolic quadrilateral isoparametric finite elements. The geometry of the specimen considered is shown in Figure 6.4.1, while the mechanical properties of the

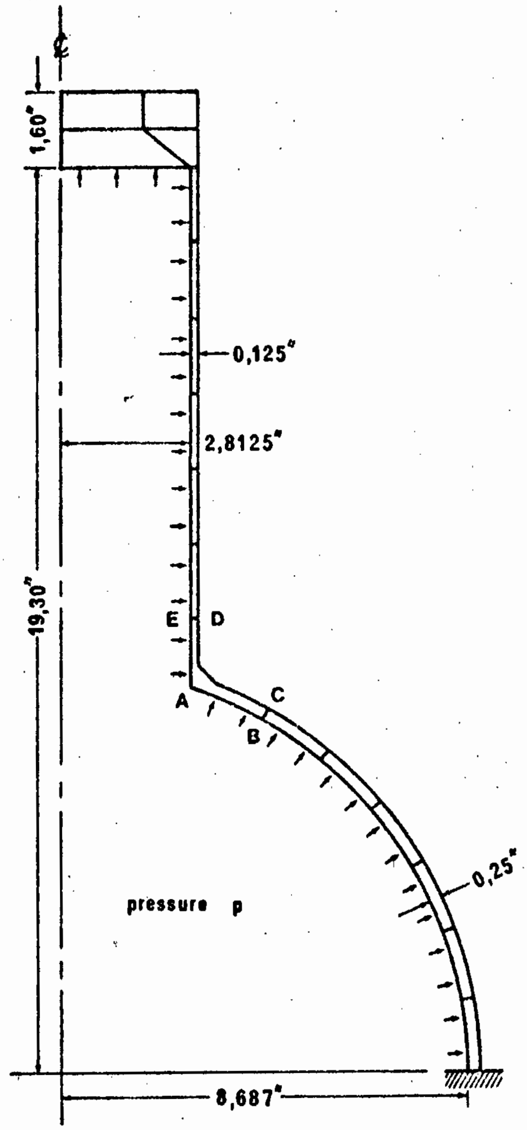


Figure 6.4.1 Spherical pressure vessel and cylindrical flush nozzle

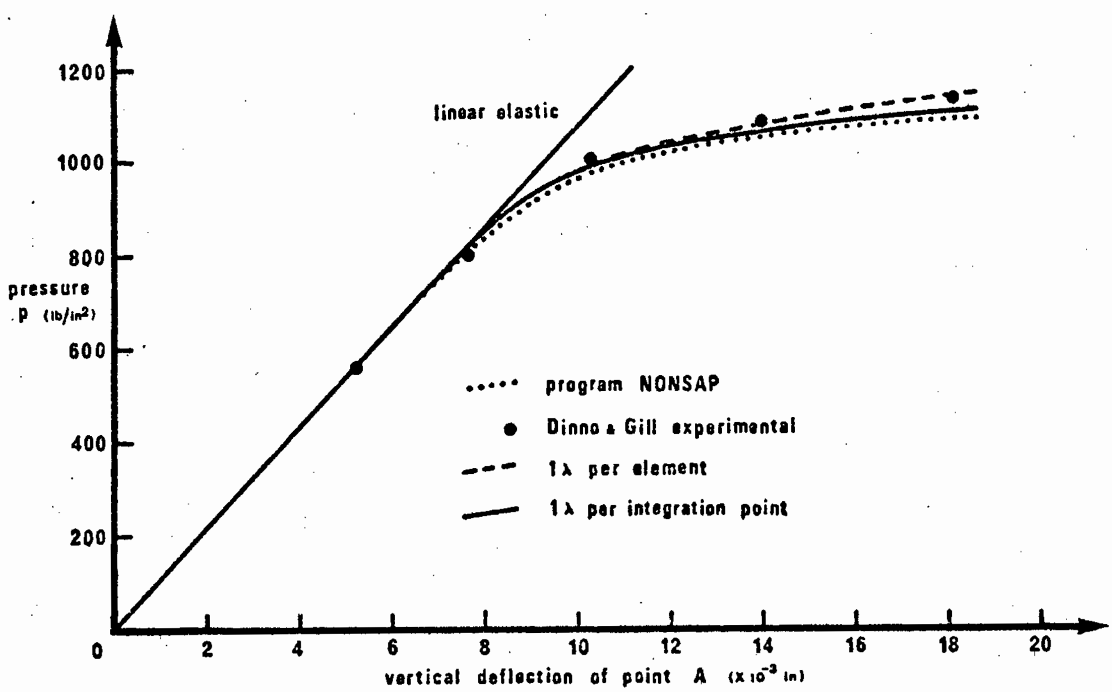


Figure 6.4.2 Pressure-displacement curve for pressure vessel - flush nozzle junction

material were given as $E = 29120000$ lb/in^2; $\nu = 0,3$ and $\sigma_o = 40540$ lb/in , with no strain hardening assumed. Loading of the axisymmetric specimen consisted of a uniform internal pressure.

Using the present method results were obtained for analyses of the vessel under increasing pressure using 26 cubic quadrilateral isoparametric elements and 209 nodes. This mesh is shown in Figure 6.4.1 with the enlargement of the weld region ABCDE shown in Figure 6.4.3(c). So as to compare results an anlysis was performed using the program NONSAP, [28]. Since this program used eight noded parabolic quadrilateral elements a mesh was used similar to that used by Nayak and Zienkiewicz, consisting of 53 elements and 225 nodes. The element discretization of the weld region is shown in Figures 6.4.3(a) and (b).

In Figure 6.4.2 vertical displacement of point A on the inside of the vessel is plotted against increasing internal pressure for various analyses. Excellent agreement between numerical and experimental results is obtained. Again use of one 'average' plastic multiplier for each element in the present method produces a stiffer analysis.

Under the loading considered plastic deformation is confined to the sphere-cylinder junction. The progression of plastic regions at increasing values of internal pressure is shown in the enlargements of the weld region in Figure 6.4.3. In this Figure (a) shows results of the analysis of Nayak and Zienkiewicz to a pressure of 1080 lb/in^2, while (b) was obtained using the program NONSAP. Plastic enclaves from the present method are shown in (c) in which the elastic-plastic interfaces corresponding to analyses using one plastic multiplier per integration point or one 'average' multiplier per element are only slightly different at high pressures. (The dashed lines at pressures 1000 and 1080 ln/in^2 correspond to the analysis using one multiplier per element). Possibly because this is essentially a thin shell problem good results are obtained using a coarse mesh of only

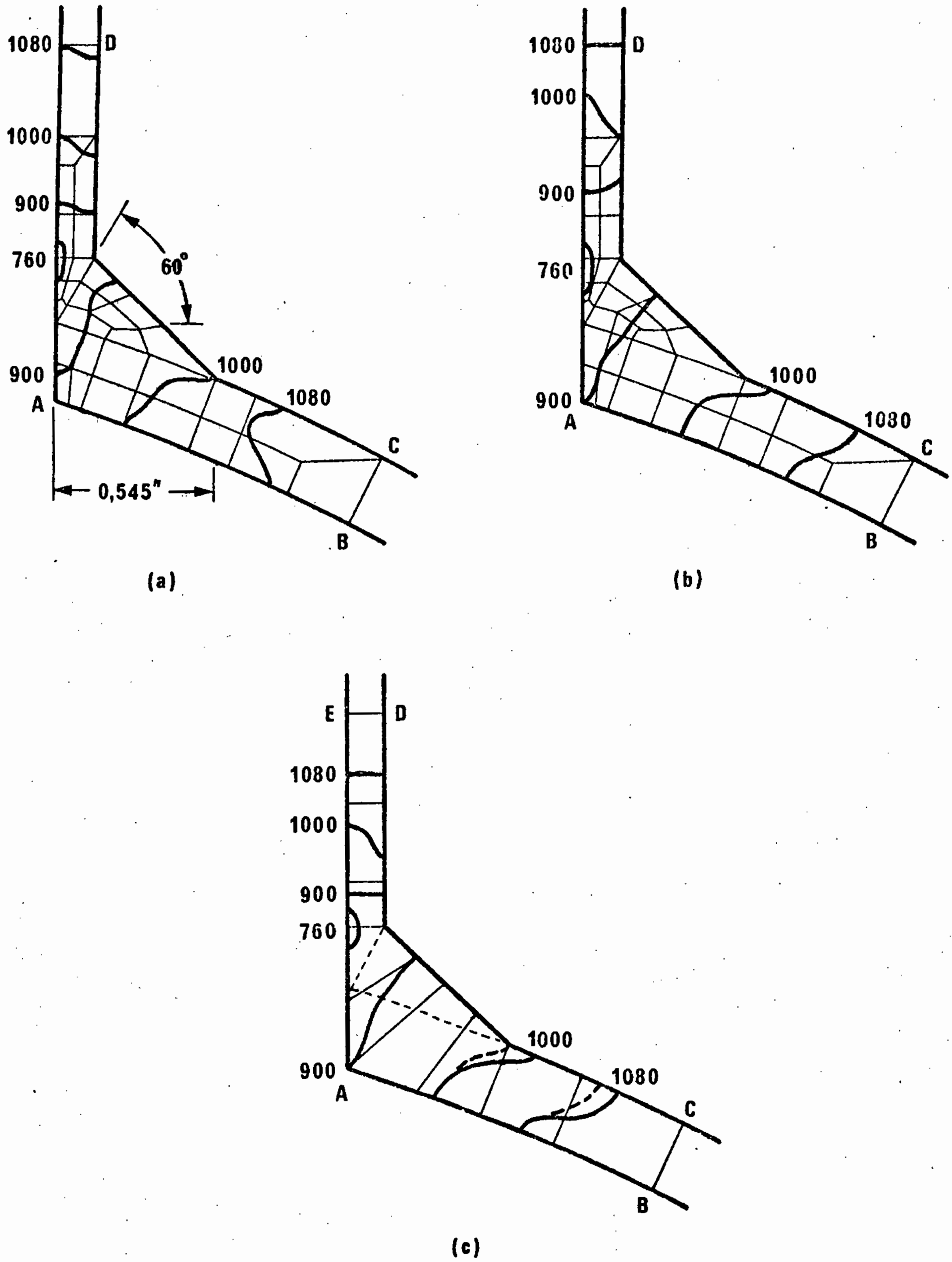


Figure 6.4.3 Plastic regions at values of internal pressure (lb/in^2)

(a) Nayak and Zienkiewicz

(b) Program NONSAP

(c) Present analysis

26 cubic quadrilateral elements. In particular use of one 'average' multiplier per element introduces negligible errors in this example - local stress concentrations as well as over-all behaviour are adequately described.

The analysis using NONSAP consisted of 30 load increments to a pressure of 1080 lb/in^2, and required 10 min 42 sec CPU time on a UNIVAC 1106 computer. The present analyses to this value of internal pressure required 40 load increments and 11 min 22 sec CPU time for the case of one multiplier per integration point; and 36 load increments and 3 min 24 sec CPU time using one multiplier per element. These analyses were also performed using a UNIVAC 1106 computer. Nayak and Zienkiewicz did not give computation time required for their analysis.

CHAPTER 7

DISCUSSION AND CONCLUSIONS

An efficient method of solution has been presented for the elasto-plastic analysis of continua in plane stress, plane strain or axial symmetry. Application of the extended minimum principle to three-dimensional continuum problems can be readily achieved by inclusion of field variable components ignored in the two-dimensional problem. The method appears attractive for three-dimensional analysis in view of the small increase in size of the system matrix in passing from the elastic to the elasto-plastic problem. In the formal quadratic programming approach the quadratic constraint on the value of the yield function is piecewise linearized, introducing a plastic multiplier and linear constraint for each yield plane of the assemblage, and thereby increasing the size of the problem manyfold.

Incremental plasticity requires determination of a complete solution for each load increment. In the conventional tangent modulus approach the entire system matrix is reformulated at each stage, while the initial strain/initial stress methods require iteration at every load increment. For the mathematical programming approach a linear complementary problem or quadratic programming problem must be solved at each stage. In the present method only part of the system matrix is reformulated for each increment. Further, the form of the system matrix is exploited to minimize computational effort, especially for load increments in which plastic deformations occur only in a small region of the assemblage.

The algorithm used to determine a solution for the present programming technique seldom requires iteration for monotonically increasing loading. In this case elastic unloading occurs infrequently and thus the best initial assumption is that all plastic regions remain plastic for the next load increment. For numerical load cycling problems investigated, reversal of the load (which causes elastic unloading of the whole structure) was found to require a single iteration before the constraints were satisfied. In general the present approach thus reduces to solving a set of simultaneous linear equations at each stage with possibly an occasional additional solution iteration. Although convergence of the algorithm has not been conclusively proved, it has been shown to be equivalent to the tangent modulus approach, which has wide acceptance. Failure to converge occurred only as the limit load was approached in the case of perfectly-plastic plane stress problems.

Exact comparisons of computational effort required by the present and other methods of analysis were not possible. But, from gross comparisons of computer times made for the numerical examples of the preceding chapter the order of magnitude of relative efficiency of the present method is clearly evident, while the order of accuracy is similar. From consideration of the form of the system matrix it is noted that problems for which the present method is especially suited are those in which plastic deformation is confined to small regions. This class of problem covers those in which stress concentrations occur, for example through abrupt change in geometry or through intersection of sections. In these problems of localized plastic deformation advantage is taken of the proposed solution method in which the system matrix is partitioned and the elastic part inverted once only.

In the tangent modulus approach, if one has a priori knowledge of the region in which plastic deformation takes place, and further if it is possible to label finite element node numbers in such a way that this region is represented near the bottom of the banded system matrix, then a saving in computation can be achieved as the system matrix can be partially inverted once only. For each load increment change in the matrix to account for plastic elements is confined to the bottom of the band. The inversion is completed, and displacement increments evaluated by back substitution. However, labelling the nodes of a structure in this way without adversely affecting the band width of the elastic system matrix is usually not possible. The feasibility of this scheme is therefore entirely problem dependent.

Anand and Shaw [30] compared the use of constant strain and linear strain triangular finite elements in elasto-plastic solutions, using a tangent modulus approach. Contrary to their conclusion the results of the present method indicate that use of higher order elements is generally more efficient in terms of accuracy and computational expenditure. The use of one 'average' plastic multiplier associated with all plastic strains within an isoparametric element gives reasonable results while offering vast savings in computation and storage requirements. Further areas of research include the possibility of improving the accuracy of this latter technique by redistributing the 'average' multiplier according to some scheme, thereby improving the approximate variation in plastic strains across an element.

REFERENCES

1. PRAGER, W., 'Fundamental theorems of a new mathematical theory of plasticity, Duke Math. Jnl., 9, 1942, 228.

2. PRAGER, W., 'Variational principles in the theory of plasticity', Proc. 6th Int. Congr. Appl. Mech., Paris, 1946.

3. HODGE, P.G. and PRAGER, W., 'A variational principle for plastic materials with strain hardening', J. Math. & Phys., 27, 1948, 1.

4. GREENBERG, H.J., 'On the variational principles of plasticity', Tech. Report A11-54, Div. of Appl. Math., Brown Univ., 1949.

5. GREENBERG, H.J., 'Complementary minimum principles for an elasto-plastic material', Quart. Appl. Math., 7, 1949, 85.

6. KOITER, W.T., 'Stress-strain relations, uniqueness and variational theorems for elasto-plastic materials with a singular yield surface', Quart. Appl. Math., 11, 1953, 350.

7. CERADINI, G., 'A maximum principle for the analysis of elastic-plastic systems', Meccanica, 1, 1966, 77.

8. MAIER, G., 'Some theorems for plastic strain rates and plastic strains', J. de Mécanique, 8, 1969. 5.

9. GALLAGHER, R.H., PADLOG, J. and BIJLAARD, P.P., 'Stress analysis of heated complex shapes', Jnl. Amer. Rocket Soc., 32, 1962, 700.

10. ANGYRIS, J.H., 'Elasto-plastic displacement analysis of three dimensional continua', Jnl. Royal Aeronaut. Soc., 69, 1965, 633.

11. ZIENKIEWICZ, O.C., VALLIAPAN, S. and KING, I.P., 'Elasto plastic solutions of engineering problems initial stress, finite element approach', Int. J. Num. Meth. Eng., 1, 1969, 75.

12. POPE, G.G., 'A discrete element method for the analysis of plane elasto-plastic stress problems', The Aeronautics Quart., 17, 1, 1966, 83.

13. SWEDLOW, J.L., 'Elastic-plastic cracked plates in plane strain', Int. Jnl. Fracture Mech., 5, 1, 1969, 33.

14. MARCAL, P.V. and KING, I.P., 'Elastic-plastic analysis of two-dimensional stress systems by the finite element method', Int. J. Mech. Sci., 9, 3, 1967, 143.

15. YAMADA, Y. YOSHIMURA, N. and SAKURAI, T., 'Plastic stress-strain matrix and its application for the solution of elastic-plastic problems by the finite element method', Int. J. Mech. Sci., 10, 1968, 343.

16. MARCAL, P.V., 'A comparative study of numerical method of elastic-plastic analysis', AIAA Jnl., 6, 1968, 157.

17. HODGE, P.G., BELYTSCHKO, T. and HERAKOVICH, C., 'Plasticity and quadratic programming', Proc. Computational Approaches in Appl. Mech., (ASME), 73, 1969.

18. SAYEGH, A.F. and RUBENSTEIN, M., 'Elastic-plastic analysis by quadratic programming', Proc. Eng. Mech. Div. ASCE., 98, (EM6), 1972, 1547.

19. GIACOMINI, S., MAIER, G. and PATERLINI, F., 'Complete solutions of elastic-plastic discretized structures by linear programming', presented at SMIRT 4, San Francisco, 1977.

20. DE DONATO, O. and MAIER, G., 'Finite element elastoplastic analysis by quadratic programming: the multistage method', presented at 2nd Int. Conf. Struct. Mech. in Reactor Technology, Berlin, 1973.

21. ANAND, S.C. and GARG, V.K., 'Shakedown of rolling wheel under hub loading', Jnl. Eng. Mech. Div., 100, (EM6), 1974, 1237.

22. DE DONATO, O. and FRANCHI, A., 'A modified gradient method for finite element elastoplastic analysis by quadratic programming', Comp. Meth. Appl. Mech. & Eng., 2, 1973, 107.

23. MARTIN, J.B., 'On the kinematic minimum principle for the rate problem in classical plasticity', J. Mech. Phys. Solids, 23, 1975, 123.

24. MARTIN, J.B. and REDDY, B.D., 'A programming approach to the solution of the rate problem in elastic plastic solids', Mechanics in Engineering, ed. R.N. Dubey and N.C. Lind, Univ. of Waterloo Press, Canada, 1977.

25. MARTIN, J.B., 'Plasticity', MIT Press, 1975.

26. VON MISES, R., 'Mechanik der festen Körper in plastisch deformablen Zustand', Göttinger Nachr. Math. Phys. Kl., 1913, 582.

27. TIMOSHENKO, S. and GOODIER, J.N., 'Theory of Elasticity', McGraw Hill, 2nd edition, 1951.

28. BATHE, K.J., WILSON, E.L. and IDLING, R.H., 'NONSAP a strucutral analysis program for static and dynamic response of nonlinear systems', Report no. UC SESM 74-3, Struc. Eng. Lab., Univ. Cal., Berkeley, 1974.

29. BATHE, K.J. and WILSON, L., 'Numerical Methods in Finite Element Analysis', Prentice Hall, 1976.

30. ANAND, S.C. and SHAW, R.H., 'Use of linear strain triangular elements in elasto-plastic solutions', to be published in Comp. Meth. Appl. Mech. and Eng.

31. THEOKARIS, P.A. and MARKETOS, E., 'Elastic-plastic strain and stress distribution in notched plates under plane stress', J. Mech. Phys. Solids, 11, 1963, 411.

32. ZIENKIEWICZ, O.C. and VALLIAPAN, S., 'Analysis of real structures for creep, plasticity and other complex constitutive laws', Strucutre, Solid Mechanics and Engineering Design, ed. M. Te'eni, Wiley Interscience, 1971, 27.

33. NEAL, B.G., 'Limit load of a cantilever in plane stress', in Engineering Plasticity, Cambridge Univ. Press, 1968, 473.

34. ZIENKIEWICZ, O.C., 'The Finite Element Method in Engineering Science', McGraw Hill, 1971.

35. DINNO, K.S. and GILL, S.S., 'An experimental investigation into the plastic behaviour of flush nozzles in spherical pressure vessels', Int. J. Mech. Sci., 7, 1965, 817.

36. NAYAK, G.C. and ZIENKIEWICZ, O.C., 'Elasto-plastic stress analysis. A generalization for various constitutive relations including strain softening', Int. Jnl. Num. Meth. Eng., 5, 1972, 113.

37. DITTMER, C.T. and MARTIN, J.B., 'A programming approach to the analysis of elasto-plastic continua by means of an extended kinematic minimum principle', presented at NATO-ASI on Plasticity and Math. Programming, Univ. Waterloo, Canada, 1977.

APPENDIX: COMPUTER PROGRAMS AND USER MANUALS

A.1. General Description

The computer programs EPTCS (Elasto-Plastic analysis using Triangular Constant Strain finite elements) and EPCQI (Elasto-Plastic analysis using Cubic Quadrilateral Isoparametric finite elements) may be used for the incremental elasto-plastic analysis of continua in plane stress plane strain or axial symmetry. The material of the body must be either elastic-perfectly plastic or linear kinematic hardening, and is assumed to obey the von Mises yield condition. Loading of the assemblage consists of a series of piecewise linear proportional load paths. The magnitudes of individual load increments are determined internally during execution of the program.

The programs are written in FORTRAN V as implemented on the UNIVAC 1106 computer. This computer has a thirty-six bit word which gives eight significant figures for arithmetic performed in single precision. Moreover, the set of simultaneous linear equations to be solved for each load increment is well conditioned (unless the limit load is approached in the elastic-perfectly plastic case). Thus, excluding special algebraic operations such as the square root function, all arithmetic in the programs EPTCS and EPCQI is performed in single precision without loss of accuracy.

The overall logic of each of the programs EPTCS and EPCQI is similar, but computational details differ. Each program consists of independent external subroutines linked by a main program. The function of each of the subroutines and the sequence in which they are called from the main program, is described in broad outline by means of the macro-flowchart in Figure A.1. Complete listings of EPTCS and EPCQI are given in sections (A.5) and (A.6) respectively.

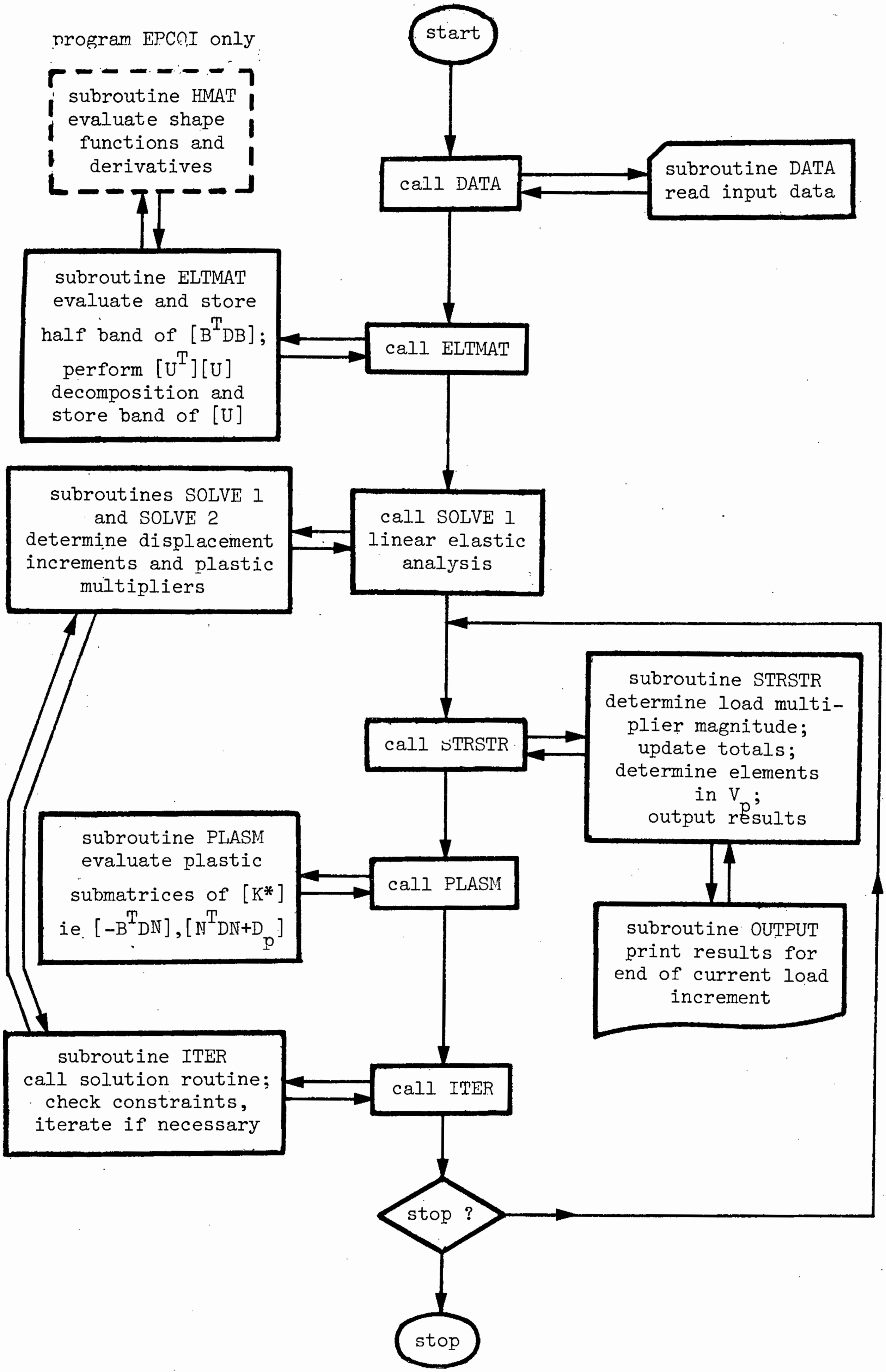


Figure A.1 Macro-flowchart for EPTCS and EPCQI

A.2. Storage Allocation, Dimensioning of Arrays, and Execution

For any realistic problem the programs require relatively large amounts of core storage. Therefore it is expedient to adjust dimensions of arrays to suit each particular analysis problem, thereby minimizing storage requirements.

To implement this dynamic dimensioning of arrays is employed where possible. Values of dynamic parameters are defined in the MAIN program, lines 5 to 11 for program EPTCS, or lines 5 to 12 for program EPCQI; (see program listings sections A.5 and A.6). The parameters have the following representation:

NDFP maximum number of displacement degrees of freedom (i.e. maximum number of nodes multiplied by two)

NECP maximum half-band width of elastic system stiffness matrix $[B^TDB]$

NPCP maximum number of plastic multipliers to be determined for any load increment

NNP maximum number of nodes

NEP maximum number of elements

ISP maximum number of stress components considered at a point in the body, i.e. ISP = 3 for plane stress, ISP = 4 for plane strain or axial symmetry

NLPP maximum number of piecewise linear proportional load paths, plus one

IOP (for program EPCQI only) maximum integration order for Gauss quadrature; maximum value 4, recommended value 3

For each analysis problem the values of these parameters should be adjusted to ensure sufficient storage allocation. During program execution dimension limits of arrays and values of parameters are checked to ensure they are compatible with the data input. If not a relevant error message is printed and the analysis terminated.

On the UNIVAC 1106 the collector limits direct core storage access to 64K words. However, with extended storage facilities approximately 130K words can be accessed in core. For this reason the two largest submatrices $[B^T DB]$ and $[-B^T DN]$ are assigned to external core storage in arrays named RKE and RKP respectively. This is achieved by the inclusion of a FORTRAN V statement COMPILER (XM = 3) at the head of each routine in which the arrays RKE and RKP appear. Further, these two arrays must be dimensioned explicitly (parameters are not permitted) in the first labelled common block EXT. Arrays RKE and RKP appear in subroutines MAIN, ELTMAT, SOLVE1, SOLVE2,STRSTR and ITER. When altering the dimension limits of the programs each of these elements must be edited and the explicit dimensions given for RKE and RKP, each a two-dimensional array. The dimension limits for RKE must be the same as the values of parameters (NDFP, NECP); while those for RKP must be the same as the values of parameters (NDFP,NPCP). (see program listings, sections A.5 and A.6).

After FORTRAN compilation of the altered symbolic elements, all relocatable elements are collected into an absolute element ABS, ensuring that the external labelled common block EXT is assigned at the top of the data bank. The programs are collected as follows:

```
@MAP,IE EPTCS.ABS
IN EPTCS.MAIN
IN EPTCS.DATA
IN EPTCS.ELTMAT
IN EPTCS.SOLVE1
IN EPTCS.SOLVE2
IN EPTCS.STRSTR
IN EPTCS.PLASM
IN EPTCS.ITER
IN EPTCS.OUTPUT
IN EXT
```

```
@MAP,IE EPCQI.ABS
IN EPCQI.MAIN
IN EPCQI.DATA
IN EPCQI.ELTMAT
IN EPCQI.HMAT
IN EPCQI.SOLVE1
IN EPCQI.SOLVE2
IN EPCQI.STRSTR
IN EPCQI.PLASM
IN EPCQI.ITER
IN EPCQI.OUTPUT
IN EXT
```

Peripheral drum storage files are used to store the half band of the elastic system matrix $[B^T DB]$, the band of the triangular decomposed matrix [U], the plastic system matrix $[-B^T DN]$, and for the isoparametric case the element deformation matrices [B']. The logical-unit numbers associated with these files are 11, 12, 13 for the program EPTCS and 11, 12,13,14 for the program EPCQI. (These logical-unit numbers are specified in the MAIN programs).

Thus typical runstreams for execution of the programs on a UNIVAC machine are as follows:

```
@RUN
@ASG,A EPTCS.
@ASG,T 11
@ASG,T 12
@ASG,T 13
@XQT EPTCS.ABS

        data

@FIN
```

```
@RUN
@ASG,A EPQCI.
@ASG,T 11
@ASG,T 12
@ASG,T 13
@ASG,T 14
@XQT EPQCI.ABS

        data

@FIN
```

Data input is in free format, separated by blanks or commas. The programs are independent of the units of the input data, and thus units of the output will be consistent with those of input data.

Turn to page A-23 for EPCQI

A.3. Program EPTCS

The program EPTCS does not have an automatic triangular finite element mesh-generation capability, and thus nodal coordinates as well as element incidences are given explicitly as data. Data input required for execution may be divided into the seven subgroups listed in the following section (A.3.1), and described in section (A.3.2). These sections should be read in conjunction with the example of data input for EPTCS listed in section (A.3.3).

A.3.1 Data Input for EPTCS

HEADING1

HEADING2

i) heading statements

TYPE

NN,NE,E,EP,RNU,SZERO,ETA,(THIK),FRAC

ii) structure statements

X_1,Y_1,X_2,Y_2. . . . .

.

.

. . . . . X_{NN},Y_{NN}

iii) nodal coordinates

$IA_{1,1},IA_{1,2},IA_{1,3}$ $IA_{2,1},IA_{2,2}$

.

.

. . . . . $IA_{NE,1},IA_{NE,2},IA_{NE,3}$

iv) element incidences

IX,IY

$N_a,N_b,N_c,\ldots,N_k,-1$

. . . . .

. . . . .

.

.

-1

v) boundary conditions

NLP,N,I_{xy},SIZE

N_i,N_i

P_x,P_y

. . . . .

. . . . .

.

.

-1

continued

```
. . . . . . . . . .   ]                     | vi) loading program
. . . . . ]                                  |
. . . . . ]                                  |
. . . . . ]                                  |
. . . . . ]                                  |
  .                                          |
  .                                          |
  .                                          |
-1                                           |
-1                    ]                      ]
```

PRINT

$IE_a, IE_b, IE_c, \ldots, IE_k, -1$

$N_a, N_b, N_c, \ldots, N_k, -1$

$IOUT_a, IOUT_b, IOUT_c, \ldots, IOUT_k, -2$

vii) output requested

A.3.2 Description of Data Input

(i) Heading Statements

HEADING1

HEADING2

where

HEADING1	each a string of alphanumeric characters of maximum length
HEADING2	eighty. Can be used to identify problem, record units employed etc. Printed at top of output.

(ii) Structure Statements

TYPE

NN,NE,E,EP,RNU,SZERO,ETA,(THIK),FRAC

where

TYPE (alpha)	either PLANE STRESS, PLANE STRAIN OR AXISYMMETRIC
NN (integer)	total number of nodes

NE (integer)	total number of elements
E (real)	Young's modulus
EP (real)	plastic modulus for linear kinematic hardening; for elastic-perfectly plastic case enter 0.0
RNU (real)	Poisson's ratio
SZERO (real)	uniaxial yield stress
ETA (real)	η, ratio of von Mises equivalent stress to yield stress for an elastic stress point to be treated as plastic. Recommended value 0.99 (see section 5.7)
THIK (real)	plate thickness if plane stress analysis. Omit for plane strain or axisymmetric analysis
FRAC (real)	for determining the limit load in the elastic-perfectly plastic case. The analysis is terminated if for three successive load increments the ratio of load increment magnitude to total load magnitude is less than the value of FRAC (typically, 0.005). Omit if material is strain hardening.

(iii) Nodal Coordinates

$X_1, Y_1, X_2, Y_2 \; . \; . \; . \; .$
.
.
$. \; . \; . \; . \; . \; X_{NN}, Y_{NN}$

where

X_i, Y_i (reals) are global Cartesian coordinates of ith node

(iv) Element Incidences

$IA_{1,1}, IA_{1,2}, IA_{1,3} \; IA_{2,1}, IA_{2,2} \; . \; . \; . \; .$
.
.
$. \; . \; . \; . \; . \; IA_{NE,1}, IA_{NE,2}, IA_{NE,3}$

where

$IA_{i,1}, IA_{i,2}, IA_{i,3}$ (integers) are node numbers of apices of ith element in counter-clockwise order.

(v) Boundary Conditions

IX,IY] (*)

$N_a, N_b, N_c, \ldots, N_k$,-1]

IX,IY]

. . . . -1]

.

.

-1

where

IX,IY (integers) are boundary conditions:

IX = 0 if constrained in global x-direction

IX = 1 if unconstrained in global x-direction

IY = 0 if constrained in global y-direction

IY = 1 if unconstrained in global y-direction

(Note in axisymmetric case radial direction coincides with global x-direction).

$N_a, N_b, N_c, \ldots, N_k$ (integers) are node numbers of nodes which have boundary conditions IX,IY of the previous line. A -1 after N_k indicates the end of the line. A data subgroup such as (*) above corresponds to each set of boundary conditions IX,IY. Only one line of data $N_a, N_b, N_c, \ldots N_k$,-1 corresponds to the boundary conditions IX,IY immediately preceeding it, and must be terminated by -1. If there are too many nodes for one line of data these must be broken into two or more subgroups (*), each having the same boundary conditions IX,IY. Totally unconstrained nodes need not appear in the boundary conditions. A line containing only -1 indicates the end of the boundary conditions.

(vi) Loading Program

```
NLP,N,Ixy,SIZE     ]
Ni,Ni    ]         |
Px,Py    ] (*)     |
. . . .  ]         |  (**)
. . . .  ]         |
.                  |
.                  |
.                  |
-1                 ]
```

where

NLP (integer) — sequential number of the proportional load path (**)

N (integer) — node number at which load is to be checked to indicate end of load path NLP

I_{xy} (integer) — coordinate direction in which load at node N is to be checked: I_{xy} = 1 for load check in x-direction; I_{xy} = 2 for load check in y-direction

SIZE (real) — load at node N (in coordinate direction I_{xy}), at which load path NLP is to be terminated

N_i,N_i (integer) — is the number of node, repeated, which carries external loads P_x,P_y applied in the global x,y directions.

Nodes which have no external loads need not appear in the loading program. A nodal load subgroup such as (*) above corresponds to each node which has any external loading for load path NLP. Should consecutive node numbers $N_i,N_{i+1},\ldots,N_k$ have the same loads P_x,P_y these can be input as

$$\left.\begin{matrix} N_i,N_k \\ P_x,P_y \end{matrix}\right] .$$

A line containing only -1 indicates the end of the load path NLP. A loading data subgroup (**) must be entered for each segment of the piecewise linear proportional load path. The end of all load path subgroups is indicated by a line containing only -1.

(vii) Output Requested

PRINT
$IE_a, IE_b, IE_c, \ldots, IE_k, -1$
$N_a, N_b, N_c, \ldots, N_k, -1$
$IOUT_a, IOUT_b, IOUT_c, \ldots, IOUT_k, -2$

where

PRINT	if results are to be given at every node and for all elements, enter ALL. If results are desired only at certain nodes and elements, enter SOME. If ALL is used the two lines of data following are omitted. If SOME is used:
IE_i (integer)	are elements for which results are to be printed
N_i (integer)	are nodes for which results are to be printed. Only one line of each is permitted, and the end of each line indicated by -1.
$IOUT_i$ (integer)	are the numbers of load increments for which results are to be printed. If $IOUT_k = -1$ the analysis will be terminated after $IOUT_{k-1}$ load increments. The end of the line is indicated by -2. In addition results are printed after termination of each proportional load path.

A.3.3 Example of Data and Results for EPTCS

The pages immediately following list data and results corresponding to a triangular constant strain finite element analysis of the deep cantilever in plane stress described in section 6.3. The element mesh consisting of 97 nodes and 158 elements is shown in Figure 6.3.2. The result of this analysis is listed in Table 6.3.1.

```
COL*FPTCS(1).DR
```

(i) Heading statements

```
DEEP BEAM
UNITS:  MM   KG
```

(ii) Structure statements

```
PLANE STRESS
97,158,1000.,0.0,0.0,32.,0.995,1.,0.005
```

(iii) Nodal coordinates

```
320.,80. 320.,40. 320.,0. 280.,80. 280.,65. 280.,40. 280.,0.
250.,80. 250.,65. 250.,45. 250.,25. 250.,0.
220.,80. 220.,65. 220.,45. 220.,25. 220.,0.
200.,80. 200.,70. 200.,55. 200.,40. 200.,20. 200.,0.
180.,80. 180.,70. 180.,60. 180.,50. 180.,35. 180.,20. 180.,0.
170.,80. 169.6875,75. 169.375,70. 168.75,60. 168.125,50.
167.1875,35. 166.25,20. 165.,0. 165.,80. 164.0625,75.
163.125,70. 162.1875,65. 160.78125,57.5 158.90625,47.5
156.5625,35. 153.75,20. 150.,0.
160.,80. 158.125,75. 156.25,70. 154.375,65. 151.5625,57.5
148.75,50. 145.,40. 141.25,30. 136.5625,17.5 130.,0.
160.,85. 155.294,80. 150.588,75. 143.529,67.5
136.47,60. 127.059,50. 112.941,35. 96.47,17.5 80.,0.
160.,90. 155.,87.1875 145.,81.5625 130.,73.125
110.,61.875 85.,47.8125 55.,30.9375 0.,0. 160.,100.
150.,97.5 135.,93.75 115.,88.75 85.,81.25 45.,71.25
0.,60. 160.,120. 140.,117.5 120.,115. 90.,111.25 50.,106.25
0.,100. 160.,140. 140.,138.75 110.,136.875 60.,137.75 0.,130.
160.,160. 140.,160. 110.,160. 60.,160. 0.,160.
```

(iv) Element incidences

```
5,1,4 2,1,5 6,2,5 3,2,6 7,3,6 9,4,8 9,5,4 10,5,9 6,5,10
11,6,10 7,6,11 12,7,11 9,8,13 14,9,13 10,9,14
15,10,14 11,10,15 16,11,15 12,11,16 17,12,16 19,13,18
14,13,19 20,14,19 15,14,20 21,15,20 16,15,21 22,16,21
17,16,22 23,17,22 25,18,24 25,19,18 26,19,25 20,19,26
27,20,26 21,20,27 28,21,27 22,21,28 29,22,28 23,22,29
30,23,29 32,24,31 25,24,32 33,25,32 26,25,33 34,26,33
27,26,34 35,27,34 28,27,35 36,28,35 29,28,36 37,29,36
30,29,37 38,30,37 40,31,39 40,32,31 41,32,40 41,33,32
42,33,41 34,33,42 43,34,42 35,34,43 44,35,43 36,35,44
45,36,44 37,36,45 46,37,45 38,37,46 47,38,46 49,39,48
49,40,39 50,40,49 50,41,40 51,41,50 51,42,41 52,42,51
52,43,42 53,43,52 53,44,43 54,44,53 45,44,54 55,45,54
46,45,55 56,46,55 47,46,56 57,47,56 59,48,58
49,48,59 60,49,59 50,49,60 51,50,60 51,60,61 52,51,61
52,61,62 53,52,62 63,53,62 54,53,63 55,54,63 55,63,64
56,55,64 65,56,64 57,56,65 66,57,65 58,67,68 59,58,68
59,68,69 60,59,69 61,60,69 61,69,70 62,61,70 71,62,70
63,62,71 64,63,71 64,71,72 65,64,72 65,72,73 66,65,73
74,66,73 67,75,76 68,67,76 69,68,76 69,76,77 70,69,77
70,77,78 71,70,78 71,78,79 72,71,79 72,79,80 73,72,80
73,80,81 74,73,81 76,75,82 76,82,83 77,76,83 77,83,84
78,77,84 78,84,85 79,78,85 79,85,86 80,79,86 80,86,87
81,80,87 83,82,88 83,88,89 84,83,89 84,89,90 85,84,90
85,90,91 86,85,91 86,91,92 87,86,92 88,93,94 89,88,94
89,94,95 90,89,95 90,95,96 91,90,96 91,96,97 92,91,97
```

(v) Boundary conditions

```
0,1
3,7,12,17,25,30,38,47,57,66 -1
0,0
74,81,87,92,93,94,95,96,97 -1
-1
```

(vi) Loading program

```
1,3,2,211.83
1,1
0.,7.
2,2
0.,34.
```

```
3,3
0.,23.
-1
-1
SOME
62,64,65,66,67,68,69,71,72,73,74,76,78,86 -1
1,2,3,7,12,17,23,30,38,47,57,66,74 -1 .
1 -2
```

(vii) Output requested (SOME through 1 -2)

```
@XQT EPTCS.ABS

@ADD,P EPTCS.DB
```

DEEP BEAM

NUMERICAL ANALYSIS USING TRIANGULAR CONSTANT STRAIN FINITE ELEMENTS

PLANE STRESS, PLANE STRAIN OR AXISYMMETRIC PROGRAM

PLANE STRESS ANALYSIS

VON MISES YIELD CONDITION

ELASTIC, PERFECTLY PLASTIC : ANALYSIS TERMINATED WHEN LOAD INCREMENT MAGNITUDE DECREASES TO .005 OF TOTAL LOAD

ELEMENTS WITH VON MISES EQUIVALENT STRESS WITHIN .995 OF YIELD STRESS TREATED AS PLASTIC ELEMENTS

UNITS: MM KG

```
STORAGE AVAILABLE  :                         MAXIMUM NUMBER OF NODES :   97
                   :                      MAXIMUM NUMBER OF ELEMENTS :  158
                   :              MAXIMUM NUMBER OF DEGREES OF FREEDOM :  194
                   : MAXIMUM HALF BAND-WIDTH OF ELASTIC MATRIX :   24
                   :              MAXIMUM NUMBER OF PLASTIC ELEMENTS :   60
                   : MAXIMUM NUMBER OF PROPORTIONAL LOAD PATHS :    1

STORAGE REQUIRED   :                                 NUMBER OF NODES :   97
                   :                              NUMBER OF ELEMENTS :  158
                   :                      NUMBER OF DEGREES OF FREEDOM :  194
                   :                 HALF BAND-WIDTH OF ELASTIC MATRIX :   24

MATERIAL CONSTANTS : ELASTIC MODULUS  = .100000+04
                   : PLASTIC MODULUS  = .000000
                   : POISSONS RATIO   = .000000
                   : YIELD STRESS     = .320000+02

PLATE THICKNESS    : .100000+01
```

COORDINATES OF NODES

NODE	X	Y	NODE	X	Y	NODE	X	Y	NODE	X	Y
1	.32000+03	.80000+02	2	.32000+03	.40000+02	3	.32000+03	.00000	4	.28000+03	.80000+02
5	.28000+03	.65000+02	6	.28000+03	.40000+02	7	.28000+03	.00000	8	.25000+03	.80000+02
9	.25000+03	.65000+02	10	.25000+03	.45000+02	11	.25000+03	.25000+02	12	.25000+03	.00000
13	.22000+03	.80000+02	14	.22000+03	.65000+02	15	.22000+03	.45000+02	16	.22000+03	.25000+02
17	.22000+03	.00000	18	.20000+03	.80000+02	19	.20000+03	.70000+02	20	.20000+03	.55000+02
21	.20000+03	.40000+02	22	.20000+03	.20000+02	23	.20000+03	.00000	24	.18000+03	.80000+02
25	.18000+03	.70000+02	26	.18000+03	.60000+02	27	.18000+03	.50000+02	28	.18000+03	.35000+02
29	.18000+03	.20000+02	30	.18000+03	.00000	31	.17000+03	.80000+02	32	.16969+03	.75000+02
33	.16937+03	.70000+02	34	.16875+03	.60000+02	35	.16812+03	.50000+02	36	.16719+03	.35000+02
37	.16625+03	.20000+02	38	.16500+03	.00000	39	.16500+03	.80000+02	40	.16406+03	.75000+02

41	.16312+03	.70000+02	42	.16219+03	.65000+02	43	.16078+03	.57500+02	44	.15891+03	.47500+02
45	.15656+03	.35000+02	46	.15375+03	.20000+02	47	.15000+03	.00000	48	.16000+03	.80000+02
49	.15812+03	.75000+02	50	.15625+03	.70000+02	51	.15437+03	.65000+02	52	.15156+03	.57500+02
53	.14875+03	.50000+02	54	.14500+03	.40000+02	55	.14125+03	.30000+02	56	.13656+03	.17500+02
57	.13000+03	.00000	58	.16000+03	.85000+02	59	.15529+03	.80000+02	60	.15059+03	.75000+02
61	.14353+03	.67500+02	62	.13647+03	.60000+02	63	.12706+03	.50000+02	64	.11294+03	.35000+02
65	.96470+02	.17500+02	66	.80000+02	.00000	67	.16000+03	.90000+02	68	.15500+03	.87187+02
69	.14500+03	.81562+02	70	.13000+03	.73125+02	71	.11000+03	.61875+02	72	.85000+02	.47812+02
73	.55000+02	.30937+02	74	.00000	.00000	75	.16000+03	.10000+03	76	.15000+03	.97500+02
77	.13500+03	.93750+02	78	.11500+03	.88750+02	79	.85000+02	.81250+02	80	.45000+02	.71250+02
81	.00000	.60000+02	82	.16000+03	.12000+03	83	.14000+03	.11750+03	84	.12000+03	.11500+03
85	.90000+02	.11125+03	86	.50000+02	.10625+03	87	.00000	.10000+03	88	.16000+03	.14000+03
89	.14000+03	.13875+03	90	.11000+03	.13687+03	91	.60000+02	.13775+03	92	.00000	.13000+03
93	.16000+03	.16000+03	94	.14000+03	.16000+03	95	.11000+03	.16000+03	96	.60000+02	.16000+03
97	.00000	.16000+03									

ELEMENT	NODES			ELEMENT	NODES			ELEMENT	NODES			ELEMENT	NODES		
1	5	1	4	2	2	1	5	3	6	2	5	4	3	2	6
5	7	3	6	6	9	4	8	7	9	5	4	8	10	5	9
9	6	5	10	10	11	6	10	11	7	6	11	12	12	7	11
13	9	8	13	14	14	9	13	15	10	9	14	16	15	10	14
17	11	10	15	18	16	11	15	19	12	11	16	20	17	12	16
21	19	13	18	22	14	13	19	23	20	14	19	24	15	14	20
25	21	15	20	26	16	15	21	27	22	16	21	28	17	16	22
29	23	17	22	30	25	18	24	31	25	19	18	32	26	19	25
33	20	19	26	34	27	20	26	35	21	20	27	36	28	21	27
37	22	21	28	38	29	22	28	39	23	22	29	40	30	23	29
41	32	24	31	42	25	24	32	43	33	25	32	44	26	25	33
45	34	26	33	46	27	26	34	47	35	27	34	48	28	27	35
49	36	28	35	50	29	28	36	51	37	29	36	52	30	29	37
53	38	30	37	54	40	31	39	55	40	32	31	56	41	32	40
57	41	33	32	58	42	33	41	59	34	33	42	60	43	34	42
61	35	34	43	62	44	35	43	63	36	35	44	64	45	36	44
65	37	36	45	66	46	37	45	67	38	37	46	68	47	38	46
69	49	39	48	70	49	40	39	71	50	40	49	72	50	41	40
73	51	41	50	74	51	42	41	75	52	42	51	76	52	43	42
77	53	43	52	78	53	44	43	79	54	44	53	80	45	44	54
81	55	45	54	82	46	45	55	83	56	46	55	84	47	46	56
85	57	47	56	86	59	48	58	87	49	48	59	88	60	49	59
89	50	49	60	90	51	50	60	91	51	60	61	92	52	51	61
93	52	61	62	94	53	52	62	95	63	53	62	96	54	53	63
97	55	54	63	98	55	63	64	99	56	55	64	100	65	56	64
101	57	56	65	102	66	57	65	103	58	67	68	104	59	58	68
105	59	68	69	106	60	59	69	107	61	60	69	108	61	69	70
109	62	61	70	110	71	62	70	111	63	62	71	112	64	63	71
113	64	71	72	114	65	64	72	115	65	72	73	116	66	65	73
117	74	66	73	118	67	75	76	119	68	67	76	120	69	68	76
121	69	76	77	122	70	69	77	123	70	77	78	124	71	70	78
125	71	78	79	126	72	71	79	127	72	79	80	128	73	72	80
129	73	80	81	130	74	73	81	131	76	75	82	132	76	82	83
133	77	76	83	134	77	83	84	135	78	77	84	136	78	84	85
137	79	78	85	138	79	85	86	139	80	79	86	140	80	86	87
141	81	80	87	142	83	82	88	143	83	88	89	144	84	83	89
145	84	89	90	146	85	84	90	147	85	90	91	148	86	85	91
149	86	91	92	150	87	86	92	151	88	93	94	152	89	88	94
153	89	94	95	154	90	89	95	155	90	95	96	156	91	90	96
157	91	96	97	158	92	91	97								

BOUNDARY CONDITIONS : 0 - CONSTRAINED
1 - UNCONSTRAINED

NODE	X,Y
3	0,1
7	0,1
12	0,1
17	0,1
23	0,1
30	0,1
38	0,1
47	0,1
57	0,1
66	0,1
74	0,0
81	0,0
87	0,0
92	0,0
93	0,0
94	0,0
95	0,0
96	0,0
97	0,0

LOADING PROGRAMME:

LOADING 1 : END LOADING 1 WHEN TOTAL LOAD AT NODE 3 IN Y-DIRECTION IS .211830+03

1	.000000	.700000+01
2	.000000	.340000+02
3	.000000	.230000+02

: COMPONENTS OF PROPORTIONAL LOAD VECTOR 1

.0000	.7000+01	.0000	.3400+02	.0000	.2300+02	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000										

REQUESTED OUTPUT OF RESULTS :-

```
PRINT RESULTS FOR ELEMENTS :-
  62  64  65  66  67  68  69  71  72  73  74  76  78  86

PRINT RESULTS FOR NODES :-
   1   2   3   7  12  17  23  30  38  47  57  66  74

RESULTS PRINTED AT END OF EACH PROPORTIONAL LOAD PATH AND AFTER LOAD INCREMENTS :-
   1
```

RESULTS AFTER 1 LOAD INCREMENTS

NEXT ELEMENTS TO UNDERGO PLASTIC DEFORMATION : 69

AT LOAD FACTOR .278873+01

LOAD VECTOR 1 : CUMULATIVE LOAD FACTOR = .278873+01

ELEMENT		STRAIN INCREMENTS ELASTIC	STRAIN INCREMENTS PLASTIC	LAMBDA	STRESS INCREMENT	CURRENT VALUES: STRAINS ELASTIC	CURRENT VALUES: STRAINS PLASTIC	CURRENT VALUES: STRAINS TOTAL	CURRENT VALUES: STRESS	CURRENT VALUES: YIELD FUNCTION	STRESS CORRECTION FACTOR
	X	-.54277-02	--		-.54277+01	-.54277-02	.00000	-.54277-02	-.54277+01		
62	Y	-.39232-02	--	--	-.39232+01	-.39232-02	.00000	-.39232-02	-.39232+01	-.98540+03	--
	XY	.44786-02	--		.22393+01	.44786-02	.00000	.44786-02	.22393+01		
	X	-.33187-02	--		-.33187+01	-.33187-02	.00000	-.33187-02	-.33187+01		
64	Y	-.28857-02	--	--	-.28857+01	-.28857-02	.00000	-.28857-02	-.28857+01	-.10072+04	--
	XY	.30536-02	--		.15268+01	.30536-02	.00000	.30536-02	.15268+01		
	X	-.33187-02	--		-.33187+01	-.33187-02	.00000	-.33187-02	-.33187+01		
65	Y	-.15653-02	--	--	-.15653+01	-.15653-02	.00000	-.15653-02	-.15653+01	-.10133+04	--
	XY	.18102-02	--		.90508+00	.18102-02	.00000	.18102-02	.90508+00		
	X	-.17976-02	--		-.17976+01	-.17976-02	.00000	-.17976-02	-.17976+01		
66	Y	-.18134-02	--	--	-.18134+01	-.18134-02	.00000	-.18134-02	-.18134+01	-.10164+04	--
	XY	.24083-02	--		.12042+01	.24083-02	.00000	.24083-02	.12042+01		
	X	-.17976-02	--		-.17976+01	-.17976-02	.00000	-.17976-02	-.17976+01		
67	Y	-.53529-03	--	--	-.53529+00	-.53529-03	.00000	-.53529-03	-.53529+00	-.10201+04	--
	XY	.13205-02	--		.66027+00	.13205-02	.00000	.13205-02	.66027+00		
	X	.00000	--		.00000	.00000	.00000	.00000	.00000		
68	Y	-.76984-03	--	--	-.76984+00	-.76984-03	.00000	-.76984-03	-.76984+00	-.10207+04	--
	XY	.19147-02	--		.95736+00	.19147-02	.00000	.19147-02	.95736+00		
	X	-.26343-01	.00000		-.26343+02	-.26343-01	.00000	-.26343-01	-.26343+02		
69	Y	-.13237-01	.00000	.00000	-.13237+02	-.13237-01	.00000	-.13237-01	-.13237+02	-.45776-04	.00000
	XY	.25911-01	.00000		.12956+02	.25911-01	.00000	.25911-01	.12956+02		
	X	-.15094-01	--		-.15094+02	-.15094-01	.00000	-.15094-01	-.15094+02		
71	Y	-.95328-02	--	--	-.95328+01	-.95328-02	.00000	-.95328-02	-.95328+01	-.72498+03	--
	XY	.12869-01	--		.64346+01	.12869-01	.00000	.12869-01	.64346+01		
	X	-.10960-01	--		-.10960+02	-.10960-01	.00000	-.10960-01	-.10960+02		
72	Y	-.48652-02	--	--	-.48652+01	-.48652-02	.00000	-.48652-02	-.48652+01	-.92474+03	--
	XY	.34234-02	--		.17117+01	.34234-02	.00000	.34234-02	.17117+01		
	Y	-.10960-01	--		-.10960+02	-.10960-01	.00000	-.10960-01	-.10960+02		
73	Y	-.68344-02	--	--	-.68344+01	-.68344-02	.00000	-.68344-02	-.68344+01	-.87701+03	--
	XY	.85684-02	--		.42842+01	.85684-02	.00000	.85684-02	.42842+01		
	X	-.81883-02	--		-.81883+01	-.81883-02	.00000	-.81883-02	-.81883+01		
74	Y	-.46250-02	--	--	-.46250+01	-.46250-02	.00000	-.46250-02	-.46250+01	-.96891+03	--
	XY	.24550-02	--		.12275+01	.24550-02	.00000	.24550-02	.12275+01		

	X	-.64840-02	--		-.64840+01	-.64840-02	.00000	-.64840-02	-.64840+01		
76	Y	-.43617-02	--	--	-.43617+01	-.43617-02	.00000	-.43617-02	-.43617+01	-.98865+03	--
	XY	.18485-02	--		.92427+00	.18485-02	.00000	.18485-02	.92427+00		
	X	-.52433-02	--		-.52433+01	-.52433-02	.00000	-.52433-02	-.52433+01		
78	Y	-.32592-02	--	--	-.32592+01	-.32592-02	.00000	-.32592-02	-.32592+01	-.10024+04	--
	XY	.90272-03	--		.45136+00	.90272-03	.00000	.90272-03	.45136+00		
	X	-.15831-01	--		-.15831+02	-.15831-01	.00000	-.15831-01	-.15831+02		
86	Y	-.18978-01	--	--	-.18978+02	-.18978-01	.00000	-.18978-01	-.18978+02	-.62627+02	--
	XY	.29462-01	--		.14731+02	.29462-01	.00000	.29462-01	.14731+02		

CURRENT PLASTIC ELEMENTS :-

69

NODE	LOAD INCREMENT DPX	DPY	TOTAL LOAD PX	PY	DISPLACEMENT INCREMENT DU	DV	TOTAL DISPLACEMENT U	V
1	.000000	.195211+02	.000000	.195211+02	-.179400+01	.405583+01	-.179400+01	.405583+01
2	.000000	.948168+02	.000000	.948168+02	-.835900+00	.406216+01	-.835900+00	.406216+01
3	.000000	.641407+02	.000000	.641407+02	.000000	.406216+01	.000000	.406216+01
7	.000000	.000000	.000000	.000000	.000000	.299526+01	.000000	.299526+01
12	.000000	.000000	.000000	.000000	.000000	.222560+01	.000000	.222560+01
17	.000000	.000000	.000000	.000000	.000000	.152839+01	.000000	.152839+01
23	.000000	.000000	.000000	.000000	.000000	.111298+01	.000000	.111298+01
30	.000000	.000000	.000000	.000000	.000000	.761039+00	.000000	.761039+00
38	.000000	.000000	.000000	.000000	.000000	.546465+00	.000000	.546465+00
47	.000000	.000000	.000000	.000000	.000000	.377537+00	.000000	.377537+00
57	.000000	.000000	.000000	.000000	.000000	.220491+00	.000000	.220491+00
66	.000000	.000000	.000000	.000000	.000000	.366498-01	.000000	.366498-01
74	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000

LOAD INCREMENT 8 ITERATION 1
ELEMENT 56 UNLOADING : LAMBDA = -.630463-03

LOAD INCREMENT 15 ITERATION 1
ELEMENT 86 UNLOADING : LAMBDA = -.959574-03

LOAD INCREMENT 18 ITERATION 1
ELEMENT 86 UNLOADING : LAMBDA = -.178643-02

LOAD INCREMENT 18 ITERATION 1
ELEMENT 87 UNLOADING : LAMBDA = -.133927-03

LOAD INCREMENT 18 ITERATION 2
ELEMENT 87 LOADING
SCALAR PRODUCT OF GRADIENT OF YIELD FUNCTION AND NORMALIZED STRESS INCREMENT VECTOR = .382127+00

LOAD INCREMENT 19 ITERATION 1

```
ELEMENT   86  UNLOADING  :  LAMBDA =  -.132379-02

LOAD INCREMENT   20   ITERATION    1
ELEMENT   86  UNLOADING  :  LAMBDA =  -.961031-03

LOAD INCREMENT   21   ITERATION    1
ELEMENT   86  UNLOADING  :  LAMBDA =  -.916677-03

LOAD INCREMENT   22   ITERATION    1
ELEMENT   86  UNLOADING  :  LAMBDA =  -.113285-02

LOAD INCREMENT   23   ITERATION    1
ELEMENT   86  UNLOADING  :  LAMBDA =  -.545818-02

LOAD INCREMENT   23   ITERATION    1
ELEMENT   87  UNLOADING  :  LAMBDA =  -.208202-02

LOAD INCREMENT   23   ITERATION    1
ELEMENT   88  UNLOADING  :  LAMBDA =  -.136440-02
```

RESULTS AFTER 26 LOAD INCREMENTS

END OF LOADING 1 : MAXIMUM LOAD REACHED AT NODE 3

AT LOAD FACTOR .604814-01

LOAD VECTOR 1 : CUMULATIVE LOAD FACTOR = .921000+01

		STRAIN INCREMENTS				CURRENT VALUES					
						STRAINS					
ELEMENT		ELASTIC	PLASTIC	LAMBDA	STRESS INCREMENT	ELASTIC	PLASTIC	TOTAL	STRESS	YIELD FUNCTION	STRESS CORRECTION FACTOR
	X	-.23677-03	-.48237-03		-.23677+00	-.30181-01	-.10786-02	-.31259-01	-.30176+02		
62	Y	-.12484-03	-.92725-04	.89462-03	-.12484+00	-.19035-01	-.20950-03	-.19245-01	-.19032+02	-.30518-04	.99994+00
	XY	-.33646-03	.74770-03		-.16823+00	.20836-01	.16999-02	.22536-01	.10416+02		
	X	-.32611-03	--		-.32611+00	-.17189-01	.00000	-.17189-01	-.17189+02		
64	Y	-.26027-03	--	--	-.26027+00	-.15327-01	.00000	-.15327-01	-.15327+02	-.55617+03	--
	XY	.28413-03	--		.14207+00	.16367-01	.00000	.16367-01	.81834+01		
	X	-.32611-03	--		-.32611+00	-.17189-01	.00000	-.17189-01	-.17189+02		
65	Y	-.92753-04	--	--	-.92753-01	-.68113-02	.00000	-.68113-02	-.68113+01	-.67398+03	--
	XY	.20948-03	--		.10474+00	.12922-01	.00000	.12922-01	.64611+01		
	X	-.14651-03	--		-.14651+00	-.87179-02	.00000	-.87179-02	-.87179+01		
66	Y	-.16692-03	--	--	-.16692+00	-.98959-02	.00000	-.98959-02	-.98959+01	-.79727+03	--
	XY	.21063-03	--		.10531+00	.13617-01	.00000	.13617-01	.68086+01		
	X	-.14651-03	--		-.14651+00	-.87179-02	.00000	-.87179-02	-.87179+01		
67	Y	-.35345-04	--	--	-.35345-01	-.24372-02	.00000	-.24372-02	-.24372+01	-.89885+03	--
	XY	.13932-03	--		.69658-01	.92702-02	.00000	.92702-02	.46351+01		
	X	.00000	--		.00000	.00000	.00000	.00000	.00000		
68	Y	-.68499-04	--	--	-.68499-01	-.41518-02	.00000	-.41518-02	-.41518+01	-.91393+03	--
	XY	.16279-03	--		.81396-01	.11126-01	.00000	.11126-01	.55629+01		
	X	-.49519-05	-.11067-01		-.49519-02	-.35379-01	-.50084+00	-.53622+00	-.33162+02		
69	Y	-.51213-05	-.26455-02	.16031-01	-.51213-02	-.23458-01	-.12892+00	-.15238+00	-.21894+02	-.45776-04	.10000+01
	XY	-.12109-04	.11292-01		-.60544-02	.16126-01	.55859+00	.57472+00	.75492+01		
	X	.15683-04	-.81933-02		.15683-01	-.33960-01	-.30596+00	-.33992+00	-.33415+02		
71	Y	-.12126-04	-.18581-02	.11576-01	-.12126-01	-.22260-01	-.99714-01	-.11197+00	-.21835+02	-.30518-04	.10000+01
	XY	.26610-04	.79638-02		.13305-01	.14964-01	.34219+00	.35715+00	.73096+01		
	X	-.32457-04	-.59350-02		-.32457-01	-.35883-01	-.18019+00	-.21608+00	-.35592+02		
72	Y	.13417-03	.80032-04	.68036-02	.13417+00	-.17402-01	.14909-01	-.24932-02	-.17284+02	-.45776-04	.99998+00
	XY	-.12232-03	.33254-02		-.61159-01	.99724-02	.83605-01	.93577-01	.49537+01		
	X	.20102-04	-.59876-02		.20102-01	-.33750-01	-.18233+00	-.21608+00	-.33399+02		
73	Y	-.20801-04	-.14747-02	.84705-02	-.20801-01	-.22495-01	-.63827-01	-.86322-01	-.22227+02	-.30518-04	.10000+01
	XY	.30888-04	.58072-02		.15444-01	.14668-01	.22097+00	.23564+00	.72301+01		
	X	-.24192-04	-.42230-02		-.24182-01	-.36188-01	-.96248-01	-.13244+00	-.35968+02		
74	Y	.92263-04	.51932-04	.46719-02	.92263-01	-.17638-01	.90729-02	-.85649-02	-.17546+02	-.15259-04	.99999+00
	XY	-.10705-03	.19975-02		-.53523-01	.84920-02	.36340-01	.44832-01	.42234+01		

	X	-.52490-05	-.23202-02		-.52490-02	-.36574-01	-.29041-01	-.65615-01	-.36491+02		
76	Y	-.34568-04	.79222-04	.24297-02	-.34568-01	-.17365-01	.21578-02	-.15207-01	-.17327+02	-.15259-04	.10000+01
	XY	-.26334-04	.71702-03		-.13167-01	.57217-02	.71496-02	.12871-01	.28547+01		
	X	-.79533-03	--		-.79533+00	-.34278-01	.00000	-.34278-01	-.34278+02		
78	Y	-.13009-03	--	--	-.13009+00	-.15282-01	.00000	-.15282-01	-.15282+02	-.13640+03	--
	XY	-.41028-04	--		-.20514-01	.19672-02	.00000	.19672-02	.98359+00		
	X	-.12092-04	--		-.12092-01	-.20861-01	-.48672-02	-.25728-01	-.20495+02		
86	Y	.60133-05	--	--	.60133-02	-.28723-01	-.11824-01	-.40547-01	-.28196+02	-.19624+02	--
	XY	-.83956-05	--		-.41978-02	.22662-01	.35395-01	.58057-01	.11064+02		

CURRENT PLASTIC ELEMENTS :-
----

41 42 43 44 45 46 48 54 55 56 57 58 59 60 61 62 69 70 71 72 73 74 75 76 77

79 87 88 90 91 105

NODE	LOAD INCREMENT DPX	DPY	TOTAL LOAD PX	PY	DISPLACEMENT INCREMENT DU	DV	TOTAL DISPLACEMENT U	V
1	.000000	.423370+00	.000000	.644700+02	-.940761-01	.213338+00	-.764891+01	.174457+02
2	.000000	.205637+01	.000000	.313140+03	-.457281-01	.213475+00	-.362323+01	.174665+02
3	.000000	.139107+01	.000000	.211830+03	.000000	.213452+00	.000000	.174655+02
7	.000000	.000000	.000000	.000000	.000000	.162675+00	.000000	.130776+02
12	.000000	.000000	.000000	.000000	.000000	.125078+00	.000000	.988035+01
17	.000000	.000000	.000000	.000000	.000000	.886060-01	.000000	.690594+01
23	.000000	.000000	.000000	.000000	.000000	.647978-01	.000000	.506720+01
30	.000000	.000000	.000000	.000000	.000000	.423805-01	.000000	.343827+01
38	.000000	.000000	.000000	.000000	.000000	.278624-01	.000000	.240680+01
47	.000000	.000000	.000000	.000000	.000000	.167080-01	.000000	.159099+01
57	.000000	.000000	.000000	.000000	.000000	.764344-02	.000000	.862167+00
66	.000000	.000000	.000000	.000000	.000000	.651903-03	.000000	.118647+00
74	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000

A.4.1 Data Input for ~~EPQCI~~ EPCQI

HEADING1

HEADING2

] i) heading statements

TYPE

NN,NE,E,EP,RNU,SZERO,ETA,IO,NLAM,(THIK),FRAC

] ii) structure statements

$1,N_1,N_2,\ldots,N_{12}$,NTYPE

$X_1,Y_1,X_2,Y_2,\ldots,X_k,Y_k$

2,....................

....

.

.

NE,...................

....

] iii) element incidences and nodal coordinates

IX,IY

$N_a,N_b,N_c,\ldots,N_k,-1$

IX,IY

.... -1

.

.

-1

] iv) boundary conditions

NLP,N,I_{xy},SIZE

IE,ISIDE,T_x,T_y,NTYPE](*)

.....................]

.

.

-1

N_i,N_i

P_x,P_y] (**)

.....

.....

.

.

-1

] (***)

continued

] v) loading program

```
..................]  ]  |
..................]  |  |
 .                   |  |
 .                   |  |
-1                   |  |
......     ]         |  |
......     ]         |  |
.....      ]         |  |
.....      ]         |  |
 .                   |  |
 .                   |  |
-1                   ]  |
 .                      |
 .                      |
-1                      ]
```

PRINT

$IE_a, IG_a, IE_b, IG_b, \ldots, IE_k, IG_k, -1$

$N_a, N_b, N_c, \ldots, N_k, -1$ — vi) output requested

$IOUT_a, IOUT_b, IOUT_c, \ldots, IOUT_k, -2$

A.4.2 Description of Data Input

(i) Heading Statements

(as for program EPTCS; see section A.3.2.(i))

(ii) Structure Statements

TYPE

NN,NE,E,EP,RNU,SZERO,ETA,IO,NLAM,(THIK),FRAC

(as for program EPTCS; see section A.3.2.(ii)), and where

IO (integer) integration order for Gauss quadrature; maximum value 4, recommended value 3.

NLAM (integer) number of plastic multipliers associated with each element: NLAM = 1 for one 'average' multiplier per element; or NLAM = 10^2 for one multiplier per integration point; (see section 4.6).

(iii) Element Incidences and Nodal Coordinates

$1, N_1, N_2, \ldots, N_{12}$, NTYPE] (*)
$X_1, Y_1, X_2, Y_2, \ldots, X_k, Y_k$]
2,.................]
...................]
.
.
NE,................]
...................]

where

$N_1, N_2, \ldots, N_{12}$ (integer) element incidences (see Figure 4.1)

NTYPE (integer) = 1 if boundaries of element are straight; ≠ 1 if any boundary of element is curved

$X_1, Y_1, \ldots, X_k, Y_k$ if NTYPE = 1 then k = 4, i.e. input global coordinates of corner nodes of element only; if NTYPE ≠ 1 then k = 12, i.e. input global coordinates of all nodes of element (see Figure 4.1)

A data subgroup such as (*) above corresponds to each isoparametric element.

(iv) Boundary Conditions

(as for program EPTCS; see section A.3.2.(v)). (page A-9)

(v) Loading Program

NLP,N,I_{xy},SIZE

(as for program EPTCS; see section A.3.2.(vi)) (page A-10)

IE,ISIDE,T_x,T_y,NTYPE](*)

where

IE (integer) number of element

ISIDE (integer) number of element side on which boundary traction T_x,T_y acts:

for side s = +1 enter 1

for side r = -1 enter 2

for side s = -1 enter 3 (see Figure 4.1)

for side r = +1 enter 4

T_x,T_y (real) boundary traction components respectively in global x, y directions: force per unit area acting along element boundary ISIDE. (for axisymmetric case do not multiply by 2π).

NTYPE (integer) if element boundary ISIDE is straight enter 1; if element boundary is arc of circle with centre the origin of global coordinate system, enter 2. In this case T_x is radius of circular arc, and T_y is magnitude of radial force per unit area. (see example of data input, section A.4.3)

A data sybgroup such as (*) above corresponds to each element side which has non-zero boundary tractions. A line containing only -1 indicates the end of boundary tractions for load path NLP. If there are no boundary tractions for load path NLP enter -1.

N_i,N_i]
P_x,P_y] nodal point loading for load path NLP (as for program EPTCS, see section A.3.2.(vi))

A line containing only -1 indicates the end of nodal point loads for load path NLP. If there are no nodal point loads for load path NLP enter -1.

A loading data subgroup (***) must be enetered for each segment of the piecewise linear proportional load path. The end of all load path subgroups is indicated by a line containing only -1.

(vi) <u>Output Requested</u>

PRINT

$IE_a, IG_a, IE_b, IG_b, \ldots, IE_k, IG_k, -1$

$N_a, N_b, N_c, \ldots, N_k, -1$

$IOUT_a, IOUT_b, IOUT_c, \ldots, IOUT_k - 2$

(as for program EPTCS; see section A.3.2.(vii), and where

IE_i, IG_i (integer) are element and integration point numbers for which stresses and strains are to be printed. Integration points are numbered for each element on lines of constant r value, beginning at the Gauss point closest to (r,s) = (-1,-1), and increasing in magnitude in the positive local s coordinate direction. For example third order numerical integration has Gauss points 1,3,7,9 respectively closest to corners (r,s) = (-1,-1), (-1,+1),(+1,-1) and (+1,+1). (see Figure 4.1)

A.4.3 Example of Data and Results for ~~EPTCS~~ EPCQI

The pages immediately following list data and results corresponding to a cubic quadrilateral finite element analysis of the deep beam in plane stress described in section 6.3. The element mesh consisting of 192 nodes and 31 elements is shown in Figure 6.3.2. The result of this analysis is listed in Table 6.3.1.

```
COL*EPCQT(1).DB
```

(i) Heading statements

```
DEEP BEAM
UNITS :  KG        MM
```

(ii) Structure Statements

```
PLANE STRESS
192,31,1000.0,0.0,0.0,32.0,0.995,3,9,1.0,0.005
```

(iii) Element incidences and nodal coordinates

```
1 32,4,1,29 24,18,3,2 17,23,30,31 1
60.,32. 0.,32. 0.,0. 60.,0.
2 35,7,4,32 25,19,6,5 18,24,33,34 1
60.,64. 0.,64. 0.,32. 60.,32.
3 38,10,7,35 26,20,9,8 19,25,36,37 1
96.,60. 0.,96. 0.,64. 60.,64.
4 41,13,10,38 27,21,12,11 20,26,39,40 1
62.,128. 0.,128. 0.,96. 60.,96.
5 44,16,13,41 28,22,15,14 21,27,42,43 1
64.,160. 0.,160. 0.,128. 62.,128.
6 60,32,29,57 52,46,31,30 45,51,58,59 1
120.,26. 60.,32. 60.,0. 120.,0.
7 63,35,32,60 53,47,34,33 46,52,61,62 1
120.,52. 60.,64. 60.,32. 120.,26.
8 66,38,35,63 54,48,37,36 47,53,64,65 1
120.,78. 60.,96. 60.,64. 120.,52.
9 69,41,38,66 55,49,40,39 48,54,67,68 1
120.,112. 62.,128. 60.,96. 120.,78.
10 72,44,41,69 56,50,43,42 49,55,70,71 1
126.,160. 64.,160. 62.,128. 120.,112.
11 89,60,57,85 80,74,59,58 73,79,86,87 1
157.46,22. 120.,26. 120.,0. 157.46,0.
12 91,63,60,88 81,75,62,61 74,80,89,90 1
157.46,44. 120.,52. 120.,26. 157.46,22.
13 94,66,63,91 82,76,65,64 75,81,92,93 1
157.46,64. 120.,78. 120.,52. 157.46,44.
14 97,69,66,94 83,77,68,67 76,82,95,96 1
160.,80. 120.,112. 120.,78. 157.46,64.
15 100,72,69,97 84,78,71,70 77,83,98,99 1
160.,160. 126.,160. 120.,112. 160.,80.
16 114,88,85,111 107,102,87,86 101,106,112,113 1
180.,20. 157.46,22. 157.46,0. 180.,0.
17 117,91,88,114 108,103,90,89 102,107,115,116 1
180.,40. 157.46,44. 157.46,22. 180.,20.
18 120,94,91,117 109,104,93,92 103,108,118,119 1
180.,60. 157.46,64. 157.46,44. 180.,40.
19 123,97,94,120 110,105,96,95 104,109,121,122 1
180.,80. 160.,80. 157.46,64. 180.,60.
20 137,114,111,134 130,125,113,112 124,129,135,136 1
210.,20. 180.,20. 180.,0. 210.,0.
21 140,117,114,137 131,126,116,115 125,130,138,139 1
210.,40. 180.,40. 180.,20. 210.,20.
22 143,120,117,140 132,127,119,118 126,131,141,142 1
210.,60. 180.,60. 180.,40. 210.,40.
23 146,123,120,143 133,128,122,121 127,132,144,145 1
210.,80. 180.,80. 180.,60. 210.,60.
24 160,137,134,157 153,148,136,135 147,152,158,159 1
260.,20. 210.,20. 210.,0. 260.,0.
25 163,140,137,160 154,149,139,138 148,153,161,162 1
260.,40. 210.,40. 210.,20. 260.,20.
26 166,143,140,163 155,150,142,141 149,154,164,165 1
260.,60. 210.,60. 210.,40. 260.,40.
27 169,146,143,166 156,151,145,144 150,155,167,168 1
260.,80. 210.,80. 210.,60. 260.,60.
28 183,160,157,180 176,171,159,158 170,175,181,182 1
```

```
320.,20. 260.,20. 260.,0. 320.,0.
29 186,163,160,183 177,172,162,161 171,176,184,185 1
320.,40. 260.,40. 260.,20. 320.,20.
30 189,166,163,186 178,173,165,164 172,177,187,188 1
320.,60. 260.,60. 260.,40. 320.,40.
31 192,169,166,189 179,174,168,167 173,178,190,191 1
320.,80. 260.,80. 260.,60. 320.,60.
```

(iv) Boundary conditions

```
0,1
17,23,29,45,51,57,73,79,85,101,111,124,129,134,147 -1
0,1
152,157,170,175,180 -1
0,0
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,22,28,44,50 -1
0,0
56,72,78,84,100 -1
-1
```

(v) Loading program

```
1,180,2,-36.8187
-1
180,180
0.,-6.908
181,181
0.,-13.720
182,182
0.,-13.432
183,183
0.,-12.952
184,184
0.,-12.280
185,185
0.,-11.416
186 186
0.,-10.360
187,187
0.,-9.112
188,188
0.,-7.672
189,189
0.,-6.040
190,190
0.,-4.216
191 191
0.,-2.200
192 192
0.,-0.284
-1
-1
```

P = -110.592

(vi) Output requested

```
SOME
16,1 16,2 16,3 17,1 17,2 17,3 18,1 18,2 18,3 19,1 19,2 19,3 -1
1,29,57,85,111,134,157,180 -1
1 -2
```

```
@XQT EPC01.ABS

@ADD,P EPC01.D8
```

DEEP BEAM

NUMERICAL ANALYSIS USING TWO DIMENSIONAL TWELVE NODE ISOPARAMETRIC FINITE ELEMENTS

PLANE STRESS, PLANE STRAIN OR AXISYMMETRIC PROGRAM

CUBIC INTERPOLATION FUNCTIONS

QUADRATIC STRAIN VARIATION

NUMERICAL INTEGRATION USING GAUSS QUADRATURE

ORDER OF INTEGRATION = 3

PLANE STRESS ANALYSIS

VON MISES YIELD CONDITION

ELASTIC, PERFECTLY PLASTIC : ANALYSIS TERMINATED WHEN LOAD INCREMENT MAGNITUDE DECREASES TO .005 OF TOTAL LOAD

INTEGRATION POINTS WITH VON MISES EQUIVALENT STRESS WITHIN .995 OF YIELD STRESS TREATED AS PLASTIC

UNITS : KG MM

STORAGE AVAILABLE	:	MAXIMUM NUMBER OF NODES :	192
	:	MAXIMUM NUMBER OF ELEMENTS :	31
	:	MAXIMUM NUMBER OF DEGREES OF FREEDOM :	384
	:	MAXIMUM HALF BAND-WIDTH OF ELASTIC MATRIX :	64
	:	MAXIMUM NUMBER OF PLASTIC ELEMENTS :	30
	:	MAXIMUM NUMBER OF PROPORTIONAL LOAD PATHS :	1

STORAGE REQUIRED	:	NUMBER OF NODES :	192
	:	NUMBER OF ELEMENTS :	31
	:	NUMBER OF DEGREES OF FREEDOM :	384
	:	HALF BAND-WIDTH OF ELASTIC MATRIX :	64

MATERIAL CONSTANTS	: ELASTIC MODULUS	=	.100000+04
	: PLASTIC MODULUS	=	.000000
	: POISSONS RATIO	=	.000000
	: YIELD STRESS	=	.320000+02

PLATE THICKNESS : .100000+01

COORDINATES OF NODES

NODE	X	Y	NODE	X	Y	NODE	X	Y	NODE	X	Y
1	.00000	.00000	2	.00000	.10667+02	3	.00000	.21333+02	4	.00000	.32000+02

5	.00000	.42667+02	6	.00000	.53333+02	7	.00000	.64000+02	8	.00000	.74667+02
9	.00000	.85333+02	10	.00000	.96000+02	11	.00000	.10667+03	12	.00000	.11733+03
13	.00000	.12800+03	14	.00000	.13867+03	15	.00000	.14933+03	16	.00000	.16000+03
17	.20000+02	.00000	18	.20000+02	.32000+02	19	.20000+02	.64000+02	20	.20000+02	.96000+02
21	.20667+02	.12800+03	22	.21333+02	.16000+03	23	.40000+02	.00000	24	.40000+02	.32000+02
25	.40000+02	.64000+02	26	.40000+02	.96000+02	27	.41333+02	.12800+03	28	.42667+02	.16000+03
29	.60000+02	.00000	30	.60000+02	.10667+02	31	.60000+02	.21333+02	32	.60000+02	.32000+02
33	.60000+02	.42667+02	34	.60000+02	.53333+02	35	.60000+02	.64000+02	36	.60000+02	.74667+02
37	.60000+02	.85333+02	38	.60000+02	.96000+02	39	.60667+02	.10667+03	40	.61333+02	.11733+03
41	.62000+02	.12800+03	42	.62667+02	.13867+03	43	.63333+02	.14933+03	44	.64000+02	.16000+03
45	.80000+02	.00000	46	.80000+02	.30000+02	47	.80000+02	.60000+02	48	.80000+02	.90000+02
49	.81333+02	.12267+03	50	.84667+02	.16000+03	51	.10000+03	.00000	52	.10000+03	.28000+02
53	.10000+03	.56000+02	54	.10000+03	.84000+02	55	.10067+03	.11733+03	56	.10533+03	.16000+03
57	.12000+03	.00000	58	.12000+03	.86667+01	59	.12000+03	.17333+02	60	.12000+03	.26000+02
61	.12000+03	.34667+02	62	.12000+03	.43333+02	63	.12000+03	.52000+02	64	.12000+03	.60667+02
65	.12000+03	.69333+02	66	.12000+03	.78000+02	67	.12000+03	.89333+02	68	.12000+03	.10067+03
69	.12000+03	.11200+03	70	.12200+03	.12800+03	71	.12400+03	.14400+03	72	.12600+03	.16000+03
73	.13249+03	.00000	74	.13249+03	.24667+02	75	.13249+03	.49333+02	76	.13249+03	.73333+02
77	.13333+03	.10133+03	78	.13733+03	.16000+03	79	.14497+03	.00000	80	.14497+03	.23333+02
81	.14497+03	.46667+02	82	.14497+03	.68667+02	83	.14667+03	.90667+02	84	.14867+03	.16000+03
85	.15746+03	.00000	86	.15746+03	.73333+01	87	.15746+03	.14667+02	88	.15746+03	.22000+02
89	.15746+03	.29333+02	90	.15746+03	.36667+02	91	.15746+03	.44000+02	92	.15746+03	.50667+02
93	.15746+03	.57333+02	94	.15746+03	.64000+02	95	.15831+03	.69333+02	96	.15915+03	.74667+02
97	.16000+03	.80000+02	98	.16000+03	.10667+03	99	.16000+03	.13333+03	100	.16000+03	.16000+03
101	.16497+03	.00000	102	.16497+03	.21333+02	103	.16497+03	.42667+02	104	.16497+03	.62667+02
105	.16667+03	.80000+02	106	.17249+03	.00000	107	.17249+03	.20667+02	108	.17249+03	.41333+02
109	.17249+03	.61333+02	110	.17333+03	.80000+02	111	.18000+03	.00000	112	.18000+03	.66667+01
113	.18000+03	.13333+02	114	.18000+03	.20000+02	115	.18000+03	.26667+02	116	.18000+03	.33333+02
117	.18000+03	.40000+02	118	.18000+03	.46667+02	119	.18000+03	.53333+02	120	.18000+03	.60000+02
121	.18000+03	.66667+02	122	.18000+03	.73333+02	123	.18000+03	.80000+02	124	.19000+03	.00000
125	.19000+03	.20000+02	126	.19000+03	.40000+02	127	.19000+03	.60000+02	128	.19000+03	.80000+02
129	.20000+03	.00000	130	.20000+03	.20000+02	131	.20000+03	.40000+02	132	.20000+03	.60000+02
133	.20000+03	.80000+02	134	.21000+03	.00000	135	.21000+03	.66667+01	136	.21000+03	.13333+02
137	.21000+03	.20000+02	138	.21000+03	.26667+02	139	.21000+03	.33333+02	140	.21000+03	.40000+02
141	.21000+03	.46667+02	142	.21000+03	.53333+02	143	.21000+03	.60000+02	144	.21000+03	.66667+02
145	.21000+03	.73333+02	146	.21000+03	.80000+02	147	.22667+03	.00000	148	.22667+03	.20000+02
149	.22667+03	.40000+02	150	.22667+03	.60000+02	151	.22667+03	.80000+02	152	.24333+03	.00000
153	.24333+03	.20000+02	154	.24333+03	.40000+02	155	.24333+03	.60000+02	156	.24333+03	.80000+02
157	.26000+03	.00000	158	.26000+03	.66667+01	159	.26000+03	.13333+02	160	.26000+03	.20000+02
161	.26000+03	.26667+02	162	.26000+03	.33333+02	163	.26000+03	.40000+02	164	.26000+03	.46667+02
165	.26000+03	.53333+02	166	.26000+03	.60000+02	167	.26000+03	.66667+02	168	.26000+03	.73333+02
169	.26000+03	.80000+02	170	.28000+03	.00000	171	.28000+03	.20000+02	172	.28000+03	.40000+02
173	.28000+03	.60000+02	174	.28000+03	.80000+02	175	.30000+03	.00000	176	.30000+03	.20000+02
177	.30000+03	.40000+02	178	.30000+03	.60000+02	179	.30000+03	.80000+02	180	.32000+03	.00000
181	.32000+03	.66667+01	182	.32000+03	.13333+02	183	.32000+03	.20000+02	184	.32000+03	.26667+02
185	.32000+03	.33333+02	186	.32000+03	.40000+02	187	.32000+03	.46667+02	188	.32000+03	.53333+02
189	.32000+03	.60000+02	190	.32000+03	.66667+02	191	.32000+03	.73333+02	192	.32000+03	.80000+02

ELEMENT	NODES										
1	32	4	1	29	24	18	3	2	17	23	30	31
2	35	7	4	32	25	19	6	5	18	24	33	34
3	38	10	7	35	26	20	9	8	19	25	36	37
4	41	13	10	38	27	21	12	11	20	26	39	40
5	44	16	13	41	28	22	15	14	21	27	42	43
6	60	32	29	57	52	46	31	30	45	51	58	59
7	63	35	32	60	53	47	34	33	46	52	61	62

8	66	38	35	63	54	48	37	36	47	53	64	65
9	69	41	38	66	55	49	40	39	48	54	67	68
10	72	44	41	69	56	50	43	42	49	55	70	71
11	88	60	57	85	80	74	59	58	73	79	86	87
12	91	63	60	88	81	75	62	61	74	80	89	90
13	94	66	63	91	82	76	65	64	75	81	92	93
14	97	69	66	94	83	77	68	67	76	82	95	96
15	100	72	69	97	84	78	71	70	77	83	98	99
16	114	88	85	111	107	102	87	86	101	106	112	113
17	117	91	88	114	108	103	90	89	102	107	115	116
18	120	94	91	117	109	104	93	92	103	108	118	119
19	123	97	94	120	110	105	96	95	104	109	121	122
20	137	114	111	134	130	125	113	112	124	129	135	136
21	140	117	114	137	131	126	116	115	125	130	138	139
22	143	120	117	140	132	127	119	118	126	131	141	142
23	146	123	120	143	133	128	122	121	127	132	144	145
24	160	137	134	157	153	148	136	135	147	152	158	159
25	163	140	137	160	154	149	139	138	148	153	161	162
26	166	143	140	163	155	150	142	141	149	154	164	165
27	169	146	143	166	156	151	145	144	150	155	167	168
28	183	160	157	180	176	171	159	158	170	175	181	182
29	186	163	160	183	177	172	162	161	171	176	184	185
30	189	166	163	186	178	173	165	164	172	177	187	188
31	192	169	166	189	179	174	168	167	173	178	190	191

BOUNDARY CONDITIONS : 0 - CONSTRAINED
1 - UNCONSTRAINED

NODE	X,Y
17	0,1
23	0,1
29	0,1
45	0,1
51	0,1
57	0,1
73	0,1
79	0,1
85	0,1
101	0,1
111	0,1
124	0,1
129	0,1
134	0,1
147.	0,1
152	0,1
157	0,1
170	0,1
175	0,1
180	0,1
1	0,0
2	0,0
3	0,0
4	0,0
5	0,0
6	0,0
7	0,0
8	0,0
9	0,0
10	0,0
11	0,0

```
 12     0,0
 13     0,0
 14     0,0
 15     0,0
 16     0,0
 22     0,0
 28     0,0
 44     0,0
 50     0,0
 56     0,0
 72     0,0
 78     0,0
 84     0,0
100     0,0
```

LOADING PROGRAMME:

LOADING 1 : END LOADING 1 WHEN TOTAL LOAD AT NODE 180 IN Y-DIRECTION IS -.368187+02

: COMPONENTS OF PROPORTIONAL LOAD VECTOR 1

.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	-.6908+01
.0000	-.1372+02	.0000	-.1343+02	.0000	-.1295+02	.0000	-.1228+02	.0000	-.1142+02	.0000	-.1036+02
.0000	-.9112+01	.0000	-.7672+01	.0000	-.6040+01	.0000	-.4216+01	.0000	-.2200+01	.0000	-.2840+00

REQUESTED OUTPUT OF RESULTS :-

```
------------------------------

PRINT RESULTS FOR :-  (ELEMENT : INTEGRATION POINT)
 16: 1   16: 2   16: 3   17: 1   17: 2   17: 3   18: 1   18: 2   18: 3   19: 1   19: 2   19: 3

PRINT RESULTS FOR NODES :-
   1  29  57  85 111 134 157 180

RESULTS REQUESTED AFTER LOAD INCREMENTS :-
   1
```

RESULTS AFTER 1 LOAD INCREMENTS

NEXT POINTS TO UNDERGO PLASTIC DEFORMATION - (ELEMENT : INTEGRATION POINT) 19: 3

AT LOAD FACTOR .169503+01

LOAD VECTOR 1 : CUMULATIVE LOAD FACTOR = .169503+01 ; {Total = 1.695 × 110.59 = 187}

ELEMENT, INT. PT.		GLOBAL COORDS	STRAIN INCREMENTS ELASTIC	STRAIN INCREMENTS PLASTIC	LAMBDA	STRESS INCREMENT	CURRENT VALUES STRAINS ELASTIC	STRAINS PLASTIC	STRAINS TOTAL	STRESS	YIELD FUNCTION	STRESS CORRECTION FACTOR
16	X	.16000+03	.41871-03	--		.41871+00	.41871-03	.00000	.41871-03	.41871+00		
1	Y	.24540+01	.11398-03	--	--	.11398+00	.11398-03	.00000	.11398-03	.11398+00	-.10218+04	--
	XY		-.16708-02	--		-.83542+00	-.16708-02	.00000	-.16708-02	-.83542+00		
16	X	.16000+03	.10766-02	--		.10766+01	.10766-02	.00000	.10766-02	.10766+01		
2	Y	.10887+02	.69527-03	--	--	.69527+00	.69527-03	.00000	.69527-03	.69527+00	-.10211+04	--
	XY		-.16330-02	--		-.81650+00	-.16330-02	.00000	-.16330-02	-.81650+00		
16	X	.16000+03	.17393-02	--		.17393+01	.17393-02	.00000	.17393-02	.17393+01		
3	Y	.19321+02	.12859-02	--	--	.12859+01	.12859-02	.00000	.12859-02	.12859+01	-.10194+04	--
	XY		-.16953-02	--		-.84764+00	-.16953-02	.00000	-.16953-02	-.84764+00		
17	X	.16000+03	.22803-02	--		.22803+01	.22803-02	.00000	.22803-02	.22803+01		
1	Y	.24229+02	.14928-02	--	--	.14928+01	.14928-02	.00000	.14928-02	.14928+01	-.10177+04	--
	XY		-.17531-02	--		-.87655+00	-.17531-02	.00000	-.17531-02	-.87655+00		
17	X	.16000+03	.34601-02	--		.34601+01	.34601-02	.00000	.34601-02	.34601+01		
2	Y	.32662+02	.21809-02	--	--	.21809+01	.21809-02	.00000	.21809-02	.21809+01	-.10116+04	--
	XY		-.20594-02	--		-.10297+01	-.20594-02	.00000	-.20594-02	-.10297+01		
17	X	.16000+03	.47279-02	--		.47279+01	.47279-02	.00000	.47279-02	.47279+01		
3	Y	.41095+02	.29220-02	--	--	.29220+01	.29220-02	.00000	.29220-02	.29220+01	-.10017+04	--
	XY		-.26378-02	--		-.13189+01	-.26378-02	.00000	-.26378-02	-.13189+01		
18	X	.16000+03	.51523-02	--		.51523+01	.51523-02	.00000	.51523-02	.51523+01		
1	Y	.45803+02	.36947-02	--	--	.36947+01	.36947-02	.00000	.36947-02	.36947+01	-.99833+03	--
	XY		-.24525-02	--		-.12263+01	-.24525-02	.00000	-.24525-02	-.12263+01		
18	X	.16000+03	.53950-02	--		.53950+01	.53950-02	.00000	.53950-02	.53950+01		
2	Y	.53549+02	.44305-02	--	--	.44305+01	.44305-02	.00000	.44305-02	.44305+01	-.99648+03	--
	XY		-.18925-02	--		-.94624+00	-.18925-02	.00000	-.18925-02	-.94624+00		
18	X	.16000+03	.62002-02	--		.62002+01	.62002-02	.00000	.62002-02	.62002+01		
3	Y	.61295+02	.52294-02	--	--	.52294+01	.52294-02	.00000	.52294-02	.52294+01	-.98722+03	--
	XY		-.21331-02	--		-.10666+01	-.21331-02	.00000	-.21331-02	-.10666+01		
19	X	.16025+03	.84958-02	--		.84958+01	.84958-02	.00000	.84958-02	.84958+01		
1	Y	.65403+02	.71563-02	--	--	.71563+01	.71563-02	.00000	.71563-02	.71563+01	-.95876+03	--
	XY		-.18784-02	--		-.93922+00	-.18784-02	.00000	-.18784-02	-.93922+00		
19	X	.16113+03	.14963-01	--		.14963+02	.14963-01	.00000	.14963-01	.14963+02		
2	Y	.71775+02	.84210-02	--	--	.84210+01	.84210-02	.00000	.84210-02	.84210+01	-.75187+03	--
	XY		-.11738-01	--		-.58691+01	-.11738-01	.00000	-.11738-01	-.58691+01		
19	X	.16200+03	.26395-01	.00000		.26395+02	.26395-01	.00000	.26395-01	.26395+02		

3	Y	.78146+02	.10272-01	.00000	.00000	.10272+02	.10272-01	.00000	.10272-01	.10272+02	-.61035-04	.00000
	XY		-.25636-01	.00000		-.12818+02	-.25636-01	.00000	-.25636-01	-.12818+02		

CURRENT PLASTIC POINTS - (ELEMENT : INTEGRATION POINT)

19: 3

NODE	LOAD INCREMENT		TOTAL LOAD		DISPLACEMENT INCREMENT		TOTAL DISPLACEMENT	
	DPX	DPY	PX	PY	DU	DV	U	V
1	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000
29	.000000	.000000	.000000	.000000	.000000	-.194631-01	.000000	-.194631-01
57	.000000	.000000	.000000	.000000	.000000	-.183088+00	.000000	-.183088+00
85	.000000	.000000	.000000	.000000	.000000	-.493479+00	.000000	-.493479+00
111	.000000	.000000	.000000	.000000	.000000	-.821990+00	.000000	-.821990+00
134	.000000	.000000	.000000	.000000	.000000	-.141792+01	.000000	-.141792+01
157	.000000	.000000	.000000	.000000	.000000	-.268781+01	.000000	-.268781+01
185	.000000	-.117093+02	.000000	-.117093+02	.000000	-.442982+01	.000000	-.442982+01

RESULTS AFTER 18 LOAD INCREMENTS

END OF LOADING 1 : MAXIMUM LOAD REACHED AT NODE 180

AT LOAD FACTOR .238755-01

LOAD VECTOR 1 : CUMULATIVE LOAD FACTOR = .532986+01 ⇒ $P = 5.32986 \times 110.59 = 589.429$ ✓

ELEMENT, INT. PT.		GLOBAL COORDS	STRAIN INCREMENTS ELASTIC	STRAIN INCREMENTS PLASTIC	LAMBDA	STRESS INCREMENT	CURRENT VALUES STRAINS ELASTIC	CURRENT VALUES STRAINS PLASTIC	CURRENT VALUES STRAINS TOTAL	CURRENT VALUES STRESS	CURRENT VALUES YIELD FUNCTION	STRESS CORRECTION FACTOR
16	X	.16000+03	.20462-04	--		.20462-01	.21930-02	.00000	.21930-02	.21930+01		
1	Y	.24540+01	.87582-05	--	--	.87582-02	.31892-03	.00000	.31892-03	.31892+00	-.92622+03	--
	XY		-.16041-03	--		-.80207-01	-.11169-01	.00000	-.11169-01	-.55847+01		
16	X	.16000+03	.32409-04	--		.32409-01	.53445-02	.00000	.53445-02	.53445+01		
2	Y	.10887+02	.57823-04	--	--	.57823-01	.38193-02	.00000	.38193-02	.38193+01	-.90545+03	--
	XY		-.14327-03	--		-.71633-01	-.11302-01	.00000	-.11302-01	-.56512+01		
16	X	.16000+03	.49681-04	--		.49681-01	.86748-02	.00000	.86748-02	.86748+01		
3	Y	.19321+02	.10874-03	--	--	.10874+00	.74779-02	.00000	.74779-02	.74779+01	-.84160+03	--
	XY		-.14290-03	--		-.71449-01	-.12442-01	.00000	-.12442-01	-.62210+01		
17	X	.16000+03	.13654-03	--		.13654+00	.11567-01	.00000	.11567-01	.11567+02		
1	Y	.24229+02	.11872-03	--	--	.11872+00	.89263-02	.00000	.89263-02	.89263+01	-.81060+03	--
	XY		-.93121-04	--		-.46561-01	-.11729-01	.00000	-.11729-01	-.58645+01		
17	X	.16000+03	.39460-03	--		.39460+00	.17775-01	.00000	.17775-01	.17775+02		
2	Y	.32662+02	.15335-03	--	--	.15335+00	.13021-01	.00000	.13021-01	.13021+02	-.62600+03	--
	XY		-.24740-03	--		-.12370+00	-.13854-01	.00000	-.13854-01	-.69272+01		
17	X	.16000+03	.71227-03	--		.71227+00	.25679-01	.00000	.25679-01	.25679+02		
3	Y	.41095+02	.19784-03	--	--	.19784+00	.17619-01	.00000	.17619-01	.17619+02	-.24581+03	--
	XY		-.47579-03	--		-.23790+00	-.18647-01	.00000	-.18647-01	-.93234+01		
18	X	.16000+03	.11607-03	.91274-03		.11607+00	.33071-01	.42495-02	.37320-01	.33056+02		
1	Y	.45803+02	.19953-04	.25685-03	.13495-02	.19953-01	.22594-01	.12421-02	.23837-01	.22584+02	-.15259-04	.99997+00
	XY		.23133-03	-.96025-03		.11567+00	-.14965-01	-.47683-02	-.19734-01	-.74792+01		
18	X	.16000+03	.79403-04	.17168-02		.79403-01	.33935-01	.33561-01	.67496-01	.33814+02		
2	Y	.53549+02	-.33168-05	.28814-03	.23553-02	-.33168-02	.20858-01	.64846-02	.27342-01	.20782+02	-.30518-04	.99999+00
	XY		.17062-03	-.15866-02		.85311-01	-.14274-01	-.40054-01	-.54328-01	-.71055+01		
18	X	.16000+03	.51300-04	.25953-02		.51300-01	.34624-01	.71654-01	.10628+00	.34334+02		
3	Y	.61295+02	-.15736-04	.21378-03	.33714-02	-.15736-01	.19328-01	.96121-02	.28940-01	.19160+02	-.30518-04	.99999+00
	XY		.12121-03	-.21412-02		.60603-01	-.13609-01	-.81751-01	-.95360-01	-.67306+01		
19	X	.16025+03	.32235-04	.34293-02		.32235-01	.33814-01	.12527+00	.15908+00	.33625+02		
1	Y	.65403+02	-.42535-04	.41506-03	.47453-02	-.42535-01	.19746-01	.16565-01	.36312-01	.19629+02	-.45776-04	.10000+01
	XY		.57088-04	-.32536-02		.28544-01	-.15071-01	-.13062+00	-.14569+00	-.74848+01		
19	X	.16113+03	-.10543-05	.58695-02		-.10543-02	.35094-01	.27602+00	.31112+00	.34213+02		
2	Y	.71775+02	-.19160-04	.38532-03	.76556-02	-.19160-01	.19247-01	.21695-01	.40942-01	.18719+02	-.45776-04	.10000+01
	XY		-.55604-05	-.48999-02		-.27802-02	-.14338-01	-.26824+00	-.28258+00	-.69164+01		

19	X	.16200+03	-.31533-05	.84248-02		-.31533-02	.36005-01	.95058+00	.48658+00	.34573+02		
3	Y	.78146+02	-.21338-04	.42314-03	.10630-01	-.21338-01	.19381-01	.28649-01	.48031-01	.18537+02	-.30518-04	.10000+01
	XY		-.11030-04	-.64681-02		-.55149-02	-.13844-01	-.41195+00	-.42579+00	-.64794+01		

CURRENT PLASTIC POINTS - (ELEMENT : INTEGRATION POINT)

13: 7   13: 8   13: 9   14: 7   14: 8   18: 1   18: 2   18: 3   18: 5   18: 6   18: 9   19: 1   19: 2   19: 3   19: 4

19: 5   19: 6   19: 7   19: 9   23: 3

	LOAD INCREMENT		TOTAL LOAD		DISPLACEMENT INCREMENT		TOTAL DISPLACEMENT	
NODE	DPX	DPY	PX	PY	DU	DV	U	V
1	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000
29	.000000	.000000	.000000	.000000	.000000	-.249011-03	.000000	-.576342-01
57	.000000	.000000	.000000	.000000	.000000	-.452886-02	.000000	-.677289+00
85	.000000	.000000	.000000	.000000	.000000	-.186929-01	.000000	-.211922+01
111	.000000	.000000	.000000	.000000	.000000	-.366278-01	.000000	-.373671+01
134	.000000	.000000	.000000	.000000	.000000	-.653602-01	.000000	-.649844+01
157	.000000	.000000	.000000	.000000	.000000	-.116072+00	.000000	-.119123+02
180	.000000	-.164932+00	.000000	-.368187+02	.000000	-.179059+00	.000000	-.190412+02

A.5 EPTCS Program Listing

COL*EPTCS(1).MAIN

```
      COMPILER (XM=3)
C
C  SET VALUES OF PARAMETERS
C
      PARAMETER  NDFP = 194
      PARAMETER  NECP = 24
      PARAMETER  NPCP = 49
      PARAMETER  NNP  = 97
      PARAMETER  NEP  = 158
      PARAMETER  ISP  = 3
      PARAMETER  NLPP = 2
C
      COMMON /EXT/ RKE(NDFP,NECP),RKP(NDFP,NPCP)
      COMMON IA,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPCS,NPC,NDFBWS,
     2LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,RMIN,IS,
     3NLT,NWRKE,NWRKP,ETA,IQ,NNEWP,NOUT,LU1,LU2,LU3,FRAC
C
C  SET LOGICAL UNIT NUMBERS OF TEMPORARY FILES
C
      LU1  =  11
      LU2  =  12
      LU3  =  13
C
C  INITIALIZE ARRAYS
C
      DIMENSION PP(NLPP,NDFP),COORD(NNP,2),IELT(NEP,3),IBC(NDFP),
     2PMAX(NLPP,2),IOUT(100),NELTPR(NEP),NODEPR(NNP),U(NDFP),
     3P(NDFP),ESTRN(NEP,ISP),PSTRN(NEP,ISP),STRS(NEP,ISP),QQ(NPCP),
     4SIGBAR(NEP,ISP),IPELTS(NEP,3),RLAM(NEP),A(NEP,ISP),RK3(NPCP,NPCP),
     5CE(NEP),DP(NDFP),DU(NDFP),DESTRN(NEP,ISP),IPPT(NEP),RMINC(NLPP),
     6DPSTRN(NEP,ISP),DSTRN(NEP,ISP),DSTRS(NEP,ISP),STRN(NEP,ISP),
     7PHI(NEP),D(4,4),VOL(NEP),B(NEP,ISP,6),DIAG(NPCP),DIAG2(NPCP),
     8BTD(NEP,6,ISP),ITRI(NEP,6),BTDN(6)
C
C  READ STRUCTURE DATA
C
      CALL DATA (NEP,NNP,NDFP,ISP,NECP,NPCP,NLPP,
     $PP,COORD,IELT,IBC,PMAX,IOUT,NELTPR,NODEPR,ITRI)
C
C  CALCULATE AND STORE ELASTIC STIFFNESS MATRIX  (BTDB)  IN TEMPORARY FILE
C
      CALL ELTMAT (NEP,NNP,NDFP,ISP,NECP,NPCP,NLPP,
     $PP,COORD,IELT,IBC,D,VOL,B,BTD,ITRI)
C
C  INITIALIZE COUNTERS FOR FIRST LOAD INCREMENT: LINEAR ELASTIC ANALYSIS
C
      LP=1
      NOUT=1
      NPPTS=0
      NPC=0
C
C  SOLVE FOR DISPLACEMENT INCREMENTS
C
    3 CALL SOLVE1 (NDFP,NPCP,NLPP,PP,DU,DIAG,QQ,RK3)
C
C  DETERMINE PLASTIC ELEMENTS
C
    1 CALL STRSTR (NEP,NNP,NDFP,ISP,NECP,NPCP,NLPP,
     $P,U,ESTRN,PSTRN,STRN,STRS,SIGBAR,PP,IELT,PMAX,IOUT,
```

```
     $NELTPR,NODEPR,D,IPELTS,RLAM,A,CF,DP,VOL,B,BTD,RMINC,
     $DU,DESTRN,DPSTRN,DSTRN,DSTRS,IPPT,PHI,ITRI)
      IF(ISTOP.EQ.-1) GO TO 2
      IF(NPPTS.EQ.0) GO TO 3
C
C  EVALUATE PLASTIC MATRICES  (-BTDN)  AND  (NTDN+DP)
C
      DO 4 J=1,NPC
      DIAG(J)=0.
      DO 4 I=1,NDF
    4 RKP(I,J)=0.
      DO 5 IE=1,NE
      IF(IPELTS(IE,1).EQ.0) GO TO 5
      CALL PLASM (NEP,TSP,IE,A,SIGBAR,VOL,D,BTDN,ENTDN,BTD)
      NC=IPELTS(IE,2)
      DO 6 I=1,6
      L=ITRI(IE,I)
    6 RKP(L,NC)=-BTDN(I)
      DIAG(NC)=ENTDN
    5 CONTINUE
C
C  APPLY BOUNDARY CONDITIONS TO PLASTIC MATRIX  (-BTDN)
C
      DO 7 I=1,NDF
      IF(IBC(I).EQ.1) GO TO 7
      DO 8 J=1,NPC
    8 RKP(I,J)=0.
    7 CONTINUE
C
C  SOLVE, CHECK KINEMATIC CONSTRAINTS ON ELASTIC AND PLASTIC REGIONS
C  AND ITERATE IF NECESSARY
C
      CALL ITER (NEP,NNP,NDFP,TSP,NECP,NPCP,NLPP,
     $TELT,IBC,D,IPELTS,RLAM,A,PP,DU,SIGBAR,VOL,B,ITRI,RMINC,
     $DQ,ESTRN,U,P,IPPT,DSTRS,DSTRN,DPSTRN,DESTRN,DP,CF,PMAX,RK3,
     $IOUT,NELTPR,NODEPR,PSTRN,STRN,STRS,PHI,DIAG,DIAG2)
C
C  APPLY NEXT LOAD INCREMENT
C
      GO TO 1
    2 STOP
      END
```

COL*EPTCS(1).DATA

```
      SUBROUTINE DATA (NEP,NNP,NDFP,ISP,NECP,NPCP,NLPP,
     $PP,COORD,IELT,IBC,PMAX,IOUT,NELTPR,NODEPR,ITRI)
      COMMON IA,NN,NF,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPCS,NPC,NDFBWS,
     2IP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,RMIN,IS,
     3NLT,NWRKE,NWRKP,ETA,IU,NNEWP,NOUT,LU1,LU2,LU3,FRAC
      DIMENSION PP(NLPP,NDFP),COORD(NNP,2),IELT(NEP,3),IBC(NDFP),
     2PMAX(NLPP,2),IOUT(100),NELTPR(NEP),NODEPR(NNP),
     3ITRI(NEP,6),PPP(2),TITLE(12),IIBC(2)
C
C  READ AND PRINT HEADING
C
      READ 101,TITLE
  101 FORMAT(12A6)
      PRINT 201,TITLE
      READ 101,TITLE
  201 FORMAT(1H1,12A6,/)
  100 FORMAT()
C
C  READ TYPE OF ANALYSIS
C
      READ 104,I,J
  104 FORMAT(2A6)
      IF(J.EQ.'STRESS') IA=-1
      IF(J.EQ.'STRAIN') IA=0
      IF(J.EQ.'METRIC') IA=1
      IF(IA.NE.0) GO TO 13
      IF(J.EQ.'STRAIN') GO TO 13
      PRINT 223
  223 FORMAT(1H0,38('*'),/,' TYPE OF ANALYSIS INCORRECTLY SPECIFIED',/,
     $' ANALYSIS TERMINATED',/,39('*'))
      STOP
C
C  READ CONSTANTS :  NUMBER OF NODES,  NUMBER OF ELEMENTS,
C  ELASTIC MODULUS,  PLASTIC MODULUS,  POISSON'S RATIO,
C  UNIAXIAL YIELD STRESS,  RATIO OF VON MISES EQUIVALENT
C  STRESS TO YIELD STRESS FOR STRESS POINT TO BE TREATED
C  AS PLASTIC,  PLATE THICKNESS IF PLANE STRESS ANALYSIS
C
   13 IF(IA.EQ.-1) READ 100,NN,NE,E,EP,RNU,SZERO,ETA,THIK,FRAC
      IF(IA.GT.-1) READ 100,NN,NE,E,EP,RNU,SZERO,ETA,FRAC
      IF(IA.EQ.0) THIK=1.
      IS=4
      IF(IA.EQ.-1) IS=3
      EPS=EP*EP
      SZEROS=SZERO*SZERO
C
C  READ NODAL COORDINATES
C
      READ 100,((COORD(I,J),J=1,2),I=1,NN)
C
C  READ ELEMENT INCIDENCES
C
      READ 100,((IELT(I,J),J=1,3),I=1,NE)
C
C  DETERMINE CONSTANTS FOR SOLUTION ROUTINE AND HALF BAND-WIDTH OF ELASTIC MATRIX
C
      NBW=0
      DO 12 I=1,NE
      J1=ABS(IELT(I,1)-IELT(I,2))
```

```
      J2=ABS(IELT(I,1)-IELT(I,3))
      J3=ABS(IELT(I,2)-IELT(I,3))
      MAX=MAX0(J1,J2,J3)
   12 IF(NBW.LT.MAX) NBW=MAX
      NBW=(NBW+1)*2
      NDF=NN*2
      NDFS=NDF-1
      NDFA=NDF+1
      NBWS=NBW-1
      NDFBWS=NDF-NBWS
      NWRKF=NDF*NBW
C
C  PRINT HEADINGS AND VALUES OF CONSTANTS
C
      PRINT 212
  212 FORMAT(//,' NUMERICAL ANALYSIS USING TRIANGULAR CONSTANT STRAIN FI
     $NITE ELEMENTS',//,' PLANE STRESS, PLANE STRAIN OR AXISYMMETRIC PRO
     $GRAM',/)
      IF(IA.EQ.-1) PRINT 215
  215 FORMAT(1H0,'PLANE STRESS ANALYSIS',/)
      IF(IA.EQ.0) PRINT 222
  222 FORMAT(1H0,'PLANE STRAIN ANALYSIS',/)
      IF(IA.EQ.1) PRINT 203
  203 FORMAT(1H0,'AXISYMMETRIC ANALYSIS',//,' VARYING CIRCUMFERENTIAL ST
     $RAINS AND STRESSES AVERAGED AT CENTROID OF ELEMENT SECTION',/)
      PRINT 211
  211 FORMAT(1H0,'VON MISES YIELD CONDITION',/)
      IF(EP.NE.0.0) PRINT 214
  214 FORMAT(1H0,'KINEMATIC HARDENING',/)
      IF(EP.EQ.0.0) PRINT 221,FRAC
  221 FORMAT(1H0,'ELASTIC, PERFECTLY PLASTIC :  ANALYSIS TERMINATED WHEN
     2 LOAD INCREMENT MAGNITUDE DECREASES TO',F6.3,'  OF TOTAL LOAD',/)
      PRINT 247,ETA
  247 FORMAT(1H0,'ELEMENTS WITH VON MISES EQUIVALENT STRESS WITHIN',
     $F6.3,'  OF YIELD STRESS TREATED AS PLASTIC ELEMENTS',/)
      PRINT 220, TITLE
  220 FORMAT(1H0,12A6,/)
      I=NLPP-1
      PRINT 213,NNP,NEP,NDFP,NFCP,NPCP,I
  213 FORMAT(1H0,'STORAGE AVAILABLE  :',19X,'MAXIMUM NUMBER OF NODES :'
     $,I5,/,20X,':',16X,'MAXIMUM NUMBER OF ELEMENTS :',I5,/,20X,
     $':',6X,'MAXIMUM NUMBER OF DEGREES OF FREEDOM :',I5,/,20X,
     $': MAXIMUM HALF BAND-WIDTH OF ELASTIC MATRIX :',I5,/,20X,':',8X,
     $'MAXIMUM NUMBER OF PLASTIC ELEMENTS :',I5,/,20X,': MAXIMUM NUMBER
     $OF PROPORTIONAL LOAD PATHS :',I5,/)
      PRINT 202,NN,NE,NDF,NBW
  202 FORMAT(1H0,'STORAGE REQUIRED   :',27X,'NUMBER OF NODES :',I5,
     $/,20X,':',24X,'NUMBER OF ELEMENTS :',I5,/,20X,':',14X,
     $'NUMBER OF DEGREES OF FREEDOM :',I5,/,20X,':',9X,
     $'HALF BAND-WIDTH OF ELASTIC MATRIX :',I5,/)
      PRINT 217,E,EP,RNU,SZERO
  217 FORMAT(1H0,'MATERIAL CONSTANTS : ELASTIC MODULUS  =',E11.6,
     $/,20X,': PLASTIC MODULUS  =',E11.6,/,20X,': POISSONS RATIO   =',
     $E11.6,/,20X,': YIELD STRESS      =',E11.6,/)
      IF(IA.EQ.-1) PRINT 216,THIK
  216 FORMAT(1H0,'PLATE THICKNESS     :',E11.6,/)
C
C  CHECK AVAILABLE STORAGE
C
      IF(NDF.LE.NDFP) GO TO 52
```

```
      PRINT 231
  231 FORMAT(1H0,60('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO FEW RO
     $WS IN ELASTIC MATRIX',/,61('*'))
      STOP
   52 IF(NBW.LE.NECP) GO TO 53
      PRINT 232
  232 FORMAT(1H0,63('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO FEW CO
     $LUMNS IN ELASTIC MATRIX',/,64('*'))
      STOP
   53 IF(NN.LE.NNP) GO TO 54
      PRINT 233
  233 FORMAT(1H0,44('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO MANY N
     $ODES',/,45('*'))
      STOP
   54 IF(NE.LE.NEP) GO TO 55
      PRINT 234
  234 FORMAT(1H0,47('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO MANY E
     $LEMENTS',/,48('*'))
      STOP
   55 IF(IS.LE.ISP) GO TO 10
      PRINT 246
  246 FORMAT(1H0,65('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO MANY R
     $OWS IN DEFORMATION MATRIX',/,66('*'))
      STOP
C
C  PRINT NODAL COORDINATES
C
   10 IF(IA.LT.1) PRINT 204
  204 FORMAT(1H0,'COORDINATES OF NODES',//,' NODE',6X,'X',11X,'Y',
     $3(10X,'NODE',6X,'X',11X,'Y'))
      IF(IA.EQ.1) PRINT 224
  224 FORMAT(1H0,'COORDINATES OF NODES',//,' NODE',6X,'R',11X,'Z',
     $3(10X,'NODE',6X,'R',11X,'Z'))
      PRINT 205,((I,(COORD(I,J),J=1,2)),I=1,NN)
  205 FORMAT(500(1H ,I3,2E12.5,3(6X,I3,2E12.5),/))
C
C  PRINT ELEMENT INCIDENCES
C
      PRINT 206
  206 FORMAT(//,' ELEMENT',7X,'NODES',3(14X,'ELEMENT',7X,'NODES'),
     $/,10X,'/',13('-'),'/',3(18X,'/',13('-'),'/'),/)
      PRINT 207,((I,(IELT(I,J),J=1,3)),I=1,NE)
  207 FORMAT(500(1H ,I4,1X,3I6,3(10X,I4,1X,3I6),/))
C
C  READ AND PRINT BOUNDARY CONDITIONS
C
      IF(IA.LT.1) PRINT 208
  208 FORMAT(//,' BOUNDARY CONDITIONS :  0  -  CONSTRAINED',
     $/,24X,'1  -  UNCONSTRAINED',/,' NODE',3X,'X,Y',/)
      IF(IA.EQ.1) PRINT 225
  225 FORMAT(//,' BOUNDARY CONDITIONS :  0  -  CONSTRAINED',
     $/,24X,'1  -  UNCONSTRAINED',/,' NODE',3X,'R,Z',/)
      DO 1 I=1,NDF
    1 IBC(I)=1
    7 READ 100,I
      IF(I.EQ.-1) GO TO 2
      READ(0,100) IIBC(1),IIBC(2)
      READ 100,I
      K=1
    3 K=K+1
```

```
      READ(0,100) (IOUT(J),J=1,K)
      IF(IOUT(K).NE.-1) GO TO 3
      K=K-1
      DO 4 I=1,K
      PRINT 209,IOUT(I),(IIBC(J),J=1,2)
  209 FORMAT(1H ,I3,4X,I1,',',I1)
      II=(IOUT(I)-1)*2
      DO 4 J=1,2
    4 IBC(II+J)=IIBC(J)
      GO TO 7
C
C READ AND PRINT LOADING PROGRAMME
C
    2 PRINT 210
  210 FORMAT(//,' LOADING PROGRAMME:')
    9 READ 100,I
      IK=IK+1
      IF(IK.LE.NLPP) GO TO 14
      PRINT 248
  248 FORMAT(1H0,49('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO MANY L
     $OAD PATHS',/,50('*'))
      STOP
   14 IF(I.LE.-1) GO TO 5
      READ(0,100) IJ,I,J,PMAX(IJ,2)
      IF(IA.LT.1.AND.J.EQ.1) JP='X'
      IF(IA.LT.1.AND.J.EQ.2) JP='Y'
      IF(IA.EQ.1.AND.J.EQ.1) JP='R'
      IF(IA.EQ.1.AND.J.EQ.2) JP='Z'
      PRINT 218,IJ,IA,I,JP,PMAX(IJ,2)
  218 FORMAT(//,' LOADING',I4,'  : END LOADING',I3,
     $'  WHEN TOTAL LOAD AT NODE',I5,'  IN ',A1,'-DIRECTION IS  'E11.6)
      L=(I-1)*2+J
      PMAX(IJ,1)=FLOAT(L)
   19 READ 100,I
      IF(I.EQ.-1) GO TO 21
      READ(0,100) N1,N2
      READ 100,PPP(1),PPP(2)
      DO 11 I=N1,N2
   11 PRINT 219,I,PPP(1),PPP(2)
  219 FORMAT(15X,I3,1X,2(2X,E11.6))
      JJ=2*(N1-1)
      II=N2-N1+1
      DO 16 I=1,II
      DO 16 J=1,2
      JJ=JJ+1
   16 PP(IJ,JJ)=PP(IJ,JJ)+PPP(J)
      GO TO 19
C
C APPLY BOUNDARY CONDITIONS TO LOAD VECTOR
C
   21 DO 6 I=1,NDF
    6 IF(IBC(I).EQ.0) PP(IJ,I)=0.
C
C PRINT RELATIVE MAGNITUDES OF COMPONENTS OF LOAD VECTOR
C
      IF(IQ.EQ.1) GO TO 9
      PRINT 237,IJ
  237 FORMAT(/,14X,': COMPONENTS OF PROPORTIONAL LOAD VECTOR',I3,
     $/,16X,38('-'),/)
      PRINT 238,(PP(IJ,I),I=1,NDF)
```

```
  238 FORMAT(1H ,200(12E10.4,/,1H ))
      GO TO 9
    5 PMAX(IK,1)=-1
C
C  READ AND PRINT OUTPUT REQUESTED
C
      PRINT 239
  239 FORMAT(/,1H0,'REQUESTED OUTPUT OF RESULTS :-',/,30('-'),/)
      READ 103,JP
  103 FORMAT(A2)
      IF(JP.NE.'AL') GO TO 23
      DO 28 IE=1,NF
   28 NELTPR(IE)=1
      DO 29 I=1,NN
   29 NODEPR(I)=1
      PRINT 240
  240 FORMAT(' PRINT RESULTS FOR ALL ELEMENTS AND ALL NODES',/)
      GO TO 30
   23 READ 100,I
      IF(I.EQ.-1) GO TO 26
      K=0
   33 K=K+1
      READ(0,100) (IOUT(J),J=1,K)
      IF(IOUT(K).NE.-1) GO TO 33
      K=K-1
      DO 25 I=1,K
      J=IOUT(I)
   25 NELTPR(J)=1
      PRINT 241
  241 FORMAT(1H ,'PRINT RESULTS FOR ELEMENTS :-')
      PRINT 244,(IOUT(J),J=1,K)
   26 READ 100,I
      IF(I.EQ.-1) GO TO 30
      K=0
   31 K=K+1
      READ(0,100) (IOUT(J),J=1,K)
      IF(IOUT(K).NE.-1) GO TO 31
      K=K-1
      DO 27 I=1,K
      J=IOUT(I)
   27 NODEPR(J)=1
      PRINT 243
  243 FORMAT(1H0,'PRINT RESULTS FOR NODES :-')
      PRINT 244,(IOUT(J),J=1,K)
  244 FORMAT(1H ,33I4)
   30 READ 100,I
      K=1
   32 K=K+1
      READ(0,100) (IOUT(J),J=1,K)
      IF(IOUT(K).NE.-2) GO TO 32
      PRINT 245
  245 FORMAT(1H0,'RESULTS PRINTED AT END OF EACH PROPORTIONAL LOAD PATH
     $AND AFTER LOAD INCREMENTS :-')
      K=K-1
      IF(IOUT(K).EQ.-1) K=K-1
      PRINT 244,(IOUT(J),J=1,K)
      RETURN
      END
```

COL*EPTCS(1).ELTMAT

```
      COMPILER (XM=3)
      SUBROUTINE ELTMAT (NEP,NNP,NDFP,ISP,NECP,NPCP,NLPP,
     $PP,COORD,IELT,IBC,D,VOL,B,BTD,ITRI)
C
C  FOR EXTENDED STORAGE GIVE EXPLICIT SIZE OF ELASTIC AND PLASTIC ARRAYS
C
       COMMON /EXT/ RKE(194,24) , RKP(194,40)
C
      COMMON IA,NN,NE,NPPIS,NDFS,NDF,NDFA,NBWS,NBW,NPCS,NPC,NDFBWS,
     2LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,RMIN,IS,
     3NLI,NWRKE,NWRKP,ETA,IU,NNEWP,NOUT,LU1,LU2,LU3,FRAC
      DIMENSION PP(NLPP,NDFP),COORD(NNP,2),IELT(NEP,3),IBC(NDFP),
     2D(4,4),VOL(NEP),B(NEP,ISP,6),BTD(NEP,6,ISP),ITRI(NEP,6),
     3AI(2),AJ(2),AL(2),BTDB(6,6)
      DOUBLE PRECISION FACT
      IF(IA.GT.-1) GO TO 18
C
C  IF PLANE STRESS ANALYSIS, EVALUATE ELEMENT ELASTICITY MATRIX  (D')
C
      DUM=E/(1.-RNU*RNU)
      D(1,1)=DUM
      D(1,2)=RNU*DUM
      D(2,2)=DUM
      D(2,1)=D(1,2)
      D(3,3)=(1.-RNU)/2.*DUM
      GO TO 17
C
C  IF PLANE STRAIN OR AXISYMMETRIC ANALYSIS, EVALUATE ELASTICITY MATRIX  (D')
C
   18 DUM=E*(1.-RNU)/(1.+RNU)/(1.-2.*RNU)
      D(1,1)=DUM
      D(1,2)=RNU/(1.-RNU)*DUM
      D(1,4)=D(1,2)
      D(2,1)=D(1,2)
      D(2,2)=DUM
      D(2,4)=D(1,2)
      D(3,3)=E/2./(1.+RNU)
      D(4,1)=D(1,2)
      D(4,2)=D(1,2)
      D(4,4)=DUM
C
C  EVALUATE ELEMENT DEFORMATION MATRICES   (B')
C ***********************************************
C
   17 DO 2 IE=1,NE
      I=IELT(IE,1)
      J=IELT(IE,2)
      L=IELT(IE,3)
      DO 1 JJ=1,2
      AI(JJ)=COORD(I,JJ)
      AJ(JJ)=COORD(J,JJ)
    1 AL(JJ)=COORD(L,JJ)
      VOL(IE)=ABS(AJ(1)*AL(2)-AL(1)*AJ(2)-AI(1)*AL(2)
     $+AL(1)*AI(2)+AI(1)*AJ(2)-AJ(1)*AI(2))
      DUM=1./VOL(IE)
C
C  FOR PLANE STRESS, PLANE STRAIN OR AXISYMMETRIC ANALYSIS :-
C
      B(IE,1,1)=AJ(2)-AL(2)
```

```
      B(IE,1,3)=AL(2)-AI(2)
      B(IE,1,5)=AI(2)-AJ(2)
      B(IE,2,2)=AL(1)-AJ(1)
      B(IE,2,4)=AI(1)-AL(1)
      B(IE,2,6)=AJ(1)-AI(1)
      B(IE,3,1)=B(IE,2,2)
      B(IE,3,2)=B(IE,1,1)
      B(IE,3,3)=B(IE,2,4)
      B(IE,3,4)=B(IE,1,3)
      B(IE,3,5)=B(IE,2,6)
      B(IE,3,6)=B(IE,1,5)
      IF(IA.EQ.1) GO TO 11
      DO 3 I=1,3
      DO 3 J=1,6
    3 B(IE,I,J)=B(IE,I,J)*DUM
      VOL(IE)=VOL(IE)/2.*THIK
      GO TO 12
C
C  IF AXISYMMETRIC ANALYSIS, EVALUATE FOURTH ROW OF ELEMENT
C  DEFORMATION MATRIX, AND EVALUATE ELEMENT VOLUME
C
   11 RB=(AI(1)+AJ(1)+AL(1))/3.
      ZB=(AI(2)+AJ(2)+AL(2))/3.
      B(IE,4,1)=(AJ(1)*AL(2)-AL(1)*AJ(2)+B(IE,2,2)*ZB)/RB+B(IE,1,1)
      B(IE,4,3)=(AL(1)*AI(2)-AI(1)*AL(2)+B(IE,2,4)*ZB)/RB+B(IE,1,3)
      B(IE,4,5)=(AI(1)*AJ(2)-AJ(1)*AI(2)+B(IE,2,6)*ZB)/RB+B(IE,1,5)
      DO 15 I=1,4
      DO 15 J=1,6
   15 B(IE,I,J)=B(IE,I,J)*DUM
      VOL(IE)=VOL(IE)*RB*3.1415927
C
C  EVALUATE ELEMENT STIFFNESS
C
   12 DO 4 I=1,6
      DO 4 J=1,IS
      DO 4 L=1,IS
    4 BTD(IE,I,J)=BTD(IE,I,J)+B(IE,L,I)*D(L,J)
      DO 5 I=1,6
      DO 5 J=I,6
      BTDB(I,J)=0.
      DO 5 L=1,IS
    5 BTDB(I,J)=BTDB(I,J)+BTD(IE,I,L)*B(IE,L,J)
      DO 6 I=1,6
      DO 6 J=I,6
      BTDB(I,J)=BTDB(I,J)*VOL(IE)
    6 BTDB(J,I)=BTDB(I,J)
C
C  ADD INTO SYSTEM ELASTIC MATRIX  (BTDB)
C
      K=1
      DO 7 I=1,3
      L=(IELT(IE,I)-1)*2
      DO 7 J=1,2
      ITRI(IE,K)=L+J
    7 K=K+1
      DO 16 I=1,6
      NR=ITRI(IE,I)
      NRS=NR-1
      DO 16 J=1,6
      NC=ITRI(IE,J)
```

```
      IF(NC.LT.NR) GO TO 16
      NC=NC-NRS
      RKE(NR,NC)=RKE(NR,NC)+BTDB(I,J)
   16 CONTINUE
    2 CONTINUE
C
C  APPLY BOUNDARY CONDITIONS TO SYSTEM ELASTIC MATRIX  (BTDB)
C
      DO 8 I=1,NDF
      IF(IBC(I).EQ.1) GO TO 8
      DO 9 J=2,NBW
    9 RKE(I,J)=0.
      RKE(I,1)=1.
      IF(I.EQ.1) GO TO 8
      II=I-1
      IK=2
      JJ=I-NBW+1
      IF(JJ.LT.1) JJ=1
      DO 10 IJ=II,JJ,-1
      RKE(IJ,IK)=0.
   10 IK=IK+1
    8 CONTINUE
C
C  STORE BAND OF SYSTEM ELASTIC MATRIX  (BTDB)  IN TEMPORARY FILE
C
      CALL NTRAN(LU1,1,NWRKE,RKE,L,22)
      IF(L.EQ.NWRKE) GO TO 27
      PRINT 820,L
  820 FORMAT(' ERROR ON WRITING SYSTEM ELASTIC MATRIX (BTDB)',
     $' :  ERROR CODE IS',I4)
      STOP
C
C  DECOMPOSE ELASTIC SYSTEM MATRIX  (BTDB)  INTO PRODUCT  (UT*U)
C
   27 DO 21 IR=1,NDFS
      IB=IR+1
      IBB=IR+NBWS
      IF(IBB.GT.NDF) IBB=NDF
      NCOLL=NBWS
      IC=2
      DO 24 I=IB,IBB
      ICC=IC
      IF(RKE(IR,IC).EQ.0.) GO TO 25
      FACT=RKE(IR,IC)/RKE(IR,1)
      IF(IR.GT.NDFBWS) NCOLL=NDFA-I
      DO 26 J=1,NCOLL
      RKE(I,J)=RKE(I,J)-FACT*RKE(IR,ICC)
   26 ICC=ICC+1
   25 NCOLL=NCOLL-1
   24 IC=IC+1
      FACT=DSQRT(RKE(IR,1))
      RKE(IR,1)=FACT
      FACT=1./FACT
      NCOLL=NBW
      IF(IR.GT.NDFBWS) NCOLL=NDFA-IR
      DO 21 J=2,NCOLL
   21 RKE(IR,J)=RKE(IR,J)*FACT
      RKE(NDF,1)=SQRT(RKE(NDF,1))
C
C  STORE BAND OF DECOMPOSED SYSTEM ELASTIC MATRIX  (U)  IN TEMPORARY FILE
```

```
C
      CALL NTRAN(LU2,1,NWRKE,RKE,L,22)
      IF(L.EQ.NWRKE) GO TO 13
      PRINT 821,L
  821 FORMAT(' ERROR ON WRITING DECOMPOSED SYSTEM ELASTIC MATRIX  (U)',
     $' :  ERROR CODE IS',I4)
      STOP
   13 RETURN
      END
```

```
COL*EPTCS(1).SOLVE1
      COMPILER (XM=3)
      SUBROUTINE SOLVE1 (NDFP,NPCP,NLPP,PP,DU,DIAG,QQ,RK3)
C
C  FOR EXTENDED STORAGE GIVE EXPLICIT SIZE OF ELASTIC AND PLASTIC ARRAYS
C
       COMMON /EXT/ RKE(194,24) , RKP(194,40)
C
      COMMON IA,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPCS,NPC,NDFBWS,
     2LP,IL,TSTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,RMIN,IS,
     3NLT,NWRKE,NWRKP,ETA,IQ,NNEWP,NOUT,LU1,LU2,LU3,FRAC
      DIMENSION DIAG(NPCP),PP(NLPP,NDFP),DU(NDFP),RK3(NPCP,NPCP),
     $QQ(NPCP)
      DOUBLE PRECISION FACT
C
C  APPLY LOAD VECTOR
C
      DO 1 I=1,NDF
    1 DU(I)=PP(LP,I)
C
C  PERFORM FORWARD- AND BACK-SUBSTITUTIONS ON (UT*U) DECOMPOSED
C  ELASTIC SYSTEM MATRIX
C
      CALL FORSUB(DU)
      CALL BAKSUB(DU)
      IF(NPC.EQ.0) GO TO 2
C
C  CALCULATE PLASTIC SUBMATRICES OF PARTITIONED UNCOUPLED SYSTEM MATRIX
C
      DO 3 I=1,NPC
      QQ(I)=0.
      DO 3 J=1,NDF
    3 QQ(I)=QQ(I)-RKP(J,I)*DU(J)
      DO 4 I=1,NPC
    4 CALL FORSUB(RKP(1,I))
      DO 5 I=1,NPC
      DO 5 J=1,NPC
      RK3(I,J)=0.
      IF(I.EQ.J) RK3(I,J)=DIAG(I)
      DO 5 K=1,NDF
    5 RK3(I,J)=RK3(I,J)-RKP(K,I)*RKP(K,J)
C
C  SOLVE FOR MULTIPLIERS USING GAUSS REDUCTION AND BACK-SUBSTITUTION
C
      IF(NPC.EQ.1) GO TO 6
      NPCS=NPC-1
      DO 7 IR=1,NPCS
      IB=IR+1
      DO 7 I=IB,NPC
      IF(RK3(IR,I).EQ.0.) GO TO 7
      FACT=RK3(IR,I)/RK3(IR,IR)
      DO 8 J=1,NPC
    8 RK3(I,J)=RK3(I,J)-FACT*RK3(IR,J)
      QQ(I)=QQ(I)-FACT*QQ(IR)
    7 CONTINUE
      DO 9 IR=NPC,2,-1
      IF(QQ(IR).EQ.0.) GO TO 9
      QQ(IR)=QQ(IR)/RK3(IR,IR)
      IB=IR-1
      DO 10 I=IB,1,-1
```

```
   10 QQ(I)=QQ(I)-RK3(I,IR)*QQ(IR)
    9 CONTINUE
    6 QQ(1)=QQ(1)/RK3(1,1)
C
C  READ PLASTIC MATRIX  (-BTDN)  FROM TEMPORARY FILE
C
      CALL NTRAN(LU3,10,2,NWRKP,RKP,L,22)
      IF(L.EQ.NWRKP) GO TO 11
      PRINT 840,L
  840 FORMAT(' ERROR ON READING SYSTEM PLASTIC MATRIX (-BTDN)',
     $' :  ERROR CODE IS',I4)
      STOP
C
C  CALCULATE RHS VECTOR OF PARTITIONED UNCOUPLED SYSTEM MATRIX
C
   11 DO 12 I=1,NDF
      DU(I)=PP(LP,I)
      DO 12 J=1,NPC
   12 DU(I)=DU(I)-RKP(I,J)*QQ(J)
C
C  SOLVE FOR DISPLACEMENTS USING FORWARD- AND BACK-SUBSTITUTION
C  ON (UT*U) DECOMPOSED ELASTIC SYSTEM MATRIX
C
      CALL FORSUB(DU)
      CALL BAKSUB(DU)
    2 RETURN
C
C  SUBROUTINE FOR FORWARD-SUBSTITUTION ON (UT*U) DECOMPOSED ELASTIC SYSTEM MATRIX
C
      SUBROUTINE FORSUB (RHS)
      DIMENSION RHS(NDFP)
      DO 1 IR=1,NDFS
      IF(RHS(IR).EQ.0.) GO TO 1
      RHS(IR)=RHS(IR)/RKE(IR,1)
      IB=IR+1
      IBB=IR+NBWS
      IF(IBB.GT.NDF) IBB=NDF
      J=2
      DO 2 I=IB,IBB
      RHS(I)=RHS(I)-RKE(IR,J)*RHS(IR)
    2 J=J+1
    1 CONTINUE
      RHS(NDF)=RHS(NDF)/RKE(NDF,1)
      RETURN
C
C  SUBROUTINE FOR BACK-SUBSTITUTION ON (UT*U) DECOMPOSED ELASTIC SYSTEM MATRIX
C
      SUBROUTINE BAKSUB (RHS)
      DIMENSION RHS(NDFP)
      DO 1 IR=NDF,2,-1
      IF(RHS(IR).EQ.0.) GO TO 1
      RHS(IR)=RHS(IR)/RKE(IR,1)
      IB=IR-1
      IBB=IR-NBWS
      IF(IBB.LT.1) IBB=1
      J=2
      DO 2 I=IB,IBB,-1
      RHS(I)=RHS(I)-RKE(I,J)*RHS(IR)
    2 J=J+1
    1 CONTINUE
```

```
RHS(1)=RHS(1)/RKE(1,1)
RETURN
END
```

```
COL*EPTCS(1).SOLVE2
         COMPILER (XM=3)
         SUBROUTINE SOLVE2 (NDFP,NPCP,NLPP,PP,DU,DIAG,QQ,RK3)
C
C   FOR EXTENDED STORAGE GIVE EXPLICIT SIZE OF ELASTIC AND PLASTIC ARRAYS
C
          COMMON /EXT/ RKE(194,24) , RKP(194,40)
C
         COMMON IA,NN,NF,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPCS,NPC,NDFBWS,
        2LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,RMIN,IS,
        3NLI,NWRKE,NWRKP,FTA,IQ,NNEWP,NOUT,LU1,LU2,LU3,FRAC
         DIMENSION DIAG(NPCP),PP(NLPP,NDFP),DU(NDFP),RK3(NPCP,NPCP),
        $QQ(NPCP)
         DOUBLE PRECISION FACT
C
C   AUGMENT WITH LOAD VECTOR
C
         DO 18 I=1,NDF
      18 DU(I)=PP(LP,I)
C
C   STORE DIAGONAL PLASTIC MATRIX  (NTDN+DP)
C
         DO 12 I=1,NPC
         QQ(I)=0.
         DO 12 J=I,NPC
      12 RK3(I,J)=0.
         DO 22 I=1,NPC
      22 RK3(I,I)=DIAG(I)
C
C   GAUSS REDUCE TO UPPER TRIANGULAR MATRIX
C   *************************************
C
C   FIRST  N  ROWS
C
         DO 1 IR=1,NDF
         IF(IR.EQ.NDF) GO TO 10
C
C   ELASTIC MATRIX  (BTDB)
C
         IB=IR+1
         IBB=IR+NBWS
         IF(IBB.GT.NDF) IBB=NDF
         NCOLL=NBWS
         IC=2
         DO 8 I=IB,IBB
         ICC=IC
         IF(RKE(IR,IC).EQ.0.) GO TO 24
         FACT=RKE(IR,IC)/RKE(IR,1)
         IF(IR.GT.NDFBWS) NCOLL=NDFA-I
         DO 9 J=1,NCOLL
         RKE(I,J)=RKE(I,J)-FACT*RKE(IR,ICC)
       9 ICC=ICC+1
         DO 16 J=1,NPC
      16 RKP(I,J)=RKP(I,J)-FACT*RKP(IR,J)
         DU(I)=DU(I)-FACT*DU(IR)
      24 NCOLL=NCOLL-1
       8 IC=IC+1
      10 IF(NPC.EQ.0) GO TO 1
C
C   PLASTIC MATRIX  (-HTDN)
```

```
19 II=II+1
14 DU(IR)=DU(IR)/RKE(IR,1)
   RETURN
   END
```

```
      COMPILER (XM=3)
      SUBROUTINE STRSTR (NEP,NNP,NDFP,ISP,NECP,NPCP,NLPP,
     $P,U,ESTRN,PSTRN,STRN,STRS,SIGBAR,PP,TELT,PMAX,IOUT,
     $NELTPR,NODFPR,D,IPELTS,RLAM,A,CF,DP,VOL,B,BTD,RMINC,
     $DU,DESTRN,DPSTRN,DSTRN,DSTRS,IPPT,PHI,ITRI)
C
C  FOR EXTENDED STORAGE GIVE EXPLICIT SIZE OF ELASTIC AND PLASTIC ARRAYS
C
       COMMON /EXT/ RKE(194,24) , RKP(194,40)
C
      COMMON IA,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPCS,NPC,NDFBWS,
     2LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,RMIN,IS,
     3NLT,NWRKE,NWRKP,ETA,IQ,NNENP,NOUT,LU1,LU2,LU3,FRAC
      DIMENSION P(NDFP),U(NDFP),ESTRN(NEP,ISP),PSTRN(NEP,ISP),
     2STRN(NEP,ISP),STRS(NEP,ISP),SIGBAR(NEP,ISP),PP(NLPP,NDFP),
     3TELT(NEP,3),PMAX(NLPP,2),IOUT(100),NELTPR(NEP),NODFPR(NNP),
     4D(4,4),IPELTS(NEP,3),RLAM(NEP),A(NEP,ISP),CF(NEP),DP(NDFP),
     5VOL(NEP),B(NEP,ISP,6),BTD(NEP,6,ISP),DU(NDFP),DESTRN(NEP,ISP),
     6DPSTRN(NEP,ISP),DSTRN(NEP,ISP),DSTRS(NEP,ISP),IPPT(NEP),
     7NENP(50),ITRI(NEP,6),RMINC(NLPP),DEL(6)
      DOUBLE PRECISION DUM
      RMIN=1.E30
C
C  FOR EACH ELEMENT :-
C ********************
C
      DO 3 IE=1,NE
C
C  CALCULATE RELATIVE MAGNITUDES OF TOTAL STRAIN INCREMENT
C
      DO 1 I=1,6
      J=ITRI(IE,I)
    1 DEL(I)=DU(J)
      DO 2 I=1,IS
      DSTRN(IE,I)=0.
      DO 2 J=1,6
    2 DSTRN(IE,I)=DSTRN(IE,I)+B(IE,I,J)*DEL(J)
      IF(IPELTS(IE,1).EQ.1) GO TO 5
C
C  FOR ELASTIC ELEMENTS CALCULATE RELATIVE MAGNITUDES OF ELASTIC STRAIN
C  AND STRESS INCREMENT VECTORS, AND SCALAR MULTIPLIER TO CAUSE STRESS
C  POINT TO REACH YIELD SURFACE. FROM ALL ELASTIC ELEMENTS DETERMINE
C  SMALLEST SCALAR MULTIPLIER
C
      DO 4 I=1,IS
      DPSTRN(IE,I)=0.
      DESTRN(IE,I)=DSTRN(IE,I)
      DSTRS(IE,I)=0.
      DO 4 J=1,IS
    4 DSTRS(IE,I)=DSTRS(IE,I)+D(I,J)*DSTRN(IE,J)
      IFACT=1
      CALL ROOT(DUM,DSTRS(IE,1),DSTRS(IE,2),DSTRS(IE,3),DSTRS(IE,4),
     $SIGBAR(IE,1),SIGBAR(IE,2),SIGBAR(IE,3),SIGBAR(IE,4))
      DD=DUM
      IF(RMIN.GT.DD) RMIN=DD
      GO TO 3
C
C  FOR PLASTIC ELEMENTS CALCULATE RELATIVE MAGNITUDES OF ELASTIC STRAIN,
C  PLASTIC STRAIN, AND STRESS INCREMENT VECTORS
```

```
      C
         5 DO 6 I=1,IS
           DPSTRN(IE,I)=RLAM(IE)*A(IE,I)
         6 DESTRN(IE,I)=DSTRN(IE,I)-DPSTRN(IE,I)
           DO 7 I=1,IS
           DSTRS(IE,I)=0.
           DO 7 J=1,IS
         7 DSTRS(IE,I)=DSTRS(IE,I)+D(I,J)*DESTRN(IE,J)
         3 CONTINUE
      C
      C  CHECK IF PROPOSED MAGNITUDE OF TOTAL LOAD IN CURRENT LOAD DIRECTION
      C  COMPLIES WITH LOADING PROGRAMME
      C
           IR=IFIX(PMAX(LP,1))
           DUM=(PMAX(LP,2)-P(IR))/PP(LP,IR)
           IF(RMIN.LT.DUM) GO TO 18
           RMIN=DUM
           ICHECK=I
        18 RMINC(LP)=RMINC(LP)+RMIN
      C
      C  MULTIPLY ALL INCREMENT QUANTITIES BY SMALLEST MULTIPLIER, AND DETERMINE
      C  CURRENT TOTALS OF LOAD, DISPLACEMENT, ELASTIC STRAIN, PLASTIC STRAIN
      C  AND STRESS
      C
           DO 17 I=1,NDF
           DU(I)=DU(I)*RMIN
           U(I)=U(I)+DU(I)
           DP(I)=RMIN*PP(LP,I)
        17 P(I)=P(I)+DP(I)
           DO 11 IE=1,NE
           DO 12 J=1,IS
           DSTRN(IE,J)=DSTRN(IE,J)*RMIN
           DESTRN(IE,J)=DESTRN(IE,J)*RMIN
           DSTRS(IE,J)=DSTRS(IE,J)*RMIN
           STRN(IE,J)=STRN(IE,J)+DSTRN(IE,J)
           ESTRN(IE,J)=ESTRN(IE,J)+DESTRN(IE,J)
           IF(IPELTS(IE,1).EQ.0) GO TO 19
           DPSTRN(IE,J)=DPSTRN(IE,J)*RMIN
           PSTRN(IE,J)=PSTRN(IE,J)+DPSTRN(IE,J)
        19 STRS(IE,J)=STRS(IE,J)+DSTRS(IE,J)
        12 SIGBAR(IE,J)=STRS(IE,J)-EP*PSTRN(IE,J)
           IF(IPELTS(IE,1).EQ.0) GO TO 11
           PLAM(IE)=RLAM(IE)*RMIN
      C
      C  IN ELASTIC-PERFECTLY PLASTIC CASE, NEUTRAL-LOADING STRESS POINTS
      C  MOVE TANGENTIAL TO YIELD SURFACE, THEREFORE CORRECT CURRENT
      C  STRESSES BY RETURNING STRESS POINT TO YIELD SURFACE
      C
           IFACT=2
           CALL ROOT(DUM,STRS(IE,1),STRS(IE,2),STRS(IE,3),STRS(IE,4),
          $PSTRN(IE,1),PSTRN(IE,2),PSTRN(IE,3),PSTRN(IE,4))
           CF(IE)=DUM
           DO 14 J=1,IS
           STRS(IE,J)=STRS(IE,J)*CF(IE)
        14 SIGBAR(IE,J)=STRS(IE,J)-EP*PSTRN(IE,J)
        11 CONTINUE
      C
      C  DETERMINE CURRENT PLASTIC ELEMENTS. FOR ELASTIC ELEMENTS CHECK RATIO
      C  OF VON MISES EQUIVALENT STRESS TO UNIAXIAL YIELD STRESS FOR STRESS
      C  POINT TO BE TREATED AS PLASTIC
```

```
C
      NPPIS=0
      NNEWP=0
      DO 15 IE=1,NE
      IF(IPELTS(IE,1).EQ.1) GO TO 10
      IF(IA.GT.-1) GO TO 9
      DUM=DSQRT(SIGBAR(IE,1)*SIGBAR(IE,1)-SIGBAR(IE,1)*SIGBAR(IE,2)
     $+SIGBAR(IE,2)*SIGBAR(IE,2)+3.*SIGBAR(IE,3)*SIGBAR(IE,3))
      GO TO 8
    9 DUM=DSQRT(SIGBAR(IE,1)*SIGBAR(IE,1)+SIGBAR(IE,2)
     $*SIGBAR(IE,2)+SIGBAR(IE,4)*SIGBAR(IE,4)-SIGBAR(IE,1)
     $*SIGBAR(IE,2)-SIGBAR(IE,2)*SIGBAR(IE,4)-SIGBAR(IE,1)
     $*SIGBAR(IE,4)+3.*SIGBAR(IE,3)*SIGBAR(IE,3))
    8 DUM=DUM/SZERO
      IF(DUM.LT.ETA) GO TO 15
      IPELTS(IE,1)=1
      NNEWP=NNEWP+1
      NEWP(NNEWP)=IE
   10 NPPIS=NPPIS+1
      IPPT(NPPIS)=IE
      IPELTS(IE,2)=NPPTS
   15 CONTINUE
      NPC=NPPIS
      NPCS=NPC-1
C
C  CHECK IF LOAD INCREMENTS BECOMING CONSISTANTLY LESS THAN 'FRAC' OF TOTAL LOAD
C
      IF(EP.GT.0.0) GO TO 20
      IF(RMIN/RMINC(LP).GT.FRAC) GO TO 21
      NKNT=NKNT+1
      IF(NKNT.LT.3) GO TO 20
      PRINT 201,FRAC
  201 FORMAT(1H0,83('*'),/,' LOAD INCREMENTS LESS THAN',F6.3,
     $'  OF TOTAL LOAD FOR PRECEDING  3  LOAD INCREMENTS',//,
     $' ANALYSIS TERMINATED',//,' RESULTS AFTER CURRENT LOAD INCREMENT L
     $ISTED BELOW',//,84('*'))
      ISTOP=-1
      IOUT(NOUT)=NLI+1
   21 NKNT=0
C
C  OUTPUT CURRENT QUANTITIES
C
   20 CALL OUTPUT (NEP,NNP,NDFP,ISP,NECP,NPCP,NLPP,
     $PSTRN,STRN,STRS,SIGBAR,PMAX,IOUT,NELTPR,NODEPR,RMINC,
     $PHI,IPELTS,RLAM,CF,DP,DU,DESTRN,DPSTRN,DSTRN,DSTRS,IPPT,
     $P,U,ESTRN,NEWP)
      IF(ISTOP.EQ.-1) GO TO 16
C
C  CHECK AVAILABLE STORAGE
C
      NWRKP=NDF*NPC
      IF(NPC.LE.NPCP) GO TO 13
      PRINT 202
  202 FORMAT(1H0,54('*'),/,' AVAILABLE STORAGE EXCEEDED :  PLASTIC MATRI
     $X TOO LARGE',//,' ANALYSIS TERMINATED',//,' RESULTS AFTER PREVIOUS
     $ LOAD INCREMENT LISTED BELOW',//,55('*'))
      IOUT(NOUT)=NLI
      NLI=NLI-1
      CALL OUTPUT (NEP,NNP,NDFP,ISP,NECP,NPCP,NLPP,
     $PSTRN,STRN,STRS,SIGBAR,PMAX,IOUT,NELTPR,NODEPR,RMINC,
```

```
     $PHI,IPELTS,RLAM,CF,DP,DU,DESTRN,DPSTRN,DSTRN,DSTRS,IPPT,
     $P,U,ESTRN,NEWP)
      STOP
C
C  READ ELASTIC MATRIX  (BTDB)   FROM TEMPORARY FILE
C
   13 IF(NPC.LE.NBW) LU=LU2
      IF(NPC.GT.NBW) LU=LU1
      CALL NTRAN(LU,10,22,2,NWRKF,RKE,L,22)
      IF(L.EQ.NWRKF) GO TO 16
      PRINT 820,L
  820 FORMAT(' ERROR ON READING SYSTEM ELASTIC MATRIX (BTDB)',
     $' :  ERROR CODE IS',I4)
      STOP
   16 RETURN
C
C  SUBROUTINE FOR DETERMINING SCALAR MULTIPLIERS FOR ELASTIC ELEMENTS,
C  AND FOR RETURNING STRESS POINTS TO YIELD SURFACE FOR PLASTIC ELEMENTS
C
      SUBROUTINE ROOT(RT,X1,X2,X3,X4,Y1,Y2,Y3,Y4)
      DOUBLE PRECISION AA,BB,CC,RT
      IF(IA.GT.-1) GO TO 2
      AA=X1*X1-X1*X2+X2*X2+3.*X3*X3
      IF(IFACT.EQ.2.AND.EP.EQ.0) GO TO 1
      BB=X1*Y1-0.5*(X1*Y2+X2*Y1)+X2*Y2+3.*X3*Y3
      CC=Y1*Y1-Y1*Y2+Y2*Y2+3.*Y3*Y3
      GO TO 4
    2 AA=X1*X1+X2*X2+X4*X4-X1*X2-X2*X4-X1*X4+3.*X3*X3
      IF(IFACT.EQ.2.AND.EP.EQ.0) GO TO 1
      BB=X1*Y1+X2*Y2+X4*Y4-0.5*(X1*Y2+X2*Y1+X2*Y4+X4*Y2
     $+X1*Y4+X4*Y1)+3.*X3*Y3
      CC=Y1*Y1+Y2*Y2+Y4*Y4-Y1*Y2-Y2*Y4-Y1*Y4+3.*Y3*Y3
    4 IF(IFACT.EQ.1) GO TO 5
      BB=-BB*EP
      CC=CC*EPS
    5 RT=(DSQRT(BB*BB-AA*(CC-SZEROS))-BB)/AA
      RETURN
    1 RT=SZERO/DSQRT(AA)
      RETURN
      END
```

```
COL*FPTCS(1).PLASM
      SUBROUTINE PLASM (NEP,ISP,IE,A,SIGBAR,VOL,D,BTDN,ENTDN,BTD)
      COMMON IA,NN,NE,NPPIS,NDFS,NDF,NDFA,NBWS,NBW,NPCS,NPC,NDFBWS,
     2LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,RMIN,IS,
     3NLT,NWRKF,NWRKP,FIA,TQ,NNEWP,NOUT,LU1,LU2,LU3,FRAC
      DIMENSION A(NEP,ISP),SIGBAR(NEP,ISP),VOL(NEP),D(4,4),
     2BTD(NEP,6,ISP),ATD(4),BTDN(6)
      DOUBLE PRECISION DUM
C
C  CALCULATE GRADIENT OF YIELD FUNCTION
C
      IF(IA.GT.-1) GO TO 1
      A(IE,1)=SIGBAR(IE,1)+SIGBAR(IE,1)-SIGBAR(IE,2)
      A(IE,2)=SIGBAR(IE,2)+SIGBAR(IE,2)-SIGBAR(IE,1)
      A(IE,3)=6.*SIGBAR(IE,3)
      GO TO 2
    1 A(IE,1)=SIGBAR(IE,1)+SIGBAR(IE,1)-SIGBAR(IE,2)
     $-SIGBAR(IE,4)
      A(IE,2)=SIGBAR(IE,2)+SIGBAR(IE,2)-SIGBAR(IE,1)
     $-SIGBAR(IE,4)
      A(IE,3)=6.*SIGBAR(IE,3)
      A(IE,4)=SIGBAR(IE,4)+SIGBAR(IE,4)-SIGBAR(IE,1)
     $-SIGBAR(IE,2)
C
C  NORMALIZE GRADIENT OF YIELD FUNCTION
C
    2 DUM=0.
      DO 3 I=1,IS
    3 DUM=DUM+A(IE,I)*A(IE,I)
      DUM=1./DSQRT(DUM)
      DO 4 I=1,IS
    4 A(IE,I)=A(IE,I)*DUM
C
C  CALCULATE  (-BTDN)  VECTOR
C
      DO 5 I=1,6
      BTDN(I)=0.
      DO 5 J=1,IS
    5 BTDN(I)=BTDN(I)+BTD(IE,I,J)*A(IE,J)
C
C  CALCULATE  (NTDN+DP)  TERM
C
      DO 6 I=1,IS
      ATD(I)=0.
      DO 6 J=1,IS
    6 ATD(I)=ATD(I)+A(IE,J)*D(J,I)
      ENTDN=EP
      DO 7 J=1,IS
    7 ENTDN=ENTDN+ATD(J)*A(IE,J)
      ENTDN=ENTDN/VOL(IE)
      RETURN
      END
```

COL*EPTCS(1).ITER

```
          COMPILER (XM=3)
          SUBROUTINE ITER (NEP,NNP,NDFP,ISP,NECP,NPCP,NLPP,
         $TELT,IBC,D,IPELTS,RLAM,A,PP,DU,SIGBAR,VOL,B,ITRI,RMINC,
         $QQ,ESTRN,U,P,IPPT,DSTRS,DSTRN,DPSTRN,DFSTRN,DP,CF,PMAX,RK3,
         $TOUT,NELTPR,NODEPR,PSTRN,STRN,STRS,PHI,DIAG,DIAG2)
C
C  FOR EXTENDED STORAGE GIVE EXPLICIT SIZE OF ELASTIC AND PLASTIC ARRAYS
C
           COMMON /EXT/ RKE(194,24) , RKP(194,40)
C
          COMMON IA,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPCS,NPC,NDFBWS,
         2LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,RMIN,IS,
         3NLI,NWRKE,NWRKP,ETA,TQ,NNEWP,NOUT,LU1,LU2,LU3,FRAC
          DIMENSION TELT(NEP,3),IBC(NDFP),D(4,4),IPELTS(NEP,3),
         2RLAM(NEP),A(NEP,ISP),PP(NLPP,NDFP),DU(NDFP),SIGBAR(NEP,ISP),
         3VOL(NEP),B(NEP,ISP,6),ITRI(NEP,6),TOUT(100),DESTRN(NEP,ISP),
         4DSTRS(NEP,ISP),DP(NDFP),DIAG(NPCP),QQ(NPCP),RK3(NPCP,NPCP),
         5DUM1(6),DUM2(4),DIAG2(NPCP),PMAX(NLPP,2),RMINC(NLPP),P(NDFP)
         $,U(NDFP)
C
C  AS ITERATION MIGHT BE NECESSARY, STORE PLASTIC MATRIX  (-RTON)
C  IN TEMPORARY FILE
C
          CALL NTRAN(LU3,10,22,1,NWRKP,RKP,L,22)
          IF(L.EQ.NWRKP) GO TO 1
          PRINT 820,L
      820 FORMAT(' ERROR ON WRITTING SYSTEM PLASTIC MATRIX (-RTON)',
         $' :  ERROR CODE IS',I4)
          STOP
        1 DO 5 I=1,NPC
        5 DIAG2(I)=DIAG(I)
C
C  INITIALISE ITERATION COUNTERS
C
          IT=1
          NLIA=NLI+1
C
C  SOLVE SYSTEM MATRIX FOR DISPLACEMENT INCREMENT VECTOR AND LAMBDAS
C
       16 IF(NPC.LE.NBW) CALL SOLVE1 (NDFP,NPCP,NLPP,PP,DU,DIAG,QQ,RK3)
          IF(NPC.GT.NBW) CALL SOLVE2 (NDFP,NPCP,NLPP,PP,DU,DIAG,QQ,RK3)
          IK=0
C
C  CHECK KINEMATIC CONSTRAINTS FOR EACH PLASTIC ELEMENT :-
C *************************************************************
C
          DO 4 IE=1,NE
          IF(IPELTS(IE,1).EQ.0) GO TO 4
          I=IPELTS(IE,2)
          RLAM(IE)=QQ(I)/VOL(IE)
          IF(IPELTS(IE,3).EQ.1) GO TO 2
C
C  (1): FOR LAMBDAS ASSUMED NON-ZERO CHECK THAT LAMBDA NON-NEGATIVE
C
          IF(RLAM(IE).GE.0.) GO TO 4
          IF(ABS(RLAM(IE)).GE.1.E-6) GO TO 30
          RLAM(IE)=0.
          GO TO 4
       30 IK=1
```

```
      IPELTS(IE,3)=1
      PRINT 201,NLTA,IT,IE,RLAM(IE)
  201 FORMAT(1H0,/,' LOAD INCREMENT ',I4,'   ITERATION ',I4,/,
     $' ELEMENT ',I4,'  UNLOADING  :  LAMBDA = ',E12.6)
      GO TO 4
C
C  (2): FOR LAMBDAS ASSUMED ZERO CHECK THAT SCALAR PRODUCT OF GRADIENT OF
C       YIELD FUNCTION AND NORMALIZED STRESS INCREMENT VECTOR NON-POSITIVE
C
C  ELEMENT NODE DISPLACEMENT INCREMENTS
C
    2 DO 6 I=1,6
      J=ITRI(IE,I)
    6 DUM1(I)=DU(J)
C
C  CALCULATE RELATIVE MAGNITUDES OF COMPONENTS OF ELASTIC STRAIN
C  INCREMENT VECTOR
C
      DO 13 I=1,IS
      DUM2(I)=0.
      DO 13 J=1,6
   13 DUM2(I)=DUM2(I)+B(IE,I,J)*DUM1(J)
C
C  CALCULATE RELATIVE MAGNITUDES OF COMPONENTS OF STRESS INCREMENT VECTOR
C
      DO 8 I=1,IS
      DUM1(I)=0.
      DO 8 J=1,IS
    8 DUM1(I)=DUM1(I)+D(I,J)*DUM2(J)
C
C  NORMALIZE RELATIVE MAGNITUDES OF COMPONENTS OF STRESS INCREMENT VECTOR
C
      DUM=0.
      DO 10 I=1,IS
   10 DUM=DUM+DUM1(I)*DUM1(I)
      DUM=1./SQRT(DUM)
      DO 19 I=1,IS
   19 DUM1(I)=DUM1(I)*DUM
C
C  CALCULATE AND CHECK SCALAR PRODUCT OF GRADIENT OF YIELD FUNCTION AND
C  NORMALIZED STRESS INCREMENT VECTOR
C
      DUM=0.
      DO 9 I=1,IS
    9 DUM=DUM+A(IE,I)*DUM1(I)
      IF(DUM.LT.0) GO TO 4
      IK=1
      IPELTS(IE,3)=0
      PRINT 202,NLTA,IT,IE,DUM
  202 FORMAT(1H0,/,' LOAD INCREMENT ',I4,'   ITERATION ',I4,/,
     $' ELEMENT ',I4,'  LOADING',/,' SCALAR PRODUCT OF GRADIENT OF YIELD
     $ FUNCTION AND NORMALIZED STRESS INCREMENT VECTOR =',F12.6)
    4 CONTINUE
C
C  IF ITERATION CONSTRAINTS SATISFIED, CONTINUE
C
      IF(IK.EQ.0) GO TO 12
C
C  IF ITERATION PROCESS NOT CONVERGING TERMINATE ANALYSIS
C
```

```
      IF(IT.LT.10) GO TO 11
      PRINT 200
  200 FORMAT(1H0,50('*'),//,' STILL IN ITERATION LOOP AFTER 10 ITERATION
     $S',//,' ANALYSIS TERMINATED',//,' TOTALS AFTER PREVIOUS LOAD INCRE
     $MENT LISTED BELOW',//,51('*'))
      GO TO 29
C
C  IF ITERATION NECESSARY, READ ELASTIC AND PLASTIC MATRICES FROM
C  TEMPORARY FILES
C
   11 IT=IT+1
      IF(NPC.LE.NBW) LU=LU2
      IF(NPC.GT.NBW) LU=LU1
      CALL NTRAN(LU,10,22,2,NWRKE,RKE,L,22)
      IF(L.EQ.NWRKE) GO TO 21
      PRINT 830,L
  830 FORMAT(' ERROR ON READING SYSTEM ELASTIC MATRIX (BTDB)',
     $' :  ERROR CODE IS',I4)
      STOP
   21 CALL NTRAN(LU3,10,22,2,NWRKP,RKP,L,22)
      IF(L.EQ.NWRKP) GO TO 17
      PRINT 840,L
  840 FORMAT(' ERROR ON READING SYSTEM PLASTIC MATRIX (-BTDN)',
     $' :  ERROR CODE IS',I4)
      STOP
   17 DO 7 I=1,NPC
    7 DIAG(I)=DIAG2(I)
C
C  FOR PLASTIC ELEMENTS ASSUMED UNLOADING, DELETE APPROPRIATE
C  ROW AND COLUMN FROM PLASTIC MATRICES  (-BTDN)  AND  (NTDN+DP)
C
      DO 14 IE=1,NF
      IF(IPELTS(IE,3).EQ.0) GO TO 14
      NC=IPELTS(IE,2)
      DO 15 I=1,NDF
   15 RKP(I,NC)=0.
      DIAG(NC)=1.
   14 CONTINUE
C
C  RE-SOLVE
C
      GO TO 16
   29 IOUT(NOUT)=NLI
      NLI=NLI-1
      DO 27 IE=1,NF
      IPELTS(IE,1)=0
      DO 27 J=1,IS
      DESTRN(IE,J)=0.
   27 DSTRS(IE,J)=0.
      DO 28 I=1,NDF
      DP(I)=0.
   28 DU(I)=0.
      CALL OUTPUT (NEP,NNP,NDEP,ISP,NECP,NPCP,NLPP,
     $PSTRN,STRN,STRS,SIGBAR,PMAX,IOUT,NELTPR,NODEPR,RMINC,
     $PHI,IPELTS,RLAM,CF,DP,DU,DESTRN,DPSTRN,DSTRN,DSTRS,IPPT,
     $P,U,ESTRN,NEWP)
      STOP
C
C  ITERATION PROCEDURE HAS CONVERGED :  RECORD CURRENT LOADING AND
C  UNLOADING PLASTIC ELEMENTS
```

```
C
   12 DO 3 IE=1,NE
      IF(IPELTS(IE,3).EQ.0) GO TO 3
      IPELTS(IE,1)=0
      IPELTS(IE,3)=0
    3 CONTINUE
      RETURN
      END
```

```
COL*EPTCS(1).OUTPUT
      SUBROUTINE OUTPUT (NEP,NNP,NDFP,ISP,NECP,NPCP,NLPP,
     $PSTRN,STRN,STRS,SIGBAR,PMAX,IOUT,NELTPR,NODEPR,RMINC,
     $PHI,IPELTS,RLAM,CF,DP,DU,DESTRN,DPSTRN,DSTRN,DSTRS,IPPT,
     $P,U,ESTRN,NEWP)
      COMMON IA,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPCS,NPC,NDFBWS,
     2LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,RMIN,IS,
     3NLI,NWRKE,NWRKP,ETA,IQ,NNEWP,NOUT,LU1,LU2,LU3,FRAC
      DIMENSION PSTRN(NEP,ISP),STRN(NEP,ISP),STRS(NEP,ISP),
     2SIGBAR(NEP,ISP),PMAX(NLPP,2),IOUT(100),NELTPR(NEP),
     3NODEPR(NNP),PHI(NEP),IPELTS(NEP,3),RLAM(NEP),CF(NEP),
     4ESTRN(NEP,ISP),NEWP(50),DP(NDFP),DU(NDFP),RMINC(NDFP),
     5DESTRN(NEP,ISP),DPSTRN(NEP,ISP),DSTRN(NEP,ISP),
     6DSTRS(NEP,ISP),IPPT(NEP),P(NDFP),U(NDFP)
      NLI=NLI+1
      IF(NLI.NE.IOUT(NOUT).AND.ICHECK.EQ.0) GO TO 7
      IF(NLI.EQ.IOUT(NOUT)) NOUT=NOUT+1
C
C  PRINT HEADINGS
C
      PRINT 200,NLI
  200 FORMAT(1H1,'RESULTS AFTER',I4,2X,'LOAD INCREMENTS',/)
      IF(ICHECK.EQ.1) GO TO 1
      PRINT 201,(NEWP(I),I=1,NNEWP)
  201 FORMAT(1H0,'NEXT ELEMENTS TO UNDERGO PLASTIC DEFORMATION :',10I5,
     $4(//,47X,10I5))
      GO TO 2
    1 I=(IFIX(PMAX(LP,1))+1)/2
      PRINT 202,LP,I
  202 FORMAT(1H0,'END OF LOADING ',I4,2X,': MAXIMUM LOAD REACHED AT NODE
     $',I4)
    2 PRINT 218,RMIN
  218 FORMAT(1H0,'AT LOAD FACTOR',E11.6)
      DO 6 I=1,LP
    6 PRINT 221,I,RMINC(I)
  221 FORMAT(1H0,'LOAD VECTOR',I3,' : CUMULATIVE LOAD FACTOR =',E11.6)
      IF(IA.EQ.1) PRINT 219
  219 FORMAT(1H0,'NOTE: CIRCUMFERENTIAL STRAINS AND STRESSES (0) CORRESP
     $OND TO CENTROIDS OF ELEMENTS',/,1H+,44X,'-',/,6('-'))
      IF(ICHECK.EQ.0) GO TO 23
      LP=LP+1
      ICHECK=0
   23 PRINT 206
  206 FORMAT(1H0,74X,'CURRENT VALUES',/,55X,'/',52('-'),'/',/,14X,'STRAI
     $N INCREMENTS',37X,'STRAINS',38X,'STRESS',/,12X,'/',18('-'),'/',14X
     $,'STRESS',4X,'/',29('-'),'/',14X,'YIELD',5X,'CORRECTION',/,
     $' ELEMENT',5X,'ELASTIC',4X,'PLASTIC',4X,'LAMBDA',4X,'INCREMENT',3X,
     $'ELASTIC',4X,'PLASTIC',5X,'TOTAL',6X,'STRESS',3X,'FUNCTION',5X,
     $'FACTOR',/)
C
C  EVALUATE YIELD FUNCTION FOR ELEMENTS REQUESTED AS OUTPUT
C
      DO 4 IE=1,NE
      IF(NELTPR(IE).NE.1) GO TO 4
      IF(IA.GT.-1) GO TO 8
      PHI(IE)=SIGBAR(IE,1)*SIGBAR(IE,1)-SIGBAR(IE,1)*SIGBAR(IE,2)
     $+SIGBAR(IE,2)*SIGBAR(IE,2)+3.*SIGBAR(IE,3)*SIGBAR(IE,3)
     $-SZEROS
      GO TO 4
    8 PHI(IE)=SIGBAR(IE,1)*SIGBAR(IE,1)+SIGBAR(IE,2)*SIGBAR(IE,2)
```

```
     $+SIGBAR(IE,4)*SIGBAR(IE,4)-SIGBAR(IE,1)*SIGBAR(IE,2)
     $-SIGBAR(IE,2)*SIGBAR(IE,4)-SIGBAR(IE,1)*SIGBAR(IE,4)
     $+3.*SIGBAR(IE,3)*SIGBAR(IE,3)-SZEROS
    4 CONTINUE
C
C  PRINT ELEMENT QUANTITIES FOR OUTPUT REQUESTED
C  *********************************************
C
      IF(IA) 9,13,10
C
C  PLANE STRESS ANALYSIS :-
C
    9 DO 12 IE=1,NE
      IF(NELTPR(IE).NE.1) GO TO 12
      IF(IPELTS(IE,1).EQ.0) GO TO 3
C
C                 PLASTIC ELEMENTS
C
      PRINT 207,DESTRN(IE,1),DPSTRN(IE,1),DSTRS(IE,1),ESTRN(IE,1),
     $PSTRN(IE,1),STRN(IE,1),STRS(IE,1),IE,DESTRN(IE,2),DPSTRN(IE,2),
     $PLAM(IE),DSTRS(IE,2),ESTRN(IE,2),PSTRN(IE,2),STRN(IE,2),STRS(IE,2)
     $,PHI(IE),CF(IE),DESTRN(IE,3),DPSTRN(IE,3),DSTRS(IE,3),ESTRN(IE,3),
     $PSTRN(IE,3),STRN(IE,3),STRS(IE,3)
  207 FORMAT(1H0,7X,'X ',2(1X,E10.5),11X,5(1X,E10.5),/,I5,3X,'Y ',
     $10(1X,E10.5),/,7X,'XY ',2(1X,E10.5),11X,5(1X,E10.5))
      GO TO 12
C
C                 ELASTIC ELEMENTS
C
    3 PRINT 205,DESTRN(IE,1),DSTRS(IE,1),ESTRN(IE,1),
     $PSTRN(IE,1),STRN(IE,1),STRS(IE,1),IE,DESTRN(IE,2),
     $DSTRS(IE,2),ESTRN(IE,2),PSTRN(IE,2),STRN(IE,2),STRS(IE,2),
     $PHI(IE),DESTRN(IE,3),DSTRS(IE,3),ESTRN(IE,3),PSTRN(IE,3),
     $STRN(IE,3),STRS(IE,3)
  205 FORMAT(1H0,7X,'X',2X,E10.5,5X,'--',15X,5(1X,E10.5),/,
     $I5,3X,'Y',2X,E10.5,5X,'--',9X,'--',4X,6(1X,E10.5),5X,
     $'--',/,7X,'XY',2X,E10.5,5X,'--',15X,5(1X,E10.5))
   12 CONTINUE
      GO TO 20
C
C  PLANE STRAIN ANALYSIS :-
C
   13 DO 21 IE=1,NE
      IF(NELTPR(IE).NE.1) GO TO 21
      IF(IPELTS(IE,1).EQ.0) GO TO 14
C
C                 PLASTIC ELEMENTS
C
      PRINT 212,DESTRN(IE,1),DPSTRN(IE,1),DSTRS(IE,1),ESTRN(IE,1),
     $PSTRN(IE,1),STRN(IE,1),STRS(IE,1),IE,DESTRN(IE,2),DPSTRN(IE,2),
     $PLAM(IE),DSTRS(IE,2),ESTRN(IE,2),PSTRN(IE,2),STRN(IE,2),STRS(IE,2)
     $,PHI(IE),CF(IE),DESTRN(IE,4),DPSTRN(IE,4),DSTRS(IE,4),ESTRN(IE,4),
     $PSTRN(IE,4),STRN(IE,4),STRS(IE,4),DESTRN(IE,3),DPSTRN(IE,3),
     $DSTRS(IE,3),ESTRN(IE,3),PSTRN(IE,3),STRN(IE,3),STRS(IE,3)
  212 FORMAT(1H0,7X,'X ',2(1X,E10.5),11X,5(1X,E10.5),/,I5,3X,'Y ',
     $10(1X,E10.5),/,8X,'Z ',2(1X,E10.5),11X,5(1X,E10.5),
     $/,7X,'XY ',2(1X,E10.5),11X,5(1X,E10.5))
      GO TO 21
C
C                 ELASTIC ELEMENTS
```

```
C
   14 PRINT 213,DESTRN(IE,1),DSTRS(IE,1),ESTRN(IE,1),
     $PSTRN(IE,1),STRN(IE,1),STRS(IE,1),IE,DESTRN(IE,2),
     $DSTRS(IE,2),ESTRN(IE,2),PSTRN(IE,2),STRN(IE,2),STRS(IE,2),
     $PHI(IE),DESTRN(IE,4),DSTRS(IE,4),ESTRN(IE,4),PSTRN(IE,4),
     $STRN(IE,4),STRS(IE,4),DESTRN(IE,3),DSTRS(IE,3),ESTRN(IE,3),
     $PSTRN(IE,3),STRN(IE,3),STRS(IE,3)
  213 FORMAT(1H0,7X,'X',2X,E10.5,5X,'--',15X,5(1X,E10.5),/,
     $I5,3X,'Y',2X,E10.5,5X,'--',9X,'--',4X,6(1X,E10.5),5X,
     $'--',/,8X,'Z',2X,E10.5,5X,'--',15X,5(1X,E10.5),
     $/,7X,'XY',2X,E10.5,5X,'--',15X,5(1X,E10.5))
   21 CONTINUE
      GO TO 20
C
C  AXISYMMETRIC ANALYSIS :-
C
   10 DO 22 IE=1,NE
      IF(NELTPR(IE).NE.1) GO TO 22
      IF(IPELTS(IE,1).EQ.0) GO TO 11
C
C              PLASTIC ELEMENTS
C
      PRINT 211,DESTRN(IE,1),DPSTRN(IE,1),DSTRS(IE,1),ESTRN(IE,1),
     $PSTRN(IE,1),STRN(IE,1),STRS(IE,1),IE,DESTRN(IE,2),DPSTRN(IE,2),
     $RLAM(IE),DSTRS(IE,2),ESTRN(IE,2),PSTRN(IE,2),STRN(IE,2),STRS(IE,2)
     $,PHI(IE),CF(IE),DESTRN(IE,4),DPSTRN(IE,4),DSTRS(IE,4),ESTRN(IE,4),
     $PSTRN(IE,4),STRN(IE,4),STRS(IE,4),DESTRN(IE,3),DPSTRN(IE,3),
     $DSTRS(IE,3),ESTRN(IE,3),PSTRN(IE,3),STRN(IE,3),STRS(IE,3)
  211 FORMAT(1H0,7X,'R ',2(1X,E10.5),11X,5(1X,E10.5),/,I5,3X,'Z ',
     $10(1X,E10.5),/,8X,'O ',2(1X,E10.5),11X,5(1X,E10.5),
     $/,1H+,7X,'-',/,7X,'RZ ',2(1X,E10.5),11X,5(1X,E10.5))
      GO TO 22
C
C              ELASTIC ELEMENTS
C
   11 PRINT 214,DESTRN(IE,1),DSTRS(IE,1),ESTRN(IE,1),
     $PSTRN(IE,1),STRN(IE,1),STRS(IE,1),IE,DESTRN(IE,2),
     $DSTRS(IE,2),ESTRN(IE,2),PSTRN(IE,2),STRN(IE,2),STRS(IE,2),
     $PHI(IE),DESTRN(IE,4),DSTRS(IE,4),ESTRN(IE,4),PSTRN(IE,4),
     $STRN(IE,4),STRS(IE,4),DESTRN(IE,3),DSTRS(IE,3),ESTRN(IE,3),
     $PSTRN(IE,3),STRN(IE,3),STRS(IE,3)
  214 FORMAT(1H0,7X,'R',2X,E10.5,5X,'--',15X,5(1X,E10.5),/,
     $I5,3X,'Z',2X,E10.5,5X,'--',9X,'--',4X,6(1X,E10.5),5X,
     $'--',/,8X,'O',2X,E10.5,5X,'--',15X,5(1X,E10.5),
     $/,1H+,7X,'-',/,7X,'RZ',2X,E10.5,5X,'--',15X,5(1X,E10.5))
   22 CONTINUE
C
C  PRINT CURRENT PLASTIC ELEMENTS
C
   20 PRINT 203
  203 FORMAT(1H0,/,' CURRENT PLASTIC ELEMENTS :-',/,27('-'))
      IF(NPPTS.NE.0) GO TO 18
      PRINT 210
  210 FORMAT(1H0,'NIL')
      GO TO 19
   18 PRINT 204,(IPPT(I),I=1,NPPTS)
  204 FORMAT(1H0,25I5,//,10(1X,25I5,//))
C
C  PRINT HEADINGS FOR NODE QUANTITIES
C
```

```
   19 IF(IA.LT.1) PRINT 208
  208 FORMAT(1H0,/,13X,'LOAD INCREMENT',16X,'TOTAL LOAD',12X,'DISPLACEME
     $NT INCREMENT',8X,'TOTAL DISPLACEMENT',/,5X,4(3X,'/',23('-'),'/'),/,
     $' NODE',7X,'DPX',11X,'DPY',11X,'PX',12X,'PY',12X,'DU',12X,'DV',
     $13X,'U',13X,'V',/)
      IF(IA.EQ.1) PRINT 215
  215 FORMAT(1H0,/,13X,'LOAD INCREMENT',16X,'TOTAL LOAD',12X,'DISPLACEME
     $NT INCREMENT',8X,'TOTAL DISPLACEMENT',/,5X,4(3X,'/',23('-'),'/'),/,
     $' NODE',7X,'DPR',11X,'DPZ',11X,'PR',12X,'PZ',12X,'DU',12X,'DV',
     $13X,'U',13X,'V',/)
C
C  PRINT NODE QUANTITIES FOR OUTPUT REQUESTED
C  *****************************************
C
      DO 5 I=1,NN
      IF(NODEPR(I).NE.1) GO TO 5
      NR=(I-1)*2
      J=NR+1
      L=NR+2
      PRINT 209,I,DP(J),DP(L),P(J),P(L),DU(J),DU(L),U(J),U(L)
  209 FORMAT(1H ,I4,8(3X,E11.6))
    5 CONTINUE
C
C  CHECK FOR TERMINATING ANALYSIS
C
    7 I=IFIX(PMAX(IP,1))
      IF(I.EQ.-1.OR.TOUT(NOUT).EQ.-1) ISTOP=-1
      RETURN
      END
```

A.6 EPCQI Program Listing

COL*EPCOT(1).MAIN

```
      COMPILER (XM=3)
C
C  SET VALUES OF PARAMETERS
C
      PARAMETER  NDFP = 384
      PARAMETER  NECP = 64
      PARAMETER  NPCP = 30
      PARAMETER  NNP  = 192
      PARAMETER  NEP  = 31
      PARAMETER  IOP  = 3
      PARAMETER  ISP  = 3
      PARAMETER  NLPP = 2
      PARAMETER  IOSP = IOP * IOP
C
      COMMON /EXT/ RKE(NDFP,NECP),RKP(NDFP,NPCP)
      COMMON IA,IO,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPC,NPCS,
     2NDFBWS,LP,IL,ISTOP,ICHECK,F,FP,EPS,RNU,SZERO,SZEROS,THIK,
     3RMIN,IS,IOS,IOSM,NLI,GP(4,4),WGT(4,4),NWRKE,NWRKP,
     4ETA,NOUT,LU1,LU2,LU3,LU4,NLAM,FRAC
C
C  SET LOGICAL UNIT NUMBERS OF TEMPORARY FILES
C
      LU1  =  11
      LU2  =  12
      LU3  =  13
      LU4  =  14
C
C  INITIALIZE ARRAYS
C
      DIMENSION PP(NLPP,NDFP),COORD(NNP,2),IELT(NEP,12),IBC(NDFP),
     2PMAX(NLPP,2),IOUT(100),NELTPR(NEP,IOSP),NODEPR(NNP),
     3H(IOSP,12),DH(IOSP,12,2),RMINC(NLPP),DIAG2(NPCP),
     4U(NDFP),FSTRN(NEP,IOSP,ISP),PSTRN(NEP,IOSP,ISP),P(NDFP),
     5STRS(NEP,IOSP,ISP),SIGBAR(NEP,IOSP,ISP),IPELTS(NEP,IOSP,3),
     6RLAM(NEP,IOSP),A(NEP,IOSP,ISP),DIAG(NPCP),QQ(NPCP),RK3(NPCP,NPCP),
     7CF(NEP,IOSP),DP(NDFP),DU(NDFP),DESTRN(NEP,IOSP,ISP),IPPT(NDFP,2),
     8DPSTRN(NEP,IOSP,ISP),DSTRN(NEP,IOSP,ISP),DSTRS(NEP,IOSP,ISP),
     9STRN(NEP,IOSP,ISP),PHI(NEP,IOSP),BTDB(24,24),D(4,4),
     9GPCRD(NEP,IOSP,2),BTDN(24),NEWP(50,2),DET(NEP,IOSP)
C
C  READ STRUCTURE DATA
C
      CALL DATA (NEP,NNP,NDFP,IOSP,ISP,NECP,NPCP,NLPP,
     $PP,COORD,IELT,IBC,PMAX,IOUT,NELTPR,NODEPR,DET,H,DH)
C
C  CALCULATE AND STORE IN TEMPORARY FILES, ELASTIC STIFFNESS MATRIX  (BTDB)
C  AND STRAIN-DISPLACEMENT MATRIX  (B')  FOR EACH INTEGRATION POINT
C
      CALL ELTMAT (NEP,NNP,NDFP,IOSP,ISP,NECP,NPCP,NLPP,
     $PP,COORD,IELT,IBC,D,DET,H,DH,GPCRD,BTDB)
C
C  INITIALIZE COUNTERS FOR FIRST LOAD INCREMENT: LINEAR ELASTIC ANALYSIS
C
      LP=1
      NOUT=1
      NPPTS=0
      NPC=0
C
C  SOLVE FOR DISPLACEMENT INCREMENTS
```

```
C
    3 CALL SOLVE1 (NDFP,NPCP,NLPP,PP,DU,DIAG,QQ,RK3)
C
C  DETERMINE PLASTIC INTEGRATION POINTS
C
    1 CALL STRSTR (NEP,NNP,NDFP,IOSP,ISP,NECP,NPCP,NLPP,
     $P,U,ESTRN,PSTRN,STRN,STRS,SIGBAR,PP,IELT,PMAX,IOUT,
     $NELTPR,NODEPR,D,DET,IPELTS,RLAM,A,CF,DP,NEWP,NNEWP,
     $DU,DESTRN,DPSTRN,DSTRN,DSTRS,GPCRD,IPPT,PHI,RMINC)
      IF(ISTOP.EQ.-1) GO TO 2
      IF(NPPTS.EQ.0) GO TO 3
C
C  REWIND TEMPORARY FILE IN WHICH INTEGRATION POINT DEFORMATION
C  MATRICES  (B')  ARE STORED
C
      CALL NTRAN(LU4,10,22)
C
C  EVALUATE PLASTIC MATRICES  (-BTDN)  AND  (NTDN+DP)
C
      DO 4 J=1,NPC
      DIAG(J)=0.
      DO 4 I=1,NDF
    4 RKP(I,J)=0.
      NW=0
      NNW=0
      DO 5 IE=1,NE
      DO 5 IG=1,IOS
      IF(IPELTS(IE,IG,1).EQ.0) GO TO 5
      N=NW-NNW
      NNW=NW+4
      CALL PLASM (NEP,IOSP,ISP,IE,IG,IPELTS,A,SIGBAR,
     $D,GPCRD,IELT,DET,WT,BTDN,ENTDN,N)
      NC=IPELTS(IE,IG,2)
      DO 6 I=1,12
      NRR=(IELT(IE,I)-1)*2
      NRE=(I-1)*2
      DO 6 J=1,2
    6 RKP(NRR+J,NC)=RKP(NRR+J,NC)-BTDN(NRE+J)*WT
      DIAG(NC)=DIAG(NC)+ENTDN*WT
    5 NW=NW+4
C
C  APPLY BOUNDARY CONDITIONS TO PLASTIC MATRIX  (-BTDN)
C
      DO 7 I=1,NDF
      IF(IBC(I).EQ.1) GO TO 7
      DO 8 J=1,NPC
    8 RKP(I,J)=0.
    7 CONTINUE
C
C  SOLVE, CHECK KINEMATIC CONSTRAINTS ON ELASTIC AND PLASTIC REGIONS
C  AND ITERATE IF NECESSARY
C
      CALL ITER (NEP,NNP,NDFP,IOSP,ISP,NECP,NPCP,NLPP,
     $IELT,IBC,D,DET,IPELTS,RLAM,A,PP,DU,NEWP,NNEWP,DIAG2,RMINC,
     $QQ,GPCRD,SIGBAR,BTDN,ESTRN,U,P,IPPT,DSTRS,DSTRN,DPSTRN,DIAG,
     $DESTRN,DP,CF,PMAX,IOUT,NELTPR,NODEPR,PSTRN,STRN,STRS,RK3)
C
C  APPLY NEXT LOAD INCREMENT
C
      GO TO 1
```

```
2 STOP
  END
```

```
      SUBROUTINE DATA (NEP,NNP,NDFP,IOSP,ISP,NFCP,NPCP,NLPP,
     $PP,COORD,IELT,IBC,PMAX,IOUT,NELTPR,NODEPR,DET,H,DH)
      COMMON IA,IO,NN,NE,NPNTS,NDFS,NDF,NDFA,NBWS,NBW,NPC,NPCS,
     2NDFBWS,LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,
     3RMIN,IS,IOS,IOSM,NLI,GP(4,4),WGT(4,4),NWRKE,NWRKP,
     4ETA,NOUT,LU1,LU2,LU3,LU4,NLAM,FRAC
      DIMENSION PP(NLPP,NDFP),COORD(NNP,2),IELT(NEP,12),IBC(NDFP),
     2PMAX(NLPP,2),IOUT(100),NELTPR(NEP,IOSP),NODEPR(NNP),DET(NEP,IOSP),
     3H(IOSP,12),DH(IOSP,12,2),PPP(2),TITLE(12),IIBC(2),
     4CD(12,2),FX(2),DPP(12,2),DERIV(2)
      DATA GP / 0.0, -.57735027, -.77459667, -.86113631,
     $0.0, .57735027, 0.0, -.33998104, 0.0, 0.0, .77459667,
     $.33998104, 0.0, 0.0, 0.0, .86113631 /
      DATA WGT /2.0, 1.0, .55555556, .34785485, 0.0, 1.0,
     $.88888889, .65214515, 0.0, 0.0, .55555556, .65214515,
     $0.0, 0.0, 0.0, .34785485 /
C
C  READ AND PRINT HEADING
C
      READ 101,TITLE
  101 FORMAT(12A6)
      PRINT 201,TITLE
      READ 101,TITLE
  201 FORMAT(1H1,12A6,/)
  100 FORMAT()
C
C  READ TYPE OF ANALYSIS
C
      READ 104,I,J
  104 FORMAT(2A6)
      IF(J.EQ.'STRESS') IA=-1
      IF(J.EQ.'STRAIN') IA=0
      IF(J.EQ.'METRIC') IA =1
      IF(IA.NE.0) GO TO 17
      IF(J.EQ.'STRAIN') GO TO 17
      PRINT 223
  223 FORMAT(1H0,38('*'),/,' TYPE OF ANALYSIS INCORRECTLY SPECIFIED',/,
     $' ANALYSIS TERMINATED',/,39('*')).
      STOP
C
C  READ CONSTANTS :  NUMBER OF NODES,  NUMBER OF ELEMENTS,
C  ELASTIC MODULUS,  PLASTIC MODULUS,  POISSON'S RATIO,
C  UNIAXIAL YIELD STRESS,  RATIO OF VON MISES EQUIVALENT
C  STRESS TO YIELD STRESS FOR STRESS POINT TO BE TREATED
C  AS PLASTIC,  INTEGRATION ORDER FOR GAUSS QUADRATURE,
C  PLATE THICKNESS IF PLANE STRESS ANALYSIS
C
   17 IF(IA.EQ.-1) READ 100,NN,NE,E,EP,RNU,SZERO,ETA,IO,NLAM,THIK,FRAC
      IF(IA.GT.-1) READ 100,NN,NE,E,EP,RNU,SZERO,ETA,IO,NLAM,FRAC
      IF(IA.EQ.0) THIK=1.
      IS=4
      IF(IA.EQ.-1) IS=3
      EPS=EP*EP
      SZEROS=SZERO*SZERO
      IOS=IO*IO
      IOSM=IOS-1
C
C  READ ELEMENT INCIDENCES AND NODAL COORDINATES
C
```

```
      DO 13 IE=1,NE
      READ 100,IE,(IELT(IE,J),J=1,12),ICS
      IF(ICS.EQ.1) GO TO 8
      READ 100,((CD(I,J),J=1,2),I=1,12)
      DO 10 J=1,12
      L=IELT(IE,J)
      COORD(L,1)=CD(J,1)
   10 COORD(L,2)=CD(J,2)
      GO TO 13
    8 READ 100,((CD(I,J),J=1,2),I=1,4)
      L=4
      DO 14 I=1,4
      J=I+1
      IF(J.EQ.5) J=1
      II=IELT(IE,I)
      JJ=IELT(IE,J)
      L=L+1
      LL=IELT(IE,L)
      L=L+1
      LLL=IELT(IE,L)
      DO 45 K=1,2
      DUM=(CD(J,K)-CD(I,K))/3.
      COORD(II,K)=CD(I,K)
      COORD(JJ,K)=CD(J,K)
      COORD(LL,K)=CD(I,K)+DUM
   45 COORD(LLL,K)=COORD(LL,K)+DUM
   14 CONTINUE
   13 CONTINUE
C
C  DETERMINE CONSTANTS FOR SOLUTION ROUTINE AND HALF BAND-WIDTH OF ELASTIC MATRIX
C
      NBW=0
      DO 12 IE=1,NE
      DO 12 I=1,11
      II=I+1
      DO 12 J=II,12
      MAX=ABS(IELT(IE,I)-IELT(IE,J))
   12 IF(NBW.LT.MAX) NBW=MAX
      NBW=(NBW+1)*2
      NDF=NN*2
      NDFA=NDF+1
      NDFS=NDF-1
      NBWS=NBW-1
      NDFBWS=NDF-NBWS
      NWRKE=NDF*NBW
C
C  PRINT HEADINGS AND VALUES OF CONSTANTS
C
      PRINT 212
  212 FORMAT(//,' NUMERICAL ANALYSIS USING TWO DIMENSIONAL TWELVE NODE
     $ISOPARAMETRIC FINITE ELEMENTS',//,' PLANE STRESS, PLANE STRAIN OR
     $AXISYMMETRIC PROGRAM',//,' CUBIC INTERPOLATION FUNCTIONS',//,
     $' QUADRATIC STRAIN VARIATION',//,' NUMERICAL INTEGRATION USING GAU
     $SS QUADRATURE',/)
      PRINT 227,IU
  227 FORMAT(1H0,'ORDER OF INTEGRATION =',I4,/)
      IF(IA.EQ.-1) PRINT 215
  215 FORMAT(1H0,'PLANE STRESS ANALYSIS',/)
      IF(IA.EQ.0) PRINT 222
  222 FORMAT(1H0,'PLANE STRAIN ANALYSIS',/)
```

```
      IF(IA.EQ.1) PRINT 203
  203 FORMAT(1H0,'AXISYMMETRIC ANALYSIS',/)
      PRINT 211
  211 FORMAT(1H0,'VON MISES YIELD CONDITION',/)
      IF(EP.NE.0.0) PRINT 214
  214 FORMAT(1H0,'KINEMATIC HARDENING',/)
      IF(EP.EQ.0.0) PRINT 221,FRAC
  221 FORMAT(1H0,'ELASTIC, PERFECTLY PLASTIC :  ANALYSIS TERMINATED WHEN
     2 LOAD INCREMENT MAGNITUDE DECREASES TO',F6.3,'  OF TOTAL LOAD',/)
      PRINT 247,ETA
  247 FORMAT(1H0,'INTEGRATION POINTS WITH VON MISES EQUIVALENT STRESS WI
     $THIN',F6.3,'  OF YIELD STRESS TREATED AS PLASTIC',/)
      PRINT 220, TITLE
  220 FORMAT(1H0,12A6,/)
      I=NLPP-1
      PRINT 213,NNP,NEP,NDFP,NECP,NPCP,I
  213 FORMAT(1H0,'STORAGE AVAILABLE   :',19X,'MAXIMUM NUMBER OF NODES :'
     $,I5,/,20X,':',16X,'MAXIMUM NUMBER OF ELEMENTS :',I5,/,20X,
     $':',6X,'MAXIMUM NUMBER OF DEGREES OF FREEDOM :',I5,/,20X,
     $': MAXIMUM HALF BAND-WIDTH OF ELASTIC MATRIX :',I5,/,20X,':',8X,
     $'MAXIMUM NUMBER OF PLASTIC ELEMENTS :',I5,/,20X,': MAXIMUM NUMBER
     $OF PROPORTIONAL LOAD PATHS :',I5,/)
      PRINT 202,NN,NE,NDF,NBW
  202 FORMAT(1H0,'STORAGE REQUIRED    :',27X,'NUMBER OF NODES :',I5,
     $/,20X,':',24X,'NUMBER OF ELEMENTS :',I5,/,20X,':',14X,
     $'NUMBER OF DEGREES OF FREEDOM :',I5,/,20X,':',9X,
     $'HALF BAND-WIDTH OF ELASTIC MATRIX :',I5,/)
      PRINT 217,E,EP,RNU,SZERO
  217 FORMAT(1H0,'MATERIAL CONSTANTS : ELASTIC MODULUS  =',E11.6,
     $/,20X,': PLASTIC MODULUS  =',E11.6,/,20X,': POISSONS RATIO   =',
     $F11.6,/,20X,': YIELD STRESS      =',E11.6,/)
      IF(IA.EQ.-1) PRINT 216,THIK
  216 FORMAT(1H0,'PLATE THICKNESS    :',F11.6,/)
C
C  CHECK AVAILABLE STORAGE
C
      IF(NDF.LE.NDFP) GO TO 52
      PRINT 231
  231 FORMAT(1H0,60('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO FEW RO
     $WS IN ELASTIC MATRIX',/,61('*'))
      STOP
   52 IF(NBW.LE.NECP) GO TO 46
      PRINT 232
  232 FORMAT(1H0,63('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO FEW CO
     $LUMNS IN ELASTIC MATRIX',/,64('*'))
      STOP
   46 IF(NN.LE.NNP) GO TO 47
      PRINT 233
  233 FORMAT(1H0,44('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO MANY N
     $ODES',/,45('*'))
      STOP
   47 IF(NE.LE.NEP) GO TO 48
      PRINT 234
  234 FORMAT(1H0,47('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO MANY E
     $LEMENTS',/,48('*'))
      STOP
   48 IF(IO*IO.LE.IOSP) GO TO 49
      PRINT 235
  235 FORMAT(1H0,56('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  INTEGRATIO
     $N ORDER TOO HIGH',/,57('*'))
```

```
      STOP
   49 IF(IS.LE.ISP) GO TO 58
      PRINT 246
  246 FORMAT(1H0,65('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO MANY R
     $OWS IN DEFORMATION MATRIX',/,66('-'))
      STOP
C
C  PRINT NODAL COORDINATES
C
   58 IF(IA.LT.1) PRINT 204
  204 FORMAT(1H0,'COORDINATES OF NODES',//,' NODE',6X,'X',11X,'Y',
     $3(10X,'NODE',6X,'X',11X,'Y'))
      IF(IA.EQ.1) PRINT 224
  224 FORMAT(1H0,'COORDINATES OF NODES',//,' NODE',6X,'R',11X,'Z',
     $3(10X,'NODE',6X,'R',11X,'Z'))
      PRINT 205,((I,(COORD(I,J),J=1,2)),I=1,NN)
  205 FORMAT(500(1H ,I3,2E12.5,3(6X,I3,2E12.5),/))
C
C  PRINT ELEMENT INCIDENCES
C
      PRINT 206
  206 FORMAT(//,' ELEMENT',28X,'NODES',/,9X,'/',68('-'),'/',/)
      PRINT 207,((I,(IELT(I,J),J=1,12)),I=1,NE)
  207 FORMAT(1H ,I4,1X,12I6)
C
C  READ AND PRINT BOUNDARY CONDITIONS
C
      IF(IA.LT.1) PRINT 208
  208 FORMAT(1H0,' BOUNDARY CONDITIONS :  0  -  CONSTRAINED',
     $/,25X,'1  -  UNCONSTRAINED',/,' NODE',3X,'X,Y',/)
      IF(IA.EQ.1) PRINT 225
  225 FORMAT(1H0,' BOUNDARY CONDITIONS :  0  -  CONSTRAINED',
     $/,25X,'1  -  UNCONSTRAINED',/,' NODE',3X,'R,Z',/)
      DO 1 I=1,NDF
    1 IBC(I)=1
    7 READ 100,I
      IF(I.EQ.-1) GO TO 2
      READ(0,100) (IIBC(J),J=1,2)
      READ 100,I
      K=1
    3 K=K+1
      READ(0,100) (IOUT(J),J=1,K)
      IF(IOUT(K).NE.-1) GO TO 3
      K=K-1
      DO 4 I=1,K
      PRINT 209,IOUT(I),(IIBC(J),J=1,2)
  209 FORMAT(1H ,I3,4X,I1,',',I1)
      II=(IOUT(I)-1)*2
      DO 4 J=1,2
    4 IBC(II+J)=IIBC(J)
      GO TO 7
C
C  READ AND PRINT LOADING PROGRAMME
C
    2 PRINT 210
  210 FORMAT(//,' LOADING PROGRAMME:')
    9 READ 100,I
      IK=IK+1
      IF(IK.LE.NLPP) GO TO 18
      PRINT 226
```

```
  226 FORMAT(1H0,49('*'),/,1H ,'AVAILABLE STORAGE EXCEEDED :  TOO MANY L
     $OAD PATHS',/,50('*'))
      STOP
   18 IF(I.EQ.-1) GO TO 5
      READ(0,100) IJ,I,J,PMAX(IJ,2)
      L=(I-1)*2+J
      PMAX(IJ,1)=FLOAT(L)
      IF(IA.LT.1.AND.J.EQ.1) JP='X'
      IF(IA.LT.1.AND.J.EQ.2) JP='Y'
      IF(IA.EQ.1.AND.J.EQ.1) JP='R'
      IF(IA.EQ.1.AND.J.EQ.2) JP='Z'
      PRINT 218,IJ,IJ,I,JP,PMAX(IJ,2)
  218 FORMAT(//,' LOADING',I4,'  : END LOADING',I3,
     $'  WHEN TOTAL LOAD AT NODE',I5,'  IN ',A1,'-DIRECTION IS  'E11.6)
   43 READ 100,I
      IF(I.LE.0) GO TO 19
C
C  FOR DISTRIBUTED LOAD CALCULATE EQUIVALENT NODAL LOADS
C
      READ(0,100) IE,ISIDE,FX(1),FX(2),ILD
      R=1.
      S=1.
      GO TO (23,24,23,24),ISIDE
   23 IF(ISIDE.EQ.3) S=-1.
      IDERIV=1
      DO 34 IX=1,IO
      R=GP(IO,IX)
   34 CALL HMAT (IOSP,H,DH,IX,R,S)
      GO TO 35
   24 IF(ISIDE.EQ.2) R=-1.
      IDERIV=2
      DO 36 IY=1,IO
      S=GP(IO,IY)
   36 CALL HMAT (IOSP,H,DH,IY,R,S)
   35 DO 44 I=1,12
      DO 44 J=1,2
   44 DPP(I,J)=0.
      IF(ILD.NE.1) RAD=FX(2)/FX(1)
      DO 42 IG=1,IO
      DO 38 I=1,2
      DERIV(I)=0.
      DO 38 K=1,12
      L=IELT(IE,K)
   38 DERIV(I)=DERIV(I)+DH(IG,K,IDERIV)*COORD(L,I)
      DUM=SQRT(DERIV(1)*DERIV(1)+DERIV(2)*DERIV(2))
      IF(IA.NE.1) GO TO 40
      THIK=0.
      DO 41 I=1,12
      L=IELT(IE,I)
   41 THIK=THIK+H(IG,I)*COORD(L,1)
      THIK=THIK*6.2831853
   40 WT=WGT(IO,IG)*THIK*DUM
      IF(ILD.EQ.1) GO TO 55
      DO 56 I=1,2
      FX(I)=0.
      DO 57 J=1,12
      K=IELT(IE,J)
   57 FX(I)=FX(I)+H(IG,J)*COORD(K,I)
   56 FX(I)=FX(I)*RAD
   55 DO 42 I=1,12
```

```
      DO 42 J=1,2
   42 DPP(I,J)=DPP(I,J)+H(IG,I)*FX(J)*WT
      DO 37 I=1,12
      II=IFLT(IE,I)*2
      PP(IJ,II-1)=PP(IJ,II-1)+DPP(I,1)
   37 PP(IJ,II)=PP(IJ,II)+DPP(I,2)
      GO TO 43
   19 READ 100,I
      IF(I.LE.0) GO TO 15
      READ(0,100) N1,N2
      IF(N1.LE.0) GO TO 15
      READ 100,(PPP(J),J=1,2)
      JJ=2*(N1-1)
      II=N2-N1+1
      DO 16 I=1,II
      DO 16 J=1,2
      JJ=JJ+1
   16 PP(IJ,JJ)=PP(IJ,JJ)+PPP(J)
      GO TO 19
C
C APPLY BOUNDARY CONDITIONS TO LOAD VECTOR
C
   15 DO 11 I=1,NDF
   11 IF(IBC(I).EQ.0) PP(IJ,I)=0.
C
C PRINT DIRECTION OF LOAD VECTOR
C
      PRINT 219,IJ
  219 FORMAT(/,14X,': COMPONENTS OF PROPORTIONAL LOAD VECTOR',I3,
     $/,16X,38('-'),/)
      PRINT 228,(PP(IJ,I),I=1,NDF)
  228 FORMAT(1H ,200(12E10.4,/,1H ))
      GO TO 9
    5 PMAX(IK,1)=-1
C
C READ AND PRINT OUTPUT REQUESTED
C
      PRINT 239
  239 FORMAT(/,1H0,'REQUESTED OUTPUT OF RESULTS :-',/,30('-'),/)
      READ 103,JP
  103 FORMAT(A2)
      IF(JP.NE.'AL') GO TO 6
      DO 28 IE=1,NE
      DO 28 IG=1,IOS
   28 NELTPR(IE,IG)=1
      DO 29 I=1,NN
   29 NODEPR(I)=1
      PRINT 240
  240 FORMAT(' PRINT RESULTS FOR ALL INTEGRATION POINTS AND ALL NODES'
     $,/)
      GO TO 30
    6 READ 100,I
      IF(I.EQ.-1) GO TO 26
      K=1
   21 K=K+2
      READ(0,100) (IOUT(J),J=1,K)
      IF(IOUT(K).NE.-1) GO TO 21
      K=K-2
      DO 25 I=1,K,2
      J=IOUT(I)
```

```
      L=IOUT(I+1)
   25 NELTPR(J,L)=1
      K=K+1
      PRINT 241
  241 FORMAT(1H ,'PRINT RESULTS FOR :-  (ELEMENT : INTEGRATION POINT)')
      PRINT 242,(IOUT(J),J=1,K)
  242 FORMAT(1H ,14(I3,':',I2,2X),//,10(1X,14(I3,':',I2,2X),//))
   26 READ 100,I
      IF(I.EQ.-1) GO TO 30
      K=1
   31 K=K+1
      READ(0,100) (IOUT(J),J=1,K)
      IF(IOUT(K).NE.-1) GO TO 31
      K=K-1
      DO 27 I=1,K
      J=IOUT(I)
   27 NODEPR(J)=1
      PRINT 243
  243 FORMAT(1H0,'PRINT RESULTS FOR NODES :-')
      PRINT 244,(IOUT(J),J=1,K)
  244 FORMAT(1H ,33I4)
   30 READ 100,I
      K=1
   32 K=K+1
      READ(0,100) (IOUT(J),J=1,K)
      IF(IOUT(K).NE.-2) GO TO 32
      PRINT 245
  245 FORMAT(1H0,'RESULTS REQUESTED AFTER LOAD INCREMENTS :-')
      K=K-1
      IF(IOUT(K).EQ.-1) K=K-1
      PRINT 244,(IOUT(J),J=1,K)
      RETURN
      END
```

COL*FPCQI(1).HMAT

```
      SUBROUTINE HMAT (IOSP,H,DH,IG,R,S)
      DIMENSION H(IOSP,12),DH(IOSP,12,2)
      RP=1.+R
      SP=1.+S
      RM=1.-R
      SM=1.-S
      RRM=1.-R*R
      SSM=1.-S*S
C
C  EVALUATE SHAPE FUNCTIONS AND DERIVATIVES
C
      R3P=1.+3.*R
      S3P=1.+3.*S
      R3M=1.-3.*R
      S3M=1.-3.*S
      R2=2.*R
      S2=2.*S
      R18=18.*R
      S18=18.*S
      BR=9.*(R*R+S*S)-10.
      H(IG,1)=RP*SP*BR
      H(IG,2)=RM*SP*BR
      H(IG,3)=RM*SM*BR
      H(IG,4)=RP*SM*BR
      H(IG,5)=SP*RRM*R3P
      H(IG,6)=SP*RRM*R3M
      H(IG,7)=RM*SSM*S3P
      H(IG,8)=RM*SSM*S3M
      H(IG,9)=SM*RRM*R3M
      H(IG,10)=SM*RRM*R3P
      H(IG,11)=RP*SSM*S3M
      H(IG,12)=RP*SSM*S3P
      DH(IG,1,1)=SP*BR+R18*RP*SP
      DH(IG,1,2)=RP*BR+S18*RP*SP
      DH(IG,2,1)=-SP*BR+R18*RM*SP
      DH(IG,2,2)=RM*BR+S18*RM*SP
      DH(IG,3,1)=-SM*BR+R18*RM*SM
      DH(IG,3,2)=-RM*BR+S18*RM*SM
      DH(IG,4,1)=SM*BR+R18*RP*SM
      DH(IG,4,2)=-RP*BR+S18*RP*SM
      DH(IG,5,1)=-R2*SP*R3P+3.*SP*RRM
      DH(IG,5,2)=RRM*R3P
      DH(IG,6,1)=-R2*SP*R3M-3.*SP*RRM
      DH(IG,6,2)=RRM*R3M
      DH(IG,7,1)=-SSM*S3P
      DH(IG,7,2)=-S2*RM*S3P+3.*RM*SSM
      DH(IG,8,1)=-SSM*S3M
      DH(IG,8,2)=-S2*RM*S3M-3.*RM*SSM
      DH(IG,9,1)=-R2*SM*R3M-3.*SM*RRM
      DH(IG,9,2)=-RRM*R3M
      DH(IG,10,1)=-R2*SM*R3P+3.*SM*RRM
      DH(IG,10,2)=-RRM*R3P
      DH(IG,11,1)=SSM*S3M
      DH(IG,11,2)=-S2*RP*S3M-3.*RP*SSM
      DH(IG,12,1)=SSM*S3P
      DH(IG,12,2)=-S2*RP*S3P+3.*RP*SSM
      DUM=1./32.
      DO 1 I=1,4
      H(IG,I)=H(IG,I)*DUM
```

```
      DO 1 J=1,2
    1 DH(IG,I,J)=DH(IG,I,J)*DUM
      DUM=9.*DUM
      DO 2 I=5,12
      H(IG,I)=H(IG,I)*DUM
      DO 2 J=1,2
    2 DH(IG,I,J)=DH(IG,I,J)*DUM
      RETURN
      END
```

```
COL*EPCOT(1).ELTMAT
      COMPILER (XM=3)
      SUBROUTINE ELTMAT (NEP,NNP,NDFP,IOSP,ISP,NECP,NPCP,NLPP,
     $PP,COORD,IELT,IBC,D,DET,H,DH,GPCRD,BTDB)
C
C  FOR EXTENDED STORAGE GIVE EXPLICIT SIZE OF ELASTIC AND PLASTIC ARRAYS
C
       COMMON /EXT/ RKE(384,64) , RKP(384,30)
C
      COMMON IA,IO,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPC,NPCS,
     2NDFBWS,LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,
     3RMIN,IS,IOS,IOSM,NLI,GP(4,4),WGT(4,4),NWRKE,NWRKP,
     4ETA,NOUT,LU1,LU2,LU3,LU4,NLAM,FRAC
      DIMENSION PP(NLPP,NDFP),COORD(NNP,2),IELT(NEP,12),IBC(NDFP),
     2D(4,4),DET(NEP,IOSP),H(IOSP,12),DH(IOSP,12,2),GPCRD(NEP,IOSP,2)
      DIMENSION B(4,24),BTD(4),BTDB(24,24),RJ(2,2),RJI(2,2)
      DOUBLE PRECISION FACT
      IF(IA.NE.-1) GO TO 18
C
C  IF PLANE STRESS ANALYSIS, EVALUATE ELEMENT ELASTICITY MATRIX  (D')
C
      DUM=E/(1.-RNU*RNU)
      D(1,1)=DUM
      D(1,2)=RNU*DUM
      D(2,2)=DUM
      D(2,1)=D(1,2)
      D(3,3)=(1.-RNU)/2.*DUM
      GO TO 17
C
C  IF PLANE STRAIN OR AXISYMMETRIC ANALYSIS, EVALUATE ELASTICITY MATRIX  (D')
C
   18 DUM=E*(1.-RNU)/(1.+RNU)/(1.-2.*RNU)
      D(1,1)=DUM
      D(1,2)=RNU/(1.-RNU)*DUM
      D(1,4)=D(1,2)
      D(2,1)=D(1,2)
      D(2,2)=DUM
      D(2,4)=D(1,2)
      D(3,3)=E/2./(1.+RNU)
      D(4,1)=D(1,2)
      D(4,2)=D(1,2)
      D(4,4)=DUM
C
C  EVALUATE SHAPE FUNCTIONS AND DERIVATIVES
C
   17 IG=0
      DO 12 IX=1,IO
      R=GP(IO,IX)
      DO 12 IY=1,IO
      S=GP(IO,IY)
      IG=IG+1
   12 CALL HMAT (IOSP,H,DH,IG,R,S)
C
C  FOR EACH ELEMENT :
C ********************
C
      DO 2 IE=1,NE
      DO 1 I=1,24
      DO 1 J=1,24
    1 BTDB(I,J)=0.
```

```
      IG=0
C
C  FOR EACH INTEGRATION POINT :
C *******************************
C
      DO 4 IX=1,ID
      DO 4 IY=1,ID
      IG=IG+1
C
C  EVALUATE DETERMINANT OF JACOBIAN MATRIX
C
      DO 15 I=1,2
      DO 15 J=1,2
      RJ(I,J)=0.
      DO 15 K=1,12
      L=IELT(IE,K)
   15 RJ(I,J)=RJ(I,J)+DH(IG,K,I)*COORD(L,J)
      DET(IE,IG)=RJ(1,1)*RJ(2,2)-RJ(1,2)*RJ(2,1)
      IF(DET(IE,IG).GT.0.1E-07) GO TO 16
      PRINT 200,IE,R,S
  200 FORMAT(1H1,41('*'),//,' ERROR : JACOBIAN NEGATIVE FOR ELEMENT ',
     $I4,//,' INTEGRATION POINT (R,S) =',F10.8,' , ',F10.8,//42('*'))
      STOP
C
C  EVALUATE INVERSE OF JACOBIAN MATRIX
C
   16 DUM=1./DET(IE,IG)
      RJT(1,1)=RJ(2,2)*DUM
      RJT(1,2)=-RJ(1,2)*DUM
      RJT(2,1)=-RJ(2,1)*DUM
      RJT(2,2)=RJ(1,1)*DUM
      K=-1
C
C  EVALUATE STRAIN-DISPLACEMENT MATRIX  (B')
C
      DO 20 I=1,12
      K=K+2
      L=K+1
      B(1,K)=0.
      B(1,L)=0.
      B(2,K)=0.
      B(2,L)=0.
      DO 21 J=1,2
      B(1,K)=B(1,K)+RJT(1,J)*DH(IG,I,J)
   21 B(2,L)=B(2,L)+RJT(2,J)*DH(IG,I,J)
      B(3,K)=B(2,L)
   20 B(3,L)=B(1,K)
C
C  CALCULATE CARTESIAN COORDINATES OF GAUSS INTEGRATION POINTS
C
      GPCRD(IE,IG,1)=0.
      GPCRD(IE,IG,2)=0.
      DO 22 I=1,12
      I1=IELT(IE,I)
      DO 22 J=1,2
   22 GPCRD(IE,IG,J)=GPCRD(IE,IG,J)+H(IG,I)*COORD(I1,J)
C
C  IF PLANE STRAIN OR AXISYMMETRIC ANALYSIS EVALUATE FOURTH ROW OF  (B')  MATRIX
C
      IF(IA) 28,29,23
```

```
   23 IF(GPCRD(IE,IG,1).GT..1E-7) GO TO 25
      DO 24 I=1,24
   24 B(4,I)=B(1,I)
      GO TO 28
   25 DUM=1./GPCRD(IE,IG,1)
      J=0
      DO 26 I=1,12
      J=J+2
      B(4,J-1)=H(IG,I)*DUM
   26 B(4,J)=0.
      GO TO 28
   29 DO 27 I=1,24
   27 B(4,I)=0.
C
C  STORE INTEGRATION POINT DEFORMATION MATRIX  (B')  IN TEMPORARY FILE
C
   28 CALL NTRAN(LU4,1,96,B,L,22)
      IF(L.EQ.96) GO TO 3
      PRINT 210,IE,IG,L
  210 FORMAT(' ERROR ON WRITING B MATRIX FOR ELEMENT',I4,
     $'  INTEGRATION POINT',I4,' :  ERROR CODE IS',I4)
      STOP
C
C  IF AXISYMMETRIC ANALYSIS DETERMINE RADIUS TO INTEGRATION POINT
C
    3 IF(IA.EQ.1) THIK=GPCRD(IE,IG,1)*6.2831853
C
C  CALCULATE WEIGHTING FACTOR AND MULTIPLIER FOR GAUSS NUMERICAL INTEGRATION
C
      WT=WGT(IO,IX)*WGT(IO,IY)*THIK*DET(IE,IG)
C
C  EVALUATE CONTRIBUTION TO ELEMENT STIFFNESS
C
      DO 4 I=1,24
      DO 5 K=1,IS
      BTD(K)=0.
      DO 5 L=1,IS
    5 BTD(K)=BTD(K)+B(L,I)*D(L,K)
      DO 4 J=I,24
      DUM=0.
      DO 6 K=1,IS
    6 DUM=DUM+BTD(K)*B(K,J)
    4 BTDB(I,J)=BTDB(I,J)+DUM*WT
      DO 30 I=1,24
      DO 30 J=I,24
   30 BTDB(J,I)=BTDB(I,J)
C
C  ADD INTO SYSTEM ELASTIC MATRIX  (BTDB)
C
      DO 2 I=1,12
      MC=IFLT(IE,I)
      DO 2 J=1,12
      NC=IFLT(IE,J)
      IF(NC.LT.MC) GO TO 2
      MNF=(MC-1)*2
      NNF=(NC-1)*2
      INF=(I-1)*2
      JNF=(J-1)*2
      DO 7 L=1,2
      IK=MNF+L
```

```
      LE=INF+L
      DO 7 LL=1,2
      LLK=NNF+LL
      IF(LLK.LT.LK) GO TO 7
      LLK=LLK-LK+1
      LLE=JNF+LL
      RKE(LK,LLK)=RKE(LK,LLK)+BTDB(LE,LLE)
    7 CONTINUE
    2 CONTINUE
C
C  APPLY BOUNDARY CONDITIONS TO SYSTEM ELASTIC MATRIX  (BTDB)
C
      DO 8 I=1,NDF
      IF(IBC(I).EQ.1) GO TO 8
      DO 9 J=2,NBW
    9 RKE(I,J)=0.
      RKE(I,1)=1.
      IF(I.EQ.1) GO TO 8
      II=I-1
      IK=2
      JJ=I-NBW+1
      IF(JJ.LT.1) JJ=1
      DO 10 IJ=II,JJ,-1
      RKE(IJ,IK)=0.
   10 IK=IK+1
    8 CONTINUE
C
C  STORE BAND OF SYSTEM ELASTIC MATRIX  (BTDB)  IN TEMPORARY FILE
C
      CALL NTRAN(LU1,1,NWRKE,RKE,L,22)
      IF(L.EQ.NWRKE) GO TO 33
      PRINT 820,L
  820 FORMAT(' ERROR ON WRITING SYSTEM ELASTIC MATRIX  (BTDB)',
     $' :  ERROR CODE IS',I4)
      STOP
C
C  DECOMPOSE ELASTIC SYSTEM MATRIX  (BTDB)  INTO PRODUCT (UT*U)
C
   33 DO 13 IR=1,NDFS
      IB=IR+1
      IBB=IR+NBWS
      IF(IBB.GT.NDF) IBB=NDF
      NCOLL=NBWS
      IC=2
      DO 14 I=IB,IBB
      ICC=IC
      IF(RKE(IR,IC).EQ.0.) GO TO 31
      FACT=RKE(IR,IC)/RKE(IR,1)
      IF(IR.GT.NDFBWS) NCOLL=NDFA-I
      DO 32 J=1,NCOLL
      RKE(I,J)=RKE(I,J)-FACT*RKE(IR,ICC)
   32 ICC=ICC+1
   31 NCOLL=NCOLL-1
   14 IC=IC+1
      FACT=DSQRT(RKE(IR,1))
      RKE(IR,1)=FACT
      FACT=1./FACT
      NCOLL=NBW
      IF(IR.GT.NDFBWS) NCOLL=NDFA-IR
      DO 13 J=2,NCOLL
```

```
   13 RKE(IR,J)=RKE(IR,J)*FACT
      RKE(NDF,1)=SQRT(RKE(NDF,1))
C
C  STORE BAND OF DECOMPOSED SYSTEM ELASTIC MATRIX  (U)  IN TEMPORARY FILE
C
      CALL NTRAN(LU2,1,NWRKE,RKE,L,22)
      IF(L.EQ.NWRKE) GO TO 11
      PRINT 821,L
  821 FORMAT(' ERROR ON WRITING DECOMPOSED SYSTEM ELASTIC MATRIX  (U)',
     $' :  ERROR CODE IS',I4)
      STOP
   11 RETURN
      END
```

COL*EPCQI(1).SOLVE1

```
      COMPILER (XM=3)
      SUBROUTINE SOLVE1 (NDFP,NPCP,NLPP,PP,DU,DIAG,QQ,RK3)
C
C  FOR EXTENDED STORAGE GIVE EXPLICIT SIZE OF ELASTIC AND PLASTIC ARRAYS
C
        COMMON /EXT/ RKE(384,64) , RKP(384,30)
C
      COMMON IA,IO,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPC,NPCS,
     2NDFBWS,LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,
     3RMIN,IS,IOS,IOSM,NLI,GP(4,4),WGT(4,4),NWRKE,NWRKP,
     4ETA,NOUT,LU1,LU2,LU3,LU4,NLAM,FRAC
      DIMENSION DIAG(NPCP),PP(NLPP,NDFP),DU(NDFP),RK3(NPCP,NPCP),
     1QQ(NPCP)
      DOUBLE PRECISION FACT
C
C  APPLY LOAD VECTOR
C
      DO 1 I=1,NDF
    1 DU(I)=PP(LP,I)
C
C  PERFORM FORWARD- AND BACK-SUBSTITUTIONS ON  (UT*U)  DECOMPOSED
C  ELASTIC SYSTEM MATRIX
C
      CALL FORSUB(DU)
      CALL BAKSUB(DU)
      IF(NPC.EQ.0) GO TO 2
C
C  CALCULATE PLASTIC SUBMATRICES OF PARTITIONED UNCOUPLED SYSTEM MATRIX
C
      DO 3 I=1,NPC
      QQ(I)=0.
      DO 3 J=1,NDF
    3 QQ(I)=QQ(I)-RKP(J,I)*DU(J)
      DO 4 I=1,NPC
    4 CALL FORSUB(RKP(1,I))
      DO 5 I=1,NPC
      DO 5 J=1,NPC
      RK3(I,J)=0.
      IF(I.EQ.J) RK3(I,J)=DIAG(I)
      DO 5 K=1,NDF
    5 RK3(I,J)=RK3(I,J)-RKP(K,I)*RKP(K,J)
C
C  SOLVE FOR MULTIPLIERS USING GAUSS REDUCTION AND BACK-SUBSTITUTION
C
      IF(NPC.EQ.1) GO TO 6
      NPCS=NPC-1
      DO 7 IR=1,NPCS
      IB=IR+1
      DO 7 I=IB,NPC
      IF(RK3(IR,I).EQ.0.) GO TO 7
      FACT=RK3(IR,I)/RK3(IR,IR)
      DO 8 J=1,NPC
    8 RK3(I,J)=RK3(I,J)-FACT*RK3(IR,J)
      QQ(I)=QQ(I)-FACT*QQ(IR)
    7 CONTINUE
      DO 9 IR=NPC,2,-1
      IF(QQ(IR).EQ.0.) GO TO 9
      QQ(IR)=QQ(IR)/RK3(IR,IR)
      IB=IR-1
```

```
      DO 10 I=IB,1,-1
   10 QQ(I)=QQ(I)-RK3(I,IR)*QQ(IR)
    9 CONTINUE
    6 QQ(1)=QQ(1)/RK3(1,1)
C
C  READ PLASTIC MATRIX  (-BTDN)  FROM TEMPORARY FILE
C
      CALL NTRAN(LU3,10,2,NWRKP,RKP,L,22)
      IF(L.EQ.NWRKP) GO TO 11
      PRINT 840,L
  840 FORMAT(' ERROR ON READING SYSTEM PLASTIC MATRIX -BTDN',
     $' :  ERROR CODE IS',I4)
      STOP
C
C  CALCULATE RHS VECTOR OF PARTITIONED UNCOUPLED SYSTEM MATRIX
C
   11 DO 12 I=1,NDF
      DU(I)=PP(LP,I)
      DO 12 J=1,NPC
   12 DU(I)=DU(I)-RKP(I,J)*QQ(J)
C
C  SOLVE FOR DISPLACEMENTS USING FORWARD- AND BACK-SUBSTITUTION
C  ON  (UT*U)  DECOMPOSED ELASTIC SYSTEM MATRIX
C
      CALL FORSUB(DU)
      CALL BAKSUB(DU)
    2 RETURN
C
C  SUBROUTINE FOR FORWARD-SUBSTITUTION ON  (UT*U)  DECOMPOSED ELASTIC SYSTEM MATRIX
C
      SUBROUTINE FORSUB (RHS)
      DIMENSION RHS(NDFP)
      DO 1 IR=1,NDFS
      IF(RHS(IR).EQ.0.) GO TO 1
      RHS(IR)=RHS(IR)/RKE(IR,1)
      IB=IR+1
      IBB=IR+NBWS
      IF(IBB.GT.NDF) IBB=NDF
      J=2
      DO 2 I=IB,IBB
      RHS(I)=RHS(I)-RKE(IR,J)*RHS(IR)
    2 J=J+1
    1 CONTINUE
      RHS(NDF)=RHS(NDF)/RKE(NDF,1)
      RETURN
C
C  SUBROUTINE FOR BACK-SUBSTITUTION ON  (UT*U)  DECOMPOSED ELASTIC SYSTEM MATRIX
C
      SUBROUTINE BAKSUB (RHS)
      DIMENSION RHS(NDFP)
      DO 1 IR=NDF,2,-1
      IF(RHS(IR).EQ.0.) GO TO 1
      RHS(IR)=RHS(IR)/RKE(IR,1)
      IB=IR-1
      IBB=IR-NBWS
      IF(IBB.LT.1) IBB=1
      J=2
      DO 2 I=IB,IBB,-1
      RHS(I)=RHS(I)-RKE(I,J)*RHS(IR)
    2 J=J+1
```

```
      1 CONTINUE
        RHS(1)=RHS(1)/RKE(1,1)
        RETURN
        END
```

```
    1 CONTINUE
      RHS(1)=RHS(1)/RKE(1,1)
      RETURN
      END
```

COL*FPCQI(1).SOLVE2

```
          COMPILER (XM=3)
          SUBROUTINE SOLVE2 (NDFP,NPCP,NLPP,PP,DU,DIAG,QQ,RK3)
C
C  FOR EXTENDED STORAGE GIVE EXPLICIT SIZE OF ELASTIC AND PLASTIC ARRAYS
C
           COMMON /EXT/ RKE(384,64) , RKP(384,30)
C
          COMMON IA,IO,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPC,NPCS,
         2NDFBWS,LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,
         3RMIN,IS,IOS,IOSM,NLI,GP(4,4),WGT(4,4),NWRKE,NWRKP,
         4FIA,NOUT,LU1,LU2,LU3,LU4,NLAM,FRAC
          DIMENSION DIAG(NPCP),PP(NLPP,NDFP),DU(NDFP),RK3(NPCP,NPCP),
         2QQ(NPCP)
          DOUBLE PRECISION FACT
C
C  AUGMENT WITH LOAD VECTOR
C
          DO 18 I=1,NDF
       18 DU(I)=PP(LP,I)
C
C  STORE DIAGONAL PLASTIC MATRIX  (NTDN+DP)
C
          DO 12 I=1,NPC
          QQ(I)=0.
          DO 12 J=I,NPC
       12 RK3(I,J)=0.
          DO 22 I=1,NPC
       22 RK3(I,I)=DIAG(I)
C
C  GAUSS REDUCE TO UPPER TRIANGULAR MATRIX
C  ****************************************
C
C  FIRST  N  ROWS
C
          DO 1 IR=1,NDF
          IF(IR.EQ.NDF) GO TO 10
C
C  ELASTIC MATRIX  (BTDB)
C
          IB=IR+1
          IBB=IR+NBWS
          IF(IBB.GT.NDF) IBB=NDF
          NCOLL=NBWS
          IC=2
          DO 8 I=IB,IBB
          ICC=IC
          IF(RKE(IR,IC).EQ.0.) GO TO 20
          FACT=RKE(IR,IC)/RKE(IR,1)
          IF(IR.GT.NDFBWS) NCOLL=NDFA-I
          DO 9 J=1,NCOLL
          RKE(I,J)=RKE(I,J)-FACT*RKE(IR,ICC)
        9 ICC=ICC+1
          DO 16 J=1,NPC
       16 RKP(I,J)=RKP(I,J)-FACT*RKP(IR,J)
          DU(I)=DU(I)-FACT*DU(IR)
       20 NCOLL=NCOLL-1
        8 IC=IC+1
       10 IF(NPC.EQ.0) GO TO 1
C
```

```
C  PLASTIC MATRIX  (-BTDN)
C
      NCOLF=1
      DO 15 I=1,NPC
      IF(RKP(IR,NCOLF).EQ.0.) GO TO 15
      FACT=RKP(IR,NCOLF)/RKE(IR,1)
      DO 17 J=NCOLF,NPC
   17 RK3(I,J)=RK3(I,J)-FACT*RKP(IR,J)
      QQ(I)=QQ(I)-FACT*DU(IR)
   15 NCOLF=NCOLF+1
    1 CONTINUE
C
C  PLASTIC MATRIX  (NTDN+DP)
C
      IF(NPC.EQ.0) GO TO 21
      IF(NPC.EQ.1) GO TO 11
      NCOLF=1
      DO 2 IR=1,NPCS
      IRA=IR+1
      IC=NCOLF+1
      DO 3 I=IRA,NPC
      IF(RK3(IR,IC).EQ.0.) GO TO 3
      FACT=RK3(IR,IC)/RK3(IR,NCOLF)
      DO 4 J=IC,NPC
    4 RK3(I,J)=RK3(I,J)-FACT*RK3(IR,J)
      QQ(I)=QQ(I)-FACT*QQ(IR)
    3 IC=IC+1
    2 NCOLF=NCOLF+1
   11 CONTINUE
C
C  BACK SUBSTITUTE
C  ***************
C
C  EVALUATE LAMBDAS
C
      QQ(NPC)=QQ(NPC)/RK3(NPC,NPC)
      IF(NPC.EQ.1) GO TO 7
      NCOLF=NPC
      DO 5 IR=NPCS,1,-1
      II=IR+1
      DO 6 J=NCOLF,NPC
      QQ(IR)=QQ(IR)-RK3(IR,J)*QQ(II)
    6 II=II+1
      NCOLF=NCOLF-1
    5 QQ(IR)=QQ(IR)/RK3(IR,NCOLF)
C
C  ELIMINATE LAMBDAS
C
    7 DO 13 I=1,NDF
      DO 13 J=1,NPC
   13 DU(I)=DU(I)-RKP(I,J)*QQ(J)
C
C  EVALUATE DISPLACEMENT INCREMENTS
C
   21 DU(NDF)=DU(NDF)/RKE(NDF,1)
      DO 14 IR=NDFS,1,-1
      II=IR+1
      NCOLL=NBW
      IF(IR.GT.NDFBWS) NCOLL=NDFA-IR
      DO 19 J=2,NCOLL
```

```
      DU(IR)=DU(IR)-RKE(IR,J)*DU(IT)
   19 II=II+1
   14 DU(IR)=DU(IR)/RKE(IR,1)
      RETURN
      END
```

COL*EPCOT(1).STRSTR

```
      COMPILER (XM=3)
      SUBROUTINE STRSTR (NEP,NNP,NDFP,IOSP,ISP,NECP,NPCP,NLPP,
     $P,U,ESTRN,PSTRN,STRN,STRS,SIGBAR,PP,TELT,PMAX,IOUT,
     $NELTPR,NODEPR,D,DET,IPELTS,RLAM,A,CF,DP,NEWP,NNEWP,
     $DU,DESTRN,DPSTRN,DSTRN,DSTRS,GPCRD,IPPT,PHI,RMINC)
C
C  FOR EXTENDED STORAGE GIVE EXPLICIT SIZE OF ELASTIC AND PLASTIC ARRAYS
C
       COMMON /EXT/ RKE(384,64) , RKP(384,30)
C
      COMMON IA,IQ,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPC,NPCS,
     2NDFBWS,LP,IL,ISTOP,ICHECK,F,FP,EPS,RNU,SZERO,SZEROS,THIK,
     3RMIN,IS,IOS,IOSM,NLI,GP(4,4),WGT(4,4),NWRKE,NWRKP,
     4FTA,NOUT,LU1,LU2,LU3,LU4,NLAM,FRAC
      DIMENSION P(NDFP),U(NDFP),ESTRN(NEP,IOSP,ISP),PSTRN(NEP,IOSP,ISP),
     2STRN(NEP,IOSP,ISP),STRS(NEP,IOSP,ISP),SIGBAR(NEP,IOSP,ISP),
     3PP(NLPP,NDFP),TELT(NEP,12),IOUT(100),NELTPR(NEP,IOSP),
     4NODEPR(NNP),D(4,4),DET(NEP,IOSP),IPELTS(NEP,IOSP,3),NEWP(50,2),
     5RLAM(NEP,IOSP),A(NEP,IOSP,ISP),RMINC(NLPP),PMAX(NLPP,2),
     6CF(NEP,IOSP),DP(NDFP),DU(NDFP),DESTRN(NEP,IOSP,ISP),
     7DPSTRN(NEP,IOSP,ISP),DSTRN(NEP,IOSP,ISP),DSTRS(NEP,IOSP,ISP),
     8GPCRD(NEP,IOSP,2),IPPT(NDFP,2),PHI(NEP,IOSP),B(4,24),DEL(24)
      DOUBLE PRECISION DUM
C
C  REWIND TEMPORARY FILE IN WHICH INTEGRATION POINT DEFORMATION
C  MATRICES  (B')  ARE STORED
C
      CALL NTRAN(LU4,10,22)
C
C  FOR EACH ELEMENT :
C *******************
C
      RMIN=1.E30
      DO 3 IE=1,NE
C
C  ELEMENT NODE DISPLACEMENTS
C
      L=0
      DO 1 I=1,12
      II=(TELT(IE,I)-1)*2
      DO 1 J=1,2
      L=L+1
    1 DEL(L)=DU(II+J)
C
C  FOR EACH INTEGRATION POINT :
C ******************************
C
      DO 3 IG=1,IOS
C
C  READ INTEGRATION POINT DEFORMATION MATRIX  (B')  FROM TEMPORARY FILE
C
      CALL NTRAN(LU4,2,96,B,L,22)
      IF(L.EQ.96) GO TO 13
      PRINT 210,IE,IG,L
  210 FORMAT(' ERROR ON READING (B) MATRIX FOR ELEMENT',I4,
     $'  INTEGRATION POINT',I4,' :  ERROR CODE IS',I4)
      STOP
   13 CONTINUE
C
```

```
C  CALCULATE RELATIVE MAGNITUDE OF TOTAL STRAIN INCREMENT
C
   35 DO 2 I=1,IS
      DSTRN(IE,IG,I)=0.
      DO 2 J=1,24
    2 DSTRN(IE,IG,I)=DSTRN(IE,IG,I)+B(I,J)*DEL(J)
      IF(IPELTS(IE,IG,1).EQ.1) GO TO 5
C
C  FOR ELASTIC INTEGRATION POINTS, CALCULATE RELATIVE MAGNITUDES OF
C  ELASTIC STRAIN AND STRESS INCREMENT VECTORS, AND SCALAR MULTIPLIER TO
C  CAUSE STRESS POINT TO REACH YIELD SURFACE.  FROM ALL ELASTIC
C  INTEGRATION POINTS DETERMINE SMALLEST SCALAR MULTIPLIER
C
      DO 4 I=1,IS
      DPSTRN(IE,IG,I)=0.
      DESTRN(IE,IG,I)=DSTRN(IE,IG,I)
      DSTRS(IE,IG,I)=0.
      DO 4 J=1,IS
    4 DSTRS(IE,IG,I)=DSTRS(IE,IG,I)+D(I,J)*DSTRN(IE,IG,J)
      IFACT=1
      CALL ROOT(DUM,DSTRS(IE,IG,1),DSTRS(IE,IG,2),DSTRS(IE,IG,3),
     $DSTRS(IE,IG,4),SIGBAR(IE,IG,1),SIGBAR(IE,IG,2),SIGBAR(IE,IG,3),
     $SIGBAR(IE,IG,4))
      DD=DUM
      IF(RMIN.GT.DD) RMIN=DD
      GO TO 3
C
C  FOR PLASTIC INTEGRATION POINTS, CALCULATE RELATIVE MAGNITUDES OF
C  ELASTIC STRAIN, PLASTIC STRAIN AND STRESS INCREMENT VECTORS
C
    5 DO 6 I=1,IS
      DPSTRN(IE,IG,I)=RLAM(IE,IG)*A(IE,IG,I)
    6 DESTRN(IE,IG,I)=DSTRN(IE,IG,I)-DPSTRN(IE,IG,I)
      DO 7 I=1,IS
      DSTRS(IE,IG,I)=0.
      DO 7 J=1,IS
    7 DSTRS(IE,IG,I)=DSTRS(IE,IG,I)+D(I,J)*DESTRN(IE,IG,J)
    3 CONTINUE
C
C  CHECK IF PROPOSED MAGNITUDE OF TOTAL LOAD IN CURRENT LOAD DIRECTION
C  COMPLIES WITH LOADING PROGRAMME
C
      IR=IFIX(PMAX(LP,1))
      DUM=(PMAX(LP,2)-P(IR))/PP(LP,IR)
      IF(RMIN.LT.DUM) GO TO 22
      RMIN=DUM
      ICHECK=1
   22 RMINC(LP)=RMINC(LP)+RMIN
C
C  MULTIPLY ALL INCREMENT QUANTITIES BY SMALLEST MULTIPLIER, AND DETERMINE
C  CURRENT TOTALS OF LOAD, DISPLACEMENT, ELASTIC STRAIN, PLASTIC STRAIN
C  AND STRESS
C
      DO 23 I=1,NDF
      DU(I)=RMIN*DU(I)
      U(I)=U(I)+DU(I)
      DP(I)=RMIN*PP(LP,I)
   23 P(I)=P(I)+DP(I)
      DO 11 IE=1,NE
      DO 11 IG=1,IOS
```

```
      DO 12 J=1,IS
      DSTRN(IE,IG,J)=DSTRN(IE,IG,J)*RMIN
      DESTRN(IE,IG,J)=DESTRN(IE,IG,J)*RMIN
      DSTRS(IE,IG,J)=DSTRS(IE,IG,J)*RMIN
      STRN(IE,IG,J)=STRN(IE,IG,J)+DSTRN(IE,IG,J)
      ESTRN(IE,IG,J)=ESTRN(IE,IG,J)+DESTRN(IE,IG,J)
      IF(IPELTS(IE,IG,1).EQ.0) GO TO 21
      DPSTRN(IE,IG,J)=DPSTRN(IE,IG,J)*RMIN
      PSTRN(IE,IG,J)=PSTRN(IE,IG,J)+DPSTRN(IE,IG,J)
   21 STRS(IE,IG,J)=STRS(IE,IG,J)+DSTRS(IE,IG,J)
   12 SIGBAR(IE,IG,J)=STRS(IE,IG,J)-EP*PSTRN(IE,IG,J)
      IF(IPELTS(IE,IG,1).EQ.0) GO TO 11
      RLAM(IE,IG)=RLAM(IE,IG)*RMIN
C
C  IN ELASTIC-PERFECTLY PLASTIC CASE, NEUTRAL-LOADING STRESS POINTS
C  MOVE TANGENTIAL TO YIELD SURFACE, THEREFORE CORRECT CURRENT
C  STRESSES BY RETURNING STRESS POINT TO YIELD SURFACE
C
      IFACT=2
      CALL ROOT(DUM,STRS(IE,IG,1),STRS(IE,IG,2),STRS(IE,IG,3),
     $STRS(IE,IG,4),PSTRN(IE,IG,1),PSTRN(IE,IG,2),PSTRN(IE,IG,3),
     $PSTRN(IE,IG,4))
      CF(IE,IG)=DUM
      DO 26 J=1,IS
      STRS(IE,IG,J)=STRS(IE,IG,J)*CF(IE,IG)
   26 SIGBAR(IE,IG,J)=STRS(IE,IG,J)-EP*PSTRN(IE,IG,J)
   11 CONTINUE
C
C  DETERMINE CURRENT PLASTIC INTEGRATION POINTS. FOR ELASTIC INTERGATION
C  POINTS CHECK RATIO OF VON MISES EQUIVALENT STRESS TO UNIAXIAL YIELD
C  STRESS FOR STRESS POINT TO BE TREATED AS PLASTIC
C
      NPC=0
      LIE=0
      NPPTS=0
      NNEWP=0
      DO 25 IE=1,NE
      DO 25 IG=1,IGS
      IF(IPELTS(IE,IG,1).EQ.1) GO TO 32
      IF(IA.GE.0) GO TO 36
      DUM=DSQRT(SIGBAR(IE,IG,1)*SIGBAR(IE,IG,1)-SIGBAR(IE,IG,1)*
     $SIGBAR(IE,IG,2)+SIGBAR(IE,IG,2)*SIGBAR(IE,IG,2)+3.*SIGBAR(IE,IG,3)
     $*SIGBAR(IE,IG,3))
      GO TO 38
   36 DUM=DSQRT(SIGBAR(IE,IG,1)*SIGBAR(IE,IG,1)+SIGBAR(IE,IG,2)
     $*SIGBAR(IE,IG,2)+SIGBAR(IE,IG,4)*SIGBAR(IE,IG,4)-SIGBAR(IE,IG,1)
     $*SIGBAR(IE,IG,2)-SIGBAR(IE,IG,2)*SIGBAR(IE,IG,4)-SIGBAR(IE,IG,1)
     $*SIGBAR(IE,IG,4)+3.*SIGBAR(IE,IG,3)*SIGBAR(IE,IG,3))
   38 DUM=DUM/SZERO
      IF(DUM.LT.ETA) GO TO 25
      IPELTS(IE,IG,1)=1
      NNEWP=NNEWP+1
      NEWP(NNEWP,1)=IE
      NEWP(NNEWP,2)=IG
   32 IF(IE.EQ.LIE.AND.NLAM.EQ.1) GO TO 8
      NPC=NPC+1
      LIE=IE
    8 NPPTS=NPPTS+1
      IPPT(NPPTS,1)=IE
      IPPT(NPPTS,2)=IG
```

```
      IPFLTS(IE,IG,2)=NPC
   25 CONTINUE
      NPCS=NPC-1
C
C  CHECK IF LOAD INCREMENTS BECOMING CONSISTANTLY LESS THAN 'FRAC' OF TOTAL LOAD
C
      IF(EP.GT.0.0) GO TO 33
      IF(RMIN/RMINC(LP).GT.FRAC) GO TO 9
      NKNT=NKNT+1
      IF(NKNT.LT.3) GO TO 33
      PRINT 201,FRAC
  201 FORMAT(1H0,83('*'),/,' LOAD INCREMENTS LESS THAN',F6.3,
     $'  OF TOTAL LOAD FOR PRECEDING  3  LOAD INCREMENTS',//,
     $' ANALYSIS TERMINATED',//,' RESULTS AFTER CURRENT LOAD INCREMENT L
     $ISTED BELOW',//,84('*'))
      ISTOP=-1
      IOUT(NOUT)=NLI+1
    9 NKNT=0
C
C  OUTPUT CURRENT QUANTITIES
C
   33 CALL OUTPUT (NEP,NNP,NDEP,IOSP,ISP,NECP,NPCP,NLPP,
     $PSTRN,STRN,STRS,SIGBAR,PMAX,IOUT,NELTPR,NODEPR,
     $PHI,IPFLTS,RLAM,CF,DP,DU,DESTRN,DPSTRN,DSTRN,DSTRS,GPCRD,IPPT,
     $P,U,ESTRN,NEWP,NNEWP,RMINC)
      IF(ISTOP.EQ.-1) GO TO 24
C
C  CHECK AVAILABLE STORAGE
C
      NWRKP=NDF*NPC
      IF(NPC.LE.NPCP) GO TO 29
      PRINT 202
  202 FORMAT(1H0,54('*'),/,' AVAILABLE STORAGE EXCEEDED :  PLASTIC MATRI
     $X  TOO LARGE',//,' ANALYSIS TERMINATED',//,' RESULTS AFTER PREVIOUS
     $ LOAD INCREMENT LISTED BELOW',//,55('*'))
      IOUT(NOUT)=NLI
      NLI=NLI-1
      CALL OUTPUT (NEP,NNP,NDEP,IOSP,ISP,NECP,NPCP,NLPP,
     $PSTRN,STRN,STRS,SIGBAR,PMAX,IOUT,NELTPR,NODEPR,
     $PHI,IPFLTS,RLAM,CF,DP,DU,DESTRN,DPSTRN,DSTRN,DSTRS,GPCRD,IPPT,
     $P,U,ESTRN,NEWP,NNEWP,RMINC)
      STOP
C
C  READ ELASTIC MATRIX  (BTDB)  FROM TEMPORARY FILE
C
   29 IF(NPC.LE.NBW) LU=LU2
      IF(NPC.GT.NBW) LU=LU1
      CALL NTRAN(LU,10,2,NWRKE,RKE,L,22)
      IF(L.EQ.NWRKE) GO TO 24
      PRINT 820,L
  820 FORMAT(' ERROR ON READING SYSTEM ELASTIC MATRIX  (BTDB)',
     $' :  ERROR CODE IS',I4)
      STOP
   24 RETURN
C
C  SUBROUTINE FOR DETERMINING SCALAR MULTIPLIERS FOR ELASTIC INTEGRATION
C  POINTS, AND FOR RETURNING STRESS POINTS TO YIELD SURFACE FOR
C  PLASTIC INTEGRATION POINTS
C
      SUBROUTINE ROOT(RT,X1,X2,X3,X4,Y1,Y2,Y3,Y4)
```

```
      DOUBLE PRECISION AA,BB,CC,RT
      IF(IA.GE.0) GO TO 2
      AA=X1*X1-X1*X2+X2*X2+3.*X3*X3
      IF(IFACT.EQ.2.AND.EP.EQ.0) GO TO 1
      BB=X1*Y1-0.5*(X1*Y2+X2*Y1)+X2*Y2+3.*X3*Y3
      CC=Y1*Y1-Y1*Y2+Y2*Y2+3.*Y3*Y3
      GO TO 4
    2 AA=X1*X1+X2*X2+X4*X4-X1*X2-X2*X4-X1*X4+3.*X3*X3
      IF(IFACT.EQ.2.AND.EP.EQ.0) GO TO 1
      BB=X1*Y1+X2*Y2+X4*Y4-0.5*(X1*Y2+X2*Y1+X2*Y4+X4*Y2
     $+X1*Y4+X4*Y1)+3.*X3*Y3
      CC=Y1*Y1+Y2*Y2+Y4*Y4-Y1*Y2-Y2*Y4-Y1*Y4+3.*Y3*Y3
    4 IF(IFACT.EQ.1) GO TO 5
      BB=-BB*EP
      CC=CC*EPS
    5 RT=(DSQRT(BB*BB-AA*(CC-SZEROS))-BB)/AA
      RETURN
    1 RT=SZERO/DSQRT(AA)
      RETURN
      END
```

```
COL*EPCQT(1).PLASM
      SUBROUTINE PLASM (NEP,IOSP,ISP,IE,IG,IPELTS,A,SIGBAR,
     $D,GPCRD,IELT,DET,WT,BTDN,FNTDN,N)
      COMMON IA,IO,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPC,NPCS,
     2NDFBWS,LP,IL,ISTOP,ICHECK,F,FP,EPS,RNU,SZERO,SZEROS,THIK,
     3RMIN,IS,IOS,IOSM,NLI,GP(4,4),WGT(4,4),NWRKF,NWRKP,
     4ETA,NOUT,LU1,LU2,LU3,LU4,NLAM,FRAC
      DIMENSION IPELTS(NEP,IOSP,3),A(NEP,IOSP,ISP),SIGBAR(NEP,IOSP,ISP),
     2D(4,4),GPCRD(NEP,IOSP,2),IELT(NEP,12),DET(NEP,IOSP),BTDN(24),
     3B(4,24),BTD(24,4),ATD(4)
      DOUBLE PRECISION DUM
C
C  CALCULATE GRADIENT OF YIELD FUNCTION
C
      IF(IA.GE.0) GO TO 40
      A(IE,IG,1)=SIGBAR(IE,IG,1)+SIGBAR(IE,IG,1)-SIGBAR(IE,IG,2)
      A(IE,IG,2)=SIGBAR(IE,IG,2)+SIGBAR(IE,IG,2)-SIGBAR(IE,IG,1)
      A(IE,IG,3)=6.*SIGBAR(IE,IG,3)
      GO TO 42
   40 A(IE,IG,1)=SIGBAR(IE,IG,1)+SIGBAR(IE,IG,1)-SIGBAR(IE,IG,2)
     $-SIGBAR(IE,IG,4)
      A(IE,IG,2)=SIGBAR(IE,IG,2)+SIGBAR(IE,IG,2)-SIGBAR(IE,IG,1)
     $-SIGBAR(IE,IG,4)
      A(IE,IG,3)=6.*SIGBAR(IE,IG,3)
      A(IE,IG,4)=SIGBAR(IE,IG,4)+SIGBAR(IE,IG,4)-SIGBAR(IE,IG,1)
     $-SIGBAR(IE,IG,2)
C
C  NORMALIZE GRADIENT OF YIELD FUNCTION
C
   42 DUM=0.
      DO 43 I=1,IS
   43 DUM=DUM+A(IE,IG,I)*A(IE,IG,I)
      DUM=1./DSQRT(DUM)
      DO 30 I=1,IS
   30 A(IE,IG,I)=A(IE,IG,I)*DUM
C
C  READ INTEGRATION POINT DEFORMATION MATRIX  (B')  FROM TEMPORARY FILE
C
      CALL NTRAN(LU4,6,N,2,96,B,L,22)
      IF(L.EQ.96) GO TO 45
      PRINT 210,IE,IG,L
  210 FORMAT(' ERROR ON READING (B) MATRIX FOR ELEMENT',I4,
     $'  INTEGRATION POINT',I4,' :  ERROR CODE IS',I4)
      STOP
C
C  CALCULATE  (BTD)  SUBMATRIX
C
   45 DO 44 I=1,24
      DO 44 J=1,IS
      BTD(I,J)=0.
      DO 44 K=1,IS
   44 BTD(I,J)=BTD(I,J)+B(K,I)*D(K,J)
C
C  CALCULATE  (BTDN)  VECTOR
C
      DO 17 I=1,24
      BTDN(I)=0.
      DO 17 J=1,IS
   17 BTDN(I)=BTDN(I)+BTD(I,J)*A(IE,IG,J)
C
```

```
C  CALCULATE  (NTDN+DP)  TERM
C
      DO 18 I=1,IS
      ATD(I)=0.
      DO 18 J=1,IS
   18 ATD(I)=ATD(I)+A(IE,IG,J)*D(J,I)
      FNTDN=FP
      DO 19 J=1,IS
   19 FNTDN=FNTDN+ATD(J)*A(IE,IG,J)
C
C  CALCULATE WEIGHTING FACTOR AND MULTIPLIER FOR GAUSS NUMERICAL INTEGRATION
C
      I=(IG-1)/IO
      IX=I+1
      IY=IG-IO*I
      IF(IA.EQ.1) THIK=GPCRD(IE,IG,1)*6.2831853
      WT=WGT(IO,IX)*WGT(IO,IY)*DET(IE,IG)*THIK
      RETURN
      END
```

COL*EPCQT(1).ITER

```
      COMPILER (XM=3)
      SUBROUTINE ITER (NEP,NNP,NDFP,IOSP,ISP,NECP,NPCP,NLPP,
     $TELT,IBC,D,DET,IPELTS,RLAM,A,PP,DU,NEWP,NNEWP,DIAG2,RMINC,
     $QQ,GPCRD,SIGBAR,BTDN,ESTRN,U,P,IPPT,DSTRS,DSTRN,DPSTRN,DIAG,
     $DESTRN,DP,CF,PMAX,TOUT,NELTPR,NODEPR,PSTRN,STRN,STRS,RK3)
C
C  FOR EXTENDED STORAGE GIVE EXPLICIT SIZE OF ELASTIC AND PLASTIC ARRAYS
C
       COMMON /EXT/ RKE(384,64) , RKP(384,30)
C
      COMMON IA,IO,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPC,NPCS,
     2NDFBWS,LP,IL,ISTOP,ICHECK,F,FP,EPS,RNU,SZERO,SZEROS,THIK,
     3RMIN,IS,IOS,IOSM,NLI,GP(4,4),WGT(4,4),NWRKE,NWRKP,
     4ETA,NOUT,LU1,LU2,LU3,LU4,NLAM,FRAC
      DIMENSION IELT(NEP,12),IBC(NDFP),D(4,4),DET(NEP,IOSP),
     2IPELTS(NEP,IOSP,3),RLAM(NEP,IOSP),DU(NDFP),DSTRS(NEP,IOSP,ISP),
     3A(NEP,IOSP,ISP),PP(NLPP,NDFP),DESTRN(NEP,IOSP,ISP),DP(NDFP),
     4GPCRD(NEP,IOSP,2),SIGBAR(NEP,IOSP,ISP),BTDN(24),TOUT(100),
     5DIAG(NPCP),DIAG2(NPCP),QQ(NPCP),RK3(NPCP,NPCP),PMAX(NLPP,2),
     6DUM1(24),DUM2(4),B(4,24),RMINC(NLPP)
C
C  AS ITERATION MIGHT BE NECESSARY, STORE PLASTIC MATRIX  (-BTDN)
C  IN TEMPORARY FILE
C
      CALL NTRAN(LU3,10,1,NWRKP,RKP,L,22)
      IF(L.EQ.NWRKP) GO TO 1
      PRINT 820,L
  820 FORMAT(' ERROR ON WRITING SYSTEM PLASTIC MATRIX (-BTDN)',
     $' :  ERROR CODE IS',I4)
      STOP
    1 DO 7 I=1,NPC
    7 DIAG2(I)=DIAG(I)
C
C  INITIALISE ITERATION COUNTERS
C
      IT=1
      NLIA=NLI+1
C
C  SOLVE SYSTEM MATRIX FOR DISPLACEMENT INCREMENT VECTOR AND LAMBDAS
C
   16 IF(NPC.LE.NBW) CALL SOLVE1 (NDFP,NPCP,NLPP,PP,DU,DIAG,QQ,RK3)
      IF(NPC.GT.NBW) CALL SOLVE2 (NDFP,NPCP,NLPP,PP,DU,DIAG,QQ,RK3)
      IK=0
C
C  CHECK KINEMATIC CONSTRAINTS FOR EACH PLASTIC INTEGRATION POINT :-
C ******************************************************************
C
      CALL NTRAN(LU4,10,22)
      NW=0
      NNW=0
      DO 4 IE=1,NE
      DO 4 IG=1,IOS
      IF(IPELTS(IE,IG,1).EQ.0) GO TO 4
      I=IPELTS(IE,IG,2)
      RLAM(IE,IG)=QQ(I)
      IF(IPELTS(IE,IG,3).EQ.1) GO TO 2
C
C  (1): FOR LAMBDAS ASSUMED NON-ZERO CHECK THAT LAMBDA NON-NEGATIVE
C
```

```
      IF(RLAM(IE,IG).GE.0.) GO TO 4
      IF(ABS(RLAM(IE,IG)).LE.1.E-4) GO TO 4
      IK=1
      IPELTS(IE,IG,3)=1
      PRINT 702,NLIA,IT
  702 FORMAT(1H0,'LOAD INCREMENT ',I4,'    ITERATION ',I4)
      PRINT 703,IE,IG,RLAM(IE,IG)
  703 FORMAT(1H ,'ELEMENT',I4,'  INTEGRATION POINT',I4,'  UNLOADING',
     $/,' LAMBDA =',F11.6)
      GO TO 4
C
C  (2): FOR LAMBDAS ASSUMED ZERO CHECK THAT SCALAR PRODUCT OF GRADIENT OF
C       YIELD FUNCTION AND DIRECTION OF STRESS INCREMENT VECTOR NON-POSITIVE
C
C  ELEMENT NODE DISPLACEMENT INCREMENTS
C
    2 L=0
      DO 6 I=1,12
      IJ=(IELT(IE,I)-1)*2
      DO 6 J=1,2
      L=L+1
    6 DUM1(L)=DU(IJ+J)
C
C  READ INTEGRATION POINT DEFORMATION MATRIX  (B')  FROM TEMPORARY FILE
C
      N=NW-NNW
      NNW=NW+4
      CALL NTRAN(LU4,6,N,2,96,B,L,22)
      IF(L.EQ.96) GO TO 10
      PRINT 210,IE,IG,L
  210 FORMAT(' ERROR ON READING (B) MATRIX FOR ELEMENT',I4,
     $'  INTEGRATION POINT',I4,' :  ERROR CODE IS',I4)
      STOP
C
C  CALCULATE RELATIVE MAGNITUDES OF COMPONENTS OF ELASTIC STRAIN
C  INCREMENT VECTOR
C
   10 DO 13 I=1,IS
      DUM2(I)=0.
      DO 13 J=1,24
   13 DUM2(I)=DUM2(I)+B(I,J)*DUM1(J)
C
C  CALCULATE RELATIVE MAGNITUDES OF COMPONENTS OF STRESS INCREMENT VECTOR
C
      DO 8 I=1,IS
      DUM1(I)=0.
      DO 8 J=1,IS
    8 DUM1(I)=DUM1(I)+D(I,J)*DUM2(J)
C
C  NORMALIZE RELATIVE MAGNITUDES OF COMPONENTS OF STRESS INCREMENT VECTOR
C
      DUM=0.
      DO 18 I=1,IS
   18 DUM=DUM+DUM1(I)*DUM1(I)
      DUM=1./SQRT(DUM)
      DO 19 I=1,IS
   19 DUM1(I)=DUM1(I)*DUM
C
C  CALCULATE AND CHECK SCALAR PRODUCT OF GRADIENT OF YIELD FUNCTION AND
C  NORMALIZED STRESS INCREMENT VECTOR
```

```
C
      DUM=0.
      DO 9 I=1,IS
    9 DUM=DUM+A(IE,IG,I)*DUM1(I)
      IF(DUM.LT.0.) GO TO 4
      IK=1
      IPELTS(IE,IG,3)=0
      PRINT 702,NLTA,IT
      PRINT 701,IE,IG,DUM
  701 FORMAT(1H ,'ELEMENT',I4,'  INTEGRATION POINT',I4,'  LOADING',
     $/,' SCALAR PRODUCT OF GRADIENT OF YIELD FUNCTION AND STRESS INCREM
     $ENT VECTOR =',F11.6)
    4 NW=NW+4
C
C  IF ITERATION CONSTRAINTS SATISFIED, CONTINUE
C
      IF(IK.EQ.0) GO TO 12
C
C  IF ITERATION PROCESS NOT CONVERGING TERMINATE ANALYSIS
C
      IF(IT.LT.10) GO TO 11
      PRINT 200
  200 FORMAT(1H0,50('*'),//,' STILL IN ITERATION LOOP AFTER 10 ITERATION
     $S',//,' ANALYSIS TERMINATED',//,' TOTALS AFTER PREVIOUS LOAD INCRE
     $MENT LISTED BELOW',//,51('*'))
      GO TO 29
C
C  IF ITERATION NECESSARY, READ ELASTIC AND PLASTIC MATRICES FROM
C  TEMPORARY FILES
C
   11 IT=IT+1
      IF(NPC.LE.NBW) LU=LU2
      IF(NPC.GT.NBW) LU=LU1
      CALL NTRAN(LU,10,2,NWRKE,RKE,L,22)
      IF(L.EQ.NWRKE) GO TO 21
      PRINT 830,L
  830 FORMAT(' ERROR ON READING SYSTEM ELASTIC MATRIX BTDB',
     $' :  ERROR CODE IS',I4)
      STOP
   21 CALL NTRAN(LU3,10,2,NWRKP,RKP,L,22)
      IF(L.EQ.NWRKP) GO TO 17
      PRINT 840,L
  840 FORMAT(' ERROR ON READING SYSTEM PLASTIC MATRIX -BTDN',
     $' :  ERROR CODE IS',I4)
      STOP
   17 DO 20 I=1,NPC
   20 DIAG(I)=DIAG2(I)
C
C  FOR PLASTIC INTEGRATION POINTS ASSUMED TO BE UNLOADING,
C  SUBTRACT CONTRIBUTION TO PLASTIC MATRICES  (-BTDN)  AND  (NTDN+DP)
C
      CALL NTRAN(LU4,10,22)
      NW=0
      NNW=0
      DO 14 IE=1,NE
      DO 14 IG=1,IOS
      IF(IPELTS(IE,IG,3).EQ.0) GO TO 14
      N=NW-NNW
      NNW=NW+4
      CALL PLASH (NEP,IOSP,ISP,IE,IG,IPELTS,A,SIGBAR,
```

```
     $D,GPCRD,IELT,DET,WT,BTDN,ENTDN,N)
      NC=IPELTS(IE,IG,2)
      DO 5 I=1,12
      NRR=(IELT(IE,I)-1)*2
      NRE=(I-1)*2
      DO 5 J=1,2
    5 RKP(NRR+J,NC)=RKP(NRR+J,NC)+BTDN(NRE+J)*WT
      DIAG(NC)=DIAG(NC)-ENTDN*WT
      IF(DIAG(NC).GT.1.E-7) GO TO 14
      DO 15 I=1,NDF
   15 RKP(I,NC)=0.
      DIAG(NC)=1.
   14 NW=NW+4
C
C  RE-SOLVE
C
      GO TO 16
   29 IOUT(NOUT)=NLI
      NLI=NLI-1
      DO 27 IE=1,NE
      DO 27 IG=1,IOS
      IPELTS(IE,IG,1)=0
      DO 27 J=1,IS
      DESTRN(IE,IG,J)=0.
   27 DSTRS(IE,IG,J)=0.
      DO 28 I=1,NDF
      DP(I)=0.
   28 DU(I)=0.
      CALL OUTPUT (NEP,NNP,NDFP,IOSP,ISP,NECP,NPCP,NLPP,
     $PSTRN,STRN,STRS,SIGBAR,PMAX,IOUT,NELTPR,NODEPR,
     $PHI,IPELTS,RLAM,CF,DP,DU,DESTRN,DPSTRN,DSTRN,DSTRS,GPCRD,IPPT,
     $P,U,FSTRN,NEWP,NNEWP,RMINC)
      STOP
C
C  ITERATION PROCEDURE HAS CONVERGED :  RECORD CURRENT LOADING AND
C  UNLOADING PLASTIC INTEGRATION POINTS
C
   12 DO 3 IE=1,NE
      DO 3 IG=1,IOS
      IF(IPELTS(IE,IG,3).EQ.0) GO TO 3
      IPELTS(IE,IG,1)=0
      IPELTS(IE,IG,3)=0
    3 CONTINUE
      RETURN
      END
```

COL*FPCQT(1).OUTPUT

```
      SUBROUTINE OUTPUT (NEP,NNP,NDFP,IOSP,ISP,NECP,NPCP,NLPP,
     2PSTRN,STRN,STRS,SIGBAR,PMAX,IOUT,NELTPR,NODEPR,
     3PHI,IPELTS,RLAM,CF,DP,DU,DESTRN,DPSTRN,DSTRN,DSTRS,GPCRD,IPPT,
     4P,U,ESTRN,NEWP,NNEWP,RMINC)
      COMMON IA,IO,NN,NE,NPPTS,NDFS,NDF,NDFA,NBWS,NBW,NPC,NPCS,
     2NDFBWS,LP,IL,ISTOP,ICHECK,E,EP,EPS,RNU,SZERO,SZEROS,THIK,
     3RMIN,IS,IOS,IOSM,NLI,GP(4,4),WGT(4,4),NWRKE,NWRKP,
     4ETA,NOUT,LU1,LU2,LU3,LU4,NLAM,FRAC
      DIMENSION P(NDFP),U(NDFP),ESTRN(NEP,IOSP,ISP),PSTRN(NEP,IOSP,ISP),
     2STRN(NEP,IOSP,ISP),STRS(NEP,IOSP,ISP),SIGBAR(NEP,IOSP,ISP),
     3PMAX(NLPP,2),IOUT(100),NELTPR(NEP,IOSP),NODEPR(NNP),PHI(NEP,IOSP),
     4IPELTS(NEP,IOSP,3),RLAM(NEP,IOSP),CF(NEP,IOSP),IPPT(NDFP,2),
     5DP(NDFP),DU(NDFP),DESTRN(NEP,IOSP,ISP),DPSTRN(NEP,IOSP,ISP),
     6DSTRN(NEP,IOSP,ISP),DSTRS(NEP,IOSP,ISP),GPCRD(NEP,IOSP,2),
     7NEWP(50,2),RMINC(NLPP)
      NLI=NLI+1
      IF(NLI.NE.IOUT(NOUT).AND.ICHECK.EQ.0) GO TO 7
      IF(NLI.EQ.IOUT(NOUT)) NOUT=NOUT+1
C
C PRINT HEADINGS
C
      PRINT 200,NLI
  200 FORMAT(1H1,'RESULTS AFTER',I4,2X,'LOAD INCREMENTS',/)
      IF(ICHECK.EQ.1) GO TO 1
      PRINT 201,((NEWP(I,J),J=1,2),I=1,NNEWP)
  201 FORMAT(1H0,'NEXT POINTS TO UNDERGO PLASTIC DEFORMATION  -  (ELEMEN
     $T : INTEGRATION POINT)',2X,6(I5,':',I2),9(//,77X,6(I5,':',I2)))
      GO TO 2
    1 I=(IFIX(PMAX(LP,1))+1)/2
      PRINT 202,LP,I
  202 FORMAT(1H0,'END OF LOADING ',I4,2X,': MAXIMUM LOAD REACHED AT NODE
     $',I4,/)
    2 PRINT 218,RMIN
  218 FORMAT(1H0,'AT LOAD FACTOR',E11.6)
      DO 6 I=1,LP
    6 PRINT 221,I,RMINC(I)
  221 FORMAT(1H0,'LOAD VECTOR',I3,' : CUMULATIVE LOAD FACTOR =',E11.6)
      IF(ICHECK.EQ.0) GO TO 23
      LP=LP+1
      ICHECK=0
   23 PRINT 206
  206 FORMAT(1H ,84X,'CURRENT VALUES',/,65X,'/',52('-'),'/',/,24X,'STRAI
     2N INCREMENTS',37X,'STRAINS',38X,'STRESS',/,' ELEMENT,',4X,
     3'GLOBAL',3X,'/',18('-'),'/',14X,
     4'STRESS',4X,'/',29('-'),'/',14X,'YIELD',5X,'CORRECTION',/,
     5' INT. PT.',4X,'COORDS',4X,'ELASTIC',4X,'PLASTIC',4X,'LAMBDA',4X,
     6'INCREMENT',3X,'ELASTIC',4X,'PLASTIC',5X,'TOTAL',6X,'STRESS',3X,
     7'FUNCTION',5X,'FACTOR'/)
C
C  EVALUATE YIELD FUNCTION FOR INTEGRATION POINTS REQUESTED AS OUTPUT
C
      DO 4 IE=1,NE
      DO 4 IG=1,IOS
      IF(NELTPR(IE,IG).NE.1) GO TO 4
      IF(IA.GE.0) GO TO 8
      PHI(IE,IG)=SIGBAR(IE,IG,1)*SIGBAR(IE,IG,1)-SIGBAR(IE,IG,1)
     $*SIGBAR(IE,IG,2)+SIGBAR(IE,IG,2)*SIGBAR(IE,IG,2)
     $+3.*SIGBAR(IE,IG,3)*SIGBAR(IE,IG,3)-SZEROS
      GO TO 4
```

```
      8 PHI(IE,IG)=SIGBAR(IE,IG,1)*SIGBAR(IE,IG,1)+SIGBAR(IE,IG,2)
       $*SIGBAR(IE,IG,2)+SIGBAR(IE,IG,4)*SIGBAR(IE,IG,4)-SIGBAR(IE,IG,1)
       $*SIGBAR(IE,IG,2)-SIGBAR(IE,IG,2)*SIGBAR(IE,IG,4)-SIGBAR(IE,IG,1)
       $*SIGBAR(IE,IG,4)+3.*SIGBAR(IE,IG,3)*SIGBAR(IE,IG,3)-SZEROS
      8 CONTINUE
C
C  PRINT INTEGRATION POINT QUANTITIES FOR OUTPUT REQUESTED :
C *************************************************************
C
        IF(IA) 9,13,10
C
C  PLANE STRESS ANALYSIS:
C
      9 DO 12 IE=1,NE
        DO 12 IG=1,IOS
        IF(NFLTPR(IE,IG).NE.1) GO TO 12
        IF(IPELTS(IE,IG,1).EQ.0) GO TO 3
C
C       PLASTIC INTEGRATION POINTS
C
        PRINT 207,IE,GPCRD(IE,IG,1),DESTRN(IE,IG,1),DPSTRN(IE,IG,1),
       2DSTRS(IE,IG,1),ESTRN(IE,IG,1),PSTRN(IE,IG,1),STRN(IE,IG,1),
       3STRS(IE,IG,1),IG,GPCRD(IE,IG,2),DESTRN(IE,IG,2),
       4DPSTRN(IE,IG,2),RLAM(IE,IG),DSTRS(IE,IG,2),ESTRN(IE,IG,2),
       5PSTRN(IE,IG,2),STRN(IE,IG,2),STRS(IE,IG,2),PHI(IE,IG),CF(IE,IG),
       6DESTRN(IE,IG,3),DPSTRN(IE,IG,3),DSTRS(IE,IG,3),
       7ESTRN(IE,IG,3),PSTRN(IE,IG,3),STRN(IE,IG,3),STRS(IE,IG,3)
    207 FORMAT(1H0,I4,3X,'X',3(1X,E10.5),11X,5(1X,E10.5),/,I5,3X,'Y',
       $11(1X,E10.5),/,7X,'XY',11X,2(1X,E10.5),11X,5(1X,E10.5))
        GO TO 12
C
C       ELASTIC INTEGRATION POINTS
C
      3 PRINT 205,IE,GPCRD(IE,IG,1),DESTRN(IE,IG,1),DSTRS(IE,IG,1),
       2ESTRN(IE,IG,1),PSTRN(IE,IG,1),STRN(IE,IG,1),STRS(IE,IG,1),
       3IG,GPCRD(IE,IG,2),DESTRN(IE,IG,2),DSTRS(IE,IG,2),
       4ESTRN(IE,IG,2),PSTRN(IE,IG,2),STRN(IE,IG,2),STRS(IE,IG,2),
       5PHI(IE,IG),DESTRN(IE,IG,3),DSTRS(IE,IG,3),ESTRN(IE,IG,3),
       6PSTRN(IE,IG,3),STRN(IE,IG,3),STRS(IE,IG,3)
    205 FORMAT(1H0,I4,3X,'X',2(1X,E10.5),5X,'--',15X,5(1X,E10.5),/,
       2I5,3X,'Y',2(1X,E10.5),5X,'--',9X,'--',4X,6(1X,E10.5),5X,
       3'--',/,7X,'XY',12X,E10.5,5X,'--',15X,5(1X,E10.5))
     12 CONTINUE
        GO TO 20
C
C  PLANE STRAIN ANALYSIS:
C
     13 DO 21 IE=1,NE
        DO 21 IG=1,IOS
        IF(NFLTPR(IE,IG).NE.1) GO TO 21
        IF(IPELTS(IE,IG,1).EQ.0) GO TO 14
C
C       PLASTIC INTEGRATION POINTS
C
        PRINT 212,IE,GPCRD(IE,IG,1),DESTRN(IE,IG,1),DPSTRN(IE,IG,1),
       2DSTRS(IE,IG,1),ESTRN(IE,IG,1),PSTRN(IE,IG,1),STRN(IE,IG,1),
       3STRS(IE,IG,1),IG,GPCRD(IE,IG,2),DESTRN(IE,IG,2),
       4DPSTRN(IE,IG,2),RLAM(IE,IG),DSTRS(IE,IG,2),ESTRN(IE,IG,2),
       5PSTRN(IE,IG,2),STRN(IE,IG,2),STRS(IE,IG,2),PHI(IE,IG),CF(IE,IG),
       6DESTRN(IE,IG,4),DPSTRN(IE,IG,4),DSTRS(IE,IG,4),ESTRN(IE,IG,4),
```

```
     7PSTRN(IE,IG,4),STRN(IE,IG,4),STRS(IE,IG,4),DESTRN(IE,IG,3),
     8DPSTRN(IE,IG,3),DSTRS(IE,IG,3),ESTRN(IE,IG,3),PSTRN(IE,IG,3),
     9STRN(IE,IG,3),STRS(IE,IG,3)
  212 FORMAT(1H0,I4,3X,'X',3(1X,F10.5),11X,5(1X,F10.5),/,I5,3X,'Y',
     211(1X,F10.5),/,8X,'Z',11X,2(1X,E10.5),11X,5(1X,E10.5),
     3/,7X,'XY',11X,2(1X,E10.5),11X,5(1X,E10.5))
      GO TO 21
C
C     ELASTIC INTEGRATION POINTS
C
   14 PRINT 213,IE,GPCRD(IE,IG,1),DESTRN(IE,IG,1),DSTRS(IE,IG,1),
     2ESTRN(IE,IG,1),PSTRN(IE,IG,1),STRN(IE,IG,1),STRS(IE,IG,1),
     3IG,GPCRD(IE,IG,2),DESTRN(IE,IG,2),DSTRS(IE,IG,2),
     4ESTRN(IE,IG,2),PSTRN(IE,IG,2),STRN(IE,IG,2),STRS(IE,IG,2),
     5PHI(IE,IG),DESTRN(IE,IG,4),DSTRS(IE,IG,4),ESTRN(IE,IG,4),
     6PSTRN(IE,IG,4),STRN(IE,IG,4),STRS(IE,IG,4),DESTRN(IE,IG,3),
     7DSTRS(IE,IG,3),ESTRN(IE,IG,3),PSTRN(IE,IG,3),
     8STRN(IE,IG,3),STRS(IE,IG,3)
  213 FORMAT(1H0,I4,3X,'X',2(1X,F10.5),5X,'--',15X,5(1X,F10.5),/,
     2I5,3X,'Y',2(1X,E10.5),5X,'--',9X,'--',4X,6(1X,F10.5),5X,
     3'--',/,8X,'Z',12X,F10.5,5X,'--',15X,5(1X,F10.5),
     4/,7X,'XY',12X,F10.5,5X,'--',15X,5(1X,E10.5))
   21 CONTINUE
      GO TO 20
C
C  AXISYMMETRIC ANALYSIS:
C
   10 DO 22 IE=1,NE
      DO 22 IG=1,IOS
      IF(NELTPR(IE,IG).NE.1) GO TO 22
      IF(IPELTS(IE,IG,1).EQ.0) GO TO 11
C
C     PLASTIC INTEGRATION POINTS
C
      PRINT 211,IE,GPCRD(IE,IG,1),DESTRN(IE,IG,1),DPSTRN(IE,IG,1),
     2DSTRS(IE,IG,1),ESTRN(IE,IG,1),PSTRN(IE,IG,1),STRN(IE,IG,1),
     3STRS(IE,IG,1),IG,GPCRD(IE,IG,2),DESTRN(IE,IG,2),
     4DPSTRN(IE,IG,2),PLAM(IE,IG),DSTRS(IE,IG,2),ESTRN(IE,IG,2),
     5PSTRN(IE,IG,2),STRN(IE,IG,2),STRS(IE,IG,2),PHI(IE,IG),CF(IE,IG)
     6DESTRN(IE,IG,4),DPSTRN(IE,IG,4),DSTRS(IE,IG,4),ESTRN(IE,IG,4),
     7PSTRN(IE,IG,4),STRN(IE,IG,4),STRS(IE,IG,4),DESTRN(IE,IG,3),
     8DPSTRN(IE,IG,3),DSTRS(IE,IG,3),ESTRN(IE,IG,3),PSTRN(IE,IG,3),
     9STRN(IE,IG,3),STRS(IE,IG,3)
  211 FORMAT(1H0,I4,3X,'R',3(1X,F10.5),11X,5(1X,F10.5),/,I5,3X,'Z',
     211(1X,F10.5),/,8X,'O',11X,2(1X,E10.5),11X,5(1X,E10.5),
     3/,1H+,7X,'-',/,7X,'RZ',11X,2(1X,F10.5),11X,5(1X,E10.5))
      GO TO 22
C
C     ELASTIC INTEGRATION POINTS
C
   11 PRINT 214,IE,GPCRD(IE,IG,1),DESTRN(IE,IG,1),DSTRS(IE,IG,1),
     2ESTRN(IE,IG,1),PSTRN(IE,IG,1),STRN(IE,IG,1),STRS(IE,IG,1),
     3IG,GPCRD(IE,IG,2),DESTRN(IE,IG,2),DSTRS(IE,IG,2),
     4ESTRN(IE,IG,2),PSTRN(IE,IG,2),STRN(IE,IG,2),STRS(IE,IG,2),
     5PHI(IE,IG),DESTRN(IE,IG,4),DSTRS(IE,IG,4),ESTRN(IE,IG,4),
     6PSTRN(IE,IG,4),STRN(IE,IG,4),STRS(IE,IG,4),DESTRN(IE,IG,3),
     7DSTRS(IE,IG,3),ESTRN(IE,IG,3),PSTRN(IE,IG,3),
     8STRN(IE,IG,3),STRS(IE,IG,3)
  214 FORMAT(1H0,I4,3X,'R',2(1X,F10.5),5X,'--',15X,5(1X,F10.5),/,
     2I5,3X,'Z',2(1X,F10.5),5X,'--',9X,'--',4X,6(1X,F10.5),5X,
```

```
     3'--',/,8X,'0',12X,F10.5,5X,'--',15X,5(1X,E10.5),
     4/,1H+,7X,'-',/,7X,'RZ',12X,E10.5,5X,'--',15X,5(1X,E10.5))
   22 CONTINUE
C
C  PRINT CURRENT PLASTIC INTEGRATION POINTS
C
   20 PRINT 203
  203 FORMAT(1H0,/,' CURRENT PLASTIC POINTS  -  (ELEMENT : INTEGRATION P
     $OINT)',/,57('-'))
      IF(NPPTS.NE.0) GO TO 18
      PRINT 210
  210 FORMAT(1H0,'NIL')
      GO TO 19
   18 PRINT 204,((IPPT(I,J),J=1,2),I=1,NPPTS)
  204 FORMAT(1H0,15(I3,':',I2,2X),//,10(1X,15(I3,':',I2,2X),//))
C
C  PRINT HEADINGS FOR NODE QUANTITIES
C
   19 IF(IA.LT.1) PRINT 208
  208 FORMAT(1H0,/,13X,'LOAD INCREMENT',16X,'TOTAL LOAD',12X,'DISPLACEME
     $NT INCREMENT',8X,'TOTAL DISPLACEMENT',/,5X,4(3X,'/',23('-'),'/'),/,
     $' NODE',7X,'DPX',11X,'DPY',11X,'PX',12X,'PY',12X,'DU',12X,'DV',
     $13X,'U',13X,'V',/)
      IF(IA.EQ.1) PRINT 215
  215 FORMAT(1H0,/,13X,'LOAD INCREMENT',16X,'TOTAL LOAD',12X,'DISPLACEME
     $NT INCREMENT',8X,'TOTAL DISPLACEMENT',/,5X,4(3X,'/',23('-'),'/'),/,
     $' NODE',7X,'DPR',11X,'DPZ',11X,'PR',12X,'PZ',12X,'DU',12X,'DV',
     $13X,'U',13X,'V',/)
C
C  PRINT NODE QUANTITIES FOR OUTPUT REQUESTED
C ******************************************
C
      DO 5 I=1,NN
      IF(NODEPR(I).NE.1) GO TO 5
      NR=(I-1)*2
      J=NR+1
      L=NR+2
      PRINT 209,I,DP(J),DP(L),P(J),P(L),DU(J),DU(L),U(J),U(L)
  209 FORMAT(1H ,I4,8(3X,E11.6))
    5 CONTINUE
C
C  CHECK FOR TERMINATING ANALYSIS
C
    7 I=IFIX(PMAX(LP,1))
      IF(I.EQ.-1.OR.IOUT(NOUT).EQ.-1) ISTOP=-1
      RETURN
      END
```

Methodology :

Since the methods of analysis are rather complex and mathematical they cannot conveniently be dealt with in lectures of this nature. Some of the more important methods will, however, be given in brief outline.

(i) Regression : The proportion of zonal trips by each mode is expressed as a function of the system trip and user characteristics, e.g.:

$$Y = a + B_1X_1 + B_2X_2 + \dots\dots\dots\dots B_nX_1$$

Where X_1 = Transit riding time and (Walk and wait and transfer) time. Driving time and terminal time.

X_2 = Median family income for zone of production.

Zone of Production:

X_3 = housing units/net residential acre.

X_4 = cars/housing unit.

X_5 = accessibility to employment (transit/car).

X_6 = 9 hr. parking cost (average rate per hour).

X_7 = 3 hr. parking cost (average rate per hour).

X_8 = employment/gross acre.

X_9 = accessibility to population (transit/car)

Y = percentage trips by transit.

(ii) Discriminant Analysis : Develop a linear function of the form

$$f = a_1 x_1 + a_2 x_2 + \dots\dots\dots\dots + a_n x_n$$

which will distinguish between travellers making different mode-choice decisions on the basis of a set of variables x defining the characteristics of the travellers of the alternative choices available to them.

Let $f = (a_1 x_1) + (a_2 x_2)$

be a discriminant function.

Where x_1 = cost difference mode 1 versus mode 2

x_2 = time difference mode 2 versus mode 1.

Choose values for a_1, a_2 such that f is small for travellers using mode 1 and large for those using mode 2.

Utility and Preference Analysis :

Individual preference for travel modes may be estimated by subjecting a set of travellers to a structured series of modal alternatives, defined in terms of varying values of relative cost, time, comfort, etc. and evaluating their "threshold" response (i.e. the point at which a change in a particular modal characteristic would bring about a change in mode choice). Ideally, such an experiment should be based on actual changes in the transportation system; it may be approximated by means of a laboratory or interview experiment. Equivalent analysis may be performed for route choice, ranking modal and route characteristics in order of importance and evaluating response to new forms of transportation.

since the water resistance had to be taken into account, it was necessary to use a voltage comparator (LM 319) which would output either 5 v or 0 v when the output from the probe itself was > 4,5 v or < 1 v respectively. This was compared with the voltage at the comparator created by either resistors R1 and R2 or resistors R3 and R4, depending on which probe was under consideration. The essential difference between the two circuits was that the potential at probe 1 was inverted.

4.4.2 RC Oscillator

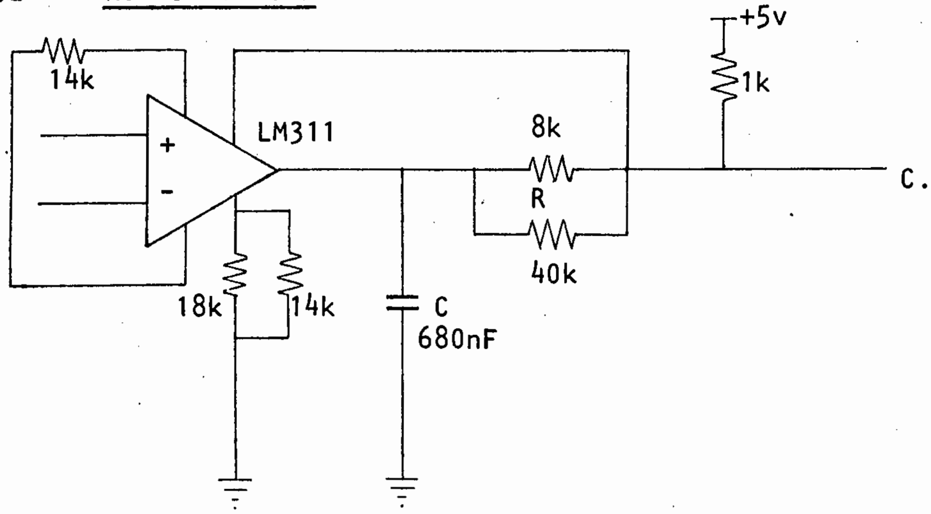


FIGURE 4.5

The RC oscillator was designed so that the temperature co-efficients of R and C cancelled each other out. To achieve this the following components were used:

metal film resistor - temperature coefficient = - 50 ppm/°C
poly-carbonate capacitor - temperature coefficient = + 50 ppm/°C

Thus,

$$\frac{1}{f} = \frac{1}{2\pi RC}$$

where

R is resistance in ohms
C is capacitance in far ads
f is frequency

13 DEC 1978