Modeling techniques in metal forming: upsetting of a forged plastic ring

Hsien-Yung Lee
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Part of the Applied Mechanics Commons

Recommended Citation
https://lib.dr.iastate.edu/rtd/8495

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University Microfilms International
300 N. Zeeb Road
Ann Arbor, MI 48106
Lee, Hsien-Yung

MODELING TECHNIQUES IN METAL FORMING: UPSETTING OF A FORGED PLASTIC RING

Iowa State University

University Microfilms International 300 N. Zeeb Road, Ann Arbor, MI 48106
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark ✓.

1. Glossy photographs or pages
2. Colored illustrations, paper or print
3. Photographs with dark background
4. Illustrations are poor copy ✓
5. Pages with black marks, not original copy
6. Print shows through as there is text on both sides of page
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements
9. Tightly bound copy with print lost in spine
10. Computer printout pages with indistinct print
11. Page(s) lacking when material received, and not available from school or author.
12. Page(s) seem to be missing in numbering only as text follows.
13. Two pages numbered. Text follows.
14. Curling and wrinkled pages
15. Other

University Microfilms International
Modeling techniques in metal forming: Upsetting of a forged plastic ring

by

Hsien-Yung Lee

A Dissertation Submitted to the Graduate Faculty in Partial Fulfillment of the Requirements for the Degree of DOCTOR OF PHILOSOPHY

Department: Engineering Science and Mechanics
Major: Engineering Mechanics

Approved:
Signature was redacted for privacy.
In Charge of Major Work
Signature was redacted for privacy.
For the Major Department
Signature was redacted for privacy.
For the Graduate College

Iowa State University
Ames, Iowa
1983
# TABLE OF CONTENTS

## I. INTRODUCTION

1

## II. REVIEW OF LITERATURE

### A. Solutions Other Than The Finite Element Method
4

### B. The Finite Element Method in Nonlinear Solid Mechanics
5

1. Major works of the finite element method in plasticity
   6
2. Major works in coupled large deformation and plasticity
   8
3. Application of finite elements mainly to metal forming
   a. Small deformation
   10
   b. Large deformation
   11

### C. Review of Ring Compression
13

## III. THE FINITE ELEMENT METHOD
18

### A. Basic Equations
18

1. Linear elastic analysis
   a. Equilibrium equations
   19
   b. Strain-displacement equations
   19
   c. Constitutive equations
   20
2. Elastic-plastic analysis
   a. Equilibrium equations
   20
   b. Strain-displacement equations
   21
   c. Constitutive equations
   21
   d. Yield criterion
   23
   e. Flow rule
   27
   f. Decomposition of strain increment
   29

### B. Basic Solution Procedures
30

1. Tangent stiffness method
   31
2. Initial stress method (Initial load method)
   31

### C. Correction for Transition Zone
35

1. Initial stress method
   36
2. Tangent stiffness method
   36

### D. Equilibrium Check
38

1. Tangent stiffness method
   39
2. Initial stress method
   40
3. Numerical errors
   40
3. Verification of finite element program 111
4. Instability 113a
5. Computer experience 114
6. Similitude requirement 114
7. Interaction of experiment and numerical method 115
8. Advantages/disadvantages of FEM and photoplasticity 116

VI. CONCLUSION 118
VII. FIGURES 120
VIII. TABLES 210
IX. BIBLIOGRAPHY 239
X. ACKNOWLEDGMENTS 248
XI. APPENDIX A - DERIVATIONS AND MATRICES 249

A. Matrices for Plasticity 250
   1. Elastic stress-strain relationship 250
   2. Strain-displacement relationship 252
   3. Stiffness matrix for an element 254
   4. Plastic matrix 256
   5. Formula r for transition zone 257

B. Matrices for Large Deformation 259
   1. Geometrically nonlinear stiffness matrix - axisymmetric 259
   2. Equivalent load - axisymmetric 261
   3. Transformation from Jaumann stress rate to Cauchy stress rate 262

C. Plastic Matrix for Modified Von Mises Criterion 264

XII. APPENDIX B - SUPPORTING INFORMATION 271

A. List and Explanation of New Macros and Input 272
   1. List of new macro programming commands 272
   2. Necessary order of macros 273
   3. Input for boundary condition change cards (BCNG macro) 273
   4. Modification of input for ELMt01 273
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.</td>
<td>Actual Input Data and Necessary JCL for Running</td>
<td>274</td>
</tr>
<tr>
<td>1.</td>
<td>Job control language (JCL) used for running MINIFEM</td>
<td>274</td>
</tr>
<tr>
<td>2.</td>
<td>Input data for case 1.a</td>
<td>276a</td>
</tr>
<tr>
<td>3.</td>
<td>Input data for case 1.b</td>
<td>279a</td>
</tr>
<tr>
<td>4.</td>
<td>Input data for case 2</td>
<td>282a</td>
</tr>
<tr>
<td>5.</td>
<td>Input data for case 3</td>
<td>287a</td>
</tr>
<tr>
<td>6.</td>
<td>Input data for case 4</td>
<td>290a</td>
</tr>
<tr>
<td>C.</td>
<td>Slicing Plan in Experiment</td>
<td>293</td>
</tr>
<tr>
<td>D.</td>
<td>Schematic of Data Flow</td>
<td>294</td>
</tr>
<tr>
<td>XIII.</td>
<td>APPENDIX C - PROGRAM LISTING</td>
<td>295</td>
</tr>
<tr>
<td>A.</td>
<td>Listing of the Finite Element Program - MINIFEM</td>
<td>295</td>
</tr>
<tr>
<td>B.</td>
<td>Listing of the Finite Element Postprocessor</td>
<td>437a</td>
</tr>
<tr>
<td>C.</td>
<td>Listing of the Strain Separation Program</td>
<td>500a</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

Metal forming is a very important industrial process. The degree of its efficiency determines both the quality and the cost of the product. Since most of the process designs are ad hoc and ingenious, a detailed look from the view of mechanics can help to optimize such processes or even break new ground.

Experiments have been conducted over the past few years on upsetting, forging and rolling using photoplasticity [1-3]. However, adequate assessment of these results is lacking since there are no adequate theoretical and experimental solutions for comparison.

Although numerical methods, especially the finite element method, have been developed to solve these problems, they themselves also need to be evaluated by comparison with experimental results. The general trend in the literature of developments in the finite element method is to give detailed descriptions of the theory. Comparisons with experiments, if made, are scanty, and when given, seldom mention details of the modeling process. At the same time, when, in the literature of experiments, comparisons are made with numerical solutions, descriptions of the solution procedures are often omitted so that it is sometimes hard to determine whether a realistic comparison has been made.

Therefore, it is seen that there is a lack of assessment for
both approaches.

Photoplasticity and finite elements are both methods that are particularly attractive since they can both be used to evaluate internal distribution of important parameters such as strains or stresses. However, they remain only models and as such need to be matched with a prototype. Birefringent plastics are used in photoplasticity, but their properties are fundamentally different from metals. The finite element method appears more adaptable, since it can be applied to different materials, be it metal or plastic when the correct material properties and process environments are given. Although some authors expressed pessimistic views on photoplasticity [4] by saying "... the likelihood of the properties of metal and birefringent plastic being the same is almost zero; the probability of just a qualitative resemblance is small." and put more faith in numerical solution by saying "... computer solutions are far more reliable and much easier to obtain." the author would like to point out, however, the feasibility of both modeling methods and their relative merits by comparing their respective results for the case of an upset forged ring.

The objective of the present research is then two-fold:

1. Develop a numerical model specifically for the photoplastic material and the accompanying forging
process.

2. Compare the results of finite element method with photoplasticity, specifically the internal parameters, such as strains and displacements, so that the relative merits of photoplasticity and finite element method can be assessed. Then, new insight might be gained into both methods and further research stimulated in both directions. Although comparisons can be made only for the deformation patterns after the load has been removed, the loaded patterns from numerical solution are also given in the hope that one day loaded photoplasticity can reach a stage that comparisons can also be made and more light can be shed on these methods.

Because of the versatility of the finite element method, it is hoped that after enough backing from experiments, the finite element method can become 'the' modeling tool.
II. REVIEW OF LITERATURE

A. Solutions Other Than The Finite Element Method

To this date, there are very few exact solutions available in plasticity. Since they do not play a major role, no specific attention will be given to them in this review. A sample of what is available can be found in the books by Hill [5], Mendelson [6] and Johnson and Mellor [7].

The first major work which had a general application to plasticity was the numerical solution by Southwell and Allen [8]. As an application of the then famous 'Relaxation Method', they solved the problems of a notched specimen in tension for plane stress and plane strain conditions using the Hencky-Mises (or better known as von Mises) yield criterion and the Prandtl-Reuss flow rule. Stress components, equivalent stress, displacement and the propagation of the yield zone were presented.

Another important contribution was the method of 'successive elastic solutions' first used by Ilyushin and then elaborated by Mendelson and others. Problems of two-dimensional stress concentrations were solved.

The above contributions used a combination of finite difference discretization, stress functions and iterative

---

1 See Chapters 9 and 10 of [6] for detailed descriptions.
solution procedures. These methods first obtain an elastic solution and then correct it to take the effect of plasticity into account until convergence is reached for a given load increment. The whole process is then repeated until some maximum deformation is reached. The most serious disadvantage of these approaches is that only very simple boundary conditions such as force or pressure loading can be applied. The same disadvantage also applies to the finite difference method which has, for example, been used to solve the problem of a thick-walled cylinder under internal pressure [9,10]. Though severe restrictions limit the wide and direct application of these methods, the underlying concept of continuous adjustment of an elastic solution to obtain an elastic-plastic one can be applied to other more powerful and flexible methods such as the boundary element method [11] and the finite element method.

B. The Finite Element Method in Nonlinear Solid Mechanics

The finite element method [12], is perhaps the most versatile of all numerical methods. It has the distinctive advantage that it can treat very complex boundary conditions.\(^2\) An excellent brief description of the history of the Finite Element Method can be found in the book by

\(^2\) Although in many cases they are neither easy nor cheap, at least they can be done.
Martin and Carey [13]. Not long after the introduction of the Finite Elements as a general method in structural analysis, efforts were made to apply it to nonlinear problems in solid mechanics. In this review, we distinguish two types of nonlinearity: material nonlinearity, or 'plasticity' and geometrical nonlinearity, or 'finite deformation'.

1. Major works of the finite element method in plasticity

Marcal treated two-dimensional elastic-plastic problems by expressing the equilibrium equations in terms of displacements [14,15] and using the finite element method together with the concept of partial stiffness. The method was incremental, used von Mises yield criterion, the Prandtl-Reuss flow rule and could handle strain-hardening problems. Three numerical examples were presented: (1) For a close-ended thick-walled cylinder in plane strain under internal pressure, the result of pressure versus outside surface strain was compared with the closed-form solution. (2) The maximum stress versus maximum strain at a point in a perforated tension strip in plane strain was compared with the experimental work of Theocaris and Marketos [16]. (3) Load versus maximum strain (at a point) and the propagation of plastic zones of notched tension specimens in various two-dimensional cases were compared with the relaxation

---

3 From the point of view of finite element method,
solution of Southwell and Allen [8]. The results showed the
same characteristics but did not agree in details.

It was not until 1968 that an explicit form of plastic
stress-strain matrix was formulated and documented by Yamada
and Yoshimura [17] for the incremental tangent stiffness
method. They 'inverted' the Prandtl-Reuss equations
'analytically' as opposed to the numerical inversion of
Marcal and King [14]. A rigorous treatment of the
transition zone was also formulated. Again, two-dimensional
notched specimens in tension served as numerical examples
and the load was pushed to near failure. Results were then
compared with slip line solutions. No unloading was allowed
but its possibility was discussed. At this stage, the
tangent stiffness method was well-established as a standard
procedure in solving elastic-plastic problems.

Then in 1969, Zienkiewicz et al. [18] adopted the
method of successive elastic solutions and proposed the, now
famous, initial stress method. By converting
plastic strains to equivalent load, this method was said to
yield rapid convergence and allow large load increments.
Notched specimens in tension were again used as numerical
examples and compared with experimental works [16] for mean
plane strain, plane stress and axisymmetric conditions are
considered as two-dimensional, whereas in photomechanics, to
be discussed in Chapter IV, the axisymmetric problem is
treated as three-dimensional.
stress versus maximum strain and with the partial stiffness method [14].

A generalization of the above works in 1972 by Nayak and Zienkiewicz [19] established the 'initial stress' method (later renamed to 'initial load' method) as a powerful algorithm for solving elastic-plastic problems. Strain-hardening, strain-softening, associated and nonassociated plasticity, transition zone and acceleration of convergence among other features were treated. Pressure versus deflection of a point in a pressurized steel vessel was compared with experiment. Mean pressure versus displacement of a point in plane strain extrusion was compared with experiment. Both showed good agreement.

With the successful formulation of the tangent stiffness and initial stress methods, the framework for plasticity problems was now firmly established. Other contributions can be considered as extensions or improvements within this framework.

2. Major works in coupled large deformation and plasticity

Not long after analysis in plasticity began, research into geometrically nonlinear problems was underway. Martin [20] pioneered the use of Lagrangian strain. Hibbit et al. [21] formulated the 'Total Lagrangian' method by transforming the equilibrium and elastic-plastic constitutive equations from the current configuration to the
initial configuration. They identified four stiffness matrices in addition to the conventional small deformation stiffness matrix. A theoretical formulation was given but not implemented numerically. The prospect of using an Eulerian formulation based on the current configuration was mentioned.

An Eulerian approach was formulated and implemented for two-dimensional situations by Osias and Swedlow [22] who investigated a necking problem. Detailed examination by the author showed that this work was really an updated formulation. By using a Galerkin's procedure and adopting Jaumann (co-rotational) rate of Cauchy stress they obtained a nonsymmetric stiffness matrix. In pursuing a symmetric stiffness matrix, McMeeking and Rice [23] obtained another Eulerian formulation by using Jaumann rate of Kirchhoff stress and rate of deformation tensor as conjugate variables. This was based on the theoretical work of Hill [24] which showed that, as a first order of approximation, the Jaumann rate of Kirchhoff stress is conjugate to the logarithmic strain. By casting the governing equations in terms of arbitrary conjugate variables, they also assessed the relative validity of various other formulations and pinpointed their errors. This formulation had the advantage that the conventional small strain elastic-plastic finite element program can be "simply and rigorously adopted to
problems involving arbitrary amounts of deformation...." A necking problem was also investigated. Their work was generalized by Nagtegaal and de Jong [25] and termed 'The Updated Lagrangian' method. A special computational technique was introduced, and a numerical example was given about the axisymmetric upsetting of a disk. The hydrostatic pressure distribution at a disk/tool interface was obtained and compared with experiment. The comparison showed large differences.

3. Application of finite elements mainly to metal forming

a. Small deformation Lee and Kobayashi investigated several problems, including plane strain and axisymmetric flat punch indentation [32] and extrusion and upsetting of a solid cylinder of pure aluminum [33, 34]. For these problems, punch pressure versus displacement, development of the plastic zone, a contour of effective plastic strain and principal stresses under load and upon removal of load were presented for the cases of sticking and zero friction.

Plane strain compression of a block under sticking and

---

The method was called 'Eulerian' by McMeeking and Rice [23] and by Osias and Swedlow [22], 'Updated Eulerian' by Yamada and Sakurai [26]. The now popular name 'Updated Lagrangian' was used by Lee, Mallet and Yang [27], Lee, Mallet and McMeeking [28], Argyris et al. [29], Yamada [30], Yamada and Hirakawa [31], Nagtegaal and de Jong [25]. The names Eulerian and Updated Eulerian in the work of Argyris et al. [29] have different meanings.
nonsticking friction was studied by Nagamatsu, Murota and Jimma [35]. The shape of the bulge, the distribution of normal pressure, the tangential stress on the die/workpiece interface and the development of plastic zones were presented. They also studied the axisymmetric upsetting of hollow cylinders (rings) [36].

Plane strain and axisymmetric extrusion under hydrostatic pressure was investigated by Iwata et al. [37]. Hydrostatic pressure versus displacement at the billet end, spread of plastic zone, effective plastic strain distribution and the stress distribution under various friction conditions were presented.

Owen et al. [38] compared the stresses in the bending of a partly yielded notched bar under plane strain as obtained by three different numerical programs. Comparison was also made with a slip line solution.

Odell [39] studied wall ironing in plane stress. Total ram force versus ram displacement was compared with upper and lower bound solutions.

b. Large deformation Wifi [40] obtained a complete solution of the stretch forming and deep drawing of a circular blank using a hemispherical punch under axisymmetric condition by employing the Total Lagrangian Method. Strains, stresses and growth of plastic zone were presented.
Lee, Mallet and Yang [27] was the first to apply the Updated Lagrangian method introduced by McMeeking and Rice [23] to obtain a complete solution for plane strain extrusion through a frictionless die. The distribution of stress at steady state was presented. An extension to axisymmetric problems was also studied [28].

Yamada and Hirakawa [31] treated the same problem but included friction. An explicit load correction matrix was formulated.

Key [41] investigated two-dimensional extrusion, rolling and sheet stretching. Limited experimental data were used for comparison.

Argyris et al. [29] used the Updated Lagrangian method and a viscous flow approach to solve the problem of extrusion of aluminum billet through a curved nozzle and the heading of a steel bolt.

Unless specifically mentioned, the results obtained were not compared with experiment.

It is clearly seen from the above review that there is a general lack of comparison of the finite element solution with experimental results. This is true for problems in plasticity, and even more so for problems that involve coupled plasticity and finite deformation. If ever made, most of the comparisons, are either limited to one particular point or to an interface or surface where
experimental data can be collected. It is unfortunate that
the sophistication of this method, which can predict the
response of a whole field, has only been checked for a small
region. Agreement at a point or a small region obviously
cannot guarantee the behavior of a whole 2- or 3-dimensional
field. This does not mean that these limited comparisons
are of no value. It simply points out the need for a
wider range of comparison when permitted by the available
methods of experiment.

It is interesting to note that the rare full field
experiment done by Theocaris and Marketos [16] was almost
the sole basis for comparison with the solutions of the
finite element method in their early stages of development
[14, 18]. Even for this case, the comparison was limited to
one point only. In the field of finite deformation, even
such a comparison is not available.

C. Review of Ring Compression

The ring compression test was first developed for
measuring the friction between a workpiece and the forming
tool by Kunogi [42]. There are two modes of flow for the
ring. Under low friction, it deforms like a solid cylinder:
both inner and outer radii move outward. Under high
friction, however, the inner radius flows inward and the
outer radius flows outward. Consequently, a neutral radius-
the radius which divides inward and outward flow—is developed for the case of high friction.

We first review major experimental works and upper bound solutions.

Application of the upper bound method for the compression of a ring was first made by Kudo [43] in 1960-61. He used Coulomb coefficient of friction in his solution. Avitzur [44] in 1964 solved the same problem but modeled friction as a constant shear factor \( m \). Male and Cockcraft [45] performed experiments to determine the classical coefficient of friction \( \mu \). Sticking to zero friction conditions were modeled by using various lubricants which included wax and metal lubricants. Calibration of the friction factor of the lubricants was obtained by plotting \% decrease of inner diameter versus \% reduction in height. The calibration curve for various friction conditions was then compared with an elementary ad hoc theoretical solution. The value of \( \mu \) was found not to be a constant during large plastic deformation.

Hawkyard and Johnson [46] in 1967 used constant \( \mu \) and stress equilibrium approach and obtained identical results with that of Avitzur [44]. Van Sant [47] obtained another simplified solution by assuming constant volume deformation and position of neutral radius. Correlation was made with the experimental results of Male and Cockcraft [45].
Male and DePierre [48], in 1970 compared the solution by Avitzur [44] with experiment and found good agreement in the shape of calibration curves. A calibration curve is the one relating % of decrease of inner diameter to the % of axial deformation of the ring for a particular value of friction. However, the actual values of m did not agree with all cases of deformation. The assumptions of the theoretical solution was to blame.

Liu [49] in 1972 considered bulging due to friction by assuming a parabolic profile for barreling and used an assumed velocity profile. Stress and strain histories at the outer equatorial surface were analyzed. Reasonably close agreement with experimental data was claimed. Lee and Altan [50] considered strain-hardening in addition to bulging by integrating the equivalent strain rate and using a realistic stress-strain curve. Calibration curves, upsetting loads and flow stress were presented and claimed to be good. Lahoti and Kobayashi in 1974 [51] used Hill's general method of analysis for metal forming to solve for bulging using m and compared the results with those of DePierre et al. [52] and Liu [49].

Nagamatsu et al. [53] in 1973 performed experiments where aluminum rings of various dimensions were deformed. Shape changes, load of compression, distribution of normal pressure and mean friction coefficient at the die/ring
interface were measured. The friction coefficient was found to increase and then decrease as deformation proceeded.

Bulging was also considered by Avitzur and Sauersine [54], in 1978, by superposing the velocity field of a disk on that of a ring without bulging. Temperature effect was also considered by Nagpal, Lahoti and Altan [55].

The Finite Element Method was first applied to ring compression in 1972 by Nagamatsu et al. [36]. They used an elastic-plastic model. Then, in 1978, Chen and Kobayashi [56] used a rigid-plastic model. For further comments on these works and upper bound solutions see [57].

The most promising work to date was the elastic-plastic finite element solution by Hartley, Sturgess and Rowe [57, 58]. By using the so called 'β-stiffness' method (β being a function of the interface shear factor m), they obtained a calibration curve and a curve of punch force required for deformation versus deformation. Change in geometry and punch force versus deformation was compared with experiments for m = 0.1 and 0.7. Good results were obtained for low friction but not when the friction was high. No flow patterns were presented.

The most serious disadvantage of the ring test for assessing friction quantitatively is the lack of analytical solutions. So far, no internal stress and strain pattern has been presented. Upper bound solutions do not have this
capability. The only two available elastic-plastic finite element solutions [36, 57, 58] also did not present these. In addition, no comparison of any internal or surface stress or strain patterns with experiments have been made.
III. THE FINITE ELEMENT METHOD

The details of the modeling technique using the finite element method are given in this section. The emphasis is on simulating, as closely as possible, the physical process of metal forming in general and ring compression in particular. A general description of the mathematical theory of plasticity is first presented. The author believes that the most crucial part in modeling is to recognize what considerations are needed for a realistic simulation. Consequently, each section below first gives the reason why a special treatment is needed and then shows how it is solved. Most proofs and derivations will be given in Appendices so as not to obscure the main line of the presentation.

A. Basic Equations

Plasticity analysis will be considered here as an extension of elasticity analysis. Hence, a parallel is first drawn between them. We use the displacement formulation by assuming time independent small deformations and an elastic-plastic material. Matrix and cartesian tensor notations will be used side by side whenever possible. The former is more suitable for computational purpose while the latter is used for conciseness and reference to literature.
1. Linear elastic analysis

The linear elastic analysis assumes a material model which follows the Hook's law. The governing equations can be classified into three categories: (a) the equilibrium equations (b) the strain-displacement equations (c) the constitutive equations (stress-strain relationship equations).

a. Equilibrium equations

\[ \sigma_{ij,j} + F_j = 0 \]  \hspace{1cm} (1a)

where \( \sigma_{ij} \) are the stress components, \( \sigma_{ij,j} \) are the partial derivatives of \( \sigma_{ij} \) with respect to coordinate \( j \) and \( F_j \) is the body force. When transformed by the principle of virtual work (which is a 'weak' form of the original equations) the standard finite element stiffness equations follow

\[ [K] \{u\} = \{f\} \]  \hspace{1cm} (1b)

where \([K]\) is the global stiffness matrix, \([u]\) the displacement vector and \([f]\) the force vector. Hook's law is seen recovered here.

b. Strain-displacement equations

\[ \varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \]  \hspace{1cm} (2a)

where \( \varepsilon_{ij} \) are the strain tensor, \( u_i \) is the displacement, \( u_{i,j} \) is the partial derivative of \( u_i \) with respect to coordinate \( j \).
where $\{\varepsilon\}$ is the vector of strain components, $[B]$ is the strain-displacement matrix.

**c. Constitutive equations**

$$\sigma_{ij} = D_{ijkl} \varepsilon_{kl}$$  \hspace{1cm} (3a)

where $D_{ijkl}$ are the elastic constants.

$$\{\sigma\} = [D] \{\varepsilon\}$$ \hspace{1cm} (3b)

where $\{\sigma\}$ is the vector of stress components, $[D]$ is the elastic matrix. Then, in terms of the strain-displacement matrix $[B],$

$$[K] = \int_Y [B]^T [D] [B] \, dv$$  \hspace{1cm} (4)

2. **Elastic-plastic analysis**

Being path dependent and nonlinear, the governing equations for elastic-plastic analysis are best cast in incremental forms. The governing equations can be classified into five categories: (a) incremental equilibrium equations (b) incremental strain-incremental displacement equations (c) constitutive equations relating incremental stress to incremental strain (d) a yield criterion to determine whether a point has yielded or not (e) a flow rule to determine the incremental plastic strain components.
a. Equilibrium equations

In the absence of body force,

$$\frac{d\sigma_{ij},j}{dt} = 0$$  \hspace{1cm} (5a)

$$[K^E]\{\Delta u\} = \{\Delta f\}$$  \hspace{1cm} (5b)

where $\sigma_{ij}$ are the incremental stress components, and $[K^E]$ is the elastic-plastic stiffness matrix, $\{\Delta u\}$ the displacement increment vector, and $\{\Delta f\}$ the force increment vector.

b. Strain-displacement equations

$$d\varepsilon_{ij} = \frac{1}{2}(du_{i,j} + du_{j,i})$$  \hspace{1cm} (6a)

where $\varepsilon_{ij}$ are the incremental strain components, $du_i$ the incremental displacement, $du_{i,j}$ the incremental displacement gradient with respect to $j$ coordinate.

$$\{\Delta \varepsilon\} = [B] \{\Delta u\}$$  \hspace{1cm} (6b)

where $\{\Delta \varepsilon\}$ is the incremental strain vector (strain rate).

c. Constitutive equations

Assuming hypoelastic material [59], we get

$$\dot{\sigma}_{ij} = L_{ijkl} \varepsilon_{kl}$$  \hspace{1cm} (7a)

---

5 $\Delta$ can be considered either as increment or derivative with respect to time or other equivalent loading parameter. Time can be considered as an artifice not necessarily in the sense of time-dependency. Hence, $\Delta u$ is often called the velocity, $\Delta \varepsilon$ the strain rate and $\Delta \sigma$ the stress rate [59].

6 A hypoelastic material has a linear stress rate versus strain rate relationship.
where $\dot{\sigma}_{ij}$ are the stress rate, $\dot{\varepsilon}_{ij}$ are the strain rate, and $L_{ijkl}$ are the elastic-plastic coefficients.

$$\{\Delta \sigma\} = [D^{EP}]\{\Delta \varepsilon\}$$  \hspace{1cm} (7b)

where $[D^{EP}]$ is the elastic-plastic matrix. Then, from eqs (5b), (6b) and (7b),

$$[k^{EP}] = \int_V [B]^T[D^{EP}][B] \, dv$$  \hspace{1cm} (8)

The total quantities of any vector

$$\text{vector} = \int_{\text{path}} d(\text{vector}) = \sum_{i=1}^{N} (\text{vector})_i$$  \hspace{1cm} (9)

where $N$ is the total number of increments (or time steps).

This is often called the 'integration process'. For example, the vector can be strain, displacement, stress etc.

From the above, it can be seen that there are two basic differences between elastic and elastic-plastic analysis:

- Elastic-plastic analysis is incremental.
- $[D]$ in linear elasticity is to be replaced by $[D^{EP}]$ in plasticity.

We now proceed to evaluate $[D^{EP}]$. If we adopt a flow theory\(^7\) of plasticity we need:

\(^7\) Another popular theory is the deformation theory of plasticity. It neglects the elastic portion of strain and assumes the total strain to be composed only of the plastic strain. The material model which follows the deformation theory is called a rigid-plastic model. For the present research, modeling of elastic unloading is necessary (see
• A yield criterion to determine the onset of yielding at a point by correlating the uniaxial phenomenon (uniaxial stress-strain curve) to multidimensional cases.

• A flow rule to determine the strain increments in the plastic zone (plastic stress-strain relationship)

d. Yield criterion We define

$$F = \bar{\sigma} - \sigma_y$$  \hspace{1cm} (10)

where $\bar{\sigma}$ is the equivalent stress, $\sigma_y$ is the yield (flow) stress and $F$ is the loading function. We also define $\sigma_0$ as the initial yield stress.

Equivalent stress $\bar{\sigma}$, also called effective stress, yield criterion or yield function determines the stress state of a point in a continuum at a given load. One example of equivalent stress is the von Mises stress as

---

The mechanism of elastic unloading is by recovering the elastic strains which are accumulated during the loading process up to the unloading point. A rigid-plastic model, being void of elastic component of strain, is incapable of modeling elastic unloading. Consequently, this model is rejected.

There are at least six different ways of defining the initial yield stress. See [60] for the relative merits. The definition used in this dissertation is the same as that in Gomide's work [51] where a secant modulus of 0.7E was used (E being the Young's modulus).
defined in eq (15a).

In practice, \( \bar{\sigma} \) is calculated from the current values of stress components. \( \sigma_y \), however, is not obtained by calculation. Its value is assigned depending on the current state of stress of a point. The relationships between the loading function, equivalent stress and the yield stress will be explained below.

Note that the one-dimensional stress-strain curve can be considered as a representation of the three-dimensional yield surface. A yield surface is the surface described by a yield function, such as the one defined in equation (15a), in the stress space. Examples of yield surfaces are given in section H and are illustrated in Fig. 13.

The possible stress-strain paths which a point in a continuum can traverse are shown in Fig. 1. The arrows in the figure indicate possible directions in which a typical point can move along the stress-strain curve.

Assuming no initial residual stress, point 0 is the starting point of the analysis (stress- and strain-free state). Let us increase the load until point 1 on the stress-strain curve is reached. This is seen as an elastic zone. So \( (\bar{\sigma}_1) < \sigma_0 \), \( (\sigma_y)_1 = \sigma_0 \) and \( F < 0 \). The subscript indicates the point marked in Fig. 1. For example, \( (\bar{\sigma}_1) \) is the equivalent stress at point 1.

When the load is increased further, point 2 is reached.
At this point, \((\sigma)_{2} = \sigma_{o}, \ (\sigma_{y})_{2} = \sigma_{o}, F = 0\). That is, the point has just reached the yield surface. This is called initial yielding.

With continued loading, point 3 can be reached. The plastic zone has been entered and \((\sigma)_{3} > \sigma_{o}, \ (\sigma_{y})_{3} = (\sigma)_{3}, F = 0\). Since \((\sigma_{y})_{2} = \sigma_{o}\), so \((\sigma)_{3} > (\sigma_{y})_{2}\) and \((\sigma_{y})_{3} > (\sigma_{y})_{2}\). That is, by increased loading, the yield strength of the point also increases. This phenomenon is called strain-hardening. The advantage of raising the yield strength of a material by strain-hardening is unfortunately accompanied by a loss in ductility.

By further increasing the load from point 3, three possible destinations can be reached:

1. **Point 5:** This is a continuation into the plastic zone. The path traversed from point 3 to point 5 is a continuation of the path from point 2 to point 3. hence, a description similar to that given for point 3 applies.

2. **Point 3:** The point becomes stationary, i.e., it remains on the previous yield surface. (Note that for a strain-hardening material, the yield surface is assumed to expand monotonically during continued loading. The expansion of the yield surface has been shown earlier through the fact that \((\sigma_{y})_{3} > (\sigma_{y})_{2}\). For this case, the yield
surface itself remains stationary under continued loading! This phenomenon is called neutral yielding and $(\sigma)_{3}^{t-1} = (\sigma)_{3}^{t}$. The superscript indicates time step (increments). That is, $(\sigma)_{3}^{t-1}$ is the equivalent stress of the point at a previous increment and $(\sigma)_{3}^{t}$ is the equivalent stress of the same point at the current increment. Physically, we can consider the state at which $(\sigma)_{3}^{t} = (\sigma)_{3}^{t-1}$ a bifurcation state. Since instability can very possibly occur at the next increment.

3. Point 4: Elastic unloading occurs, that is, $(\sigma)_{4}^{t} < (\sigma)_{3}^{t}$ and $(\sigma)_{3}^{t} > \sigma_{0}$. This state is detected when the current equivalent stress $(\sigma)_{4}^{t}$ is less than the previous yield stress $(\sigma)_{3}^{t}$. Then by definition of strain-hardening, $(\sigma)_{4}^{t} = (\sigma)_{3}^{t}$. That is, for continued loading, in order for the particular point which was unloaded to again enter into the plastic zone, the value of the equivalent stress for the subsequent time step must be greater than $(\sigma)_{4}^{4}$.

As far as the author is aware, the conditions 2 and 3 above have not been clearly elucidated and explained by experiments.
The above discussions can be summarized as follows:

point 0 $F = \sigma = \sigma_y = 0$ stress free state (11a)

point 1 $F < 0, \sigma < \sigma_y, \sigma_y = \sigma_o$ not yielding, in the elastic zone. (11b)

point 2 $F = 0, \sigma = \sigma_o = \sigma_y$ initial yielding. (11c)

point 3 $F = 0, \sigma = \sigma_y > \sigma_o$ continued (post) yielding. (11d)

point 4 $F < 0, \sigma < \sigma_y, \sigma_y > \sigma_o$ elastic unloading of a previously yielded point (11e)

point 5 (same as point 3) (11f)

The possibility of 3 above makes the process path dependent.

e. Flow rule

$$d\varepsilon_{ij}^p = d\lambda \frac{\partial Q}{\partial \sigma_{ij}}$$ (12a)

$$\{d\varepsilon^p\} = d\lambda \{\frac{\partial Q}{\partial \sigma}\}$$ (12b)

where $d\varepsilon_{ij}^p$ is the plastic strain increment, $\{d\varepsilon^p\}$ is the plastic strain increment vector, $d\lambda$ is an instantaneous constant which usually varies with the loading history, $Q$ is the plastic potential. Plastic potential is similar to a strain energy density function.

We define the deviatoric stress $\sigma'_{ij}$ as

$$\sigma'_{ij} = \sigma_{ij} - 1/3\delta_{ij} \sigma_{ii}$$ (13)

where $\delta_{ij}$ is the Kronecker delta defined by

$$\delta_{ij} = 0, i \neq j; \delta_{ij} = 1, i = j.$$ 

The second invariant of the deviatoric stress $J_2'$ is defined as
Equivalent stress $\bar{\sigma}$ for the von Mises yield criterion is defined as

$$\bar{\sigma} = \left[ \frac{2}{3}(\sigma_x - \sigma_y)^2 + \frac{2}{3}(\sigma_y - \sigma_z)^2 + \frac{2}{3}(\sigma_z - \sigma_x)^2 \ight.\
\left. + 3\tau_{xy}^2 + 3\tau_{yz}^2 + 3\tau_{xz}^2 \right]^{1/2}$$

(15a)

where

$$\sqrt{\frac{3}{2} J_2} \equiv \sqrt{\frac{3}{2} \sigma_{ij} \sigma_{ij}}$$

Since $\sigma_{ij}$ is independent of hydrostatic pressure, von Mises yield criterion is said to be a pressure independent yield criterion.

Equivalent (effective) plastic strain increment $d\varepsilon_p$ is defined as

$$d\varepsilon_p = \sqrt{2}/3 \left\{ (d\varepsilon_{x}^P - d\varepsilon_{y}^P)^2 + (d\varepsilon_{y}^P - d\varepsilon_{z}^P)^2 + 6((d\varepsilon_{xy}^P)^2 +
(d\varepsilon_{yz}^P)^2 + (d\varepsilon_{zx}^P)^2 + (d\varepsilon_{z}^P - d\varepsilon_{x}^P)^2 \right\}^{1/2}$$

(15b)

where the superscript $p$ indicates that the components of strain are plastic ones.

Associated plasticity is defined for the situation where the flow rule (eq (12)) is evaluated with the plastic potential $Q$ equivalent to the yield criterion $\bar{\sigma}$. For example, when $Q$ is replaced by eq (15a), we obtain the flow rule associated with von Mises yield criterion. That is [6],
\[
d\varepsilon_{ij}^P = \sigma'_{ij} \ d\lambda = \frac{3}{2} \ \frac{d\varepsilon_{ij}^P}{\sigma'} \sigma'_{ij} \quad (16a)
\]
\[
\{d\varepsilon^P\} = \frac{3}{2} \ \frac{d\varepsilon_{ij}^P}{\sigma'} \ \{\sigma'\} \quad (16b)
\]

where \(\{\sigma'\}\) is the vector of deviatoric stress, and the famous Prandtl-Reuss equations are recovered.

Eq (16) implies the following [6]:

- Volume is constant in the plastic range because

\[
d\varepsilon_{ii}^P = d\varepsilon_{x}^P + d\varepsilon_{y}^P + d\varepsilon_{z}^P = 0
\]

- Von Mises yield criterion is used.

- Directions of the principal stress and principal plastic strain increments are the same.

f. Decomposition of strain increment

Assuming that the incremental strain can be decomposed into elastic and plastic parts, we have

\[
d\varepsilon_{ij} = d\varepsilon_{ij}^e + d\varepsilon_{ij}^P \quad (17)
\]

where \(d\varepsilon_{ij}^e\) are the elastic strain increments and \(d\varepsilon_{ij}^P\) are the plastic strain increments. And we find that [17]

\[
d\varepsilon_{ij}^P = \sigma''_{ij} \ d\lambda + \frac{d\sigma'_{ij}}{2G} \quad (18a)
\]

This gives the complete strain rate-stress relationship for elastic-plastic analysis. A more convenient form of eq (18a) is obtained by solving for \(d\lambda\) and 'inverting' the above equations [18, 19]. Then,
when von Mises yield criterion and its associated flow rule are used. Where \([D^{ep}]\), \([D]\), \(Q\) are defined in eqs. (7b), (3b) and (12) respectively.

And \(H' = \text{strain-hardening parameter}\)

\[
\varepsilon = \frac{d\sigma}{d\varepsilon^p}
\]

Eq (18b) can be expressed as

\[
[D^{ep}] = [D] - [D] \frac{\partial Q}{\partial \sigma} \frac{\partial F^T}{\partial \sigma} \frac{\partial Q}{\partial \sigma} \frac{[D]}{\partial \sigma}
\]

(18c)

Where \([D]\) is called the elastic matrix and \([D^{p}]\) is called the plastic matrix. The explicit form of \([D^{ep}]\) and other matrices for two-dimensional cases are given in Appendix A.

When the plastic potential \(Q\) is different from the yield criterion in evaluating the flow rule (eq (12)), we have the condition of nonassociated plasticity. This is to be explained later in section I of this chapter.

B. Basic Solution Procedures

There are two basic solution procedures for integrating the governing equations (18a). Both methods are incremental in nature. The first, or the 'tangent stiffness method', does not in principle require iteration while for the second, or the 'initial stress method' iteration is mandatory.
1. Tangent stiffness method

In a method analogous to the first order Euler's forward method for solving ordinary differential equations, the incremental equilibrium equations (5b) can be written as

\[ [K_{i-1}^{ep}] \Delta u_i = \{ \Delta f \}_i \]  

(19)

where the subscript \( i \) is the number of increment. Since this is a step by step incremental process, \( i \) is often referred to as the 'time step'.

Refer to Fig. 2(a) and eq (19). The instantaneous slope of the stress-strain curve at a specific time step \( i \) is represented by the stiffness matrix \( [K_{i-1}^{ep}] \) from the previous time step \( (i-1) \). It is used to predict the displacement increment \( \Delta u_i \) at the current time step \( i \) to generate a piecewise linear approximation to the continuous physical deformation process. Fig. 2(b) shows the skeleton of the procedure. Since the stiffness matrix needs to be reformed at the start of every increment, the procedure is also called the Newton-Raphson method [12].

2. Initial stress method (Initial load method)

Combining the incremental equilibrium equations (5b) & (8) gives

\[ \int_V [B]^T [D_p^{ep}] [B] \, dV \{ \Delta u \} = \{ \Delta f \} \]  

(20a)

But since

\[ [D_p^{ep}] = [D] - [D_p] \]  

(18c)
We can substitute eq (18c) into eq (20a) and obtain

\[
\left( \int_V [B]^T [D] [B] \, dV \right) \{\Delta u\} = \{\Delta f\} + \left( \int_V [B]^T [D^P] [B] \, dV \right) \{\Delta u\} \tag{20b}
\]

Substituting eq (4) into eq (20b), we obtain

\[
[K] \{\Delta u\} = \{\Delta f\} + \left( \int_V [B]^T [D^P] [B] \, dV \right) \{\Delta u\} \tag{20c}
\]

We define the plastic stress increment \(\{\Delta \sigma^P\}\) as follows:

\[
\{\Delta \sigma^P\} = [D^P] [B] \{\Delta u\} = [D^P] \{\Delta \varepsilon\} \tag{20d}
\]

So that

\[
[K] \{\Delta u\} = \{\Delta f\} + \left( \int_V [B]^T \{\Delta \sigma^P\} \, dV \right) \tag{21a}
\]

\[
= \{\Delta f\} + \{\Delta f\}_{il} \tag{21b}
\]

where \(\{\Delta f\}_{il}\) is called the initial load and is defined by,

\[
\{\Delta f\}_{il} = \int_V [B]^T \{\Delta \sigma^P\} \, dV \tag{22}
\]

The initial load is obtained from the plastic stress increment \(\{\Delta \sigma^P\}\) (eq (21a)). The plastic stress increment can be considered as some initial residual stress (produced by plasticity), hence sometimes, the name 'initial stress method' is employed. In general, iteration within an increment is mandatory. A schematic of this iterative process is shown in Fig. 3.

Upon being given an increment of load, \(\{\Delta f\}_i\), the objective is to find the corresponding displacement...
increment $[\Delta u]_i$. The path a-b-c in Fig. 3 illustrates the operation of a typical iteration.

For one iteration in Fig. 3, assume $\Delta \sigma_1't$ is the initial (unbalanced) stress at the beginning of the iteration. (The superscript used in this paragraph and Fig. 3 is not the deviatoric stress.) Consider the triangle a-b-d. Since the slope $K$, the elastic stiffness, is known, we can find the correct strain increment $\Delta e_1$ corresponding to $\Delta \sigma_1't$. Knowing the slope from a to c (since a-c lies on the stress-strain curve), the stress increment $\Delta \sigma_1'P$ for the iteration can be found (from triangle a-c-d). The new 'initial stress' is now $\Delta \sigma_1'D$. It can be obtained by subtracting from $\Delta \sigma_1't$. A detailed procedure for obtaining the initial stress will be given in the next section when it is coupled with the transition zone.

Any iteration process needs a criterion to test convergence. Termination of convergence is achieved when the ratio of the square of norm$^9$

\begin{equation*}
\text{ratio} = \left( \frac{\|(\Delta u)_{i-1}\|^2}{\|\{(\Delta u)_{i}\}\|^2} \right)
\end{equation*}

where the subscript $i$ refers to the index of iteration.

$^9$ Alternatively or combinatorially a displacement convergence criterion can be adopted. For example, the ratio can be replaced by

\begin{equation*}
\text{ratio} = \left( \frac{\|\{(\Delta u)_{i-1}\}\|^2}{\|\{(\Delta u)_{i}\}\|^2} \right)
\end{equation*}
where the superscript \( il \) stands for initial load. Fig. 4 shows the skeleton of flow for the initial stress method. Since the original (elastic) stiffness is used throughout the analysis, and provided that the decomposed stiffness matrix is stored, only a re-resolution (forward and backward substitution) is needed for each iteration. This is considerably faster than reforming the stiffness matrix and decomposing it. The method is also called the modified Newton-Raphson method [12] since the 'slope' (stiffness matrix) does not change during the analysis.

In eq (21a), the plastic stress increment \( \Delta \sigma^P \) is used as the 'initial stress'. The framework of analysis would be essentially the same if other types of 'initial stresses' take the place of \( \Delta \sigma^P \). That is, we can redefine the initial load to be

\[
\{ \Delta f \}^{il} = \int_V [B]^T \{ \Delta \sigma \}^{il} \, dV
\]

where \( \{ \Delta \sigma \}^{il} \) can be any kind of initial stress, such as: thermal stress, residual stress, in situ stress, creep stress, stress from finite deformation etc. Conceptually, any stress that arises in addition to the elastic stress as obtained by simple mechanical loading can be considered as an initial stress.
C. Correction for Transition Zone

When a continuum is deforming, yielding will occur sequentially according to the magnitude of the equivalent stress. That is, initial yielding will occur at a point when the magnitude of the equivalent stress \( \bar{\sigma} \) is equal to the initial yield stress \( \sigma_0 \) (eq (11c)). However, in finite element analysis a fixed load increment is usually applied. It is not known beforehand what the magnitude of the load increment is required to cause yielding at a sampling point.\(^{10}\) Hence, a point which was elastic can enter into the plastic zone with an equivalent stress that is higher than the initial yield stress instead of being equal to it.

Different approaches have been used to treat this problem. Yamada and Yoshimura [17] used a scaling method in the tangent stiffness approach while an iterative method was used by Nayak and Zienkiewicz [19] for an initial stress approach. The present research uses a noniterative approach for both solution procedures.

\(^{10}\) Numerical integration is usually required for evaluating the stiffness matrix in eq (8). When a Gaussian integration method is used, the points at which integration is performed are called Gauss points. This approach is commonly used for isoparametric elements. For simple triangular elements, integration is usually performed at the centroid of the triangle. Sampling points are defined, however, as the points where stresses and strains are evaluated. The usual practice is to let sampling points coincide with integration points.
1. **Initial stress method**

Refer to the schematic shown in Fig. 5. If the current stress point is A, then an increment of load may move point A to point B which is not on the stress-strain curve, i.e., it is beyond the initial yield surface. The correct point for this iteration should be C. The detailed step by step solution is shown in Fig. 6 where matrix and vector notations are dropped. Note that we have bypassed the steps for finding E by finding C directly. This eliminates the need for global scaling to be discussed in the next section. Linear hardening is used here for illustrative purpose only. In general, for nonlinear strain-hardening material we like to keep point C close to E so that the slope change from E to C will not deviate too much from the true slope. In other words, a small increment of load is preferred.

2. **Tangent stiffness method**

This treatment is attributed to Yamada and Yoshimura [17]. As shown in Fig. 7, the correct point which is just on the initial yield surface is point C. For a given load increment, it is possible for many points in a body to go beyond the initial yield surface. The amount of overshoot however is usually different for each point. But in order for yielding to occur sequentially, only the points which have the lowest magnitude of equivalent stress are allowed to reach the yield surface. Hence, a ratio $r$ (Ref. Appendix
A) is calculated for all points inside the transition zone (i.e., the points which have gone beyond the initial yield surface) and the minimum of the ratios is calculated to scale all global incremental values such as stress, strain, displacement etc., for example:

\[ \{\Delta f\}_\text{new} = r_{\text{min}} \times \{\Delta f\}_\text{old} \]

Experience shows that the number of points which have the minimum ratio is usually one! That means, for most of the cases, given one load increment, there is at most one point which can reach the initial yield surface. If the magnitude of the load increment is small, there may not be any point at all which will reach the yield surface for this increment. The restriction that only one point can yield (reach the initial yield surface) for a load increment means that the minimum number of increments needed for all elements to yield is approximately the same as the number of total sampling points in the domain. (A counter example is an axisymmetric thick-walled cylinder under internal pressure. In this case, all points which have the same value of radial coordinate will yield at the same time.)

This restriction is partially relaxed by considering points whose equivalent stresses fall within a certain percentage of the initial yield stress as having reached the yield surface. For example, for a point if \(0.99 \sigma_0 < \overline{\sigma} < 1.01 \sigma_0\), then the point is considered to
have yielded. By this approach, more points can yield at
the same time within an increment. This allows fewer load
increments to be used in the analysis. Consequently, it
becomes more economical.

D. Equilibrium Check

Consider the total equilibrium equations obtained by
integrating eq (20a) over the path of loading

$$\sum_{i=1}^{N} \int_{V} [B]^T [D^eP][B] \, dV \{\Delta u\} = \sum_{i=1}^{N} \{\Delta f\} \quad (25a)$$

where $i = 1$ to $N$ is the index for any of $N$ increments. $N$ is
the total number of increments.

From equation (25a)

$$\int_{V} [B]^T \{\sigma\} \, dV = \{f\} \quad (25b)$$

Let

$$\psi = \text{residual force} = \int_{V} [B]^T \{\sigma\} \, dV - \{f\} \quad (26)$$

$\psi$ is defined as the residual or total unbalanced force. It
should be zero if equilibrium is satisfied. In practice,
this means that reactions should be equal to zero for those
nodes with no prescribed tractions or displacements, and
should conform to the boundary conditions for prescribed
nodes. Deviation\textsuperscript{11} from the stress-strain curve however is

\textsuperscript{11} This deviation was apparently not noticed in some of
likely to occur for at least three reasons. These three reasons are explained according to three situations at which the deviation can occur. The first situation can occur when the tangent stiffness method is used, the second one can occur when the initial stress method is used and the last situation can occur when computer arithmetics is involved.

1. **Tangent stiffness method**

By virtue of eq (5b), the stiffness matrix (slope) $[K^e]$ is constant within an increment. Consequently, we can consider the total integrating process (eq (25b)) as a linear piecewise operation. That is, the total path is divided into small increments or pieces and within each increment, the process is linear (eq (5b)). Since this process is only an approximation, drifting from the true response can occur as shown in Fig. 8(a). Consider three consecutive points $i-1$, $i$, $i+1$ on the dashed line. The dashed line shows the deviated path while the solid line shows the path of true response. The effect of deviation is accumulative, since the slope for the increment $i$ to $i+1$ depends on the position of $i$ and $i$ is predicted by the increment $(i-1, i)$. $A-B$ in the figure indicates the typical

---

the early works. For remarks concerning this see Knothe and Muller [52].
unbalanced force at a point. From the figure, it seems that if the material follows a linear hardening law then there will be no drifting because it corresponds to a linear approximation. However, drifting is often observed even for this case. The reason is that the one-dimensional schematic used here is really an idealization for illustrative purpose only. Its interpretation cannot be carried too far as the actual domain is three-dimensional in nature. For example, if we have a perfectly plastic material, then according to the one-dimensional idealization the initial stress method will never converge! This is obviously ridiculous.

2. Initial stress method

Since a convergence criterion has to be employed, a 'true' response cannot be obtained in a strict sense. For example, if in Fig. 8(b), the true response is point C, iteration may stop at point B either because of a loose convergence criterion or because a maximum number of iteration has been exceeded. Clearly equilibrium has not been achieved. The difference BC is then calculated as the unbalanced force.

3. Numerical errors

Numbers are never exactly represented in a computer. Manipulations of these inexact numbers bring in errors by truncation, approximation etc. Since the equilibrium
equations (5b) are solved many times as simultaneous equations, numerical errors can occur and lead to unbalanced force.

Regardless of the source, the problem is treated by calculating the residual force at the start of each increment by eq (26), and adding it to the present load increment. In the literature, this method is also called 'one-step correction'. Alternatively, when the initial stress method is used, the unbalanced force can be corrected for each iteration. This is not done presently in this dissertation.

E. Acceleration of Convergence

For the initial stress method, the iteration process may converge very slowly under certain circumstances. Hence, in order to save computation time an accelerating scheme is desired. While Nayak and Zienkiewicz [19] used an iterative scheme called the 'a-stiffness method', we adopt the modified Aitken accelerator first proposed by Boyle and Jennings [63]. It is essentially an extrapolation process based on the assumption that the 'error' (residual force in our case) decays exponentially. The accelerator is defined as follows
\[
S = \frac{(\delta_2 - \delta_1)^T (-\delta_2)}{(\delta_2 - \delta_1)^T (\delta_2 - \delta_1)}
\]

where \(\delta_1\) and \(\delta_2\) are two consecutive incremental displacement vectors. Superscript T indicates transpose of the vectors. Note that the multiplying operation above is essentially a 'dot' operation of two vectors and renders \(S\) as a scalar.

The extrapolation process is shown schematically in Fig. 9. Although the formula for the accelerator \(S\) is due to Boyle and Jennings [63], the accelerating scheme is formulated by the author for the present research. It proceeds as follows by starting from a typical point A:

- Obtain \(\delta_1\) by reaching point B.
- Use unbalanced force BC to obtain \(\delta_2\) by the procedures in Fig. 4 for example. Calculate \(S\) by eq (27) from \(\delta_1\) and \(\delta_2\).
- Obtain accelerated displacement increment \(\delta = (1+S)\delta_2\).
- The correct point is seen as F. So the correct total stress is \(\sigma_B + \Delta\sigma^{ep}\) where \(\Delta\sigma^{ep}\) is calculated from \((1+S)\Delta\varepsilon_2\).
- The total displacement is \(x_{i+1} + \delta\). The total strain is \(\varepsilon_B + (1+S)\Delta\varepsilon_2\).
- The new unbalanced force is then \(\Delta\sigma_2 - \Delta\sigma^{ep}(FH)\).

Note that the intermediate state D is never calculated.
explicitly. Although this approach is similar to the work of Atrek [64], his intention was only to obtain accelerated convergence for one-dimensional case so that the process of obtaining the new residual load (HF in Fig. 9) is ambiguous. When the accelerator becomes negative, the process becomes meaningless, and the accelerating process is suspended until the accelerator becomes positive again [64]. Also note that the acceleration is performed within an increment. Since the increment itself is already made very small, this process will not have the seeming disadvantage of using a large increment.

F. Large Deformation

Traditionally, two views of mechanics are employed: solid mechanics and fluid mechanics. Within this framework, three types of analysis can be used depending on the coordinate system and the configuration that are being used. The following shows the classification of names used for analysis:

<table>
<thead>
<tr>
<th>Name for Analysis Systems</th>
<th>Fluid Mechanics &amp; Solid Mechanics</th>
<th>Solid Mechanics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Coordinate</td>
<td>Configuration</td>
</tr>
<tr>
<td>Eulerian</td>
<td>Spatial</td>
<td>Deformed</td>
</tr>
<tr>
<td>Lagrangian</td>
<td>Material</td>
<td>Undeformed</td>
</tr>
<tr>
<td>Convective</td>
<td>Material (solid mechanics only)</td>
<td>Undeformed or deformed</td>
</tr>
</tbody>
</table>
A 'spatial coordinate' is defined when the coordinate system is fixed in space. A 'material coordinate' is defined when the coordinate system is fixed on the system of particles. 'Configuration' is defined as the state of the physical object at a certain time. Configuration is in general not important in fluid mechanics because usually a region fixed in space is of interest. The identification of coordinate system, on the other hand, is of interest. Since all the laws of mechanics are expressed for systems, i.e. for material coordinates, there is the need to relate laws written for material coordinates to laws written for spatial coordinates. In solid mechanics, however, both configuration and coordinate are important, since a distinction has to be drawn between deformed and undeformed states and the history of the particles which are continuously deforming is of interest. This greatly complicates the analysis.

All the above terms follow standard terminology. Their definitions can be found for example in the books by Fung [65], Malvern [59] and Mason [66]. However, special note needs to be taken of the 'convective coordinate' because of its extensive use in the derivation of equations. It is defined as a coordinate system that is embedded in the body and retains its value while deforming [65]. It can be considered as a special case of a material coordinate.
A straightforward analysis in large deformations is usually carried out by a Lagrangian analysis (i.e., material coordinate and undeformed configuration are used). The second Piola-Kirchhoff stress \([65, 59, 66]\) and Green strain \([59, 65, 66]\) are used. However, when coupled with plasticity, which is expressed in terms of true stress and true strain, a current configuration (deformed) is desirable where a rate type (of all the parameters) rather than total type needs to be used (refer to section A of this chapter.)

There are several approaches to deal with the problem of coupled large deformation and plasticity. One way is to transform all the equations of plasticity from the current configuration to the undeformed configuration. This method is referred to as the Total Lagrangian Method \([21]\). A second way is just the opposite of the above. That is to transform the Lagrangian equations to the current (deformed) configuration. This method is called an Eulerian formulation. The third way is to adopt an artifice by updating the reference configuration to the current configuration, where the reference configuration is the configuration of the previously deformed state. This method is called the 'Updated Lagrangian Method'.

Theoretically speaking, a continuous updating is possible. But in an incremental finite element approach, finite increments are actually used, hence different degrees
of approximations must be adopted. This will be described in the following paragraphs.

We will exclude the Total Lagrangian Method from our discussion because of the high cost associated with implementing and executing such a method on computers.

The now widely accepted name 'Updated Lagrangian' analysis can be stated as

- During an increment, a Lagrangian analysis is used.
- At the end of an increment, the variables are updated to the current configuration.

By updating the configuration we have extended the classical views of Eulerian and Lagrangian analysis. We first focus our attention on what is going on during one increment. After a finite increment of deformation, two different configurations will invariably be generated ($V^t$ and $V^{t+\Delta t}$), where the superscript indicates a time-like parameter. When we consider $V^t$ as the 'undeformed' or 'reference' configuration, then we must consider $V^{t+\Delta t}$ as the 'deformed' or 'current' configuration. It is obvious that $V^t$ is not the same as the traditionally termed 'undeformed', 'reference' configuration $V^0$. They coincide only at the beginning of the deformation process. It would be difficult to perform a strict Eulerian analysis because usually the current configuration for calculating stress and strain is not available. Hence, a Lagrangian analysis is preferred,
and reference is made to the material coordinate at the
start of each increment. In practice, when using finite
elements, a fixed orthogonal coordinate system is usually
employed which will be the reference for both the deformed
and undeformed configurations. If a convective coordinate
is to be used, then at state $V_{t+\Delta t}$ the original orthogonal
coordinate at $V_t$ will become curvilinear (direction is
messed up!) and is not suitable for finite element
analysis. Since we still desire to use the orthogonal
coordinate for the analysis of the next increment, a
convective coordinate may have its place in derivation but
may not be practical.

For a Lagrangian analysis during an increment, the
conjugate stress rate and strain rate are the second Piola-
Kirchhoff stress rate $\dot{S}_{ij}$ and the Green strain-rate $\dot{\eta}_{kl}$. They are related by

$$\dot{S}_{ij} = L^{ijkl} \dot{\eta}_{kl}$$

But when plasticity is involved, $L^{ijkl}$ needs to be
calculated from true stress and true strain, which can be
interpreted as referring to the current configuration. The
most important consideration in forming constitutive
relations is that invariance (objectivity) must be enforced.
The Jaumann rate of Cauchy stress or Kirchhoff stress is
usually employed. These are the stress rates viewed from a
rotated coordinate system, and hence are objective to any rigid body rotation. When properly transformed, \( L_{ijkl} \) will become symmetric if the Jaumann rate of Kirchhoff stress is used (which implies the existence of a strain rate energy) and nonsymmetric if the Jaumann rate of Cauchy stress is used.

Hence, incremental displacement can be obtained through finite element approximation, and then Green strain rate and second Piola-Kirchhoff stress can be updated. The Cauchy stress is then transformed from the total second Piola-Kirchhoff stress by standard transformation [59]. The Cauchy stress can then be used to formulate \( L_{ijkl} \) in the constitutive equation for the analysis of next increment. Also, coordinates of the nodal points of the finite element mesh and the Green strain are updated. This completes the analysis of an increment [57].

The seeming advantage of the above formulation is that large increments may be used because the full form of the Green strain rate is used. Since the Green strain rate contains all nonlinear terms of the rate type, large increments of load can be used. However, the plasticity analysis requires small increments so the response will not deviate from the stress-strain curve. If small increments are used to satisfy the requirement of plasticity, then we can omit the nonlinear terms in the Green strain rate. The
omission of these nonlinear terms implies the condition of small deformation analysis for an increment. In other words, the full nonlinear terms of the Green strain rate are meaningful only when a significant difference between the deformed and undeformed states arises (large increment is used). When we interpret the thus linearized strain rate as the rate of deformation tensor, then the name Updated Eulerian or Eulerian method is not inappropriate (see Chapter II).

The present research is then based upon this argument for using a linearized strain rate for the analysis of an increment. For small increments, we will adopt the rate form of virtual work principle by Hill [24] and following Yamada [30], and McMeeking and Rice [23]:

$$\int_{V^0} \{\dot{\epsilon}\}^T \{\dot{s}\} \, dV^0 = \int_{V^0} \delta \{\dot{u}\}^T \{\dot{X}_o\} \, dV^0 + \int_{S^0} \delta \{\dot{u}\}^T \{\dot{F}\} \, dS^0 \quad (28)$$

where integration is over the reference configuration, $V^0$, $S^0$ are the volume and area at reference configuration, $\{\dot{u}\}$ is the virtual velocity gradient, $\{\dot{s}\}$ is the nominal stress rate, $\{\dot{X}_o\}$ and $\{\dot{F}_o\}$ the body force and surface traction referred to the reference configuration. The term of body force $\{\dot{X}_o\}$, since it does not enter into the present problem, will be omitted henceforth for brevity.

The constitutive equation relating stress rate and strain rate suitable for this analysis is
\[ \tau_{ij} = L_{ijkl} D_{kl} \]  
\[ \{t\} = [D^\text{ep}]\{\dot{e}\} \]

where \( \tau_{ij} \) is the Jaumann (co-rotational) rate of Kirchhoff stress and \( D_{kl} \) the rate of deformation tensor. \([D^\text{ep}]\) is the conventional small deformation elastic-plastic matrix. Hill [68] has shown that \( \tau_{ij} \) is conjugate to the logarithmic strain rate as a first degree of approximation, \( \{s\} \) is related to \( \{t\} \) [31] by

\[ \{s\} = \{t\} + [D^*] \{e\} \]

Substituting into equation (28) one obtains the finite element equation:

\[ ([K^\text{ep}] + [K^G])\{\dot{u}\} = \{\dot{F}\} \]

where \([K^\text{ep}]\) is the conventional small strain elastic-plastic stiffness matrix, \([K^G]\) is the geometrical nonlinear stiffness matrix and is given by [31] as

\[ [K^G] = \int_V [B_e]^T [D^*] [B_e] \, dv \]

For a typical increment, the displacement increment is obtained, and the rate of deformation tensor and the Jaumann

\[ \text{12} \] The more general stress rate is the Jaumann rate of Cauchy stress, which will lead to a nonsymmetric matrix.
rate of Kirchhoff stress are calculated. Since the Jaumann rate of Kirchhoff stress is viewed from a rotated axis (the rotation that has occurred during the increment), it needs to be transformed back to the Cauchy stress rate. Then total stresses, strains and coordinates etc. are updated at the end of the increment. When using the initial load method,

$$\{\Delta f\}_{il} = \{\Delta f\}_{ilp} + \{\Delta f\}_{gn}$$

where \(\{\Delta f\}_{ilp}\) is the initial load from plasticity (eq (22)) and \(\{\Delta f\}_{gn}\) is the initial load due to finite deformation. The explicit forms of the geometric nonlinear stiffness G matrix \([K]\), the geometric nonlinear initial load \(\{\Delta f\}_{gn}\) and the equations of transformation from the Jaumann rate of Kirchhoff stress to the Cauchy stress rate are given in Appendix A.

McMeeking and Rice's [23] original motivation and emphasis in pursuing this type of analysis was to find the formulation which admits a rate potential and in turn leads to a symmetric stiffness matrix, so that the conventional small strain program can be easily adopted. The formulation is indeed attractive. However, the symmetry condition may not be as important as the authors claimed. Since an initial load scheme is used, the nonsymmetry can be contained at the element level, hence a global nonsymmetrical stiffness matrix is never explicitly formed. By this scheme, no extra
computer time is needed in dealing with the nonsymmetry. (As will be shown later in section I of this chapter, for nonassociated plasticity, the stiffness matrix derived is also nonsymmetrical. This nonsymmetry is handled in the same way.) Also, by assuming hypoelasticity, a rate potential is not required.

G. Interfacial Friction

There are two basic concepts for measuring the effect of friction: Coulomb friction and interface shear factor. Coulomb coefficient of friction \( \mu \) is the most well-known and defined as

\[
\mu = \frac{\tau}{\sigma_n}
\]  

(33)

where \( \sigma_n \) is the normal stress at the interface and \( \tau \) is the shear stress at the interface due to friction. The interface shear factor, \( m \), is a relatively new concept. It is defined (34) as:

\[
m = \frac{\tau^*_{\text{(friction layer)}}}{\tau^*_{\text{(work piece)}}}
\]  

(34)

where \( \tau^*_{\text{o}} \) is the initial yield stress in shear for the friction layer, (a friction layer is a layer of lubricant at the tool/workpiece interface), \( \tau^*_{\text{o}} \) is the initial yield stress in shear for the workpiece (a ring in our case). The initial yield stress in shear for any material can be
obtained from a simple torsion test.

The interface shear factor $m$ is used widely in 'upper bound' solutions for metal forming problems [44, 69]. When $m = 0$, the process is frictionless. When $m = 1$, there is no sliding at the tool/workpiece interface, and this is called 'sticking' friction. The relative merits of $\mu$ and $m$ are discussed in [70, 71]. We adopt $m$ in our analysis since it is easier to use in modeling as shown below.

The two characteristic patterns for the flow of a ring in axial compression are show in Fig. 10, where $\tau$ indicates the interface shear stress caused by friction. In the right hand portion of the figure, the friction is low and all the radial flow is outward. In the left hand portion of the figure, the friction is high and the inward portion flows inward and the outward portion flows outward. A neutral radius is defined as the radius at which the radial displacement is zero, i.e., the neutral radius is the boundary which separates inward and outward flow. From the definition of tangential strain $\varepsilon_\theta$, a neutral radius is also the one where $\varepsilon_\theta = 0$. The major problem with calculating the deformations of the ring is that the position of the neutral radius is not known a priori. Once the neutral radius is known, a mixed boundary condition (displacement and traction input on the same part of the boundary) can be specified.
For sticking friction, the interfacial friction is so high that the top layer of the workpiece is virtually bonded to the die. This can be modeled by viewing friction as a constraint on the workpiece and by adding an additional layer of friction elements, which have very high stiffness when compared to that of the workpiece at the interface. This is shown in Fig. 11.

A quantitative form for the total range of friction was proposed by Hartley et al. [58] as follows:

$$\beta = \frac{m}{1-m}$$

where $\beta$ is a multiplication factor for friction. They call this approach the '\(\beta\)-stiffness' method. In summary, when $m = 0$, $\beta = 0$ frictionless

$0 < m < 1$ intermediate friction

$m = 1$, $\beta = \infty$ sticking friction.

This approach is one of many possible ways for satisfying the friction boundary conditions, i.e. $\beta = 0$ when $m = 0$ and $\beta = \infty$ when $m = 1$. It has no other physical or theoretical support. However, it should be adequate for frictionless and sticking friction.

In contrast to the finite element work by Hartley et al. [58], who used elastic-plastic elements for the friction layer, the author of this dissertation used an elastic layer and adopted the name 'modified $\beta$-stiffness method' for the process. The reasons for this approach are:
The actual physical phenomenon occurring at the interface is complicated. It is not always easy to maintain a constant level of lubrication. Hence, it is a common practice that the lubricant be replaced periodically during the deformation process. Using an elastic-plastic layer throughout the deformation process is definitely a simplified representation of the actual process at the interface. This approach has the disadvantage that after most of the elastic-plastic friction elements undergo yielding, instability can occur as these elements break down. That is, elastic unloading can occur for these elements! The underlying physical mechanism of this instability is not well understood at this moment. There is not enough physical evidence showing that the friction layer actually behaves in a simple elastic-plastic way. By keeping the friction layer elastic, we are assuming a constant restraint (stiffness) on the workpiece. This is similar to a physical process where a constant friction is maintained as mentioned above in this paragraph. In this approach, the friction layer will remain stable.

H. Elastic Unloading

In the photoplasticity technique used by Gomide [61], the unloaded deformed model was sliced and the analysis was performed on the slices. Here, unloading refers to a process
where the workpiece is removed from the loading machine. Since the measurements were performed on an 'unloaded model', the photoplastic patterns were from residual deformations.

In order to have a realistic numerical model to compare with the experiments, unloading also needs to be implemented in the numerical model. Apparently, this was overlooked by the work of Friere, et al. [72].

In this dissertation, the input to the numerical program for the upsetting of a forged ring is in increments of displacement rather than force since the latter is not known beforehand. The analysis for the case of unloading is relatively simple if known forces or pressures are applied, because by adding a force or pressure which has an equal and opposite magnitude to the total force or pressure at the end of the loading step, the unloading process can be accomplished. For a displacement input, the unloading process is more difficult to model. A rigorous procedure which requires incremental unloading is called for [34]. In this dissertation, the author used a simplified approach: one step unloading. For displacement input, this is accomplished by changing the boundary conditions of nodes from prescribed displacement to prescribed force input, where the forces, by virtue of the change of boundary conditions, are revealed at the nodes, which are then used
as unbalanced or residual forces for the input of the unloading step (Fig. 12). This type of one step unloading is commonly used in analytical solutions [73, p.476] and is acceptable if only a few points undergo further yielding during the unloading process. This should be true for most cases and also true for the photoplastic modeling processes.

At the end of the unloading process, for a 'complete' unloading, all external and internal forces should be equal to zero. The external forces at the start of the unloading process are generated by the change of boundary conditions mentioned above. The internal forces, if present, are due to the situation as discussed in section D of this chapter. It is desirable that this latter force be equal to zero so that only the external force is present. However, in order to save computational time, this latter force is usually not zero. This can be caused, for example, by allowing too large a tolerance during the iteration process. Decreasing tolerances can improve this situation, however, at the expense of increasing computation time. Hence, we see that due to the need of the unloading process, a more stringent requirement needs to be imposed on the accuracy obtained in the loading process.
The general assumption in classical plasticity theory that yielding is independent of hydrostatic pressure is approximately true for metals. It is in general not true for polymers.

The influence of pressure on yielding was proposed independently by Schleicher and von Mises according to Hill ([5], p. 21). Hence, the name modified von Mises was used. A brief historical account of the yield criterion was given by Raghava et al. [74]. By performing tests on several glassy [74] and crystalline [75] polymers, they demonstrated that the same yield criterion could be used for different types of polymers when macroscopic yielding was the only concern.

The material 'Laminac' used by Gomide [61] also seems to follow this yield criterion. This is confirmed from the investigation of initial yield surface by Zachary and Riley [76] and Freire and Riley [77].

A mathematical construction of the yield surface was formulated by Tschoegl [78]. The pictorial construction shown in Fig. 13 is Meldahl's [79]. The length of the axis, measured from the origin of the $\sigma_1, \sigma_2, \sigma_3$ coordinate system, of the yield surface of the modified von Mises criterion (hypothesis of Schleicher in Fig. 13) is a measure of the hydrostatic pressure $(\sigma_1 + \sigma_2 + \sigma_3)/3$. For von Mises
yield criterion (hypothesis of the maximum energy of deformation in Fig. 13) since the radius of the yield surface (a cylinder) remains constant along the axis of the cylinder, it is concluded that yielding for von Mises yield criterion is independent of hydrostatic pressure. For modified von Mises yield criterion, the radius of the yield surface increases in the direction of increasing hydrostatic pressure (in the quadrant of \(-\sigma_1, -\sigma_2, -\sigma_3\)). Consequently, the higher the hydrostatic pressure, the more difficult it is for yielding to occur. The yield surface for modified von Mises yield criterion converges to a limiting point for certain triaxial tension states, beyond which the yield surface is not defined. This is never reached in a compression process.

When reduced for one-dimensional case, this yield criterion indicates that the yield stress is different in compression from that in tension.

For the modified von Mises yield criterion [74, 75, 78]:

\[
F = \frac{1}{\sqrt{2}} \left( (\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6\tau_{xy}^2 + 6\tau_{yz}^2 + 6\tau_{xz}^2 \right)^{\frac{1}{2}} - \left\{ CT - (\sigma_x + \sigma_y + \sigma_z)(C - T) \right\}^{\frac{1}{2}}
\]

\[
= \sqrt{2} - \left\{ CT - J_1(C - T) \right\}^{\frac{1}{2}}
\]

\[
= \bar{\sigma} - \left\{ CT - J_1(C - T) \right\}^{\frac{1}{2}}
\]

(36)
where $F$ is the loading function (ref. eq 10), $J_2$ is the second invariant of stress components, $J_1$ is the first invariant of stress components, and $C$ and $T$ are the absolute values of yield stress in compression and tension. Note that the $J_1$ is a pressure term. Physically, we can consider the continuum to be made up of inhomogeneous materials (points) where each material (point) has a different varying yield stress $[CT - (\sigma_x + \sigma_y + \sigma_z)(C-T)]^{\frac{1}{2}}$. The state of stress and strain for a particular point depends on the instantaneous hydrostatic pressure, as dictated by the boundary conditions, and on the instantaneous constraint exerted by the neighboring points. In practice, the word 'points' refer to the sampling points of the elements (ref. section C of this chapter).

While both associated and nonassociated plasticity can be used, we chose the latter form by using $\sqrt{J_2}$ as the yield potential $Q$ (as in eq 12) in order to satisfy incompressibility requirement in the plastic region (see section A of this chapter). The incompressibility requirement is based on the observation that Poisson's ratio for rigid Laminac is 0.485 [61].

Note that the yield potential $Q$ gives the subsequent yield surface after initial yielding. For a certain material, the forms of initial yield surface and subsequent
yield surface should be determined by experiments. For associated plasticity, since the subsequent yield surface is assumed to be the same as the initial one, usually only the initial yield surface needs to be determined. For nonassociated plasticity, the subsequent yield surface should also be determined. However, it should be noted that there is no experimental evidence on the shape of subsequent yield surface for the material 'Laminac'. This experimental evidence should be the theoretical basis for choosing a yield potential. Let the yield potential be

\[ Q = \sqrt{J_2} \] (37)

The most general form of the plastic matrix \([D^P]\) is given by Zienkiewicz [12] as

\[
[D^P] = \frac{\partial Q}{\partial \sigma} \{\sigma\} \{\sigma\}^T \frac{[D]}{A + \{\partial F/\partial \sigma\}^T [D] \{\partial F/\partial \sigma\}} \] (38)

Assumptions and evaluations of \([D^P]\) are given in Appendix A.
IV. RESULTS

A. Numerical Results

1. Description of the finite element program

   a. Background and development of the program

The author had developed an axisymmetric plasticity finite element program from scratch as part of his master's thesis [10]. It used a triangular ring element, and the von Mises as well as Tresca yield criteria with their associated flow rules, to investigate the residual stresses in a yielded open-ended thick-walled cylinder after internal pressure was released (autofrettage). This program was adapted for the present research. Most of the features mentioned in Chapter III were first implemented and successfully tested using that program.

In the initial design stage of that program, the best data structure and programming practices were not followed closely. The increasing complexity of the multiple features required in the present research, together with their delicate interactions, soon rendered the maintenance and further modification of the original program difficult. The future need for three-dimensional analysis further spurred the author to seek a more versatile but satisfactory and economical program.

   Based on the author's extensive and sometimes painful
experiences, several criteria were set up for the candidate program:

- A source code must be available. This is based on the following reasons: (a) Any program, no matter how well written, needs continuous modification to suit individual needs. A program which comes only in the form of an object or executable code will never do. (b) Errors are always present in any fair-sized program. The validity of functions that the program is claimed to be capable of performing can only be checked when the source code is available.

- The program must be specifically written by a researcher in a specific field who is also well versed with computer science.

- The program must be nonproprietary. Most of the proprietary programs are expensive, and given without source code. Usually, the cost of modifications and improvement of the programs, when suggested or requested by the customer, are all charged to the customer!

- Good programming practices must be adhered to.

- Documentation must exist, be available, be well written and complete. Any program without adequate documentation is of little worth. Sparsely
scattered comments (or none at all) and cursory descriptions of inputs can render an excellent program useless.

- It must be easy to expand or modify the program. In general, the time spent in modifying and maintaining an existing program is an important measure of its longevity. This time is often greater than that required in the initial development.
- The program must be inexpensive to purchase or lease and run.
- The program must be capable of doing plasticity and/or 3-dimensional problems.

It was the goal of this search for a program which will satisfy most, if not all, of these criteria.

After searching through survey papers and computer information centers and other references, no program was found that met the above requirements\(^\text{13}\) with respect to 3-dimensional plasticity and the peculiar material properties encountered in the photoplasticity research for which this program had to be written. Finally, a program

\[\text{------------------}\]

\(^{13}\) The search was not exhaustive as the author was restricted by the available time. Being a collector of GOOD computer programs from operating systems to scientific utilities to games, the author welcomes and appreciates any contribution.
written by Taylor in [12] was chosen as a starting point. It satisfied approximately all the criteria except the last one. Yet, there are provisions even for that. The source code is available, the author is a specialist with good reputation, the program is free to be copied and costs nothing. In addition, the program also has the following specific attractions:

- It has a library of elements. Provisions for adding new elements into the libraries already exist. The use of numerically integrated elements are superior in programming compared to the ordinary triangular elements when extensions into different types of elements and dimensions in space are required.

- It makes very efficient use of memory. Memory locations are allocated only when needed by using ('simulated') dynamic memory management. Hence, the size of the problem is limited only by the computing environment (disk or core memory size). Also, there is no need to change the code to accommodate problems of different magnitudes.

- It has efficient equation solvers which use memory economically. Specifically, the method of skyline (profile) storage (always better than the conventional banded storage) was used. The program
also embraces both a symmetric and an unsymmetric equation solver.

- A macro language for solution procedures and for input and output makes it convenient to include combinations of features.
- Above all, the program was designed to advance further research so that the possibilities of future expansion and modification had been well thought out by the original author.

The program has the structure and potential of a general purpose program. The original program has been described in detail [12].

In the field of solid mechanics, the original program has the capability of elastic analysis using plane stress and plane strain elements. For the present research, the author of this dissertation added the additional capability of elastic-plastic and finite deformation analysis for plane stress, plane strain and axisymmetrical situations. An elastic-plastic three-dimensional (8-noded brick) element was also added to the library. Most of the routines were either extensively modified or expanded with some rewritten and many new ones added. The number of macros were more than doubled to meet increasing needs. A list of revised and new macros is given in Appendix B. Special routines to deal with oblique boundary restraint and continous boundary
condition change were also incorporated. Although not all of these features were used for the present dissertation, they were included to make further research in this area easier. A listing of the program is given in Appendix C.

b. **Postprocessing**

The output from a nonlinear finite element program is enormous mainly due to its incremental nature. Even though sometimes the main interest is in the final or residual state of deformation, the history of the deformation still needs to be monitored. The original program has provided means for selected output in stress, strain and displacements at desired time steps. These provisions greatly decrease the amount of output.

To exploit the capability of the finite element method for obtaining information over a whole field, it is desirable to have a contour plot of selected variables at different time steps over the entire or selected domain. Another need is to manipulate the output data of the finite element program to obtain more information, for example in producing simulated isochromatic fringe orders. The contouring ability is a necessity rather than a decoration.

While contouring over a domain where data are specified on a rectangular grid (uniformly or nonuniformly spaced) is fairly common and easy, it is not so easy when data are scattered across the two-dimensional domain. As far as the
author is aware, all the locally available routines belong to the former category and hence can not be used.

Contouring is closely related to interpolation, because the coordinates of a specific contour line need to be located. Interpolation in one-dimensional space is elementary, but when the domain is a two-dimensional space with randomly spaced data, the more complicated bivariate interpolation is desired. The most widely used method for interpolation over an irregular domain is the distance weighted least-square method [80]. It is based on the idea that near neighbors of a point to be interpolated have a higher influence (and hence weight) than distant ones. A direct bivariate interpolation over an irregular domain was formulated by Akima [81].

Both schemes map the irregular domain to a rectangular one by interpolation and extrapolation, and then contour plots are drawn over the mapped rectangular domain. When these schemes were tested for this research, they produced poor results. It seemed that the test cases used by [80, 81] are really 'random' data—e.g. data from surveys of land—where the accuracy may not be important, and the general pattern of the interpolated values may not follow a logical pattern. The data we have, however, do have a definite physical meaning and follow a logical pattern. The results from these formulations are poor because they often give
spurious numbers, for example a negative interpolated or extrapolated number from a positive data set! The distance weighted least-square method [80] has the disadvantage that, for every point interpolated, a complete least-square analysis which involves all the data points is needed. In the author's opinion, if the spirit of this algorithm is to be followed, then there should be some means to use only selected neighbors in the least square analysis. This also involves a lot of computer time. For these reasons, both of these schemes were then dropped. For further discussion of the comparison of these two methods see [82].

The author considered the data obtained from the finite element analysis such as stress, strain etc. as being 'correct' and 'exact' as far as the input to the contouring program is concerned. The problem of contouring over an irregular domain is finally dealt with by using a simple yet effective method developed by the author. In this procedure, the whole domain is first triangulated into a number of triangles. An inverse linear interpolation is then performed for each contour level over each triangle by following the simple formula given in [83]. This eliminates the complex interpolation used in Akima's work [84]. Coordinates interpolated in this way are 'correct' if the original data are 'good' and the distribution of their location in the domain is
'reasonable'. Contour lines for a particular contour value are connected triangle by triangle, and they are unique for a given data set.

Connecting contour lines is usually a difficult task. In the present method, they are connected automatically without further processing. This is most effective when plots are made on a graphic terminal since there is no need for 'raising', 'lowering' or 'moving' a physical pen. In general, a triangular cell as a data base is superior to a rectangular one, where a condition called 'degeneration' can occur, i.e., contour lines can be nonunique and sometimes will even cut across each other when they are connected. This anomaly is sometimes seen in 'packaged' contour programs. Such degradation can be eliminated by further triangulation of the rectangular cell as implemented by the author on local PDP-11 computers.

For this research, a finite element postprocessor was developed by the author on a local VAX-11/780 supermini computer. An Interactive Graphic Library (IGL), supplied by the computation center, provided basic routines such as drawing a line etc. for plotting on Tektronics plotting hardwares or their compatibles. The postprocessor is interactive, menu driven and of pipeline hierarchy (can go up and down the pipes-or stations as the popular UNIX operating system calls them) with simulated dynamic memory management.
Data such as principal stresses, strains, their directions, equivalent stress, displacement and simulated isochromatic fringes can be processed. An original and a deformed mesh can be plotted together or separately. Displacements can be amplified to emphasize deformation. This is very useful in small deformation analysis, but is also helpful for large deformation analysis since it can indicate the approximate trends of deformation. The size of a graph can be arbitrarily distorted and scaled to suit special needs. Specific contour values and dash/solid line patterns can also be chosen for the purpose of distinction of lines. Input data can be formatted or unformatted. Formatted data are mandatory when the input data set to the contour program comes from a different computer, because the internal representation of numbers is likely to be different. Unformatted data will speed up the processing if the input data set is from VAX itself. For a detailed description of the program and other options see the listings and comments in Appendix C.

Note that, purposely, no smoothing or extrapolation is done in the postprocessing. The author believes that smoothing should be done only in the final stage of analysis when the analyst is sure of the detailed pattern of results. Smoothing should not destroy any meaningful characteristic in the original. The idea is to give a 'raw' but correct
picture of what the data really represent by sacrificing somewhat the aesthetic part.

c. General description of cases and plots  This subsection describes the way cases and figures are organized for presentation. Table 1.1 gives a summary of the numerical cases studied. Table 1.2 gives the material properties of the photoplastic material, rigid 'Laminac' polyester, which is used in numerical modeling [61]. Table 2.1 gives the correspondence of figure number and numerical cases. The numbering of the figures are assigned as follows. The first group of figures are for loaded response (Fig. 16 to Fig. 30), where strain, fringe and equivalent stress are presented case by case. The second group of figures (Fig. 31 to Fig. 40) are for unloaded response, where strain and isochromatic fringe are organized case by case. The third group of figures (Fig. 41 to Fig. 45) are mesh plots for both loaded and unloaded response (superposed). Table 2.2 gives the correspondence of figure number and experimental results under different assumptions when analyzing the data. The isochromatic fringe data are obtained from Gomide [61].

Refer to Table 2.1. The way data are plotted when classified according to parameters is as follows:

• Mesh plots (Fig. 41 to Fig. 45): deformed (dashed lines) and undeformed (solid lines) meshes are
plotted together to show the effect of deformation. For undeformed mesh plots, the domain used is the shaded region in Fig. 14, i.e., the friction elements are not indicated. This is also the domain on which modeling is done. (Since the elements used are axisymmetric, the shaded region actually represents one-half of the physical ring.)

- Strain plots (Figs. 16, 19, 21, 24, 26, 29, 31, 33, 35, 37, 39): \( \varepsilon_1 \) and \( \varepsilon_2 \) are the principal strains in the R-Z plane. In the absence of shear strain, \( \varepsilon_1 \) corresponds to \( \varepsilon_r \) and \( \varepsilon_2 \) corresponds to \( \varepsilon_z \). Then \( \varepsilon_3 \) is \( \varepsilon_\theta \) by virtue of axisymmetry.

- Fringe plots (Figs. 17, 20, 22, 25, 27 and 30): \( N_\theta \) and \( N_\phi \) are simulated fringes as computed from distributions of principal strains. They are calculated according to eq. (39).

- Equivalent stress plots (Figs. 18, 23, 28): For these plots, minimum value = \( \sigma_0 \), the initial yield stress. In this way, the elastic zone, if any, can be detected.

The domain of the strain and fringe plots is the one formed by the Gauss (sampling) points and not the one formed by the nodes. This domain (region formed by the dashed lines in Fig. 15.1(a)) is always smaller than the one of the mesh plots (shaded region in Fig. 14) because the Gauss
points are always inside the element (Fig. 15.1(a)).

For small deformation analysis (cases 1.a, 1.b, 2, 3), the undeformed domain is used for plotting. The dimension of inner radius $R_i$ is 30 units, outer radius $R_o$ is 60 units and the height $h$ is 40 units. The plotting domain, as formed by the Gauss points, is located at some distances $a$, $b$ and $c$ from the actual domain. For the four-noded linear element, $a = 1.15$ units, $b = 1.44$ units and $c = 0.288$ units. For the eight-noded quadratic element, $a = 0.72$ units, $b = 1.44$ units and $c = 0.36$ units.

For large deformation analysis (cases 4 and 5), since the coordinates of the nodes (mesh) are updated after each increment, the deformed domain is used for plotting. The $a$, $b$ and $c$ values will be slightly different from those of the undeformed region. These values will change from increment to increment. However, because the change is not significant enough to influence the interpretation of results, these values will not be specified.

Refer to Table 2.2. The domain of plotting for experimental data is shown in Fig. 15.2. The data are collected over part of the deformed domain. This domain (region bounded by dashed line in Fig. 15.2), being rectangular, is smaller than the actual deformed one. The actual deformed domain is indicated partly by the gap e in Fig. 15.2. The domain is characterized by the following
dimensions [61]:

1. For A-8%, \(a = 5.2\) units, \(b = 1\) unit, \(c = 3.8\) units, \(d = 1.4\) units and \(e = 1\) unit.
2. For A-16%, \(a = 4.7\) units, \(b = 4.7\) units, \(d = 3.25\) units and \(e = 2.2\) units.

For loaded response, since no comparison with experiment was available, only one half of the ring was used in plotting.

For the unloaded numerical solution, two operations were performed on the data when plotted.

1. A mirror reflection of the plots (which in their original form showing only one half of the ring) is made artificially about the equator \((Z = 0)\) to occupy the full domain of the ring.
2. The size of the figures are adjusted artificially to be the same size as the experimental ones.

The first operation is desirable since the full size of the ring was used in the collection of experimental data, and the data collected were not always symmetrical about the equator. The second operation obviously distorted somewhat the shape of the original plots as produced by the numerical solution. The distortion, however, is not significant. By recognizing the fact that the physical domains over which the numerical and experimental data are available are not the same, the above operations attempt to make the
comparison between FEM and the results of the model from the photoplastic experiments easier. Since the friction elements are not part of the ring, they are not included in the plots.

The contour plots of all figures are at 10 levels. Each contour interval is \( = \frac{(\text{maximum value} - \text{minimum value})}{10} \). Ideally speaking, there should be 11 contour lines. However, since the minimum and maximum values are usually only points, most figures will not show all 11 lines. Refer to Table 3.1. This table is constructed for numerical results when loaded. '\% of deformation' column gives the percent of reduction in height. Minimum and maximum values correspond to contour lines number 1 and number 11 respectively. 'Increment' corresponds to the value of the interval between two consecutive contour lines.

The strains of Table 3.1 are actually principal strains \( \varepsilon_1 \), \( \varepsilon_2 \), \( \varepsilon_3 \) as computed by the finite element program. The fringe orders are calculated by the finite element postprocessor according to

\[
N_\theta = \frac{(\varepsilon_r - \varepsilon_z)t}{f} \quad (39a)
\]
\[
N_z = \frac{(\varepsilon_r - \varepsilon_\theta)t}{f} \quad (39b)
\]

where \( N \) = isochromatic fringe order.

\( t \) = thickness of a given slice.

\( f \) = unloaded material strain-optical constant.
The value $f$ is obtained from calibration [61]. The actual values used are as follows: $t = 0.04$ inches, $f = 0.003$ fringe/inch. According to the convention of photomechanics, fringe order is assigned a positive value. However, when the above equations are used, by virtue of the relative magnitude of the strains, fringe order can become negative. This effect is of course quite artificial. A negative fringe order, when encountered, can simply be interpreted as a positive one. That is, instead of decreasing from zero to a negative number, it increases from zero to a positive number. Since the region of negative fringe order is usually small (Figs. 22 & 30), the ordinary convention is not followed for these two figures.

Table 3.2 is similar to Table 3.1. It gives the information on the range, increment etc. but for unloaded data. For evaluating strains from experimental data, different assumptions are made. These will be discussed in section B of this chapter. The contour plots in these figures were made with a locally available computer program AGRAPH.

The contour plots of all numerical unloaded responses uses the range (maximum value - minimum value) of the corresponding experimental results. For example, if a ring was compressed to 8% in the experiment and unloaded, the numerical analysis of cases of 8% compression and unloaded
will use the same range as that used in the experiment. These ranges are marked with an asterisk * in Table 3.2. In this way, comparisons can be made between FEM & experiment, not only qualitatively, but also quantitatively because the lines of the corresponding plots indicate the same values. For the same reason, the sizes of the numerical plots are adjusted artificially to the same size as the experimental plots (as mentioned before). For numerical data, the contour plots are all made by using the finite element postprocessor developed for this purpose.

2. Description of cases

Refer to Table 1.1 and Table 1.2. In the following description, case 1 will be explained in detail. For other cases, only the important differences in modeling parameters from case 1 will be elaborated. A summary of modeling information is given in Table 1.1. Complete modeling information of all cases can be found in section B of Appendix B.

a. Case 1: simple plastic analysis Case 1 uses 42 4-noded isoparametric linear elements, of which six are elastic friction elements, the remaining 36 elastic-plastic elements form the ring. Two Gauss points (Fig. 15.1(a)) are used in each direction (r and z) for the stiffness formulation as well as the evaluation of stress and strain. This makes 144 effective sampling points (4 *
The geometry is that of a 'standard' ring with the ratios of outside diameter:inside diameter:height equal to 6:3:2. Taking advantage of symmetry, only one half of the ring is modeled by providing rollers on the line of symmetry \((Z = 0)\) at the equator. The geometry with boundary conditions is shown in Fig. 15.1(b). Von Mises yield criterion and its associated flow rule (Prandtl-Reuss) is used. The material is assumed to be isotropic, homogeneous with Young's modulus \(E = 120,000 \text{ psi}\), initial yield strength \(\sigma_0 = 4100 \text{ psi}\) and the strain-hardening parameter \(H' = 8400 \text{ psi}\). The ratio of \(H'\) to \(E\) is equal to 0.07. These values are taken from Gomide [61]. The top friction layer assumes a stiffness 1000 times that of the ring for sticking friction modeling. Also, the horizontal degrees of freedom of the friction layer are eliminated. A fairly large increment of displacement is first applied at the top nodes of the billet until incipient plastic deformation is about to occur. 42 equal increments of displacement are then applied to reach the maximum deformation of 8%. The initial stress method is used and the tolerance of convergence is set to \(1.0 \times 10^{-9}\). The maximum number of iterations within an increment is 30. Reactions are monitored just before unloading to make sure that static equilibrium has indeed been satisfied. Unloading is then imposed by changing the input displacement boundary conditions to force input and
residual stress/strain patterns are then obtained. (Refer to section H. of Chapter III.) Small deformation is assumed. The input data was submitted from a VAX computer and executed on a local Itel mainframe computer. The output from the finite element program was again routed back through a local computer network from Itel to VAX for postprocessing. The input data with necessary JCL (Job Control Language) are given in section B of Appendix B. A schematic showing the computing environment and data flow is given in section D of Appendix B.

The maximum deformation under load is 8% reduction in height, i.e., in overall axial length, for case (1.a) and 16% for case (1.b). The principal strain distributions just before unloading are given in Figs. 16 & 24. The condition after unloading is given in Figs. 31 & 33. Fringe patterns just before unloading are given in Figs. 17 & 25 and after unloading in Figs. 32 & 34. The first figure of the above figure pair is for 8% -case (1.a), and the last is for 16% - case (1.b). Original and deformed meshes just before and after unloading are given in Figs. 41 & 44. Equivalent stress distributions are given in Fig. 18 at successive stages of deformation of 4.6%, 6.25%, 8% and 16% respectively.

b. Case 2: quadratic element Case 2 uses 54 8-noded quadratic isoparametric elements with 193
nodes. 6 elements are elastic friction elements. The remaining 48 elastic-plastic elements form the ring. The geometry with boundary conditions is given in Fig. 15.1(b). Maximum deformation is 8%. The rest of the parameters are similar to case 1.

Principal strain distributions before and after unloading are given in Figs. 21 & 37, whereas fringe patterns are given in Figs. 22 & 38. Original and deformed mesh before and after unloading are given in Fig. 43. Equivalent stress distributions are given in Fig. 23 at successive stages of deformation of 5% and 8%.

c. Case 3: nonassociated plasticity  Case 3 assumes that the material follows the modified von Mises yield criterion with the condition of nonassociated plasticity. The ratio of the absolute values of compressive and tensile yield stress $C/T$ is 2.1 [61]. The maximum deformation is 8%. The rest of the input is similar to case 1.

Principal strain distributions before and after unloading are given in Figs. 19 & 35. Fringe patterns are given in Figs. 22 and 38. Original and deformed mesh before and after unloading are given in Fig. 42.

d. Case 4: finite deformation  Case 4 assumes large deformation by using the Updated Lagrangian method. The maximum deformation is 16%. The rest of the input is similar to case 1.
Principal strain distribution before and after unloading are given in Figs. 26 & 39, whereas fringe patterns are given in Figs. 27 & 40. Original and deformed mesh before and after unloading are given in Fig. 45. Equivalent stress distributions are given in Fig. 28 at successive stages of deformation of 8.7% and 16%.

e. Case 5: quadratic and large deformation

Case 5 uses the same mesh as case 2 (quadratic element) with the assumption of von Mises yield criterion and large deformation. It had a convergence problem and terminated prematurely at 12.1% deformation. However, part of the results were salvaged at 8.7% deformation. A discussion of this will be given in Chapter V.

Principal strains and fringes of the salvaged results are given in Figs. 29 & 30.

B. Experimental Results

The experiment referred to in this dissertation was the work done by Gomide [61]. Partial results had been previously published [85]. Rings made of Laminac polyester resin of different geometry and with different friction at the interface were compressed axially beyond the elastic limit by an MTS machine. After removal from the machine, the specimens were sliced to analyze the three-dimensional residual strain patterns by the method of photoplasticity.
A schematic showing the slicing plans is given in section C of Appendix B. Although different friction factors were used in the experiment, only sticking friction at 8% and 16% yielded satisfactory results in that the results are more symmetrical with respect to the equator \((Z = 0)\) than the rest.

The basic information obtained from photoplastic analysis consists of the isochromatic fringes viewed in the \(\theta\) and \(z\) (axial) direction, i.e., \(N_\theta\) and \(N_z\). From these two pieces of experimental data, principal strains \(\epsilon_1, \epsilon_2, \epsilon_3\) can be derived from different assumptions. These assumptions will be described later in this section (eq (41) to (45)) as well as in section A.2 of Chapter V (for the motivation behind the assumptions). The analysis of data in the present dissertation is not quite the same as that in Gomide [61]. For discussions of differences from the previous work see Chapter V.

The two equations (39a) and (39b) contain three unknowns, \(\epsilon_r\), \(\epsilon_\theta\) and \(\epsilon_z\). A third equation is then needed. This comes from the observation that the Poisson's ratio of the material \(v = 0.485\) which is close to the condition of incompressibility: Poisson's ratio = 0.5.

\[
\epsilon_r + \epsilon_\theta + \epsilon_z + \epsilon_r \epsilon_\theta + \epsilon_\theta \epsilon_z + \epsilon_z \epsilon_r + \epsilon_r \epsilon_\theta \epsilon_z = 0 \tag{40}
\]

Two types of assumptions can be made regarding (1) \(\epsilon_r > \epsilon_\theta\) or \(\epsilon_r < \epsilon_\theta\) in eq (39b). (2) whether second order terms in eq
(40) should be retained. (Third order term is omitted.) When the second order terms are retained, the condition is called 'finite (large) deformation'. When they are ignored, the analysis is referred to as 'small defomation'. It will be shown later in Chapter V that the conditions $e_r > e_\theta$ and $e_r < e_\theta$ can coexist for the cases studied in this dissertation. Physically, the fringe pattern $N_\theta$ is a picture of one vertical slice, taken in the $\theta$ direction. The fringe $N_z$ needs several pictures taken in the axial direction. Several horizontal slices are needed through the thickness of the ring. Hence, the plot of $N_z$ is really a composite of pictures rather than one single picture. The plot of $e_\theta$ is also a composite figure of the distribution of $e_\theta$ in the radial direction for different heights ($z$).

The fringe distribution of $N_\theta$ and $N_z$ are given in Figs. 45, 47 for 8% deformation (designated as A-8%) and in Fig. 50 for 16% deformation (designated as A-16%).

The equations used in strain separation under different assumptions are as follows:

1. Small deformation

(i) $e_r > e_\theta$

\[
\begin{align*}
\varepsilon_r & = (K_\theta N_{\theta} + K_z N_z)/3.0 \\
\varepsilon_z & = (K_z N_{z} - 2K_\theta N_{\theta})/3.0 \\
\varepsilon_\theta & = (K_\theta N_{\theta} - 2K_z N_z)/3.0
\end{align*}
\]
where

\[
K_\theta = \frac{f_\varepsilon}{t_\theta} \\
K_z = \frac{f_\varepsilon}{t_z}
\]

\(\varepsilon_r\), \(\varepsilon_z\) and \(\varepsilon_\theta\) are shown in Fig. 52 for 16% deformation, and in Fig. 49 for 8% deformation.

(ii) \(\varepsilon_r < \varepsilon_\theta\)

\[
\varepsilon_r = \frac{(K_\theta N_\theta - K_z N_z)}{3.0} \\
\varepsilon_z = \frac{-2K_\theta N_\theta + K_z N_z}{3.0} \\
\varepsilon_\theta = \frac{(K_\theta N_\theta + 2K_z N_z)}{3.0}
\]

when partly corrected (refer to Chapter V section A.2.a.1) for \(\varepsilon_r < \varepsilon_\theta\), \(\varepsilon_r\), \(\varepsilon_z\) and \(\varepsilon_\theta\) are shown in Fig. 53 for 16% deformation.

The method of selection of (i) or (ii) above will be described in section A.2 of Chapter V.

2. Large deformation

(i) \(\varepsilon_r > \varepsilon_\theta\)

\[
\varepsilon_r = \frac{(-B + \sqrt{B^2 - 12C})}{6} \\
\varepsilon_z = \varepsilon_r - K_\theta N_\theta \\
\varepsilon_\theta = \varepsilon_r - K_z N_z
\]

where

\[
B = 3 - 2K_z N_z - 2K_\theta N_\theta \\
C = K_z K_\theta N_\theta N_z - K_z N_z - K_\theta N_\theta
\]

\(\varepsilon_r\), \(\varepsilon_z\), \(\varepsilon_\theta\) are given in Fig. 54 for 16% deformation, and in
Fig. 48 for 8% deformation.

(ii) $\varepsilon_r < \varepsilon_\theta$

$$\varepsilon_r = \frac{-B + \sqrt{B^2 - 12C}}{6}$$

$$\varepsilon_z = \varepsilon_r - K_\theta N_\theta$$

$$\varepsilon_\theta = \varepsilon_r + K_z N_z \quad (45)$$

where

$$B = 3 + 2(K_z N_z - K_\theta N_\theta)$$

$$C = K_z N_z - K_\theta N_\theta - K_\theta N_\theta K_z N_z$$

$\varepsilon_r', \varepsilon_z', \varepsilon_\theta'$ when corrected are given in Fig. 51 for 16% deformation.
V. DISCUSSIONS

A. Discussion of Results

1. Propagation of yield zone

One of the difficult tasks in any analytic solution of elastic-plastic problems is the determination of plastic zones (enclaves) when the geometry or boundary condition is not simple. "It is...usually of such an awkward shape that even the stress distribution in the elastic region can only be obtained by laborious numerical methods" (p. 70, [5]). An example of simple elastic-plastic boundary is that of a circular cylinder subjected to internal pressure where the plastic zone propagates uniformly across the radius.

The plastic zone in a ring under axial compression with sticking friction between platen and ring is of an awkward shape but poses no threat at all in analysis. Yielding first occurs near the outside corner and then occurs near the equator. The yield zone then propagates upward toward the platen-ring interface. It also propagates toward the outside equator. This can be seen from Figs. 18, 23 and 28, where the distribution of equivalent stress, as expressed by the contour lines, follows approximately the same pattern. The pattern of propagation is in agreement with that reported by Nagamatsu et al. [36]. The region between contour level 1 (which is equivalent to the initial yield strength
of 4100 psi) and the top boundary of the graph in Fig. 18 (for 4.6%, 6.25% and 8% deformation) & Fig. 23, and the circular region near the top inner corner in Fig. 28 (for 8.7% deformation) are elastic zones. In other words, an elastic zone (dead zone) exists near the ring-platten interface for 8% and 8.7% deformation. The ring is already totally plastic at 16% deformation as shown in the last graph of Figs. 18 and 28.

Local irregularity of the equivalent stress distribution can be seen in the early stages of deformation (for example, the regions indicated by the arrows in Fig. 18 at 4.6%). It smooths out somewhat as deformation proceeds. The part of the domain which becomes plastic can be considered as a 'different' material which among other things has a lower modulus of 'elasticity'. Consequently, the constraint exerted on a particle by its neighbors may no longer be the same after plastic deformation. Hence, local irregularity is commonly seen. However, the material may become more or less 'homogeneous' again when all particles are plastic. (See also section I of Chapter III.)

The figures presented do not follow the entire path of deformation from initial yielding to final deformation. For example, for the four consecutive graphs in Fig. 18, we can see that the plastic zone extends from the center of the graphs upward toward the ring-platen interface as
deformation proceeds from 4.6% to 16%. The previous history of yield (before 4.6% deformation) is not presented because outputs are obtained only at selected time steps in order to minimize the amount of output. The previous history of the propagation of yield zone, though not presented, was observed during the execution. This was done by temporarily suspending the execution of the program at selected time steps (interactively!!). The current values of equivalent stress etc. were examined from memory to determine the current region of yield zone. An on-line debugger available at the computer center was used to accomplish this. These observations were made when running both the recently developed finite element program (see Appendix C) and the modified version of the program [10].

The yield zone usually exhibited a saw-toothed shape when triangular elements were used in the finite element analysis. With isoparametric elements, as used for the present research, partial yielding can occur within an element. As a consequence, the shape is usually smoother. A smoother shape of the yield zone can be obtained when using triangular elements by using the similar contouring techniques discussed in Chapter IV. Note also that the quadratic elements give a smoother shape than the rest of the plots by virtue of the facts that (a) more data points are used, and (b) higher order terms are included in the
shape function. No plots of equivalent stress for case 3 (nonassociated plasticity) are presented because each point now has a different yield stress, which makes a graphical presentation more difficult.

2. Strains and fringes
   
a. Methods of data analysis of experiment
   
The strain distribution in the ring were obtained by using different assumptions as stated in section B of Chapter IV. This subsection will discuss why the assumptions are used and in what way they are different from the previous works [61, 85].

1) Small deformation analysis
   
   There are two possibilities concerning the relative magnitude of \( \varepsilon_r \) and \( \varepsilon_\theta \). They are: \( \varepsilon_r > \varepsilon_\theta \) or \( \varepsilon_r < \varepsilon_\theta \). The first possibility (\( \varepsilon_r > \varepsilon_\theta \)) was chosen by Gomide [61] as valid over the whole domain. He first separated the strains by using both assumptions. The radial displacement was obtained by integrating \( \varepsilon_r \). This calculated displacement was then compared with the actual measured one. He found that \( \varepsilon_r > \varepsilon_\theta \) gives a better comparison, hence this assumption was adopted for his analysis for all specimens. However, this method was not rigorous enough for at least the following reasons:

   - The integrated displacement which serves as the 'equilibrium' check for strain analysis gives only
a gross view of deformation. It gives no
indication of how the internal strain should be
distributed. That is, it is possible for different
internal distributions of strain to give the same
amount of overall displacement. The uniqueness of
strain is not guaranteed by this type of analysis.

- The strains in eq (39a) are principal ones and are
  not strain components in the coordinate system.
  Hence, the integration of 'radial strain' is
  actually performed on $\varepsilon_1$. This is at best an
  approximation.

- Compatibility should be checked if possible.
  Particularly, since we can find radial displacement
  (when the angle of isoclinics is small) across the
  radius from $\varepsilon_\theta$, then we can double check whether
  the gradient $du/dr$ as found from $\varepsilon_r$ is consistent
  with the displacement distribution.

A detailed examination of the finite element results
(for example, the plot of $N_x$ in Fig. 22) shows that for part
of the region near the outer radius, $\varepsilon_r < \varepsilon_\theta$. A check on
the photoplastic results also shows this possibility.
Compatibility is not satisfied for part of the region. That
is, across the radius of the ring, a positive radial strain
was sometimes found with decreasing displacements! This
phenomenon was corrected by assuming $\varepsilon_r < \varepsilon_\theta$. The results
are given in Figs. 51 & 53. Since isoclinics are not recorded in the original experiment, corrections are made only near the equator where the strains are almost principal ones. That is, $\varepsilon_r$ is assumed to be the same as $\varepsilon_1$ etc.

For the results presented in this dissertation, it seems that only cases 2 & 5 (both used quadratic elements) exhibit this phenomenon. This is definitely not accidental. First, this phenomenon was observed in many other parametric runs of the same problem when different mesh and elements were used (ref. subsection 2 of General Discussion in this chapter). Secondly, the mesh using quadratic elements is a finer mesh than the mesh using the linear elements. There are more nodes at the outer side of the ring, which can be used to calculate strains etc. This region is exactly where $\varepsilon_r < \varepsilon_0$ occurs. This explains why by using the mesh formed by quadratic elements, which is of higher order, and of a finer mesh, the phenomenon is 'caught'. Although this is not a very large region, it affects the contour plots significantly ($\varepsilon_0$ plot of Fig. 52 versus Fig. 53). Hence, the $\varepsilon_0$ plot of Fig. 7 in the work of Gomide and Burger [85] is not correct. An interesting point is that the patterns of $\varepsilon_r$ and $\varepsilon_z$ are not significantly affected.

The separation of the strains was performed by using the computer program whose listing is given in Appendix C. The procedures for the correction are as follows.
1. Calculate strains $\epsilon_r$, $\epsilon_z$, $\epsilon_\theta$ by eqs (41) and (43) respectively.
2. Calculate radial displacement $u$ from $\epsilon_\theta$.
3. Calculate displacement gradient $du/dr$ from $\epsilon_r$.
4. Pick one set of strains from procedure 1 above which shows a consistent pattern of $u$ and $du/dr$ over the radius near the equator to be the correct ones. This part is done manually.

2) Large deformation analysis

One advantage of the method of photoplasticity is that the strain-optic law (eq (39)), once verified by calibration is valid for the amount of deformation actually observed. That is, the strain-optic law itself does not assume whether the deformation is small or not. However, when strains are related to displacement, a distinction between small and large deformation needs to be made. In the original analysis [61] by Gomide, second order terms in the eq (40) are used because under this assumption, the integrated displacement as calculated from strains agrees better with measured values.

Gomide used a Lagrangian analysis throughout; however, the author of this dissertation believes that an Eulerian analysis is more adequate. Lagrangian analysis requires the knowledge of the undeformed geometry. For a point of interest in the current configuration, we can obtain the
fringe order. But in order to evaluate the displacement according to the Lagrangian analysis, we need to know the coordinates of the point in the undeformed configuration. In other words, we need to know where this point came from. For an Eulerian analysis, all we have to know is the current configuration, since displacements will be evaluated in the current configuration. This configuration is always available in photoplasticity. If a grid is inscribed on the specimen, then both configurations are available, and Lagrangian as well as Eulerian analysis can be performed. For three-dimensional photoplastic analysis, imbedding grids in the model is clearly not a feasible task. Consequently, only Eulerian analysis should be employed.

For axisymmetric Eulerian analysis, the physical components of strain are:

\[
\varepsilon_r = \frac{\partial u}{\partial r} - \zeta \left( \left( \frac{\partial u}{\partial r} \right)^2 + \left( \frac{\partial v}{\partial z} \right)^2 \right) = (46a)
\]

\[
\varepsilon_\theta = \frac{u}{r} - \zeta \left( \frac{u}{r} \right)^2 = (46b)
\]

\[
\varepsilon_z = \frac{\partial v}{\partial z} - \zeta \left( \left( \frac{\partial v}{\partial z} \right)^2 + \left( \frac{\partial u}{\partial z} \right)^2 \right) = (46c)
\]

\[
\varepsilon_{rz} = \zeta \left( \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} - \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} \right) = (46d)
\]

Note that in the evaluation, Christoffel symbols have been used. In the above equations, coordinates \( r \) and \( z \) are those of the current configuration. Lagrangian strains on the
other hand can be obtained from the above equations by replacing all minus signs by plus signs and then coordinates will refer to the original undeformed configuration.

The displacement gradient can be solved from eq (46a) when $\varepsilon_r$ is assumed to be a principal strain, i.e., $\frac{\partial \varepsilon}{\partial \varepsilon} = 0$

$$\frac{\partial u}{\partial \varepsilon} = 1 - \sqrt{1 - 2\varepsilon_r}$$  \hspace{1cm} (47a)

And radial displacement $u$ is calculated from eq (46b).

$$u = r(1 - \sqrt{1 - 2\varepsilon_\theta})$$  \hspace{1cm} (47b)

Similar to small deformation analysis, corrections are made whenever possible according to consistency in the distribution of $u$ and $du/dr$ by using the Eulerian analysis. The corrected plots for large deformation are given in Figs. 51 & 54. Separation of the strains is performed using eqs (44) & (45).

The correction for $\varepsilon < \varepsilon_\theta$ is not made for 8% deformation since Gomide's data is not symmetrical with respect to the line $Z = 0$. Hence, it is more difficult to locate the equatorial plane where correction is done. For 16% deformation (A-16%), all comparisons and discussions will refer to the corrected plots for large deformation only (Figs. 51 & 54).

Rigorously speaking, the way calibration is done should be consistent with the assumption of large or small
deformation. For an Eulerian analysis, calibration should be performed over the current configuration. For Gomide's analysis [61], calibration is performed over the undeformed region. For the Eulerian analysis performed by the author, the strain-optic coefficient \( f_\varepsilon^{\text{Eulerian}} \) should be different from that of \( f_\varepsilon^{\text{Lagrangian}} \). However, \( f_\varepsilon^{\text{Lagrangian}} \) calibrated by Gomide is used in the current analysis for the sake of simplicity.

b. Comparisons of results

Comparisons of distributions of strains and isochromatic fringe orders are given in Tables 4 and 5. Several types of comparisons can be seen from these tables.

1. Unloaded numerical results for each case are compared with experiments. (The experimental results are all unloaded.)

2. Loaded numerical results for each case are compared with unloaded numerical results for each case respectively.

3. Numerical cases are compared with each other.

The first type of comparison can give us a feeling for the adequacy of both numerical and experimental modeling. The second type of comparison can give us a feeling whether unloading will make significant changes in the pattern of results. The third type of comparison can give us a feeling for the influence of different modeling techniques on the
results.

Because comparison possibilities are numerous, and contour plots are used, several key conditions (as shown in Tables 4 & 5) are specified as follows:

1. Is the general pattern of the contour plots the same? These patterns are as follows for different parameters. For $\varepsilon_z$, $\varepsilon_z^*$, $N_\theta$, and $N_\theta^*$, the magnitude increases from the ring-platen interface toward the equator and the outer corner. It decreases toward the outer equator. For $\varepsilon_0$, it increases monotonically from the inner radius toward the outer radius.

2. What is the specific distribution of the magnitude of various parameters at the equator from inner radius toward outer radius? Two conditions are applicable. The magnitude either decreases all the way, or first increases and then decreases.

3. At what point are the various parameters concentrated along the radius at the equator?

4. Is there concentration of various parameters near outside corner?

5. Where is the maximum value of a particular parameter?

6. What is the relative magnitude of a particular
parameter when compared with some predefined basis? For the loaded numerical solution, the simple plastic case is used as the basis for comparison. For the unloaded solution, the experimental case is used as the basis for comparison.

Overall speaking, the results for all types of comparison agreed well, especially for $\varepsilon_z, \varepsilon_\theta, N_\theta$. So only peculiarities will be discussed here.

1. Refer to condition 2 in Table 4(a) and 5(a). In experimental results, $\varepsilon_r$ along the equator first increases and then decreases (Figs. 48, 49 and 51). This pattern is consistent for 8% and 16% results. However, when using simple plastic analysis (cases 1.a and 1.b) which uses the linear elements, the numerical solution (Figs. 31 and 33) gives an $\varepsilon_r$ which decreases all the way from inner radius to outer radius. The response agrees more when the model uses the quadratic elements (case 2, Fig. 37) or large deformations (case 4, Fig. 39) or nonassociated plasticity (Case 3, Fig. 35). For the nonassociated case (Case 3, Fig. 35), better match is reached only when unloaded. The degree of increasing and decreasing pattern of $\varepsilon_r$ is more prominent for nonassociated plasticity (case 3) than the other two models (cases 2 and 4) when unloaded.
These two models (cases 2 and 4), however, do exhibit the increasing-decreasing pattern when loaded (Figs. 21 and 26) and unloaded (Figs. 37 and 39).

2. Refer to the distribution of $\varepsilon_\theta$ in Tables 4(c) and 5(c). Numerical results of $\varepsilon_\theta$ show clearly that a neutral radius is present in all cases (for example, Fig. 16) and its position is almost the same for different models. This indicates a correct friction modeling as far as the presence of inward and outward flow is concerned. The unloaded neutral positions are shifted slightly from the loaded ones (for example, Figs. 16 and 31). Numerical results, of the unloaded response, however, consistently predict a higher magnitude of $\varepsilon_\theta$ near the equator than the experimental results (for example, compare Fig. 51 with Fig. 33). The best match occurs for a large deformation assumption (case 4, Figs. 51 and 39).

3. Refer to a, b, d, e of Tables 4 and 5 concerning the concentration of $\varepsilon_\tau$, $\varepsilon_\sigma$, $N_\theta$, $N_z$. Quadratic elements (case 2) and large deformation analysis (case 4) consistently give a maximum concentration at the outer corner (for example, see Figs. 22 and 38 for quadratic elements, Figs. 27 and 40 for large deformation analysis). The simple plastic case (case 1) gives a maximum concentration also at
middle equator (for example, see Figs. 17 and 32). For experimental results, maximum concentration is always at middle equator, with almost no concentration at outer corner for 8% analysis (for example, see Fig. 46). This agrees consistently with nonassociated analysis (case 3). For case 3, under loaded situation, there is less concentration at the outer corner than in other numerical cases (compare Fig. 20 with Fig. 17). For unloaded situation, the concentration is almost gone for 8% deformation (Figs. 35 and 36).

4. Refer to Tables 4(e) and 5(e) for $N_z$. In the experimental results (Figs. 47 and 50), the magnitude of $N_z$ consistently increases first and then decreases along the radius near the equator, while all numerical cases show a consistently decreasing pattern (for example, see Figs. 18 and 34).

c. Discussion of comparisons Refer to point 1 in the above subsection concerning the peculiar distribution of $\varepsilon_r$ along the equator. A linear element is characterized by using a displacement function

$$\phi = a_1 + a_2 r + a_3 z + a_4 rz$$

where $a_1$, $a_2$, $a_3$, $a_4$ are constants. so

$$\varepsilon_r = a_2 + a_4 z$$

which shows that $\varepsilon_r$ varies only with $z$ but not with $r$ within
an element. In another words, if the orientation of an element is such that the Gauss points, say points a' and b' in Fig. 15.1(a), are of the same height \( z \), then \( \varepsilon_r \) for points a' and b' will be the same. In this way, effective sampling points are reduced by half. This is exactly the mesh used in the analysis. If the mesh remains the same throughout the analysis, then this trend is true throughout the analysis. This means that details of the strain distribution can be hidden within an element. (That is, \( \varepsilon_r \) is constant for a constant \( z \).) For a quadratic element, however, \( \varepsilon_r \) can vary linearly within an element by the same argument as above. Hence, a quadratic element can give more detailed information. Also, if the mesh is updated constantly during the analysis, then \( \varepsilon_r \) will not be constant for a constant height \( z \). This explains why when using quadratic elements (case 2) and large deformation analysis (case 4) a better response is obtained that agrees more with experimental results. The region and degree of such an agreement for numerical results is smaller than the experimental results. It needs to be kept in mind that for experiments there are only four effective rows of data for one half of the ring, while for numerical solution, 14 rows of data are available and hence can give more detailed information. As a consequence, we do not expect the numerical and experimental methods to give exactly the same result.
It is noticed that case 2 (quadratic element) gives a higher concentration of parameters near the outside corner than other cases. This is because a quadratic element, being of a higher order, can distort more than a linear element. It will then be more susceptible to plastic deformation. Since a yield criterion can also be considered as a criterion for distortional energy, that is, the higher the distortional energy, the easier for yielding to occur. The outer corner is the first place for yielding to begin. Hence for quadratic elements, it will give a higher concentration at the corner.

Refer to point 3 of the above subsection concerning the lesser concentration at the outer corner for nonassociated analysis (case 3). According to the modified von Mises criterion, the higher the hydrostatic pressure, the more difficult it is for a point to yield (ref. section I of Chapter III). The highest hydrostatic pressure obviously occurs at the tool/ring interface. For the left to middle part of this interface, the distortion is already small. This can be seen from the plots and discussions of equivalent stress (ref. section 1 of this chapter). Hence, this region is not much affected by hydrostatic pressure. For the outer corner, where distortion is large, the effect of hydrostatic pressure is dramatic. This explains point 3 above.

The explanation for point 1 of the previous subsection
for modified von Mises yield criterion, however, is not so obvious. The unloaded response shows the accumulated effect of plastic deformation, which is more complicated for the nonassociated case than the other cases (see section I of Chapter III).

Refer to point 4 of the previous section concerning the distribution of $N_z$. Physically, $N_z$ gives the maximum shear strain in the $r-\theta$ plane (note that $\varepsilon_{r\theta}$ is zero by axisymmetry) by equation (39b). In this equation $\varepsilon_r$ and $\varepsilon_\theta$ are the in-plane principal strains (secondary principal strains). It can be shown by simple algebraic manipulation below that for a given pattern of $\varepsilon_r$ and $\varepsilon_\theta$ across the radius, different patterns of $N_z$ can occur depending on the relative magnitude of $\varepsilon_r$ and $\varepsilon_\theta$.

For two consecutive points along the radius where $r$ of point 1 is smaller than $r$ of point 2, or $(r)_1 < (r)_2$ where subscript denotes the point:

$$N_z = C(\varepsilon_r - \varepsilon_\theta)$$

where $C$ is a constant. For the sake of discussion, let $C = 1$. For all cases, $(\varepsilon_\theta)_2 > (\varepsilon_\theta)_1$. Then,

(i). If $(\varepsilon_r)_1 > (\varepsilon_r)_2$

$$N_z)_1 = (\varepsilon_r)_1 - (\varepsilon_\theta)_1$$

$$N_z)_2 = (\varepsilon_r)_2 - (\varepsilon_\theta)_2$$
\[ \Delta N_z = (N_z')_1 - (N_z')_2 \\
= ((\varepsilon_r')_1 - (\varepsilon_r')_2) - ((\varepsilon_\theta')_1 - (\varepsilon_\theta')_2) \\
= \Delta \varepsilon_r - \Delta \varepsilon_\theta = \Delta \varepsilon_r + |\Delta \varepsilon_\theta| > 0 \quad (50) \]

(ii) If \((\varepsilon_r')_1 < (\varepsilon_r')_2\), then by the same manipulation as (i) above:

\[ \Delta N_z = \Delta \varepsilon_r + |\Delta \varepsilon_\theta| < 0 \quad (51) \]

That is, if \(\varepsilon_r\) has the possibility of increasing across the radius, the pattern of \(N_z\) can decrease or increase depending on whether \(|\Delta \varepsilon_r|\) is smaller or greater than \(|\Delta \varepsilon_\theta|\).

A decreasing pattern of \(N_z\) near the equator with an increasing-decreasing pattern of strain is also observed in another parametric study by the author using a different material - CAB (cellulosic acetate butyrate). This is the same as the experimental results obtained by Gomide [61]. The pattern of \(N_z\) is observed by using an image processor. This method is not subjected to fringe interpretation since maximum fringe value is less than 0.5.

Consequently, the seeming difference of distribution of \(N_z\) near the inner equator between the numerical and experimental results is a measure of the subtle difference between the magnitude of \(\varepsilon_r\) and \(\varepsilon_\theta\). It does not seem to
have a definite physical significance relevant to the current problem at hand.

3. Change of geometry
   
a. Deformed bulge and profile  All bulge profiles are alike from the mesh plots (Figs. 41 to 45). Table 6 gives the minimum and maximum radii for numerical and experimental results. The amplified mesh plots (Figs. 41(c) and 43(c)) clearly show that at still larger deformations, the contact area will increase when folding occurs at the outer corner of the ring, which in turn will increase friction as often observed in experiment.

b. Displacement near equator  Radial displacement at the equator predicted from the numerical analysis (cases 1.b and 4) and calculated for experimental results of 16% deformation using an Eulerian analysis are given in Fig. 55. Only small difference exists near the inner radius where the experimental result shows more of a curvature. This curvature, in turn, gives a more pronounced increase of radial strain for this region as discussed in section 2 of this chapter. It is interesting to see that the minimum and maximum displacements agree very well.
c. decrease of inner diameter versus deformation

The change of internal diameter versus deformation for numerical analysis when loaded is compared with experimental results in Hartley's work [58] and numerical solution of Avitzur [69] as shown in Table 7 and Fig. 56. The present numerical analysis for m = 1 (sticking friction) is consistent with itself and the finite element solution [58] for m = 0.9. But the curve is flatter when compared with experiment for sticking friction. It is apparent that true sticking friction is not achieved. The unloaded curve when compared with Gomide's experiment is given in Fig. 57 and agrees well.

B. General Discussion

1. of retained deformation/incompressibility

One of the characteristics of the material used in photoplasticity is its time dependency. For the particular experiment involved, it is manifested in relaxation of fringe orders and hence strains after unloading. The deformation retained after the load is removed is smaller because of relaxation even when the model is cooled rapidly after unloading. The retained deformation is actually due to the removal of two types of strain, elastic and creep (time-dependent) strain. (This material has a higher elastic strain than ordinary metal such as aluminum, hence the retained deformation due to elastic recovery will be smaller
than ordinary metal.) The finite element method, which uses the same stress-strain curve as the photoplastic material, should also indicate this higher recovery. However, in the finite element model, a creep/relaxation model is not used, so the final deformation should not be as low as the experiment. However, the numerical solution gives a retained deformation not higher than the experiment (refer to Table 7) for at least the following reasons:

1. It is found during parametric study that a more rigorous tolerance of convergence will give a higher retained deformation. Possibly the tolerance is not tight enough.

2. Another more fundamental characteristic of the displacement formulation of the finite element method is a 'too stiff' situation. In the displacement formulation, a continuum is replaced by a model of finite elements. The original continuum has an infinite degrees of freedom while the model has finite degrees of freedom. These finite degrees of freedom are determined by the number of nodes used in the mesh. Consequently, the model has a higher constraint (because of less freedom) than the continuum. A higher constraint gives a stiffer stiffness matrix consequently a lower displacement. Thus,
the displacement formulation of the finite element method is often referred to as a lower bound solution. A lower bound solution is of course more dangerous than an upper one.

3. It has also been shown that 'locking' can occur when a fully plastic situation is reached for an element [86]. Specifically, the shape function itself exerts different degrees of constraint on the model due to the requirement of incompressibility. A different element arrangement and different types of shape functions can sometimes lessen this phenomenon. This is probably why the quadratic elements and the Updated Lagrangian formulation give a better response. A rigorous solution to this problem has been suggested [86]; however, the author has not yet seen an application of this solution other than the originators and their coworkers. The author considers their approach cumbersome in practicality because a different formulation needs to be implemented for each specific type of element. A better way probably should be sought.

4. It is assumed that no yielding occurs during unloading (see Chapter III). This assumption may
not be true when an excessively low deformation was observed when using the modified von Mises yield criterion. A step by step unloading involving iteration may be needed, but the cost would obviously be much higher.

5. The general trend in the upset forging of rings is that higher friction leads to more pronounced bulging. The lower value of bulge in the finite element solutions may be another indication that sticking friction has not been achieved in the numerical modeling.

2. Parametric studies

Results presented in Chapter IV are only selected ones from many cases of studies. The total cases actually run after the program was developed exceeded 60. Most of them are for verification and parametric studies. Since most of them are of an elementary nature, they will be stated without presenting the output. The present results could not have been obtained if these studies had not been first performed successfully.

1. Elastic solutions of various geometry and loading/boundary conditions were first performed. This is a prerequisite of any elastic-plastic analysis since a continuous adjustment of elastic solution was used one way or another.
The development of a neutral radius was verified for various friction levels. These trends follow the general rule in Chapter III. The initial general pattern of bulge profile also agrees with Nagamatsu's findings [53], that a double bulge was present.

Different meshes had been designed to 'catch' some of the anomaly observed in the above section but to no avail.

According to the modified-δ stiffness method, the stiffness of the friction layer should be infinite. The actual number 1000 was chosen somewhat arbitrarily when the stiffness was progressively increasing and beyond a threshold no significant change in results occurred. The threshold is approximately 100.

Physically, the thickness of the friction layer can be very small when compared with the height of the ring. However, for a certain range of height, the constraint together with the high stiffness of the layer remains about the same. As a rule of thumb, the thicker the friction layer, hence more material, the higher the constraint is. The reasoning is similar to the reason that a higher constraint exists in plane
strain than in plane stress situations. A reasonable height is then chosen.

6. Tolerances and maximum allowable iterations within an increment were chosen so that the magnitude of reactions would be satisfactory.

3. Verification of finite element program

The antecedent of the present program was the program in an M.S. thesis [10] by the author. This thesis compared the finite element solution with finite difference solutions as well as an analytical solution for autofrettage of a thick-walled cylinder. The program [10] did not include large deformation theory or a modified von Mises yield criterion or the transition zone analysis. The predicted residual stress patterns across the thickness of the wall matched very well.

This program [10] was then used to predict the deformations of the current problem of the upset forging of a plastic ring. The results of the strain patterns of the ring which was unloaded from 16% deformation under sticking friction, when compared with Gomide's experiment [61] showed discrepancies mainly in the distribution of $N_z$ and $\varepsilon_r$ near the line of symmetry ($z=0$). In this analysis, a simple plastic model was adopted which used a von Mises yield criterion, Prandtl-Reuss flow rule and incorporated the
effect of elastic unloading.

At this stage, further refinement of the program was desired in order to have a better match with Gomide's experimental results. The refinement would include a modified von Mises yield criterion and large deformation analysis. It was then decided to develop a new program to implement the desired refinement (Ref. p. 62).

The newly developed program was first tested for elastic solutions which included a plane-stress simple tensile test, axial compression of an axisymmetrical solid cylinder and a plane stress 90 degree notched specimen under tension. The results were found satisfactory (Ref. previous subsection 'Parametric Studies').

For elastic-plastic problems (small deformation theory) the upset forging of the plastic ring under sticking friction was analyzed and compared with the results of the previous program as mentioned above. The result of comparison was satisfactory. In this analysis, isoparametric elements instead of triangular elements were used, transition zone analysis was incorporated (case 1.b).

For the current problem of upset forging of a plastic ring, there are no analytical/numerical or experimental predictions available for comparison.

The closest equivalent was the elastic-plastic finite element prediction by Hartley et al. [58]. This was for the
upset forging of a ring similar to the one studied in this dissertation. A calibration curve which relates \% of axial deformation to \% decrease of inner diameter when calculated by the present program was compared to Hartley's solution [58] in Fig. 56 and agreed well. For a specific value of the interface friction factor, the calibration curve was compared with an experiment and found to be satisfactory [58].

Up to this point of comparison, the results from the present program was considered satisfactory. From here onward, the only independent solution for the upset forging of a plastic ring is the photoplastic solution of Gomide [61]. An attempt to obtain a good match between the numerical results and Gomide's experimental results was the goal of the present research.

When the results are compared with the present program, the 3-dimensional strain fields are similar. They disagree in certain details especially in the distribution of $N_z$ at the line of symmetry ($Z=0$) near the inner radius.

With the level of agreement between numerical and experimental predictions of the strain field the two methods should be evaluated with respect to their relative accuracy. At this stage, it is clear that while neither may be exact, the numerical prediction is more advanced than the experimental one. The next step in the iterative process
toward good prediction of the forming process is the refinement of the photoplastic experiment with respect to better temperature control, a material which does not have a large elastic recovery with less dependence on pressure in yielding.

The verification of nonlinear finite element program is by no means easy. Differences are often found when comparisons are made either with numerical or experimental solutions. For example, in an attempt to check the validity of different nonlinear finite element programs [87], the results of the upsetting of a solid cylinder were compared and large differences were found among them.

Furthermore, most of the comparisons of nonlinear finite element method with experiments are done by numerical analysts. Relatively few comparisons are done by experimentalists. Two examples from literature are given below to show the comparisons done by experimentalists.

Upsetting of a cylinder when unloaded was compared with the finite element program NONSAP by Friere et al. [72]. Poor results were found in the comparison of radial displacement near the equator. Actually that study compared an unloaded experiment with loaded numerical solution. This was not adequate, especially with a material like Laminac! The discrepancy is bound to be more significant if an unloaded numerical solution were obtained. The researchers
blamed the discrepancy on "... influence of different yield criterion."

In another attempt [89], predictions of pressures for gross yielding of thick-walled cylinders under external pressure was given by using the finite element program ADINA. Comparisons with experimental and other analytical results showed that the method was the least satisfactory. The authors gave 12 considerations, both experimental and numerical, as possible sources of errors. On the numerical side, they suspected numerical accuracy, unloading during pressure tests and yield criterion. Finally, "... results ... show that existing yield criteria can be in considerable error. The lack of agreement between analysis and experiment ... is attributed mainly to the plasticity theories used." Hence they suggested that a pressure-dependent theory of yield might give better agreement. Note that some of these considerations have been incorporated into the author's present program.

4. Instability

Instability in finite element methods in plasticity is often observed. Its source can be either due to an inadequate modeling technique or an actual physical instability.

A physical instability develops when a great or catastrophic response occurs by a given small input or
perturbation. Sometimes a small force/displacement input gives rise to an excessive displacement and consequently unreasonable stresses and strains, which usually indicate a nearly singular stiffness matrix. Other times the symptom is manifested in the fact that the stiffness matrix cannot be decomposed. When using, say, the Cholesky method, the square root term becomes negative, indicating a non positive-definite matrix. Also sometimes many points are unloaded even when the load itself is increasing. For example, when a thick-walled cylinder is subjected to internal pressure, instability occurs when the plastic front propagates a little over one half the wall thickness, which indicates that the outer elastic layer can no longer hold the plastic zone. The same would apply when a solid cylinder is upset in a frictionless situation, or a necking problem occurs in a uniaxial tensile specimen.

On the other hand, the instability caused by the inadequate modeling technique can usually be eliminated by proper adjustment of input parameters such as the mesh, the tolerance etc. Case 5 is believed to be such a one. A close examination of the output shows that convergence is not reached in an increment due to too large an increment. The unbalanced force thus accumulated then caused the eventual collapse of the particular run. Decreasing the size of increment in preventing instability is often done in
parametric studies since for these preliminary runs accuracy is usually not important while low cost is. This particular case was not rerun because of inhibitive cost and is presented only to show that in modeling, usually not all is well.

5. Computer experience

Although it is a great temptation to use a 'packaged' program for research (the author had tried ANSYS), eventually one has to develop or modify according to needs since there are always deficiencies, ambiguity and inconveniences in the programs. In my opinion, only the 'mechanical' parts of the program should be used without writing one's own. For example, the various storage considerations, solutions of simultaneous equations, derivation of standard shape functions etc. can be learned once and for all. After these things are mastered, if efficiency is needed then we can appeal to specialists. The 'non-mechanical' parts such as recognizing the modeling needs and translating existing principles from mechanics to finite element methods in meeting these needs are the crux of research.

6. Similitude requirement

The present work is not an 'analogue' solution [4] since the finite element method uses the mathematical theory
of plasticity while unloaded photoplasticity uses only the relationship of strain-dielectric tensor and strain-displacement which can be considered as definitions. Although differences in material properties between photoplastic material and metal exist, the finite element method can meet this deficiency when it is properly validated.

Although the so called photoelastoviscoplasticity is claimed to be superior to photoplasticity [88], we offer a challenge here that a verification or comparison other than an 'analogue' solution be made to support the claim.

7. Interaction of experiment and numerical method

Not only is a rigorous comparison helpful in assessing values of both photoplasticity and the finite element method, it also provides some helpful aid in practicality, especially when contour plots are drawn. For example, in one of the author's experiments of upsetting of a ring with a purpose to verify the patterns observed by Gomide [61], he forgot to ask the technician to mark and distinguish which edge of a slice was the inner radius. The slice is a tiny and transparent specimen, and it is not very easy without marking to find the right position. However, aided by the contour plots of the fringes calculated from the finite element program, the correct position was found easily simply by making contour plots of the slice on the imaging
system and comparing. The two approaches, experimental and numerical can give each other a very good idea whether a 'ball park' solution has been reached. This usually saves a lot of guessing and can eliminate uncertainty.

8. Advantages/disadvantages of FEM and photoplasticity

The advantages of the finite element methods are:
1. Able to model different materials and processing environment.
2. Can trace the detailed history of deformation and predict future trends.
3. Once developed for a particular problem, results can be obtained repeatedly with ease.
4. Can predict loaded and unloaded response.

The disadvantages of the finite element methods are:
1. It is very expensive in developing and running.
2. It needs a mathematical theory of plasticity and finite deformation. Many assumptions used still needs experimental verification.

The author believes that with the advent of the 32-bit microcomputer, a dedicated computer can reduce the running cost to a minimum.

The advantages of the unloaded photoplasticity are:
1. No need to have a mathematical theory of plasticity. Being phenomenological, it represents the true response for the model
material and processing environment.

2. It is the only feasible experimental method in finding internal three-dimensional strain distribution in the plastic region.

The disadvantages of the unloaded photoplasticity are:

1. Difficult to find or prepare commercially available materials that have both good mechanical and optical properties which satisfy rigorous similitude requirements.

2. Only unloaded response can be investigated.

3. Experimental setup is not always easy.
VI. CONCLUSION

Finite element programs have been developed to model the upsetting of a forged plastic ring. Results are compared with photoplasticity and other numerical, experimental results. In the author's opinion the following conclusions can be drawn from the investigation:

1. Both finite element and photoplastic methods give similar pattern of results in most of the variables.

2. The agreement of results from any methods in outward change in geometry will not guarantee the same for all internal variables. Specifically, contributions to strain or other distributions can come from various sources in a certain process. Such can be different material models, large or small deformation assumptions, loading or unloading and different finite element approximations such as type of mesh used and different data analysis method in experiment. Hence, a proper identification of these factors is important in the modeling process. A whole field comparison of parameters is more convincing than a point-wise one.

3. It is not fit to compare results of loaded and unloaded responses, especially when one of the
material is plastic with a large elastic recovery.

4. Different C/T ratios can make significant difference in regions where hydrostatic pressure is large, especially near the work/tool interface which can obscure the region where failure may occur.

5. Compatibility should be checked in addition to total equilibrium in photoplasticity, and isoclinics should be recorded. An Eulerian rather than Lagrangian analysis should be used for three-dimensional strain analysis.

6. When both methods are still in a stage of art, they can complement and verify each other in metal forming processes.
VII. FIGURES
Figure 1. Possible paths in elastic-plastic analysis
slope \[ = \left[ K^{ep} \right]_{i-1} \]

slope \[ = \left[ K \right]_{i} \]

\( i \) = current time step (increment)

\( i-1 \) = previous time step (increment)

Fig. 2(a) One-dimensional illustration of tangent stiffness method
APPLY LOAD \( \{\Delta f\} \)

OBTAIN \( \{\Delta u\}_i = [K^e_p]^{-1}_{i-1} \{\Delta f\}_i \)

\( \{\Delta e\}_i = [B] \{\Delta u\}_i \)

\( \{\Delta \sigma\}_i = [D^e_p]_{i-1} \{\Delta e\}_i \)

FORM \( [K^e_p]_i \)

WHERE \( i = \) index of increment

**Fig. 2(b) Schematic of the tangent stiffness method**
Figure 3. One-dimensional illustration of initial stress method for one increment $i$
Figure 4. Flow sequence of initial stress method within an increment

SUPERSCRIPT $i$ = ITERATION

$il$ = INITIAL LOAD
Figure 5. One-dimensional illustration of transition zone correction of initial stress method
Given a load increment from A to B, assume B is in transition zone. Stresses and strains are calculated as follows:

1. obtain $\Delta \varepsilon_1$
2. calculate $\Delta \sigma_1 = D \Delta \varepsilon_1$
3. $\sigma_1 = \sigma_0 + \Delta \sigma_1$ where $\sigma_0$ is the stress at the beginning of the increment (point A).
4. find ratio $r$ by the formula given in Appendix A as according to Yamada [17].
5. calculate $\Delta \sigma_{1p} = D_{1p} (1-r) \Delta \varepsilon_1$
6. $\Delta \sigma_{1p} = (1-r) \Delta \sigma_1 - \Delta \sigma_{1p} = (1-r) (\Delta \sigma_1 - D_{1p} \Delta \varepsilon_1)$
7. $\varepsilon_1 = \varepsilon_0 + \Delta \varepsilon_1 = \varepsilon$
8. $\sigma = \sigma_1 - \Delta \sigma_{1p}$

We see that the correct point C has been found. The next iteration will no longer require the transition zone correction described in this figure, that is, $r$ is no longer calculated, hence it can be set to zero.

Figure 6. Algorithm of transition zone correction of initial stress method
Figure 7. One-dimensional illustration of transition zone correction of tangent stiffness method.
Figure 8. Equilibrium check: (a) tangent stiffness method (b) initial stress method.
Figure 9. One-dimensional illustration of acceleration of convergence of initial stress method
Figure 10. Flow modes of a ring when compressed axially: (a) Flow in left portion is for high friction at upper and lower interface (b) Flow at right is for low friction at the upper and lower interface
\[ m = \frac{\tau^*}{\tau_0} \]

\[ \beta = \frac{m}{1-m} \]

- \( m = 0 \Rightarrow \beta = 0 \) 
  NO FRICTION
- \( 0 < m < 1 \) 
  INTERMEDIATE FRICTION
- \( m \rightarrow 1, \beta \rightarrow \infty \) 
  STICKING FRICTION

Figure 11. Modeling of friction: modified \( \beta \) stiffness method.
Figure 12. Modeling of elastic unloading: (a) Loaded, displacement input (b) Unloaded, force input. (Reaction force revealed at the nodes where the boundary condition has been changed). (shaded region is the friction layer)
Figure 13. Example of Yield Surfaces [79]
Figure 14. Domain of modeling and mesh plots (shaded region $\text{shaded}$) $h$, $R_i$, $R_o$ are undeformed dimension.
Figure 15.1 (a) Description of geometry, element and domain of plotting.
One half of the ring is modeled.
node a: horizontal degree of freedom restrained.
node b: horizontal degree of freedom restrained; displacement input in the vertical (negative z) direction.

Figure 15.1(b) Description of boundary conditions for sticking friction.
Figure 15.2 Domain of plotting for experimental data

$h = 40$ units, $R_i = 30$ units, $R_o = 60$ units.

<table>
<thead>
<tr>
<th></th>
<th>A-8%</th>
<th>A-16%</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>5.2 units</td>
<td>4.7 units</td>
</tr>
<tr>
<td>b</td>
<td>1.0 units</td>
<td>1.7 units</td>
</tr>
<tr>
<td>c</td>
<td>3.8 units</td>
<td>4.7 units</td>
</tr>
<tr>
<td>d</td>
<td>1.4 units</td>
<td>3.3 units</td>
</tr>
<tr>
<td>e</td>
<td>1.0 units</td>
<td>2.2 units</td>
</tr>
</tbody>
</table>
Figure 16. Strain distribution at 8% - case la. loaded: linear element, von Mises yield criterion, small deformation (See Table 3.1)
Figure 17. Fringe distribution at 8% - case 1a. loaded
(See Table 3.1)
Figure 18. Equivalent stress distribution at different % of deformation - case l.a & l.b loaded
(See Table 3.1)
Figure 18. (Continued)
Figure 19. Strain distribution at 8% - case 3 loaded: linear element, modified von Mises yield criterion, small deformation (See Table 3.1)
Figure 20. Fringe distribution at 8% - case 3 loaded
(See Table 3.1)
Figure 21. Strain distribution at 8% - case 2 loaded: quadratic element, von Mises yield criterion, small deformation (See Table 3.1)
Figure 22. Fringe distribution at 8% - case 2 loaded (See Table 3.1)
Figure 23. Equivalent stress distribution at different % of deformation - case 2 loaded
Figure 24. Strain distribution at 16% - case 1.6 loaded
Figure 25. Fringe distribution at 16% - case 1.b loaded
Figure 26. Strain distribution at 16% - case 4 loaded: linear element, von Mises yield criterion, large deformation
Figure 27. Fringe distribution at 16% - case 4 loaded
Figure 28. Equivalent stress distribution at different %
of deformation - case 4 loaded
Figure 29. Strain distribution at 8.7% - case 5 loaded: quadratic element, von Mises yield criterion, large deformation
Figure 30. Fringe distribution at 8.7% - case 5 loaded quadratic element, von Mises yield criterion, large deformation
Figure 31. Strain distribution at 8% - case 1.a unloaded, linear element, von Mises yield criterion, small deformation
Figure 31. (Continued)
Figure 31. (Continued)
Figure 32. Fringe distribution at 8% - case 1.a, unloaded
Figure 32. (Continued)
Figure 33. Strain distribution - case l.b, unloaded
Figure 33. (Continued)
Figure 33. (Continued)
Figure 34. Fringe distribution - case 1.b, unloaded
Figure 34. (Continued)
Figure 35. Strain distribution - case 3, unloaded: linear element, modified von Mises yield criterion, small deformation
Figure 35. (Continued)
Figure 35. (Continued)
Figure 36. Fringe distribution - case 3, unloaded
Figure 36. (Continued)
Figure 37. Strain distribution - case 2, unloaded: quadratic element, von Mises yield criterion, small deformation
Figure 37. (Continued)
Figure 37. (Continued)
Figure 38. Fringe distribution – case 2, unloaded
Figure 38. (Continued)
Figure 39. Strain distribution - case 4, unloaded: linear element, von Mises yield criterion, large deformation
Figure 39. (Continued)
Figure 40. Fringe distribution - case 4, unloaded: linear element, von Mises yield criterion, large deformation
Figure 40. (Continued)
Figure 41. Meshplot of case 1.a - linear element, von Mises yield criterion, small deformation: (a) Loaded at 8%, deformed and undeformed mesh (b) Unloaded, deformed and undeformed mesh (c) Loaded at 8%, deformed mesh with displacement amplified 5 times.
Figure 42. Mesh of case 3 - linear element, modified von Mises yield criterion, small deformation:
(a) Loaded at 8%, deformed and undeformed mesh
(b) Unloaded deformed and undeformed mesh
Figure 43. Mesh of case 2 - quadratic element, von Mises yield criterion, small deformation: (a) Loaded at 8%, deformed & undeformed mesh (b) Unloaded deformed & undeformed mesh (c) Loaded at 8%, displacement amplified 5 times
Figure 44. Mesh of case 1.b - linear element, von Mises yield criterion, small deformation: (a) Loaded at 16%, deformed and undeformed mesh (b) Unloaded, deformed and undeformed mesh
Figure 45. Mesh of case 4 - linear element, von Mises yield criterion, large deformation: (a) Loaded at 16%, deformed and undeformed mesh (b) Unloaded, deformed and undeformed mesh
Figure 46. Fringe N_g distribution at 8% - experiment
Figure 47. Fringe $N_z$ distribution at 8% - experiment
Figure 48. Experimental strain distribution: 8%, large deformation, $\varepsilon_r > \varepsilon_\theta$
Figure 48. (Continued)
Figure 49. Experimental strain distribution: 8%, small deformation, $\varepsilon_r > \varepsilon_\theta$
Figure 49. (Continued)
Figure 49. (Continued)
Figure 50. Experimental fringe distribution at 16%
Figure 50. (Continued)
Figure 51. Experimental strain distribution: 16%, large deformation, $\varepsilon_r < \varepsilon_0$ partly valid
Figure 51. (Continued)
Figure 51. (Continued)
Figure 52. Experimental strain distribution: 16%, small deformation, $\varepsilon_r > \varepsilon_\theta$
Figure 52. (Continued)
Figure 52. (Continued)
Figure 53. Experimental strain distribution: 16%, small deformation, $\varepsilon_r < \varepsilon_\theta$ partly valid
Figure 53. (Continued)
Figure 53. (Continued)
Figure 54. Strain distribution of experiment: 16%, large deformation, $\varepsilon_r > \varepsilon_\theta$
Figure 54. (Continued)
Figure 54. (Continued)
Figure 55. Comparison of unloaded radial displacement distribution at equator (Z = 0)
Figure 56. Comparison of % of decrease of inner diameter versus % of deformation for loaded response
Figure 57. Comparison of % of decrease of inner diameter versus % of deformation for unloaded response
VIII. TABLES
Table 1.1 A Summary of Numerical Cases Studied

<table>
<thead>
<tr>
<th>Case Number</th>
<th>1.a</th>
<th>1.b</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of Axial Deformation</td>
<td>8</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Name Used in Dissertation</td>
<td>Simple Plastic</td>
<td>Simple Plastic</td>
<td>Quadratic</td>
</tr>
<tr>
<td>Total No. of Elements (^a)</td>
<td>42</td>
<td>42</td>
<td>54</td>
</tr>
<tr>
<td>Total No. of Nodes</td>
<td>56</td>
<td>56</td>
<td>193</td>
</tr>
<tr>
<td>Type of Elements</td>
<td>4-noded Linear</td>
<td>4-noded Linear</td>
<td>8-noded Quadratic</td>
</tr>
<tr>
<td>Yield Criterion</td>
<td>von Mises</td>
<td>von Mises</td>
<td>von Mises</td>
</tr>
<tr>
<td>Flow Rule</td>
<td>Associated</td>
<td>Associated</td>
<td>Associated</td>
</tr>
<tr>
<td>Magnitude of Deformation</td>
<td>Small</td>
<td>Small</td>
<td>Small</td>
</tr>
<tr>
<td>Is Accelerator Used?</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Total No. of Increments</td>
<td>43</td>
<td>43</td>
<td>43</td>
</tr>
<tr>
<td>Max. No. of Iterations/Increment</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Unloaded at Max. Deformation?</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

\(^a\) Total number of elements includes the friction layer.
Table 1.1 - (Continued)

<table>
<thead>
<tr>
<th>Case Number</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of Axial Deformation</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Name Used in Dissertation</td>
<td>Nonassociated</td>
<td>Large Deformation</td>
</tr>
<tr>
<td>Total No. of Elements</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>Total No. of Nodes</td>
<td>56</td>
<td>56</td>
</tr>
<tr>
<td>Type of Elements</td>
<td>4-noded Linear</td>
<td>4-noded Linear</td>
</tr>
<tr>
<td>Yield Criterion</td>
<td>Modified von Mises</td>
<td>von Mises</td>
</tr>
<tr>
<td>Flow Rule</td>
<td>Nonassociated</td>
<td>Associated</td>
</tr>
<tr>
<td>Magnitude of Deformation</td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td>Is Accelerator Used?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Total No. of Increments</td>
<td>43</td>
<td>43</td>
</tr>
<tr>
<td>Max. No. of Iterations/Increment</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Unloaded at Max. Deformation?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 1.1 - (Continued)

<table>
<thead>
<tr>
<th>Case Number</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of Axial Deformation</td>
<td>12</td>
</tr>
<tr>
<td>Name Used in Dissertation</td>
<td>Large Deformation &amp; Quadratic Element</td>
</tr>
<tr>
<td>Total No. of Elements</td>
<td>54</td>
</tr>
<tr>
<td>Total No. of Nodes</td>
<td>193</td>
</tr>
<tr>
<td>Type of Elements</td>
<td>8-noded Quadratic</td>
</tr>
<tr>
<td>Yield Criterion</td>
<td>von Mises</td>
</tr>
<tr>
<td>Flow Rule</td>
<td>Associated</td>
</tr>
<tr>
<td>Magnitude of Deformation</td>
<td>Large</td>
</tr>
<tr>
<td>Is Accelerator Used?</td>
<td>Yes</td>
</tr>
<tr>
<td>Total No. of Increments</td>
<td>30</td>
</tr>
<tr>
<td>Max. No. of Iterations/increment</td>
<td>30</td>
</tr>
<tr>
<td>Unloaded at Max. Deformation?</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 1.2 Material Properties of Rigid Laminac Polyester [61]

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young's Modulus (E)</td>
<td>120,000 psi</td>
</tr>
<tr>
<td>Initial Yield Strength in Tension ((σ₀)_T, T)</td>
<td>4,100 psi</td>
</tr>
<tr>
<td>Initial Yield Strength in Compression ((σ₀)_C, C)</td>
<td>8,610 psi</td>
</tr>
<tr>
<td>C/T = (σ₀)_C/σ₀)_T</td>
<td>2.1</td>
</tr>
<tr>
<td>Poisson's Ratio (ν)</td>
<td>0.3</td>
</tr>
<tr>
<td>Strain-Hardening Parameter (H')</td>
<td>8,400 psi</td>
</tr>
<tr>
<td>H'/E</td>
<td>0.07</td>
</tr>
</tbody>
</table>

a Since the Poisson's ratio for 'Laminac' in the elastic range is not available, the value of steel is used instead.
Table 2.1 Correspondence of Figure Number and Numerical Cases

<table>
<thead>
<tr>
<th>Condition</th>
<th>Case</th>
<th>1.a</th>
<th>1.b</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loaded</td>
<td>Strain</td>
<td>16</td>
<td>24</td>
<td>21</td>
<td>19</td>
<td>26</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>Fringe</td>
<td>17</td>
<td>25</td>
<td>22</td>
<td>20</td>
<td>27</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Equivalent Stress</td>
<td>18</td>
<td>18</td>
<td>23</td>
<td>NA</td>
<td>28</td>
<td>NA</td>
</tr>
<tr>
<td>Unloaded</td>
<td>Strain</td>
<td>31</td>
<td>33</td>
<td>37</td>
<td>35</td>
<td>39</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>Fringe</td>
<td>32</td>
<td>34</td>
<td>38</td>
<td>36</td>
<td>40</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>Mesh</td>
<td>41</td>
<td>44</td>
<td>43</td>
<td>42</td>
<td>45</td>
<td>NA</td>
</tr>
</tbody>
</table>

*NA - Not applicable or not available.*
Table 2.2 Correspondence of Figure Number and Assumptions of Experimental Data Analysis

<table>
<thead>
<tr>
<th></th>
<th>$\varepsilon_r &gt; \varepsilon_\theta$ Small Deformation</th>
<th>$\varepsilon_r &lt; \varepsilon_\theta^a$ Small Deformation</th>
<th>$\varepsilon_r &gt; \varepsilon_\theta$ Large Deformation</th>
<th>$\varepsilon_r &lt; \varepsilon_\theta^a$ Large Deformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-8%</td>
<td>Strain</td>
<td>49</td>
<td>NA$^b$</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>$N_\theta$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$N_z$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-16%</td>
<td>Strain</td>
<td>52</td>
<td>53</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>$N_\theta$, $N_z$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$^a$Partial correction (See section B in Chapter IV).

$^b$NA - not applicable.
Table 3.1: Range of Parameters of Plots: Loaded

A. Equivalent Stress (von Mises Stress, \( \sigma \)) in ksi.

<table>
<thead>
<tr>
<th>Case</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of Deformation</td>
<td>4.6</td>
<td>6.25</td>
<td>8</td>
</tr>
<tr>
<td>Figure No.</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>41.0</td>
<td>41.0</td>
<td>41.0</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>43.3</td>
<td>45.7</td>
<td>48.2</td>
</tr>
<tr>
<td>Increment</td>
<td>0.23</td>
<td>0.469</td>
<td>0.72</td>
</tr>
</tbody>
</table>
Table 3.1 - (Continued)

B. Strain

<table>
<thead>
<tr>
<th>Case</th>
<th>l.a</th>
<th>l.b</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of Deformation</td>
<td>8%</td>
<td>16%</td>
</tr>
<tr>
<td>Figure No.</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td>Strains</td>
<td>( \varepsilon_r )</td>
<td>( \varepsilon_\theta )</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>0.00358</td>
<td>-0.0174</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>0.0897</td>
<td>0.0222</td>
</tr>
<tr>
<td>Increment</td>
<td>0.0086</td>
<td>0.00396</td>
</tr>
</tbody>
</table>
Table 3.1 - (Continued)

<table>
<thead>
<tr>
<th>Case</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of Deformation</td>
<td>8%</td>
<td>8%</td>
</tr>
<tr>
<td>Figure No.</td>
<td>21</td>
<td>19</td>
</tr>
<tr>
<td>Strains</td>
<td>$\varepsilon_r$</td>
<td>$\varepsilon_\theta$</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>0.00105</td>
<td>-0.01711</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>0.132</td>
<td>0.0223</td>
</tr>
<tr>
<td>Increment</td>
<td>0.013</td>
<td>0.0039</td>
</tr>
</tbody>
</table>
Table 3.1 - (Continued)

<table>
<thead>
<tr>
<th>Case</th>
<th>% of Deformation</th>
<th>Figure No.</th>
<th>Strains</th>
<th>Minimum Value</th>
<th>Maximum Value</th>
<th>Increment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\epsilon_r$</td>
<td>$\epsilon_\theta$</td>
<td>$\epsilon_z$</td>
<td>$\epsilon_r$</td>
</tr>
<tr>
<td>4</td>
<td>16%</td>
<td>26</td>
<td>0.0144</td>
<td>-0.04</td>
<td>-0.374</td>
<td>-0.0006</td>
</tr>
<tr>
<td>5</td>
<td>12%</td>
<td>29</td>
<td>0.02216</td>
<td>0.00916</td>
<td>-0.03</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Table 3.1 - (Continued)

C. Isochromatic Fringe Orders

<table>
<thead>
<tr>
<th>Case</th>
<th>1.a</th>
<th>1.b</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of Deformation</td>
<td>8%</td>
<td>16%</td>
<td>8%</td>
</tr>
<tr>
<td>Figure No.</td>
<td>17</td>
<td>25</td>
<td>22</td>
</tr>
<tr>
<td>Fringes</td>
<td>$N_\theta$</td>
<td>$N_z$</td>
<td>$N_\theta$</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>0.518</td>
<td>0.0444</td>
<td>1.07</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>2.82</td>
<td>1.43</td>
<td>5.96</td>
</tr>
<tr>
<td>Increment</td>
<td>0.23</td>
<td>0.1386</td>
<td>0.489</td>
</tr>
<tr>
<td>Case</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>------</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>% of Deformation</td>
<td>8%</td>
<td>16%</td>
<td>12%</td>
</tr>
<tr>
<td>Figure No.</td>
<td>20</td>
<td>27</td>
<td>30</td>
</tr>
<tr>
<td>Fringes</td>
<td>$N_0$</td>
<td>$N_z$</td>
<td>$N_0$</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>0.633</td>
<td>0.0392</td>
<td>1.18</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>2.5</td>
<td>1.15</td>
<td>7.48</td>
</tr>
<tr>
<td>Increment</td>
<td>0.1867</td>
<td>0.111</td>
<td>0.63</td>
</tr>
</tbody>
</table>
Table 3.2 Range of Parameters of Plots: Unloaded

A. Strains

<table>
<thead>
<tr>
<th>Case</th>
<th>Figure No.</th>
<th>Deformation</th>
<th>$\varepsilon_r &lt; \varepsilon_0$ corrected?</th>
<th>Parameters</th>
<th>Minimum Value</th>
<th>Maximum Value</th>
<th>Increment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>49</td>
<td>Small</td>
<td>No</td>
<td>$\varepsilon_r$</td>
<td>0.007</td>
<td>0.1067</td>
<td>0.00997</td>
</tr>
<tr>
<td></td>
<td>48</td>
<td>Large$^a$</td>
<td>No</td>
<td>$\varepsilon_\theta$</td>
<td>-0.03375</td>
<td>0.0155</td>
<td>0.005475</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\varepsilon_z$</td>
<td>-0.0982</td>
<td>-0.0159</td>
<td>0.00915</td>
</tr>
</tbody>
</table>

$^a$Range also used for unloaded numerical plots.
Table 3.2 - (Continued)

<table>
<thead>
<tr>
<th>Case</th>
<th>A-16%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure No.</td>
<td>52</td>
</tr>
<tr>
<td>Deformation</td>
<td>Small</td>
</tr>
<tr>
<td>$\varepsilon_r &lt; \varepsilon_0$ corrected?</td>
<td>No</td>
</tr>
<tr>
<td>Parameters</td>
<td>$\varepsilon_r$</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>0.006</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>0.2533</td>
</tr>
<tr>
<td>Increment</td>
<td>0.02473</td>
</tr>
</tbody>
</table>
Table 3.2 - (Continued)

<table>
<thead>
<tr>
<th>Case</th>
<th>Figure No.</th>
<th>Deformation</th>
<th>$\epsilon_r &lt; \epsilon_\theta$ corrected?</th>
<th>Parameters</th>
<th>Minimum Value</th>
<th>Maximum Value</th>
<th>Increment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-16%</td>
<td>54</td>
<td>Large</td>
<td>No</td>
<td>$\epsilon_r$</td>
<td>0.006</td>
<td>0.2733</td>
<td>0.0267</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\epsilon_\theta$</td>
<td>-0.0573</td>
<td>0.0437</td>
<td>0.0101</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\epsilon_z$</td>
<td>-0.2296</td>
<td>-0.01901</td>
<td>0.0234</td>
</tr>
<tr>
<td></td>
<td>51</td>
<td>Large$^a$</td>
<td>Yes</td>
<td>$\epsilon_r$</td>
<td>0.0042</td>
<td>0.2733</td>
<td>0.02691</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\epsilon_\theta$</td>
<td>-0.0573</td>
<td>0.0457</td>
<td>0.0103</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\epsilon_z$</td>
<td>-0.2296</td>
<td>-0.01901</td>
<td>0.0234</td>
</tr>
</tbody>
</table>

$^a$Range also used for unloaded numerical plots.
Table 3.2 - (Continued)

B. Isochromatic Fringe Orders

<table>
<thead>
<tr>
<th>Case</th>
<th>A-8%&lt;sup&gt;a&lt;/sup&gt;</th>
<th>A-16%&lt;sup&gt;a&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure No.</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>Parameters</td>
<td>(N_0)</td>
<td>(N_z)</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>0.19</td>
<td>0.0</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>2.69</td>
<td>1.63</td>
</tr>
<tr>
<td>Increment</td>
<td>0.25</td>
<td>0.163</td>
</tr>
</tbody>
</table>

<sup>a</sup>The range is also used for all unloaded numerical plots.
Table 4 Comparison of Plots: 8%

(a) $\varepsilon_r$

<table>
<thead>
<tr>
<th>Case</th>
<th>1.a - Simple Plastic</th>
<th>2 - Quadratic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Decrease</td>
<td>Decrease</td>
</tr>
<tr>
<td>3</td>
<td>Inside Equator</td>
<td>Inside Equator</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Inside Equator</td>
<td>Inside Equator</td>
</tr>
<tr>
<td>6</td>
<td>Basis</td>
<td>Low</td>
</tr>
</tbody>
</table>

*a Condition is specified on page 97.

*b Figure numbers are given in Tables 2.1 and 2.2.
Table 4 - (Continued)

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition</th>
<th>3 - Nonassociated</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Loaded Unloaded</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Decrease</td>
<td>Increase Decrease</td>
<td>Increase Decrease</td>
</tr>
<tr>
<td>3</td>
<td>Inside Equator</td>
<td>Middle Equator</td>
<td>Middle Equator</td>
</tr>
<tr>
<td>4</td>
<td>Less</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Inside Equator</td>
<td>Middle Equator</td>
<td>Middle Equator</td>
</tr>
<tr>
<td>6</td>
<td>Lower</td>
<td>Lower</td>
<td>Basis</td>
</tr>
</tbody>
</table>
Table 4 - (Continued)

(b) $\varepsilon_z$

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition</th>
<th>l.a - Simple Plastic</th>
<th>2 - Quadratic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Outside Corner</td>
<td>Outside Corner &amp; Middle Equator</td>
<td>Outside Corner</td>
</tr>
<tr>
<td>6</td>
<td>Basis</td>
<td>Higher</td>
<td>Narrower</td>
</tr>
</tbody>
</table>
Table 4 - (Continued)

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition</th>
<th>3 - Nonassociated</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Less</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Middle</td>
<td>Middle</td>
<td>Equator</td>
</tr>
<tr>
<td></td>
<td>Equator</td>
<td>Equator</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Wider</td>
<td>Higher</td>
<td>Basis</td>
</tr>
</tbody>
</table>
Table 4 - (Continued)

(c) $\epsilon_\theta$

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition</th>
<th>1.a - Simple Plastic</th>
<th>2 - Quadratic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Basis</td>
<td>Higher</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition</th>
<th>3 - Nonassociated</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Narrower</td>
<td>Narrower</td>
<td></td>
</tr>
<tr>
<td>Case</td>
<td>1.a - Simple Plastic</td>
<td>2 - Quadratic</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
<td>Loaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Middle</td>
<td>Middle</td>
<td>Inside</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Middle</td>
<td>Middle</td>
<td>Outside Corner Corner</td>
</tr>
<tr>
<td>6</td>
<td>Basis</td>
<td>Lower</td>
<td>Wider</td>
</tr>
</tbody>
</table>
Table 4 - (Continued)

<table>
<thead>
<tr>
<th>Case</th>
<th>3 - Nonassociated</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Middle Inside</td>
<td>Middle</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Middle Inside</td>
<td>Middle</td>
</tr>
<tr>
<td>6</td>
<td>Narrower Low</td>
<td>Twice as</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low</td>
</tr>
</tbody>
</table>
Table 4 - (Continued)

(e) $N_z$

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition</th>
<th>1.a - Simple Plastic</th>
<th>2 - Quadratic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Decrease</td>
<td>Decrease</td>
<td>Decrease</td>
</tr>
<tr>
<td>3</td>
<td>Inside</td>
<td>Inside</td>
<td>Inside</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Inside</td>
<td>Inside</td>
<td>Outside</td>
</tr>
<tr>
<td></td>
<td>Equator</td>
<td>Equator</td>
<td>Corner</td>
</tr>
<tr>
<td>6</td>
<td>Basis</td>
<td>Lower</td>
<td>Wider</td>
</tr>
</tbody>
</table>
Table 4 - (Continued)

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition 3 - Nonassociated</th>
<th>Experiment Unloaded</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Decrease</td>
<td>Decrease</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Inside</td>
<td>Inside</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Less</td>
</tr>
<tr>
<td>5</td>
<td>Inside</td>
<td>Inside</td>
</tr>
<tr>
<td></td>
<td>Equator</td>
<td>Equator</td>
</tr>
<tr>
<td>6</td>
<td>Lower</td>
<td>Twice as</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td></td>
</tr>
</tbody>
</table>
### Table 5 Comparison of Plots: 16%

(a) $\varepsilon_r$

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition</th>
<th>1.b - Simple Plastic</th>
<th>4 - Large Deformation</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
<td>Loaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Decrease</td>
<td>Decrease</td>
<td>Increase</td>
<td>Increase</td>
</tr>
<tr>
<td></td>
<td>Inside</td>
<td>Inside</td>
<td>Inside</td>
<td>Inside</td>
</tr>
<tr>
<td></td>
<td>Equator</td>
<td>Equator</td>
<td>Middle</td>
<td>Middle</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Inside</td>
<td>Inside</td>
<td>Outside</td>
<td>Outside</td>
</tr>
<tr>
<td></td>
<td>Equator &amp;</td>
<td>Equator &amp;</td>
<td>Corner</td>
<td>Corner</td>
</tr>
<tr>
<td></td>
<td>Outside</td>
<td>Outside</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Corner</td>
<td>Corner</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Basis</td>
<td>Lower</td>
<td>Higher</td>
<td>Lower</td>
</tr>
</tbody>
</table>
Table 5 - (Continued)

(b) $\varepsilon_z$

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition</th>
<th>1.b - Simple Plastic</th>
<th>4 - Large Deformation</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
<td>Loaded</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Outside</td>
<td>Outside</td>
<td>Outside</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corner</td>
<td>Corner</td>
<td>Corner</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Basis</td>
<td>Lower</td>
<td>Lower</td>
</tr>
</tbody>
</table>
Table 5 - (Continued)

(c) $\epsilon_0$

<table>
<thead>
<tr>
<th>Case</th>
<th>1.b - Simple Plastic</th>
<th>4 - Large Deformation</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
<td>Loaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Basis</td>
<td>Narrower</td>
<td>Wider</td>
</tr>
<tr>
<td>Case</td>
<td>Condition</td>
<td>1.b - Simple Plastic</td>
<td>4 - Large Deformation</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>---------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Middle</td>
<td>Middle Inside</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Middle Equator &amp; Outside Corner</td>
<td>Outer Corner</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Basis</td>
<td>Lower</td>
</tr>
</tbody>
</table>
Table 5 - (Continued)

(e) $N_z$

<table>
<thead>
<tr>
<th>Case</th>
<th>1.b - Simple Plastic</th>
<th>4 - Large Deformation</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Loaded</td>
<td>Unloaded</td>
<td>Loaded</td>
</tr>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Decrease</td>
<td>Decrease</td>
<td>Decrease</td>
</tr>
<tr>
<td>3</td>
<td>Inside</td>
<td>Inside</td>
<td>Inside</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Inside</td>
<td>Inside</td>
<td>Inside</td>
</tr>
<tr>
<td>6</td>
<td>Basis</td>
<td>Slightly\textsuperscript{a} Lower</td>
<td>Same</td>
</tr>
</tbody>
</table>

\textsuperscript{a}Or almost the same.
Table 6 Minimum and Maximum Radii at Equator (inch.)

<table>
<thead>
<tr>
<th></th>
<th>Finite Element</th>
<th></th>
<th>8%</th>
<th>16%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Loaded</td>
<td></td>
<td>Case 1.a</td>
<td>Case 2</td>
</tr>
<tr>
<td></td>
<td>Unloaded</td>
<td></td>
<td>29.364</td>
<td>29.367</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>61.34</td>
<td>61.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r_{\text{min}}</td>
<td>29.487</td>
<td>29.491</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r_{\text{max}}</td>
<td>60.955</td>
<td>60.842</td>
</tr>
<tr>
<td></td>
<td>Experiment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unloaded</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>r_{\text{min}}</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>r_{\text{max}}</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>% Decrease</td>
<td>% Deformation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>------------</td>
<td>---------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Finite Element</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loaded</td>
<td>0.8</td>
<td>4.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>6.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.12</td>
<td>8.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.06</td>
<td>5.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.12</td>
<td>8.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>4.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.21</td>
<td>6.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.9</td>
<td>8.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unloaded</td>
<td>1.71</td>
<td>4.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.7</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.42</td>
<td>2.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Experiment</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unloaded</td>
<td>3.2</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7 % Decrease of Inner Radius versus % of Deformation
Table 7 - (Continued)

<table>
<thead>
<tr>
<th></th>
<th>% Decrease</th>
<th>% Deformation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Finite Element</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loaded</td>
<td>2.1</td>
<td>4.40</td>
</tr>
<tr>
<td></td>
<td>2.6</td>
<td>5.80</td>
</tr>
<tr>
<td></td>
<td>8.7</td>
<td>16.0</td>
</tr>
<tr>
<td></td>
<td>8.7</td>
<td>16.0</td>
</tr>
<tr>
<td>Unloaded</td>
<td>3.94</td>
<td>5.20</td>
</tr>
<tr>
<td></td>
<td>11.0</td>
<td>11.0</td>
</tr>
<tr>
<td><strong>Experiment</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unloaded</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11.0</td>
<td></td>
</tr>
</tbody>
</table>
IX. BIBLIOGRAPHY


X. ACKNOWLEDGMENTS

The author is grateful to Dr. C. P. Burger for the many helpful suggestions, encouragement and counseling he provided throughout the graduate program.

Thanks are also due to Drs. L. W. Schmerr, T. J. Ruldolphi, T. H. Okiishi and S. Bahadur for serving on the graduate committee.

The author thanks Dr. H. J. Weiss for providing support in teaching, research assistantships and computer time which make this dissertation possible.

Last but not least, the author thanks his wife Huey-Jiun Yu Lee for the tremendous support throughout the graduate study.
XI. APPENDIX A - DERIVATIONS AND MATRICES

A. Matrices for Plasticity

1. Elastic stress-strain relationship

   (i). General formula

   \[
   \begin{bmatrix}
   \sigma_{11} \\
   \sigma_{22} \\
   \sigma_{33} \\
   \sigma_{12}
   \end{bmatrix} =
   \begin{bmatrix}
   D_1 & D_2 & D_2 & 0 \\
   D_2 & D_1 & D_2 & 0 \\
   D_2 & D_2 & D_1 & 0 \\
   0 & 0 & 0 & D_3
   \end{bmatrix}
   \begin{bmatrix}
   \varepsilon_{11} \\
   \varepsilon_{22} \\
   \varepsilon_{33} \\
   \varepsilon_{12}
   \end{bmatrix}
   \]

   (ii). Axisymmetric

   \[
   \begin{bmatrix}
   \sigma_r \\
   \sigma_z \\
   \sigma_\theta \\
   \tau_{rz}
   \end{bmatrix} = \frac{E(1-v)}{(1+v)(1-2v)}
   \begin{bmatrix}
   1 & \frac{v}{1-v} & \frac{v}{1-v} & 0 \\
   1 & \frac{v}{1-v} & 0 & 0 \\
   S & 1 & 0 & 0 \\
   M & \frac{1-2v}{2(1-v)} & \gamma_{rz}
   \end{bmatrix}
   \begin{bmatrix}
   \varepsilon_r \\
   \varepsilon_z \\
   \varepsilon_\theta \\
   \gamma_{rz}
   \end{bmatrix}
   \]

   \[D_1 = \frac{E(1-v)}{(1+v)(1-2v)}, \quad D_2 = \frac{v}{1-v} D_1, \quad D_3 = G = \frac{E}{2(1+v)}\]

Terms set to zero for plane stress in practice.
(iii). Plane Strain

\[
\begin{align*}
\begin{bmatrix}
\sigma_x \\
\sigma_y \\
\sigma_z \\
\tau_{xy}
\end{bmatrix}
= \begin{bmatrix}
\epsilon_x \\
\epsilon_y \\
0 \\
\gamma_{xy}
\end{bmatrix}
\end{align*}
\]

same as (ii)

(iv). Plane Stress

\[
\begin{align*}
\begin{bmatrix}
\sigma_x \\
\sigma_y \\
0 \\
\tau_{xy}
\end{bmatrix}
= \frac{E}{(1-\nu^2)} \begin{bmatrix}
1 & \nu & \nu & 0 \\
\nu & 1 & \nu & 0 \\
\nu & \nu & 1 & 0 \\
0 & 0 & 0 & \frac{1-\nu}{2}
\end{bmatrix}
\begin{bmatrix}
\epsilon_x \\
\epsilon_y \\
\epsilon_z \\
\gamma_{xy}
\end{bmatrix}
\end{align*}
\]

\[D_1 = \frac{E}{(1-\nu^2)}, \quad D_2 = D_1\nu, \quad D_3 = G = \frac{E}{2(1+\nu)}\]
2. **Strain-displacement relationship**

   (i). General form

\[
\begin{align*}
\{\varepsilon\} &= [B]^{4 \times 2} \{\varepsilon\}^{2 \times 1} \\
&= \sum_{i=1}^{\text{NEL}} [B]_i \{\varepsilon\}_i
\end{align*}
\]

Where

\[
[B]_i = \begin{bmatrix}
    B_1 & 0 \\
    0 & B_2 \\
    B_3 & 0 \\
    B_2 & B_1
\end{bmatrix}
\]

(ii). Axisymmetric

\[
\begin{align*}
\begin{bmatrix}
    \varepsilon_r \\
    \varepsilon_z \\
    \varepsilon_\theta \\
    \gamma_{rz}
\end{bmatrix}
&= \begin{bmatrix}
    \frac{\partial u}{\partial r} \\
    \frac{\partial v}{\partial z} \\
    \frac{u}{r} \\
    \frac{\partial u}{\partial z} + \frac{v}{r}
\end{bmatrix}
\begin{bmatrix}
    N_i,r & 0 \\
    0 & N_i,z \\
    N_i^2 & 0 \\
    N_i,z & N_i,r
\end{bmatrix}
\begin{bmatrix}
    u \\
    v
\end{bmatrix}
\end{align*}
\]

Terms set to \( \frac{N_i}{r} \) for axisymmetric case only.
(iii). Plane strain

\[
\begin{bmatrix}
\varepsilon_x \\
\varepsilon_y \\
0 \\
\gamma_{xy}
\end{bmatrix}
= \begin{bmatrix}
\frac{\partial u}{\partial x} \\
\frac{\partial v}{\partial y} \\
0 \\
\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}
\end{bmatrix}
\begin{bmatrix}
N_{i,x} \\
0 \\
0 \\
N_{i,y}
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
\]

(iv). Plane stress

Same as above.

\[
\varepsilon_z = -\frac{v}{E} (\sigma_x + \sigma_y) \quad \text{(not recorded)}
\]

Since \( \sigma_z \varepsilon_z = 0 \), it does not contribute to strain energy, hence no need to record \( \varepsilon_z \).

Where \( i = 1 \) to NEL, NEL is number of nodes/ele.
3. **Stiffness matrix for an element**

\[
[K]_{2\times \text{NEL}, 2\times \text{NEL}} = \int_V [B]^T [D] [B] \, dv
\]

\[
= (\sum_{i=1}^{\text{NEL}} \sum_{j=1}^{\text{NEL}} [K_{ij}]_{2\times 2}) \, v
\]

Where

\[
v = \text{volume of an element}
\]

\[
dv = 2\pi r \, dr \, dz \quad \text{(axisymmetric)}
\]

\[
dv = dx \, dy \times \text{thickness} \quad \text{(plane stress, plane strain)}
\]

And

\[
[K_{ij}]_{2\times 2} = [B_i]^T [D] [B_j] = [B_i]^T [Q_j]
\]

Where \([Q_j] = [D] [B_j]\)

\[
= \begin{bmatrix}
D_1 N_{j,x} + D_2 \frac{N_j}{r} & D_2 N_{j,y} \\
D_2 (N_{j,x} + \frac{N_j}{r}) & D_1 N_{j,y} \\
D_2 N_{j,x} + D_1 \frac{N_j}{r} & D_2 N_{j,y} \\
D_3 N_{j,y} & D_3 N_{j,x}
\end{bmatrix}
\]
\[
\begin{bmatrix}
Q_{11} & Q_{12} \\
Q_{21} & Q_{22} \\
Q_{31} & Q_{32} \\
Q_{41} & Q_{42}
\end{bmatrix}
\]

Hence

\[
[k_{ij}]_{2 \times 2} = \\
\begin{bmatrix}
N_{i,x}Q_{11} + \frac{N_{i,y}Q_{31}}{r} + N_{i,y}Q_{41} & N_{i,x}Q_{12} + \frac{N_{i,x}Q_{32}}{r} + N_{i,y}Q_{42} \\
N_{i,x}Q_{21} + N_{i,x}Q_{41} & N_{i,y}Q_{22} + N_{i,x}Q_{42}
\end{bmatrix}
\]

Where

\[
\frac{N_{i}}{r}
\]
is set to zero for plane stress and plane strain

\[x = r, y = z\] for axisymmetric case
4. **Plastic matrix**

\[
[D_P] = \begin{bmatrix}
D_{11} & D_{12} & D_{13} & D_{14} \\
D_{22} & D_{23} & D_{24} & \\
D_{33} & D_{34} & \\
& & & D_{44}
\end{bmatrix} \text{ (constant)}
\]

Where

\[
D_{11} = \sigma_1'^2, \quad D_{12} = \sigma_1'^2, \quad D_{13} = \sigma_1'^2, \quad D_{14} = \sigma_1'^2
\]

\[
D_{22} = \sigma_2'^2, \quad D_{23} = \sigma_2'^2, \quad D_{24} = \sigma_2'^2
\]

\[
D_{33} = \sigma_3'^2, \quad D_{34} = \sigma_3'^2
\]

\[
D_{44} = \sigma_4'^2
\]

And prime (') indicates deviatoric stress (\(\sigma'_i = \sigma_i\)).

For plane stress and plane strain, the third row and third column are set to zero.

\[
\text{constant} = \frac{3G}{\frac{2}{\sigma} \left( \frac{H}{3G} + 1 \right)}
\]

Also for plane stress, set \(\sigma'_3 = -\sigma'_1 - \sigma'_2\)
5. Formula of $r$ for transition zone

$$r = \frac{\Gamma + \sqrt{\Gamma^2 + 4(\Delta \sigma_{ij})^2}}{2(\sigma_{ij})^2}$$

Where

- $\bar{\sigma}_i$ = equivalent stress at the end of the present increment
- $\bar{\sigma}_{i-1}$ = OP
- $\Delta \sigma_{ij}$ = equivalent stress at the start of the present increment
- $\Delta \sigma_{ij} = \frac{1}{\sqrt{2}} \left\{ (\Delta \sigma_1 - \Delta \sigma_2)^2 + (\Delta \sigma_2 - \Delta \sigma_3)^2 + (\Delta \sigma_3 - \Delta \sigma_1)^2 + 6\Delta \sigma_{12}^2 + 6\Delta \sigma_{23}^2 + 6\Delta \sigma_{31}^2 \right\}^{\frac{1}{2}}$
  = equivalent stress increment due to the present incremental stress for von Mises yield criterion
- $\Delta \bar{\sigma} = \bar{\sigma}_i - \sigma_{i-1}$ = PS
- $\Gamma = \frac{2}{\sigma_{ij}} - 2\bar{\sigma}_{i-1} \Delta \bar{\sigma} - \Delta \bar{\sigma}$
  = initial yield stress
- $\sigma_{i-1}$ = OP
- $\sigma_{i-1}$ = OR = OP + PS
- $\sigma_{i-1} = \bar{\sigma}_{i-1} + \Delta \bar{\sigma}$

See Figure A.1 for explanation of symbols.
Figure A.1 Explanation of symbols in transition zone
B. Matrices for Large Deformation

1. Geometrically nonlinear stiffness matrix - axisymmetric

\[
[K^G_{ij}] = [B_{e_i}]^T [D^*_G] [B_{e_j}] = [B_{e_i}]^T [Q_j]
\]

Where

\[
[Q]_j = [D^*_G] [B_{e_j}]
\]

And \([B_{e_i}]\) is defined as follows:

\[
\{\dot{e}\} = \begin{bmatrix} \dot{e}_{11} \\ \dot{e}_{12} \\ \dot{e}_{21} \\ \dot{e}_{22} \\ \dot{e}_{33} \end{bmatrix} = \begin{bmatrix} \dot{e}_{rr} \\ \dot{e}_{rz} \\ \dot{e}_{zr} \\ \dot{e}_{zz} \end{bmatrix} = [B_{e_i}] \{\dot{u}\}
\]

\[
= \begin{bmatrix} N_{i,r} & 0 \\ N_{i,z} & 0 \\ 0 & N_{i,r} \\ 0 & N_{i,z} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix}
\]
Where

\[ [D^*] = \]

\[
\begin{bmatrix}
-\sigma_{11} & -\sigma_{12} & 0 & 0 & 0 \\
-\sigma_{12} & -\frac{1}{2}(\sigma_{11} - \sigma_{22}) & -\frac{1}{2}(\sigma_{11} + \sigma_{22}) & 0 & 0 \\
0 & -\frac{1}{2}(\sigma_{11} + \sigma_{22}) & \frac{1}{2}(\sigma_{22} - \sigma_{11}) & -\sigma_{12} & 0 \\
0 & 0 & -\sigma_{12} & -\sigma_{22} & 0 \\
0 & 0 & 0 & 0 & -\sigma_{23}
\end{bmatrix}
\]

And so

\[
[Q] = \begin{bmatrix}
Q_{11} & Q_{12} \\
Q_{21} & Q_{22} \\
Q_{31} & Q_{32} \\
Q_{41} & Q_{42} \\
Q_{51} & Q_{52}
\end{bmatrix}
\]

Where

\[ Q_{11} = -\sigma_{11} N_{j,x} - \sigma_{12} N_{j,y} \]
\[ Q_{12} = 0 \]
\[ Q_{21} = -\sigma_{12} N_{j,x} + \frac{1}{2}(\sigma_{11} - \sigma_{22}) N_{j,y} \]
\begin{align*}
Q_{22} &= -\frac{1}{2}(\sigma_{11} + \sigma_{22})N_j, x \\
Q_{31} &= -\frac{1}{2}(\sigma_{11} + \sigma_{22})N_j, y \\
Q_{32} &= \frac{1}{2}(\sigma_{22} - \sigma_{11})N_j, x - \sigma_{12}N_j, y \\
Q_{41} &= 0 \\
Q_{42} &= -\sigma_{12}N_j, x - \sigma_{22}N_j, y \\
Q_{51} &= -\sigma_{33}N_j
\end{align*}

Hence,

\begin{align*}
[K_{ij}^G] &= [B_{e_i}]^T [Q]_j = \\
&= \begin{bmatrix}
N_i, x Q_{11} + N_i, y Q_{21} + \frac{N_i Q_{51}}{r} & N_i, y Q_{22} \\
N_i, x Q_{31} & N_i, x Q_{32} + N_i, y Q_{42}
\end{bmatrix}
\end{align*}

And \([K^G] = \int_v [K_{ij}^G] \, dv\)

2. Equivalent load - axisymmetric

\begin{align*}
\{\Delta f\} &= - \left( \int_v [B_e]^T [D^*_G] [B_e] \, dv \right) \{\Delta u\}
\end{align*}

\begin{align*}
\text{NELNEL} \\
&= \sum_{i=1}^{2x1} \sum_{j=1}^{2x1} \{\Delta f\}_{ij}
\end{align*}
Where

\[
\{\Delta f\}_{ij} = - \int_{\Omega} \left[ B_e \right]_i^T \left[ D^*_G \right] \left[ B_e \right]_j \, dv \{\Delta u\}
\]

\[
= - \int_{\Omega} \left[ B_e \right]_i^T \left[ \Omega \right] \_j \, dv
\]

And \( \left[ \Omega \right] \_j = [Q] \_j \{\Delta u\} \)

\[
\begin{pmatrix}
Q_1'
Q_2'
Q_3'
Q_4'
Q_5'
\end{pmatrix}
= \begin{pmatrix}
Q_{11} \cdot \dot{u}
Q_{21} \cdot \dot{u} + Q_{22} \cdot \dot{v}
Q_{31} \cdot \dot{u} + Q_{32} \cdot \dot{v}
Q_{42} \cdot \dot{v}
Q_{51} \cdot \dot{u}
\end{pmatrix}
\]

Hence,

\[
\{\Delta f\}_{ij} = - \begin{pmatrix}
N_{i,x} Q_1' + N_{i,y} Q_2' + \frac{N_i}{r} Q_5'
n_i, x Q_3' + N_{i,y} Q_4'
\end{pmatrix} \, dv
\]

3. **Transformation from Jaumann stress rate to Cauchy stress rate**

\[
\{t\} = \{\dot{s}\} - \left[ \dot{D}^*_G \right] \{\epsilon\}
\]

\[
\{\dot{\sigma}\} = \{\dot{s}\} - \left[ \dot{D}^*_G \right] \{\dot{\epsilon}\}
\]
Hence,
\[
\{\ddot{s}\} = \{\ddot{t}\} + ([D^*_G] - [D^*_G]) \{\hat{e}\}
\]

Where \([D^*_G] - [D^*_G]\) is given by Yamada [31].

Now let \(\omega_{12} = \kappa (\dot{e}_{12} - \dot{e}_{21})\)

Then,
\[
\begin{align*}
\Delta \sigma_{11} &= \Delta t_{11} - 2\omega_{12}\sigma_{12} \\
\Delta \sigma_{12} &= \Delta t^*_{12} + \sigma_{11} \omega_{12} - \sigma_{22} \omega_{12} \\
\Delta \sigma_{21} &= \Delta \sigma_{12} \\
\Delta \sigma_{22} &= 2\Delta t^*_{22} \omega_{12} \\
\Delta \sigma_{33} &= \tau_{33}
\end{align*}
\]

Where
\[
\begin{align*}
1 &+ r \\
2 &+ z \\
3 &+ \theta
\end{align*}
\]
C. Plastic Matrix for Modified Von Mises Criterion

\[
[D]^P = \frac{\partial Q}{\partial \sigma} \cdot \frac{\partial F^T}{\partial \sigma} + \frac{\partial F}{\partial \sigma} \cdot \frac{\partial Q}{\partial \sigma} \cdot [D] \cdot \frac{\partial [D]}{\partial \sigma}
\]

Where

\[
A = \frac{\partial F}{\partial \sigma} \cdot \frac{1}{\lambda}
\]

\[
Q = \left[ \frac{1}{2} (\sigma_x - \sigma_y)^2 + \frac{1}{2} (\sigma_y - \sigma_z)^2 + \frac{1}{2} (\sigma_z - \sigma_x)^2 + 3\tau_{xy}^2 + 3\tau_{yz}^2 + 3\tau_{xz}^2 \right]^\frac{1}{2}
\]

\[
F = Q - (CT - (\sigma_x + \sigma_y + \sigma_z)(C - T))^\frac{1}{2}
\]

Now let us evaluate term by term:

\[
\left\{ \frac{\partial F}{\partial \sigma} \right\} = \left\{ \frac{\partial F}{\partial \sigma} \right\} + \left\{ \frac{\partial F}{\partial \sigma} \right\}
\]

And

\[
\frac{\partial F}{\partial J_1} = \frac{C - T}{2\sqrt{CT - (\sigma_x + \sigma_y + \sigma_z)(C - T)}}
\]

\[
\left\{ \frac{\partial J_1}{\partial \sigma} \right\} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}
\]
\[
\begin{bmatrix}
3\sigma' \\
3\sigma' \\
3\sigma' \\
6\tau & xy \\
6\tau & yz \\
6\tau & zx
\end{bmatrix}
\]

\[
\frac{\partial F}{\partial J_2} = \frac{1}{2\sqrt{J_2}} = \frac{1}{2\sigma}
\]

So

\[
\begin{bmatrix}
1 \\
1 \\
1 \\
1 \\
0 \\
0
\end{bmatrix}
\]

\[
= C_1 \begin{bmatrix}
1 \\
1 \\
1 \\
1 \\
0 \\
0
\end{bmatrix} + \begin{bmatrix}
\sigma' \\
\sigma' \\
\sigma' \\
6\tau & xy \\
6\tau & yz \\
6\tau & zx
\end{bmatrix}
\]

Where

\[
\begin{bmatrix}
\sigma' \\
\sigma' \\
\sigma' \\
6\tau & xy \\
6\tau & yz \\
6\tau & zx
\end{bmatrix} = \frac{3\sigma' - \frac{3\partial J_2}{\partial \sigma}}{2\sigma} = \frac{3}{2\sigma} \begin{bmatrix}
\sigma' \\
\sigma' \\
\sigma' \\
6\tau & xy \\
6\tau & yz \\
6\tau & zx
\end{bmatrix}
\]

And

\[
C_1 = \frac{C - T}{2\sqrt{C-T} - (\sigma_x + \sigma_y + \sigma_z)(C - T)}
\]
\[ \frac{\partial F^T}{\partial \sigma} [D] = \frac{3D_3}{\sigma} \{ \sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx} \} \]

+ \[ C_1(D_1 + 2D_2) \right] (C - T) \{1, 1, 1, 0, 0, 0\} \]

And

\[ [D] \left\{ \frac{\partial \Omega}{\partial \sigma} \right\} = \frac{\partial \Omega^T}{\partial \sigma} [D] = \frac{3D_3}{\sigma} \left\{ \begin{array}{c} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{array} \right\} \]

Since \([D]^T = [D]\).

Hence,

\[ \{ \frac{\partial F}{\partial \sigma} \}^T [D] \left\{ \frac{\partial \Omega}{\partial \sigma} \right\} = \frac{9D_3}{\sigma} \left( \frac{1}{2} (\sigma_x^2 + \sigma_y^2 + \sigma_z^2) + \tau_{xy}^2 + \tau_{yx}^2 + \tau_{zx}^2 \right) \]

\[ = 3D_3 \]

Where the relationships

\[ \sigma_x' + \sigma_y' + \sigma_z' = 0 \]
and
\[
\overline{\sigma} = \sqrt{3} \left[ \frac{1}{2} (\sigma_x^2 + \sigma_y^2 + \sigma_z^2) + \tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2 \right]^{\frac{1}{2}}
\]

have been used.

Hence,
\[
[D] \left\{ \frac{\partial \Omega}{\partial \sigma} \right\}^T \frac{\partial F}{\partial \sigma} [D] =
\]

\[
\begin{bmatrix}
\sigma_x^2 & \sigma_x' \sigma_y' & \sigma_x' \sigma_z' & \sigma_x' \tau_{xy} & \sigma_x' \tau_{yz} & \sigma_x' \tau_{zx} \\
\sigma_y' \sigma_x' & \sigma_y^2 & \sigma_y' \sigma_z' & \sigma_y' \tau_{xy} & \sigma_y' \tau_{yz} & \sigma_y' \tau_{zx} \\
\sigma_z' \sigma_x' & \sigma_z' \sigma_y' & \sigma_z^2 & \sigma_z' \tau_{xy} & \sigma_z' \tau_{yz} & \sigma_z' \tau_{zx} \\
\frac{9D^2}{3} S & \tau_{xy} & \tau_{xy} & \tau_{xy}^2 & \tau_{yz} & \tau_{yz} \\
\frac{9D^2}{3} M & \tau_{zx} & \tau_{zx} & \tau_{zx} & \tau_{zx}^2
\end{bmatrix}
\]

\[
+ \frac{3(D_1 + 2D_2)(C - T)D_3 C_1}{\overline{\sigma}} [D_{MV}]
\]

Where
\[
[D_{MV}] =
\]
Now we are ready to evaluate $A$.

$$A = - \frac{\partial F}{\partial k} \mathbf{d}k \frac{1}{\Lambda} = - \frac{\partial F}{\partial k} \{\sigma\}^T \frac{\partial \Omega}{\partial \sigma}$$

where $\{\sigma\}^T = \{\sigma_x \sigma_y \sigma_z \tau_{xy} \tau_{yz} \tau_{zx}\}$.

We can express

$$F = f(J_2) - \sigma_y (J_1, k)$$

$$\mathbf{d}k = \sigma_y \mathbf{d}\varepsilon_{up} = \sqrt{CT - (\sigma_x + \sigma_y + \sigma_z)(C - T)} \mathbf{d}\varepsilon_{up}$$

Where

$\mathbf{d}k =$ plastic strain energy density in uniaxial loading

$\mathbf{d}\varepsilon_{up} =$ uniaxial plastic strain increment

$\sigma_y =$ flow stress
So

\[ \frac{\partial F}{\partial k} = \frac{\partial \sigma_y}{\partial k} \frac{(J_1, k)}{\partial k} = \frac{\partial \sigma_y}{\partial \varepsilon_{up}} \frac{\partial \varepsilon_{up}}{\partial k} = \frac{\partial \sigma_y}{\partial \varepsilon_{up}} \frac{1}{\sigma_y} = \frac{H'}{\sigma_y} \]

Where \( H' \) is the slope of flow stress versus plastic strain increment as seen in Figure A.2.

But \( \{\sigma\}^T \{\sigma_{eq}\} = \sigma \), so

\[ A = \frac{H'}{\sigma_y} \sigma = H' \], since upon yielding \( \sigma = \sigma_y (F = 0) \)

Here we have assumed that \( H' \) in compression is the same as \( H' \) in tension as shown in Figure A.2 (b).

Collecting terms one gets:

\[ [D_P] = [D_P]^* + \frac{3C_1(D_1 + 2D_2)D_3}{(H' + 3G)\sigma} [D_{MV}] \]

Where \([D_P]^*\) is the plastic matrix for pressure independent material (von Mises yield criterion).
Figure A.2 A comparison of plastic work density $d_k$ of pressure independent (a) and pressure dependent (b) process (cross-hatched area)
### A. List and Explanation of New Macros and Input

#### 1. List of new macro programming commands

<table>
<thead>
<tr>
<th>Columns 1 to 4</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCE</td>
<td>Acceleration of displacement convergence for initial stress method only. (opt.)</td>
</tr>
<tr>
<td>BCNG</td>
<td>Enforce boundary condition change during execution. (opt.)</td>
</tr>
<tr>
<td>FCNG</td>
<td>Test force convergence for initial stress only (must have for the initial stress method).</td>
</tr>
<tr>
<td>INCR</td>
<td>Enable incremental (tangential stiffness) method. (must have for incremental analysis).</td>
</tr>
<tr>
<td>LOCA</td>
<td>Set memory location and input for locally defined axis.</td>
</tr>
<tr>
<td>NONL</td>
<td>Set memory location for nonlinear analysis - plasticity and/or large deformation.</td>
</tr>
<tr>
<td>PLAS</td>
<td>Execute algorithm for plasticity analysis.</td>
</tr>
<tr>
<td>POST</td>
<td>Write postprocessing data on Tape30.</td>
</tr>
<tr>
<td>UNLD</td>
<td>Perform mandatory unloading for plasticity analysis.</td>
</tr>
<tr>
<td>ROLL</td>
<td>Input geometry information and set CUVCH logical flag for modeling rolling.</td>
</tr>
<tr>
<td>UPCO</td>
<td>Update coordinate (can use with or without curvature correction).</td>
</tr>
<tr>
<td>ROBC</td>
<td>Set boundary condition change for rolling analysis.</td>
</tr>
<tr>
<td>WYLB</td>
<td>Used with POST, all post processing data are written as 'formatted' on Tape31.</td>
</tr>
<tr>
<td>UPLR</td>
<td>Perform Updated Lagrangian (large deformation) analysis.</td>
</tr>
</tbody>
</table>
2. Necessary order of macros

a. If POST is used, place it before NONL.
b. If BCNG is used, place it before NONL.
c. WYLB needs to be placed before POST.
d. Always have a LOOP, NEXT pair.
e. If stress output is needed for UNLD operation, must use STRE after SOLV.
f. If we need to find the percent of deformation, put DISP before STRE.

3. Input for boundary condition change cards (BCNG macro)

a. First Card (Format 16I5)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 5</td>
<td>NBN</td>
<td>Maximum number of nodes that has the possibility of boundary condition change.</td>
</tr>
</tbody>
</table>

b. Second Card - input boundary condition. Format (16I5)

<table>
<thead>
<tr>
<th>Columns</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 5</td>
<td>N</td>
<td>Boundary node number</td>
</tr>
<tr>
<td>6 to 10</td>
<td>NG</td>
<td>Auto increment indicator and value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii. NG &gt; 1 Starting auto increment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iii. NG &lt; 0 Terminating auto increment.</td>
</tr>
<tr>
<td>11 to 15</td>
<td>IDF(1)</td>
<td>1 = change needed.</td>
</tr>
<tr>
<td>16 to 20</td>
<td>IDF(2)</td>
<td>Degree of freedom of node N whose</td>
</tr>
<tr>
<td>21 to 25</td>
<td>IDF(3)</td>
<td>boundary condition is to be changed.</td>
</tr>
</tbody>
</table>

c. Terminating input with a blank card.

4. Modification of input for ELMt01

a. First Card (Format = 3F10.0, 3I5, F5.0, 2F10.0)
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 10</td>
<td>E</td>
</tr>
<tr>
<td>11 to 20</td>
<td>v</td>
</tr>
<tr>
<td>21 to 30</td>
<td>ρ</td>
</tr>
<tr>
<td>31 to 35</td>
<td>L</td>
</tr>
<tr>
<td>36 to 40</td>
<td>K</td>
</tr>
<tr>
<td>41 to 45</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>= 0</td>
</tr>
<tr>
<td></td>
<td>&gt; 0</td>
</tr>
<tr>
<td>46 to 50</td>
<td>D(8)</td>
</tr>
<tr>
<td></td>
<td>&lt; 0</td>
</tr>
<tr>
<td>51 to 60</td>
<td>D(9)</td>
</tr>
<tr>
<td>61 to 70</td>
<td>D(10)</td>
</tr>
</tbody>
</table>

b. Second Card

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 10</td>
<td>( % ) 0</td>
</tr>
<tr>
<td></td>
<td>= 0</td>
</tr>
<tr>
<td>11 to 20</td>
<td>C-T</td>
</tr>
</tbody>
</table>

B. Actual Input Data and Necessary JCL for Running

1. **Job control language (JCL) used for running MINIFE**
//D439LEE JOB 13512, JONAH
/*KEY XX
/*JOBPARM CARDS=99999, LINES=99999
//S1 EXEC PGM=MINIFEM, REGION=256K, TIME=(60,)
/*STEPLIB DD DSN=J.13512.LOADMOD, DISP=SHR
//FT06F001 DD SYSOUT=A, DCB=(RECFM=FBA, LRECL=133, BLKSIZE=133)
//FT11F001 DD DSN=&TEMP1, UNIT=SCRTCH,
  // SPACE=(TRK, (20, 10)), DCB=(RECFM=VBS, BLKSIZE=804, LRECL=50),
  // DISP=(NEW, PASS)
//FT12F001 DD DSN=&TEMP2, UNIT=SCRTCH,
  // SPACE=(TRK, (20, 10)), DCB=(RECFM=VBS, LRECL=1596, BLKSIZE=1600),
  // DISP=(NEW, PASS)
//FT14F001 DD DSN=&TEMP3, UNIT=SCRTCH,
  // SPACE=(TRK, (20, 10)), DCB=(RECFM=VBS, BLKSIZE=804, LRECL=100),
  // DISP=(NEW, PASS)
//FT16F001 DD DSN=&TEMP4, UNIT=SCRTCH,
  // SPACE=(TRK, (20, 10)), DCB=(RECFM=VBS, BLKSIZE=804, LRECL=X),
  // DISP=(NEW, PASS)
//FT18F001 DD DSN=&TEMP5, UNIT=SCRTCH,
  // SPACE=(TRK, (20, 10)), DCB=(RECFM=VBS, BLKSIZE=804, LRECL=X),
  // DISP=(NEW, PASS)
//FT30F001 DD SYSOUT=A
//FT31F001 DD SYSOUT=B
//FT05F001 DD *
2. Input data for case 1.a
FEAP
RING OF 6:3:2 LAMINAC 42 ELEMENTS 56 NODES (CASE 1.a)

56
42 2 2 2 4 0

COOR

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>30.0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>60.0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>30.0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>60.0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>30.0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>60.0</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>30.0</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>60.0</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>30.0</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>60.0</td>
</tr>
<tr>
<td>36</td>
<td>1</td>
<td>30.0</td>
</tr>
<tr>
<td>42</td>
<td>0</td>
<td>60.0</td>
</tr>
<tr>
<td>43</td>
<td>1</td>
<td>30.0</td>
</tr>
<tr>
<td>49</td>
<td>0</td>
<td>60.0</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>30.0</td>
</tr>
<tr>
<td>56</td>
<td>0</td>
<td>60.0</td>
</tr>
</tbody>
</table>

ELEM

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>8</td>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>13</td>
<td>14</td>
<td>21</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>15</td>
<td>16</td>
<td>23</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>20</td>
<td>21</td>
<td>28</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>22</td>
<td>23</td>
<td>30</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>27</td>
<td>28</td>
<td>35</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>29</td>
<td>30</td>
<td>37</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>34</td>
<td>35</td>
<td>42</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>36</td>
<td>37</td>
<td>44</td>
</tr>
</tbody>
</table>
MATE
  1  1  NONLINEAR PLASTIC
  1200.0  0.30  2  2  +1  -1  84.0  41.0
  2  1  INTERFACE PROPERTY
  1200000.0  0.30  2  2  +1  +1  84000.0  41000.0
BOUND
  1  1  0  -1
  7  0  0  1
  43  1  -1  -1
  49  0  1  1
  50  1  -1
  56  0  1
FORC
  43  1  -0.6
  49  0  -0.6
END
MACR
DT  1.0
PROP  1.0
POST
BCNG
NONL
TANG
LOOP  42
TIME
LOOP  40
FORM
FCNV
SOLV
PLAS
NEXT
DISP 14
STRE 14
NEXT
UNLD
TANG
FORM
SOLV
DISP
STRE
END

1 1 1.01 40.0 0.04
7
43 1 1 1
49 -1 1 1

STOP
3. Input data for case 1.b
FEAP | RING OF 6:3:2 LAMINAC 42 ELEMENTS 56 NODES (CASE 1.b)

| 56 | 42 | 2 | 2 | 2 | 4 | 0 |

**COOR**

| 1  | 1  | 30.0 |
| 7  | 0  | 60.0 |
| 8  | 1  | 30.0 | 1.0 |
| 14 | 0  | 60.0 | 1.0 |
| 15 | 1  | 30.0 | 3.0 |
| 21 | 0  | 60.0 | 3.0 |
| 22 | 1  | 30.0 | 6.0 |
| 28 | 0  | 60.0 | 6.0 |
| 29 | 1  | 30.0 | 10.0 |
| 35 | 0  | 60.0 | 10.0 |
| 36 | 1  | 30.0 | 16.0 |
| 42 | 0  | 60.0 | 16.0 |
| 43 | 1  | 30.0 | 20.0 |
| 49 | 0  | 60.0 | 20.0 |
| 50 | 1  | 30.0 | 23.0 |
| 56 | 0  | 60.0 | 23.0 |

**ELEM**

| 1  | 1  | 1  | 2  | 9  | 8  | 1  |
| 6  | 1  | 6  | 7  | 14 | 13 |    |
| 7  | 1  | 8  | 9  | 16 | 15 | 1  |
| 12 | 1  | 13 | 14 | 21 | 20 |    |
| 13 | 1  | 15 | 16 | 23 | 22 | 1  |
| 18 | 1  | 20 | 21 | 28 | 27 |    |
| 19 | 1  | 22 | 23 | 30 | 29 | 1  |
| 24 | 1  | 27 | 28 | 35 | 34 |    |
| 25 | 1  | 29 | 30 | 37 | 36 | 1  |
| 30 | 1  | 34 | 35 | 42 | 41 |    |
| 31 | 1  | 36 | 37 | 44 | 43 | 1  |
| 36 | 1  | 41 | 42 | 49 | 48 |    |
MATE
1 1 NONLINEAR PLASTIC
1200.0 0.30 2 2 +1 -1 84.0 41.0
2 1 INTERFACE PROPERTY
1200000.0 0.30 2 2 +1 +1 84000.0 41000.0

BOUND
1 1 0 -1
7 0 0 1
43 1 -1 -1
49 0 1 1
50 1 -1
56 0 1

FORC
43 1 -0.6
49 0 -0.6

END
MACR
DT 1.0
PROP 1.0
POST
BCNG
NONL
ACCE
TANG
LOOP 43
TIME
LOOP 30
FORM
FCNV
SOLV
PLAS
NEXT
DISP  20
STRE  20
NEXT
DISP
STRE
REAC
UNLD
TANG
FORM
SOLV
DISP
STRE
END

\[ \begin{array}{cccc}
1 & 1 & 1.01 & 45.0 & 0.10 \\
7 & & & & \\
43 & 1 & 1 & 1 \\
49 & -1 & 1 & 1 \\
\end{array} \]

STOP
4. Input data for case 2
<table>
<thead>
<tr>
<th>ELEM</th>
<th>1</th>
<th>1</th>
<th>3</th>
<th>23</th>
<th>21</th>
<th>2</th>
<th>15</th>
<th>22</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>25</td>
<td>23</td>
<td>4</td>
<td>16</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>7</td>
<td>27</td>
<td>25</td>
<td>6</td>
<td>17</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1</td>
<td>7</td>
<td>9</td>
<td>29</td>
<td>27</td>
<td>8</td>
<td>18</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1</td>
<td>11</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>19</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1</td>
<td>11</td>
<td>13</td>
<td>33</td>
<td>31</td>
<td>12</td>
<td>20</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1</td>
<td>21</td>
<td>23</td>
<td>43</td>
<td>41</td>
<td>22</td>
<td>35</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1</td>
<td>23</td>
<td>25</td>
<td>45</td>
<td>43</td>
<td>24</td>
<td>36</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1</td>
<td>25</td>
<td>27</td>
<td>47</td>
<td>45</td>
<td>26</td>
<td>37</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1</td>
<td>27</td>
<td>29</td>
<td>49</td>
<td>47</td>
<td>28</td>
<td>38</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>1</td>
<td>29</td>
<td>31</td>
<td>51</td>
<td>49</td>
<td>30</td>
<td>39</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>1</td>
<td>31</td>
<td>33</td>
<td>53</td>
<td>51</td>
<td>32</td>
<td>40</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>1</td>
<td>41</td>
<td>43</td>
<td>63</td>
<td>61</td>
<td>42</td>
<td>55</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>1</td>
<td>43</td>
<td>45</td>
<td>65</td>
<td>63</td>
<td>44</td>
<td>56</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1</td>
<td>45</td>
<td>47</td>
<td>67</td>
<td>65</td>
<td>46</td>
<td>57</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>1</td>
<td>47</td>
<td>49</td>
<td>69</td>
<td>67</td>
<td>48</td>
<td>58</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>1</td>
<td>49</td>
<td>51</td>
<td>71</td>
<td>69</td>
<td>50</td>
<td>59</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>1</td>
<td>51</td>
<td>53</td>
<td>73</td>
<td>71</td>
<td>52</td>
<td>60</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>1</td>
<td>61</td>
<td>63</td>
<td>83</td>
<td>81</td>
<td>62</td>
<td>75</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1</td>
<td>63</td>
<td>65</td>
<td>85</td>
<td>83</td>
<td>64</td>
<td>76</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>1</td>
<td>65</td>
<td>67</td>
<td>87</td>
<td>85</td>
<td>66</td>
<td>77</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>1</td>
<td>67</td>
<td>69</td>
<td>89</td>
<td>87</td>
<td>68</td>
<td>78</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>1</td>
<td>69</td>
<td>71</td>
<td>91</td>
<td>89</td>
<td>70</td>
<td>79</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>1</td>
<td>71</td>
<td>73</td>
<td>93</td>
<td>91</td>
<td>72</td>
<td>80</td>
<td>92</td>
</tr>
</tbody>
</table>
MATE
1 1 ELASTO-PLASTIC, VON-MISES CRITERION.
1200.0 0.30 2 2 +1 -1 84.0 41.0
0.0
2 1 ELASTIC FRICTION LAYER
1200000.0 0.30 2 2 +1 +1 84000.0 41000.0
BOUND
  1  1  0  -1
  13 0  0  1
  161 1  -1  -1
  173 0  1  1
  174 1  -1  0
  193 0  1  0

FORC
  161 1  -0.6
  173 0  -0.6

END
MACR
DT  1.0
PROP 1.0
WYLB
POST
BCNG
NONL
ACCE
TANG
LOOP  43
TIME
LOOP  30
FORM
FCNV
SOLV
PLAS
NEXT
DISP  20
STRE  20
NEXT
DISP
STRE
REAC
UNLD
5. **Input data for case 3**
FEAP  RING OF 6:3:2 LAMINAC 44 ELEMENTS 56 NODES (CASE 3)

56  42  2  2  2  4  0

NPR

COOR

1  1  30.0
7  0  60.0
8  1  30.0  1.0
14  0  60.0  1.0
15  1  30.0  3.0
21  0  60.0  3.0
22  1  30.0  6.0
28  0  60.0  6.0
29  1  30.0 10.0
35  0  60.0 10.0
36  1  30.0 16.0
42  0  60.0 16.0
43  1  30.0 20.0
49  0  60.0 20.0
50  1  30.0 23.0
56  0  60.0 23.0

ELEM

1  1  1  2  9  8  1
6  1  6  7 14 13 1
7  1  8  9 16 15 1
12  1 13 14 21 20 1
13  1 15 16 23 22 1
18  1 20 21 28 27 1
19  1 22 23 30 29 1
24  1 27 28 35 34 1
25  1 29 30 37 36 1
30  1 34 35 42 41 1
31  1 36 37 44 43 1
MATE
1 1 NONLINEAR PLASTIC
1200.0 0.30 2 2 +1 -1 84.0 27.5
1.0 21.5
2 1 INTERFACE PROPERTY
1200000.0 0.30 2 2 +1 +1 84000.0 41000.0
BOUND
1 1 0 -1
7 0 0 1
43 1 -1 -1
49 0 1 1
50 1 -1
56 0 1
FORC
43 1 -0.6
49 0 -0.6
END
MACR
DT 1.0
PROP 1.0
WYLB
POST
BCNG
NONL
ACCE
TANG LOOP 43
TIME LOOP 30
FORM FCNV SOLV PLAS NEXT
DISP 14 STRE 14 NEXT
DISP STRE REAC UNLD TANG FORM
SOLV DISP STRE END

        1    1    1.01   45.0   0.04
7
43    1    1    1
49  -1.0    1    1

STOP
6. Input data for case 4
**FEAP**  RING OF 6:3:2 LAMINAC 42 ELEMENTS 57 NODES (CASE 4)

<table>
<thead>
<tr>
<th>NOPR</th>
<th>56</th>
<th>42</th>
<th>2</th>
<th>2</th>
<th>2</th>
<th>4</th>
<th>0</th>
</tr>
</thead>
</table>

**COOR**

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>30.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>60.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>30.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>60.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>30.0</td>
<td>3.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>60.0</td>
<td>3.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>30.0</td>
<td>6.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>60.0</td>
<td>6.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>30.0</td>
<td>10.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>60.0</td>
<td>10.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>1</td>
<td>30.0</td>
<td>16.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>0</td>
<td>60.0</td>
<td>16.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>1</td>
<td>30.0</td>
<td>20.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>0</td>
<td>60.0</td>
<td>20.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>30.0</td>
<td>23.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>0</td>
<td>60.0</td>
<td>23.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ELEM**

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>14</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>8</td>
<td>9</td>
<td>16</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>13</td>
<td>14</td>
<td>21</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>15</td>
<td>16</td>
<td>23</td>
<td>22</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>20</td>
<td>21</td>
<td>28</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>22</td>
<td>23</td>
<td>30</td>
<td>29</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>27</td>
<td>28</td>
<td>35</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>29</td>
<td>30</td>
<td>37</td>
<td>36</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>34</td>
<td>35</td>
<td>42</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>36</td>
<td>37</td>
<td>44</td>
<td>43</td>
<td>1</td>
</tr>
</tbody>
</table>
MATE
  1  1  NONLINEAR PLASTIC
  1200.0  0.30  2  2  +1  -1  84.0  41.0
  2  1  INTERFACE PROPERTY
  1200000.0  0.30  2  2  +1  +1  84000.0  41000.0

BOUND
  1  1  0  -1
  7  0  0  1
  43  1  -1  -1
  49  0  1  1
  50  1  -1
  56  0  1

FORC
  43  1  -0.6
  49  0  -0.6

END
MACR
DT  1.0
PROP  1.0
POST
BCNG
NONL
ACCE
UPLR
TANG
LOOP  43
TIME
LOOP  30
FORM
FCNV
SOLV
PLAS
NEXT
DISP  20
STRE  20
NEXT
DISP
STRE
REAC
UNLD
TANG
FORM
SOLV
DISP
STRE
END

1  1  1.01  45.0  0.10

  7
43  1  1  1
49 -1  1  1

STOP
C. Slicing Plan in Experiment

*STAR VALUES ARE AFTER COMPRESSION AND REMOVAL FROM MACHINE, OTHER VALUES ARE PRIOR TO COMPRESSION.

3 Sketch showing (a) geometry of the ring before and after compression, how the radial and circumferential slices were taken, and coordinates orientation, (b) detailed sketch of the circumferential slices.
D. Schematic of Data Flow

Output via print queue (monitoring out):
- Echo of input data
- Actual number of iterations
- Actual sequence of macros executed
- Magnitude of unbalanced force
- Warning or error messages.

Output via punch queue:
- Post-processing data (stresses, strains, displacement etc.).
XIII. APPENDIX C - PROGRAM LISTINGS

A. Listing of the Finite Element Program - MINIFEM
PROGRAM MINIFEM

C..............SET PROGRAM CAPACITY & MAX MUST AGREE WITH DIMENSION OF M
COMMON M(20000)
COMMON/PSIZE/ MAX
MAX = 20000
CALL PCONTR
STOP
END

C

BLOCK DATA
COMMON/CDATA/ 0.HEAD(20)•NUMNP,NUMEL•NUMMAT•NEN•NEQ•IPR
COMMON/LABEL/PDIS(6),A(6),BC(2),DI(6),CD(3),TE(3),FD(3)
DATA A/2H.1,2H.2,2H.3,2H.4,2H.5,2H.6/,CD/4H COO,4HRDN,4HATES/
DATA TE/4H TEM,4HPERA,4HTURE/,FD/4H FOR,4HCE/D,4HISPL/
DATA PDIS/4H(II0),2H, ,4HF13,4H4, ,4H6E13,4H,4) /
DATA BC/4H B,C.2H, /,DI/4H DIS,2HPL,4H VEL,2H0C,4H ACC,2HEL/
DATA 0/1H0/,IPR/1/
END
SUBROUTINE ACCELE (DR,UAC,NFD)

C**********************************************************************************
C PROGRAM NAME: ACCELERATION.
C PURPOSE : USING MODIFIED AITKEN'S METHOD FOR ACCELERATING
DISPLACEMENT CONVERGENCE FOR PLASTICITY USING
THE "INITIAL STRESS METHOD".
C LOGICAL FLOW: CALCULATE ACCELERATOR S IF POSSIBLE ( THAT IS
C INPUT : CURRENT DISPLACEMENT INCREMENT ::DR
LAST DISPLACEMENT INCREMENT ::UAC
C OUTPUT : UPDATED DR AND UAC IF APPLICABLE.
C**********************************************************************************
C DIMENSION DR(1),UAC(1)
COMMON/ACDATA/ S
COMMON/DEBUG/ IDEBUG
C
DO 10 I = 1,NFD
UAC(I) = DR(I) - UAC(I)
10 CONTINUE
DENOM = DOT(UAC,UAC,NEO)
IF(DENOM.EQ.0.0) GO TO 25
UPPER = DOT(UAC,DR,NEO)
S = -UPPER/DENOM
IF(IDESUB.GE.7) WRITE(6,100) S
100ddd..NEGATIVE S IS MEANINGLESS.
IF(S.LE.0.0) GO TO 25
DO 20 I = 1,NEO
DR(I) = DR(I) * (S + 1.0)
20 CONTINUE
   GO TO 30
25 S = 0.0
C
30 DO 40 I = 1, NEQ
   UAC(I) = DR(I)
40 CONTINUE
100 FORMAT(/20X,'VALUE OF ACCELERATOR S IS ',5X,F8.3)
RETURN
END
SUBROUTINE ACTCOL(A,B,JDIAG,NEQ,AFAC,BACK)
LOGICAL AFAC,BACK
COMMON/ENGYS/ AENGY
DIMENSION A(1),B(1),JDIAG(1)
C
C............ACTIVE COLUMN PROFILE SYMMETRIC EQUATION SOLVER.
C
C............FACTOR A TO U*D*U REDUCE B
AENGY = 0.0
JR = 0
DO 600 J = 1,NEQ
   JD = JDIAG(J)
   JH = JD - JR
   IS = J - JH + 2
   IF(JH-2) 600,300,100
100  IF(.NOT.AFAC) GO TO 500
   IE = J - 1
   K = JR + 2
   ID = JDIAG(IS-1)
   ....REDUCE ALL EQUATIONS EXCEPT DIAGONAL
   DO 200 I = IS,IE
      IR = ID
      ID = JDIAG(I)
      IH = MIN(I,D-IR-1,I-IS+1)
      IF(IH.GT.0) A(K) = A(K)-DOT(A(K-IH)*A(ID-IH),IH)
200  K = K + 1
C............REDUCE DIAGONAL TERM
300  IF(.NOT.AFAC) GO TO 500
   IR = JR + 1
   IE = JD - 1
   K = J - JD
   DO 400 I = IR,IE
      ID = JDIAG(K+1)
      IF(A(ID).EQ.0.0) GO TO 400
      D = A(ID)
      A(I) = A(I)/A(ID)
      A(JD) = A(JD) - D * A(I)
400  CONTINUE
C********** REDUCE RIGHT HAND SIDE.
  500  IF(BACK) B(J) = B(J) - DOT(A(JR+1), B(IS-1), JH-1)
  600  JR = JD
        IF(.NOT. BACK) RETURN
C********** DIVIDED BY DIAGONAL PIVOTS.
  700  DO I = 1, NEQ
       ID = JDIAG(I)
       IF(A(ID) .NE. 0.0) B(I) = B(I) / A(ID)
       AENGY = AENGY + 3(I) * B(I) * A(ID)
  700  AENGY = AENGY + 3(I) * B(I) * A(ID)
C********** BACK SUBSTITUTE.
  800  J = NEQ
       JD = JDIAG(J)
  800  D = B(J)
       J = J - 1
       IF(J .LE. 0) RETURN
       JR = JDIAG(J)
       IF(JD - JR .LE. 1) GO TO 1000
       IS = J - JD + JR + 2
       K = JR - IS + 1
       DO I = IS, J
       900  B(I) = B(I) - A(I+K) * D
  900  B(I) = B(I) - A(I+K) * D
  1000 JD = JR
       GO TO 800
       END
SUBROUTINE ADDSTF(A,B,C,S,P,JDIAG,LD,NST,NEL,AFL,BFL,CFL)

C

C ASSEMBLE GLOBAL ARRAYS

C

LOGICAL AFL,BFL,CFL

DIMENSION A(1),B(1),JDIAG(1),P(1),S(NST,1),LD(1),C(1)

DO 200 J = 1,NEL
  K = LD(J)
  IF(K.EQ.0) GO TO 200
  IF(BFL) B(K) = B(K) + P(J)
  IF(.NOT.AFL.AND..NOT.CFL) GO TO 200
  L = JDIAG(K) - K
  DO 100 I = 1,NEL
    M = LD(I)
    IF(M.GT.K.OR.M.EQ.0) GO TO 100
    M = L + M
    IF(AFL) A(M) = A(M) + S(I,J)
    IF(CFL) C(M) = C(M) + S(J,I)
  100 CONTINUE

200 CONTINUE

RETURN

END
SUBROUTINE BCHNG(ID, IX, F, JDIAG, NDF, NENI, NAD, NA, NE,
  N, NBN, NDFI, IPBCN, B, BB)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C NAME : BOUNDARY CHANGE ROUTINE
C
C PURPOSE: MODIFY THE ID AND F ARRAY FOR CHANGE IN BOUNDARY
C CONDITIONS SUCH AS FROM PRESCRIBED DISPLACEMENT TO
C PRESCRIBED FORCE AND VICE VERSA.
C
C INPUT : ID, F, IX, IPBCN, B
C
C OUTPUT : ID, F, B, JDIAG, NEQ, NE
C
C SUBROUTINES CALLED :
C
BCSTOR, PROFIL, SETMEM
C
C CALLED BY : "OMACR" OF MACRO 'BCNG'
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
COMMON/CDATA/ 0, HEAD(20), NUMNP, NUMEL, NUMMAT, NEN, NEQ, IPR
COMMON/RLOD/ TPROP, PROP
COMMON/ROLL/XR(3), ROLLRA, FRFAC, OMEGA, CUVC, FRFL
LOGICAL CUVC, FRFL
DIMENSION ID(NDF,1), F(NDF,1), IX(1), JDIAG(1), IPBCN(NDFI,1),
  B(1), BB(1)
C
C................MAPPING THE B ARRAY USING THE OLD ID ARRAY. SINCE
C B DOES NOT CONTAIN PRESCRIBED DISPLACEMENT, WE HAVE
C TO CHANGE THE LENGTH ACCOMODATING THE CHANGE.
C

C
J = 1
DO 60 I = 1, NUMNP
  DO 10 II = 1, NBN
    IF(IPBCN(I, II) .EQ. 1) GO TO 30
10  CONTINUE
C
C NODE WHOSE BOUNDARY CONDITION IS NOT TO BE CHANGED
C
DO 20 JJ = 1, NDF
  IF(ID(JJ, I) .EQ. 0) GO TO 20
C
C FORCE INPUT, STORE THE DISPLACEMENT VALUE.
C
K = IABS(ID(JJ, I))
BB(J) = B(K)
J = J + 1
20  CONTINUE
GO TO 60
C
C NODE WHOSE BOUNDARY "IS" TO BE CHANGED.
C
30  DO 50 JJ = 1, NDF
    IF(IPBCN(JJ+1, II) .NE. 0) GO TO 40
C
C THE D.O.F. WHOSE B.C. IS NOT TO BE CHANGED
C
    IF(ID(JJ, I) .EQ. 0) GO TO 50
    K = IABS(ID(JJ, I))
    BB(J) = B(K)
    J = J + 1
    GO TO 50
C
C THE D.O.F. WHOSE B.C. "IS" TO BE CHANGED
C
40  IF(ID(JJ, I) .GT. 0) GO TO 50
DISPLACEMENT INPUT CHANGED TO FORCE INPUT. THE CURRENT TOTAL DISPLACEMENT IS INSERTED.

BB(J) = 0.0
BB(J) = TRP(F(JJ,I))
J = J + 1

50 CONTINUE
60 CONTINUE

C
C RESTORE ID AND F FROM TAPE16, MODIFY THEM.
C
CALL BCSTOR(ID,F,N,1,2)
DO 500 I = 1,NUMNP
  DO 100 J = 1,NBN
    IF(IPBCN(I,J).EQ.I) GO TO 200
  100 CONTINUE
  GO TO 500
C
A BOUNDARY NODE IS FOUND
C
200 DO 400 K = 1,ND
  IF(IPBCN(K+1,J).EQ.0) GO TO 400
C
THE D.O.F. WHOSE B.C. NEEDS TO BE CHANGED IS FOUND
C
IF(ID(K,I).EQ.0) GO TO 300
  ID(K,I) = 0
C
DISPLACEMENT INPUT TO FORCE INPUT. NOTE THAT THE VALUE OF FORCE HAS TO BE ZERO UNLESS MODIFIED.
C
F(K,I) = 0.0
GO TO 400

300 ID(K,I) = 1
C
SPECIAL CASE FOR ROLLING. FORCE INPUT TO DISPLACE. INPUT.
IF(.NOT.CUVCH.OR.JJ.NE.1) GO TO 400
F(K,I) = OMEGA

400 CONTINUE
500 CONTINUE

C**********SAVE NEW VALUES ON TAPE16, RECALCULATE JDIAG, NEQ
C
CALL BCSTOR(ID,F,N,1,1)
CALL PROFIL(JDIAG,ID,IX,NDF,NEN1,NAD)

C**********RECALCULATE POSTIONS FOR STIFFNESS.
C      ** STIFFNESS MATRIX HAS TO BE THE LAST ONE TO BE DIMENSIONED
C             IN THE ENTIRE PROGRAM **
C
NOFF = NE - NA
IF(JDIAG(NEQ).LE.NOFF) GO TO 600
NE = NA + JDIAG(NEQ)
CALL SETMEM(NE)

C**********STORE MAPPED DISPLACEMENT BACK TO ORIGINAL ARRAY(B).
C
600 DO 700 I = 1,NEQ
      B(I) = BB(I)
700 CONTINUE
RETURN
END
SUBROUTINE BCINPU (IPBCN, NDF1, NDF, NTOTAL)

NAME : BOUNDARY CONDITION INPUT ROUTINE.
PURPOSE : HANDLES LINEAR INTERPOLATION FOR BOUNDARY NODES AND ITS DEGREE OF FREEDOM INPUT. ALSO TO BE USED AS INPUT TOOL FOR LOCALLY ORIENTED COORDINATES.

MAJOR VARIABLE :

N = BOUNDARY NODE (LOCALLY ORIENTED NODE) NUMBER.

NG = AUTO INCREMENT INDICATOR AND VALUE.
  (1) > 1 = STARTING THE AUTO INCREMENT, NG IS THE INCREMENT.
  (2) NEGATIVE = TERMINATING AUTO INCREMENTING.

IDF = DEGREE OF FREEDOM ASSOCIATED WITH THE NODE N (FOR BOUNDARY NODE INPUT )
  = NO. OF TRANSFORMATION MATRIX FOR LOCALLY ORIENTED NODES INPUT.

NDF = DEGREE OF FREEDOM FOR BOUNDARY NODE INPUT.
  = 1 FOR LOCALLY ORIENTED NODE.

NDF1 = NDF + 1

IPBCN = ARRAY OF (NDF+1, NBN) WHERE NBN IS THE MAXIMUM NUMBER OF BOUNDARY NODES (FOR BOUNDARY NODES INPUT) AND NBN IS THE MAXIMUM NUMBER OF NODES THAT ARE LOCALLY ORIENTED.


C

IPBCN(1,J) STORES THE BOUNDARY NODE NUMBER (LOCALLY
ORIENTED NODE NUMBER).

IPBCN(2,J) TO IPBCN(NDF+1,J) STORES THE ASSOCIATED
DEGREE OF FREEDOM FOR BOUNDARY NODE INPUT

IP3CN(2,J) STORES THE NUMBER OF TRANSFORMATION
MATRIX.

NTOTAL = THE ACTUAL NUMBER OF NODES GENERATED (INPUT).

NOTE (1) : TERMINATING THE INPUT WITH A BLANK ROW.

(2) : THE IDF VALUE FOR AUTOINCREMENTING IS BASED ON THE
TERMINATING ONE (I.E. NG = -1). IT IS RECOMMENDED
THAT THEY BE THE SAME FOR STARTING (NG = 1) AND
TERMINATING (NG NEGATIVE) CARD.

DIMENSION IPBCN(NDF,1), IDF(3)

I = 1
10 READ(5,1000) N,NG,(IDF(J),J = 1,NDF)
NTOTAL = I - 1
IF(N.EQ.0) RETURN

IF(NG.NE.0) GO TO 20

C...NO AUTO INCREMENT DESIRED.

IPBCN(1,1) = N
DO 15 J = 1,NDF
   IPBCN(J+1,1) = IDF(J)
15 CONTINUE
I = I + 1
GO TO 10
C........AUTO_INCREMENT_DESIRED
C
20 IF(NG.LT.0) GO TO 30
C
C STARTING_NODE_OF_AUTOINCREMENT_FOUND.(NG > 1)
C
NHEAD = N
INC = NG
IPBCN(1,1) = N
DO 25 J = 1,NDF
   IPBCN(J+1, I) = IDF(J)
25 CONTINUE
I = I + 1
GO TO 10
C
C ENDING_NODE_OF_AUTOINCREMENT_FOUND.(NG_NEGATIVE)
C
30 NEND = N
IGAP = NEND - NHEAD
NITE = IGAP/INC
DO 45 JJ = 1,NITE
   IPBCN(1, I) = NHEAD + INC*JJ
   DO 40 J = 1,NDF
   40 IPBCN(J+1, I ) = IDF(J)
   I = I + 1
45 CONTINUE
C
C ERROR_CHECK WHETHER_THE_INCREMENT_SIZE_CONFORMS_WITH_NODES.
C
IF(IPBCN(1,I-1). NE . NEND) GO TO 50
GO TO 10
C
50 WRITE(6,1001)
   CALL_EXIT
C
1000 FORMAT(1615)
1001 FORMAT(/5X,'** FATAL ERROR IN BOUNDARY NODE INPUT **'/10X.
     1 * STARTING/END NODE NUMBER AND INCREMENT NO MATCH*/)
END
SUBROUTINE BCSTOR (ID, F, N, IPR, IBSW)

NAME : BOUNDARY CONDITION STORE OR RETRIEVE.

INPUT :

ID = BOUNDARY CONDITION ARRAY BEFORE "PROFL" IS CALLED

F = FORCE/DISPLACEMENT INPUT

IPR = DEGREE OF PRECISION

N = NDF = NUMNP

IBSW = 1 INDICATES WRITE (STORE ) ACTION.

= 2 INDICATES READ(RESTORE) ACTION.

CALLED BY :

"PCONTR" (IBSW = 1) UNCONDITIONALLY.

"BCHNG" (IBSW = 1 AND 2) - CALLED BY "PMACR" OF "BCNG" MACRO.

LOGICAL UNIT USED : TAPE16

DIMENSION ID(1), F(1)
REWIND 16
NIPR = N = IPR
GO TO (10, 20), IBSW

10 WRITE(16) (ID(J), J=1,N), (F(J), J=1,NIPR)
REWIND 16
RETURN
C

20 READ(16) (ID(J), J = 1,N),(F(J), J = 1,NIPR)
REWIND 16
RETURN
C
END
SUBROUTINE DISLD(I1,NILC,NDF,UL,NLOC,TRANF,NME2,IDSW)
C
NAME : DISPLACEMENT CONVERSION FOR LOCAL COORDINATES.
C
OPERATION : UL = L * UL'
C
MAJOR VARIABLES :
C
NTLC,NDF,NLOC,TRANF (FOR DESCRIPTION SEE "STCNV")
C
IDSW = DISPLACEMENT SWITCH
C
   = 1   UL = L * UL'
C
   = 2   P* = LT * P (CALLED BY "ELMTON")
C
II = IX(I,N)  NODE NUMBER OF THE ELEM. N
C
  I = I, NEN THE ITH NODE OF ELE. N.
C
UL = DISPLACEMENT OF THE NODES OF NTH ELE. (NDF,NEN)
C
TEMPU = WORKING AREA FOR DISPLACEMENT.
C
CALLED BY : "PFORM" INSIDE THE ELEMENT LOOP. ACTIVATED IF
C
LQFL = .TRUE. (THERE IS LOCALY COORDINATE).
C
DIMENSION NLOC(2,1),TRANF(NME2,1),TEMPU(6),UL(NDF,1)
C
DO 1071 JL = 1,NILC
   IF(NLOC(1,JL).EQ.I1) GO TO 1072
1071 CONTINUE
RETURN
C  FOUND A LOCAL NODE

1072  JTR = NLOC(2,JL)
    DO 1074 KL = 1,NDF
        TEMPU(KL) = 0.0
    DO 1073 ML = 1,NDF
        IF(IDSW.EQ.1) LT = ML + NDF*(KL-1)
        IF(IDSW.EQ.2) LT = KL + NDF*(ML-1)
        TEMPU(KL) = TEMPU(KL) + TRANF(LT,JTR)*UL(ML,I)
1073  CONTINUE
1074  CONTINUE

C  DEPOSIT TEMPU INTO UL ARRAY
C
    DO 1075 KL = 1,NDF
    1075  UL(KL*I) = TEMPU(KL)
RETURN
END
FUNCTION DOT(A,B,N)
C
C..............VECTOR DOT PRODUCT
C
DIMENSION A(1),B(1)
DOT = 0.0
DO 100 I = 1,N
100   DOT = DOT + A(I)*B(I)
RETURN
END
SUBROUTINE DSTAR8 (EPGR, TSIG, DE)

NAME : DO * B * DU

NOTE : ONLY WORK FOR TWO-DIMENSIONAL CASE

DIMENSION EPGR(3,3), TSIG(1), DE(3,3)

DE(1,1) = -TSIG(1)*EPGR(1,1) - TSIG(4)*EPGR(1,2)
DE(1,2) = -TSIG(4)*EPGR(1,1) + 0.5*(TSIG(1) - TSIG(2))*EPGR(1,2)
1 = 0.5*(TSIG(1) + TSIG(2))*EPGR(2,1)
DE(2,1) = -0.5*(TSIG(1) + TSIG(2))*EPGR(1,2)
1 = 0.5*(TSIG(2) - TSIG(1))*EPGR(2,1)
2 = -TSIG(4)*EPGR(2,2)
DE(2,2) = -TSIG(4)*EPGR(2,1) - TSIG(2)*EPGR(2,2)
DE(3,3) = -TSIG(3) * EPGR(3,1)

RETURN
END
SUBROUTINE ELMLIB(D,U,X,I,T,S,P,I,J,K,ISW)
  1  TSIG,TEPS,RATIO,EQSIG,FT,
  2  UAC,UNSIG,NLOC,TRANF,NDM2,NGAUP,NTOT)
  
C ELEMENT LIBRARY

COMMON/ELDATA/ DM,N,MA,MC1,IEL,NEL
DIMENSION P(K),S(K,K),D(I),U(I),X(I),T(I),UNSIGY(1),
  1  TSIG(I),TEPS(I),RATIO(I),EQSIG(I),FT(I),UAC(I),
  2  NLOC(I),TRANF(I)
IF(IEL.LE.0.OR.IEL.GT.4) GO TO 400
  IF(ISW.LT.3) GO TO 30
  DO 20 L = 1,K
    P(L) = 0.0
  DO 20 M = 1,K
    S(L,M) = 0.0
  20 GO TO 1,2,3,4,IEL
  1 CALL ELMTO1(D,U,X,I,T,S,P,I,J,K,ISW,
  1  TSIG,TEPS,RATIO,EQSIG,FT,
  2  UAC,UNSIG,NLOC,TRANF,NDM2,NGAUP,NTOT)
  GO TO 10
  2 CALL ELMTO2(D,U,X,I,T,S,P,I,J,K,ISW,
  1  TSIG,TEPS,RATIO,EQSIG,FT,
  2  UAC,UNSIG,NLOC,TRANF,NDM2,NGAUP,NTOT)
  GO TO 10
  3 CALL ELMTO3(D,U,X,I,T,S,P,I,J,K,ISW,
  1  TSIG,TEPS,RATIO,EQSIG,FT,
  2  UAC,UNSIG,NLOC,TRANF,NDM2,NGAUP,NTOT)
  GO TO 10
  4 CALL ELMTO4(D,U,X,I,T,S,P,I,J,K,ISW,
  1  TSIG,TEPS,RATIO,EQSIG,FT,
  2  UAC,UNSIG,NLOC,TRANF,NDM2,NGAUP,NTOT)
  10 RETURN
  400 WRITE(6,4000) IEL
  STOP
  4000 FORMAT(5X,'**FATAL ERROR 04** ELEMENT CLASS NUMBER',I3,' INPUT')
END
SUBROUTINE ELMTO1 (D,UL, XL, IX, TL, S, P, NDF, NDM, NST, ISW,
  1 TSIG, TEPS, RATIO, EOSIG, FT, UAC, UNSIGY,
  2 NLDC, TRANF, NDM2, NGAUP, NTOT)

C         COMMON/CDATA/0.HEAD(20), NUMNP, NUMEL, NUMMAT, NEN, NEQ, IPR
C         COMMON/ELDATA/ N,M, NA, MCT, IEL, NEL
C         COMMON/EPSRES/ROSIG(6)
C         COMMON/CLOCA/LOFL, NME2, NTLC, NDTAL, NTR
C         COMMON/DEBUG/ IDBG
C         COMMON/PDDATA/POFL, FMFL
LOGICAL
  ACFL, ELAS, INCFL, ISFL, NDFL, TRANS, UNFL, POFL, LOFL
  FRFL, CUVCV, FMFL, PMFL, UPLRF
C         COMMON/NLDATA/ ACFL, ELAS, INCFL, ISFL, NDFL, TRANS, UNFL
COMMON/PMDATA/PWFL
COMMON/UPDATE/UPLRF
COMMON/CROLL/XR(3), ROLLRA, RFAC, OMEGA, CUVCV, FRFL
DIMENSION D(8, MA), UL(NDF, 1), XL(NDM, 1), IX(1), TL(1), S(NST, 1), P(1)
  1 SHP(3, 9), SG(9), TG(9), WG(9), SIG(7), EPS(7), WD(9)
  2 TSIG(NDM2, 1), TEPS(NDM2, 1), RATIO(1), EOSIG(1), FT(1)
  3 UAC(1), EP(6, 6), UNSIGY(1), NLDC(2, 1), TRANF(1), DE(3, 3)
  4 EPGR(3, 3)
DATA WD/4HPLAN, 4HST, 4HRESS, 4HPLAN, 4HE ST, 4HRAIN,
  1 4HAXIS, 4HYMME, 4HTRIC/
DATA NTAPE1/30/, NTAPE2/31/

C         GO TO CORRECT ARRAY PROCESSOR.
C
GO TO (1.2, 3.4, 5.4, 4) * ISW

C         INPUT MATERIAL PROPERTIES.
C         D(8, MA) .GE. 0  --- ELASTIC MATERIAL
C         D(8, MA) .LT. 0  --- NONLINEAR MATERIAL
C         D(9, MA) = 4 (STRAIN HARDENING PARAMETER)
C         D(10, MA) = SIGY (INITIAL YIELD STRESS)
C         D(11, MA) = C - T FOR MODIFIED VON MISES YIELD CRITERION
D(7) = I

I .LT.0 -- PLANE STRESS. I .EQ.0 -- PLANE STRAIN.
I .GT.0 -- AXISYMMETRIC.

READ(5,1000) E,XNU,D(4),L,K,I,D(8),D(9),D(10)
IF(D(8).LT.0) READ(5,1000) D(11),D(12)
IF(D(11).EQ.0.) PMFL = .TRUE.
D(7) = FLOAT(I)
IF(I.GE.0.) I = 2
IF(I.LT.0) I = 1

D(1) = E*(1.+(1-I)*XNU)/(1.+XNU)/(1.-I*XNU)
D(2) = XNU*D(1)/(1.+(1-I)*XNU)
D(3) = E/2./(1.+XNU)
L = MIN0(3*MAX0(1,L))
D(5) = L
K = MIN0(3*MAX0(1,K))
D(6) = K
LINT = 0
IIND = I

IF(D(7).EQ.0.0) IIND = 4
IF(D(7).GT.0.0) IIND = 7
IIND2 = IIND+2
WRITE(6,2000) (WD(J),J=IIND,IIND2),E,XNU,D(4),L,K
IF(D(8).LT.0) WRITE(6,2003) D(9),D(10)
IF(D(11).NE.0.) WRITE(6,2005) D(12)
RETURN
RETURN

L = D(5)
IF(L.LT.LINT) CALL PGAUSS(L,LINT,SG,TG,WG)

FOR EACH INTEGRATION PT. COMPUTE CONTRIBUTION TO STIFF.

IF(D(7).LE.0) PI2 = 1.0
DO 100 L = LINT
    CALL SHAPE(SG(L),TG(L),XL,SHP,XSJ,NDM,NEL,IX.,FALSE.)
    RRINV = 0.0
    IF(D(7).LE.0) GO TO 70
**SPECIAL TREATMENT FOR AXISYMMETRIC CASE.**

```fortran
RR = 0.0
DO 60 JAX = 1, NEL
   RR = RR + SHP(3,JAX)*XL(1,JAX)
60
PI2 = 6.28318 / RR
RRINV = 1./RR

DO 70 JAX = 1, NEL
   DV = XSJ*WG(L)*PI2
   D1 = D(1)*DV
   D2 = D(2)*DV
   D3 = D(3)*DV

C CHECK IF ANY NONLINEAR PROCESS.
C
IF (ELAS .OR. D(8) .GE. 0) GO TO 80
IF (RATID(L) .GE. 0.0) GO TO 80
C
C NONLINEAR MATERIAL, POST YIELD ELEMENT.
C
CALL EPMAT(NDM2, DV, D, EQSIG(L), TSIG(1, L), EP)
C
C FOR EACH J NODE COMPUTE DEPB = DEP*B.
C
DO 120 J = 1, NEL
   DEPB = DEP*B

C CHECK ANY FRICTION EFFECT
C
IF (.NOT. FRFL) GO TO 79
 DO 77 JK = 1, NTOTAL
    IF (NLOC(1, JK) .EQ. IX(J) .AND. NLOC(2, JK) .NE. 0)
       GO TO 78
1
77 CONTINUE
GO TO 70
```
C

78  FRTM = FRFAC
79  CONTINUE
C

DUM1 = SHP(3,J)*RRINV
DB11 = (EP(1,1)*SHP(1,J) + EP(1,3)*DUM1 + EP(1,4)*SHP(2,J)) * FRTM
DB12 = (EP(1,2)*SHP(2,J) + EP(1,4)*SHP(1,J)) * FRTM
DB21 = (EP(2,1)*SHP(1,J) + EP(2,3)*DUM1 + EP(2,4)*SHP(2,J)) * FRTM
DB22 = (EP(2,2)*SHP(2,J) + EP(2,4)*SHP(1,J)) * FRTM
C

C........... FOR PLANE STRESS, THE THIRD ROW OF D MATRIX IS ZERO.
C

DB31 = 0.0
DB32 = 0.0
IF(DC7).LT.0 GO TO 75
DB31 = (EP(3,1)*SHP(1,J) + EP(3,3)*DUM1 + EP(3,4)*SHP(2,J)) * FRTM
DB32 = (EP(3,2)*SHP(2,J) + EP(3,4)*SHP(1,J)) * FRTM
C

75  CONTINUE
C

DB41 = (EP(4,1)*SHP(1,J) + EP(4,3)*DUM1 + EP(4,4)*SHP(2,J)) * FRTM
DB42 = (EP(4,2)*SHP(2,J) + EP(4,4)*SHP(1,J)) * FRTM
DO 110 I = 1,J

DUM2 = SHP(3,I)*RRINV
S(I+I-1,J+J-1) = S(I+I-1,J+J-1) + SHP(1,I)*DB11
+ DUM2*DB31 + SHP(2,I)*DB41
1 S(I+I-1,J+J-1) = S(I+I-1,J+J-1) + SHP(1,I)*DB12
+ DUM2*DB32 + SHP(2,I)*DB42
2 S(I+I-1,J+J-1) = S(I+I-1,J+J-1) + SHP(2,I)*DB21
+ SHP(1,I)*DB41
2
CONTINUE TO TOO
FOR EACH J NODE COMPUTE DB = D*B (ELASTIC)

DO 140 J = 1,NEL
    FRTM = 1.0
    CONTINUE
    CHECK ANY FRICTION EFFECT
    IF (.NOT. FRL) GO TO 89
    DO 87 JK = 1,NTOTAL
        IF (NLOC(L1,J).EQ.IX(J) .AND. NLOC(L2,JK).NE.0)
            GO TO 88
    CONTINUE
    GO TO 89
    FRTM = FRFAC
    CONTINUE
    DUM1 = SHP(3,J)*RINV
    DB11 = (D1*SHP(1,J)+D2*DUM1)*FRTM
    DB12 = (D2*SHP(2,J))*FRTM
    DB21 = (D2*(SHP(1,J)+DUM1))*FRTM
    DB22 = (D1*SHP(2,J))*FRTM

FOR PLANE STRESS, THE THIRD COLUMN OF D MATRIX IS ZERO.

IF(D(7).LT.0) GO TO 85
DB31 = (D2*SHP(1,J)+D1*DUM1)*FRTM
DB32 = (D2*SHP(2,J))*FRTM

CONTINUE
DB41 = (D3*SHP(2,J))*FRTM
DB42 = (D3*SHP(1,J))*FRTM

C***** FOR EACH I NODE COMPUTE S = BT*DB
DO 130 I = 1,J
   DUM2 = SHP(3,I)*RRINV
   S(I+1-1,J+1-I) = S(I+1-1,J+1-I)+SHP(1,I)*DB11
       +DUM2*DB31+SHP(2,I)*DB41
   S(I+1-1,J+1) = S(I+1-1,J+1)+SHP(1,I)*DB12
       +DUM2*DB32+SHP(2,I)*DB42
   S(I+1,J+1-I) = S(I+1,J+1-I)+SHP(2,I)*DB21
       +SHP(1,I)*DB41
   S(I+1,J+1) = S(I+1,J+1)+SHP(2,I)*DB22
       +SHP(1,I)*DB42
130 CONTINUE

140 CONTINUE
100 CONTINUE

C
C CONVERT FOR LOCALLY ORIENTED NODE(S).
C
IF(LOFL) CALL STCNV(NLOC,IX,TRANF,S,NST,NEL,NDM)
C
C......... COMPUTE LOWER TRIANGULAR PART BY SYMMETRY.
C
NL = NEL+NEL
DO 200 I = 2,NL
   DO 200 J = 1,I
   S(I,J) = S(J,I)
200 RETURN
C
4 L = D(5)
IF(ISW.EQ.4) L = D(6)
   IF(L.LT.LINT) CALL PGAUSS(L,LINT,SG,TG,WG)
C
**C............COMPUTE ELEMENT STRESSES, STRAINS AND FORCES.**

**C**

\[
\text{DO 600 } L = 1, \text{LINT}
\]

**C............COMPUTE ELEMENT SHAPE FUNCTIONS.**

**C**

\[
\text{CALL SHAPE(SG(L),TG(L),XL,SHP,XSJ,NDM,NEL,IX,.FALSE.)}
\]

**C............UNLOAD AND PRINT STRESS.**

**C**

\[
\text{IF(ISW.EQ.4.AND.UNFL) GO TO 618}
\]

\[
\text{IF(ISW.EQ.4.AND.(.NOT.ELAS)) GO TO 618}
\]

**C.............COMPUTE STRAINS AND COORDINATES.**

**C**

\[
\text{DO 410 } I = 1, 4
\]

\[
\text{EPS(I) = 0.0}
\]

\[
\text{CONTINUE}
\]

\[
\text{XX = 0.0}
\]

\[
\text{YY = 0.0}
\]

\[
\text{DO 415 } J = 1, \text{NEL}
\]

\[
\text{XX = XX +SHP(3,J)XL(1,J)}
\]

\[
\text{YY = YY +SHP(3,J)XL(2,J)}
\]

\[
\text{CONTINUE}
\]

\[
\text{PI2 = 1.0}
\]

\[
\text{RRINV = 1.0/XX}
\]

\[
\text{IF(D(7).LE.0) RRINV = 0.0}
\]

\[
\text{IF(D(7).GT.0) PI2 = 6.283180X}
\]

**C**

**C.............FOR UNLOAD AND REACTION PHASE NO NEED TO CALCULATE**

**C**

**STRESS AND STRAIN AGAIN, USE THE TOTAL STRESS INSTEAD.**

**C**

**C (REF. "RESID"). EXCEPT IN PRINTING STAGE )**

**C**

\[
\text{IF(UNFL.AND.ISW.EQ.4) GO TO 418}
\]

\[
\text{IF(UNFL.OR.NDFL) GO TO 421}
\]

**C**

**C**

\[
\text{CONTINUE}
\]
DO 420 J = 1,NEL
  EPS(1) = EPS(1) + SHP(1,J)*UL(1,J)
  EPS(2) = EPS(2) + SHP(2,J)*UL(2,J)
  EPS(3) = EPS(3) + SHP(3,J)*UL(1,J)*RRINV
  EPS(4) = EPS(4) + SHP(1,J)*UL(2,J) + SHP(2,J)*UL(1,J)
CONTINUE
C
IF(ELAS) GO TO 422
C
421 IF(ISW.EQ.7) GO TO 422
    CALL RESID(D,1.0,SIG,EPS,EOSIG,L,TSIG(I,L),
                         RATIO(L),NDM2)
    GO TO 620
C............compute elastic stresses.
C
422 CONTINUE
  SIG(1) = D(1)*EPS(1) + D(2)*(EPS(2) + EPS(3))
  SIG(2) = D(1)*EPS(2) + D(2)*(EPS(3) + EPS(1))
  SIG(3) = D(1)*EPS(3) + D(2)*(EPS(1) + EPS(2))
  SIG(4) = D(3)*EPS(4)
C............for plane stress, the third stress is zero (SIG-Z)
C
  IF(D(7).LT.0) SIG(3) = 0.0
C
  IF(UNFL.AND.ISW.EQ.4) GO TO 618
  IF(ISW.EQ.7) GO TO 615
    IF(ISW.EQ.6) GO TO 620
    CALL PSTRES(SIG,SIG(5),SIG(6),SIG(7),0)
C............output stresses and strains.
C
MCT = MCT-2
  IF(MCT.GT.0) GO TO 430
    WRITE(6,2001) O,HEAD
    MCT = 50
430 WRITE(6,2002) N,MA,XX,YY,SIG,(EPS(J),J=1,4)
GO TO 600
C............COMPUTE INTERNAL FORCES.
C
620 DV=XSJ*WG(L)*PI2
J1 = 1
C
IF(UPLRFL.AND. .NOT.ISFL) CALL WRTBDU
1
IF(ULFLRFL.AND.ISFL) CALL GETBDU(EPGR)
IF(ULFLRFL) CALL OSTARBCEPGR.TS IG(1.L).OE)
C
DO 610 J = 1,NEL
C
C CHECK ANY FRICTION EFFECT
C
IF(.NOT.FPFL) GO TO 606
DO 602 JK = 1,NTOTAL
1
IF(NLDC(1.JK).EQ.1X(J).AND.NLOC(2.JK)
+NE.0) GO TO 604
602 CONTINUE
GO TO 606
C
C MODIFY THE NODE FOR FRICTION EFFECT IN ROLLING
C
604 P(J1)=P(J1)-(SHP(1,J)*ROSIG(1)+SHP(3,J)*
RRINV*ROSIG(3)+SHP(2,J)*ROSIG(4))*DV
P(J1+1)=P(J1+1)-(SHP(2,J)*ROSIG(2)+SHP(1,J)*
ROSIG(4))*DV
GO TO 608
C
606 CONTINUE
P(J1)=P(J1)-(SHP(1,J)*SIG(1)+SHP(3,J)*RRINV*SIG
(3)+SHP(2,J)*SIG(4))*DV
**P(J1+1)=P(J1+1)-(SHP(2,J)*SIG(2)+SHP(1,J)*SIG(4))*DV**

C

IF(UPLRFL) CALL GEODF(De,J1,SHP,DV,P,RRINV)

C

608 CONTINUE
610 J1 = J1+NDF
GO TO 600

C

615 CALL PRATID(EPS,SIG,D,TSIG(I,L),RATIO(L),EQSIG
1 (L),UNSIGY(L),NDM2)
IF(UPLRFL) CALL WRTBDU(NDF,NEL,RRINV,UL,SHP,EPGR)
GO TO 600

C

C NONLINEAR STRESS/STRAIN.
C

618 IF(ISW.NE.4.AND..NOT.UNFL) GO TO 6170
DO 6160 J = 1,NDM2
TSIG(J,L)=TSIG(J,L)+SIG(J)
TEPS(J,L)=TEPS(J,L)+EPS(J)
6160 CONTINUE

C

6170 XX = 0.0
YY = 0.0
DO 6180 J=1,NEL
XX = XX+SHP(3,J)*XL(1,J)
YY = YY+SHP(3,J)*XL(2,J)
6180 CONTINUE

CALL PSTRFS(TSIG(1,L),SIG(5),SIG(6),SIG(7),0)
CALL PSTRESS(TEPS(1,L),EPS(5),EPS(6),EPS(7),1)
IF(POFL) GO TO 5310
MCT = MCT-2
IF(MCT.GT.0) GO TO 5300
WRITE(6,2001) 0,HEAD
MCT = 50
5300       WRITE (6, 2002) NTAPE, XX, YY, SIG(J, L), J=1, 4),
           (SIG(J), J=5, 7), (EPS(J, L), J=1, 4),
           (EPS(J), J=5, 7)
GO TO 600
C
C..............DATA OUTPUT FOR POSTPROCESSOR
C
5310 IF(.NOT.FMFL)
1 WRITE (NTAPE1) XX, YY, SIG(5), SIG(6), SIG(7),
   EPS(5), EPS(6), EPS(7), EQSIG(L)
C
1 IF(FMFL) WRITE (NTAPE2, 2004) XX, YY, SIG(5), SIG(6), SIG(7),
   EPS(5), EPS(6), EPS(7), EQSIG(L)
600 CONTINUE
RETURN
C
C..............COMPUTE CONSISTENT MASS MATRIX.
C
5 L = D(5)
1 IF (L .LE. LINT) CALL PGAUSS (L, LINT, SG, TG, WG)
DO 500 L = 1, LINT
C..............COMPUTE SHAPE FUNCTIONS.
   CALL SHAPE (SG(L), TG(L), XL, SHP, XSJ, NDM, NEL, IX, .FALSE.)
   DV = WG(L) * XSJ * D(4)
C..............FOR EACH NODE J COMPUTE DB = RHO * SHAPE * DV.
   J1 = 1
   DO 500 J = 1, NEL
      W11 = SHP(3, J) * DV
   C..............FOR EACH NODE K COMPUTE MASS MATRIX (UPPER TRIANGULAR PART).
      K1 = J1
      DO 510 K = J, NEL
         S(J1, K1) = S(J1, K1) + SHP(3, K) * W11
      K1 = K1 + NDF
510
500 J1 = J1 + NDF
C... COMPUTE MISSING PARTS AND LOWER PART BY SYMMETRY.
NSL = NEL*NDN
DO 520 J = 1,NSL,NDN
   DO 520 K = J,NSL,NDN
      S(J+1,K+1) = S(J,K)
      S(K,J) = S(J,K)
   520 S(K+1,J+1) = S(J,K)
C
C... FORMATS FOR INPUT AND OUTPUT.
C
RETURN
1000 FORMAT(3F10.0,3I5,F5.0,2F10.0)
2000 FORMAT(/5X,3A4,1X,'TWO DIMENSIONAL ELEMENT '//'
   1 10X,'MODULUS','E18.5/10X,'POISSON RATIO','F18.5/10X,
   2 10X,'DENSITY','E18.5/10X,'GAUSS PTS/DIR','I3/10X,'STRESS PTS','I6)
2001 FORMAT(/5X,'FLEM MATERIAL','I3X,'1-COORD','I3X,'2-COORD','
   1 4X,'11-STRS','4X,'22-STRS','4X,'33-STRS','4X,'12-STRS','
   2 5X,'1-STRN','5X,'2-STRN','5X,'ANGLE','4X,'11-STRN','
   3 4X,'22-STRN','4X,'33-STRN','4X,'12-STRN')
2002 FORMAT(/5X,'NONLINEAR PLASTICITY MATERIAL'//'
   1 'HARDENING=',F16.5/
2003 FORMAT(/5X,'MODIFIED VON-MISES YIELD CRITERION'//'
   1 'C - T = ',G11.4/)
   1 'PARAMETER=',F16.5/10X,'YIELD STRESS ',F16.5/
2004 FORMAT(8G10.3)
2005 FORMAT(/5X,'MODIFIED VON-MISES YIELD CRITERION'//'
   1 'C - T = ',G11.4/)
SUBROUTINE ELMT04(D,UL,XL,IX,TL,S,P,NDF,NDM,NST,ISW)

C THREE DIMENSIONAL ELASTIC ELEMENT ROUTINE.

C COMMON/CDATA/0.HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
C COMMON/ELDATA/DM,NA,MCT,IEL,NEL
C COMMON/EPRES/ROSIG(6)
C COMMON/CLOCA/LOFL,NDME2,NTLC,NTOT,NTR
C COMMON/PDATA/POFL
C COMMON/CRoll/XR(3),ROLLRA,FRFAC,OMEGA,CUVCH,FRFL
C LOGICAL ACFL,ELAS,INCF,ISFL,NDFL,TRANS,UNFL,POFL,LOFL
C COMMON/NLDATA/ACFL,ELAS,INCF,ISFL,NDFL,TRANS,UNFL
DATA NTAPE1/30/
DIMENSION D(1),UL(NDF,1),XL(NDM,1),IX(1),TL(1),S(NST,1),
P(1),SHP(4,8),SG(8),TG(8),UG(8),WG(8),SIG(9),
EPS(9),
TSIG(NDM2,1),TEPS(NDM2,1),RATIO(1),EQSIG(1),FT(1),
UAC(1),EO(6,6),UNSIGY(1),NLOC(2,1),TRANS(1)
C
C GO TO (1,2,3,4,4).ISW
C
C READ(5,1000) E,XNU,D(4),L,K,IEMPTY,D(8),D(9),D(10)
D(1)= E*(1-XNU)/(1+XNU)/(1-2*XNU)
D(2)= XNU*D(1)/(1-XNU)
D(3)= E/2.*(1+XNU)
L = MIN0(2.*MAX0(1.L))
D(5)= L
K = MIN0(2.*MAX0(1.K))
D(6) = K
LINT = 0
WRITE(6,2000) F,XNU,D(4),L,K
IF(D(8),LT.0) WRITE(6,2000) D(9),D(10)
RETURN
C 2 RETURN
C 3 L = D(5)
   IF(L*L,L.NE.LINT) CALL GAUSS3(L,LINT,SG,TG,UG,WG)
C C.............FOR EACH INTEGRATION PT. COMPUTE CONTRIBUTION TO STIFFNESS.
C DO 100 L = 1,LINT
   CALL SHAPE3(SG(L),TG(L),UG(L),XL,SHP,XSJ,NDM,NEL,IX,.FALSE.)
   DV = XSJ*WG(L)
   D1 = D(1)*DV
   D2 = D(2)*DV
   D3 = D(3)*DV
C C.............CHECK IF ANY NONLINEAR PROCESS.
C IF(ELAS.OR.D(8).GE.0) GO TO 80
   IF(RATIO(L).GE.0.0) GO TO 80
C C TRULY NONLINEAR, POST YIELD POINT
C CALL EPMAT(NDM2,DV,D,EOSIG(L),TSIG(1,L),EP)
C FOR EACH J NODE COMPUTE DEPB = DEP*B
C DO 100 J = 1,NEL
   FRTM = 1.0
C C CHECK ANY FRICTION EFFECT
C IF(.NOT.FRFL) GO TO 79
   DO 77 JK = 1,NTOTAL
      IF(NLOC(1,JK).EQ.IX(J).AND.NLOC(2,JK).NE.0)
         GO TO 78
   77 CONTINUE
C 78 C.
CONTINUE
GO TO 79

C
FRTM = FRFAC
CONTINUE
J3 = J * J

C
DB11 = (EP(1,1) * SHP(1,.J) + EP(1,4) * SHP(2,.J) + EP(1,6) * SHP(3,.J)) * FRTM
1
DB12 = (EP(1,2) * SHP(2,.J) + EP(1,4) * SHP(1,.J) + EP(1,5) * SHP(3,.J)) * FRTM
1
DB13 = (EP(1,3) * SHP(3,.J) + EP(1,5) * SHP(2,.J) + EP(1,6) * SHP(1,.J)) * FRTM
1

C
DB21 = (EP(2,1) * SHP(1,.J) + EP(2,4) * SHP(2,.J) + EP(2,6) * SHP(3,.J)) * FRTM
1
DB22 = (EP(2,2) * SHP(2,.J) + EP(2,4) * SHP(1,.J) + EP(2,5) * SHP(3,.J)) * FRTM
1
DB23 = (EP(2,3) * SHP(3,.J) + EP(2,5) * SHP(2,.J) + EP(2,6) * SHP(1,.J)) * FRTM
1

C
DB31 = (EP(3,1) * SHP(1,.J) + EP(3,4) * SHP(2,.J) + EP(3,6) * SHP(3,.J)) * FRTM
1
DB32 = (EP(3,2) * SHP(2,.J) + EP(3,4) * SHP(1,.J) + EP(3,5) * SHP(3,.J)) * FRTM
1
DB33 = (EP(3,3) * SHP(3,.J) + EP(3,5) * SHP(2,.J) + EP(3,6) * SHP(1,.J)) * FRTM
1

C
DB41 = (EP(4,1) * SHP(1,.J) + EP(4,4) * SHP(2,.J) + EP(4,6) * SHP(3,.J)) * FRTM
1
DB42 = (EP(4,2) * SHP(2,.J) + EP(4,4) * SHP(1,.J) + EP(4,5) * SHP(3,.J)) * FRTM
1
DB43 = (EP(4,3) * SHP(3,.J) + EP(4,5) * SHP(2,.J) + EP(4,6) * SHP(1,.J)) * FRTM
1

C
DB51 = (EP(5,1) * SHP(1,.J) + EP(5,4) * SHP(2,.J) + EP(5,6) * SHP(3,.J)) * FRTM
1
DO 100 J = 1, NEL
   FRTM = 1.0
   CHECK ANY FRITCITON EFFECT
   IF (.NOT. FRFL) GO TO 89
       DO 87 JK = 1, NTOTAL
         IF (NLOC(1*JK) .EQ. IX(J) .AND. NLOC(2*JK) .NE. 0) GO TO 88
       CONTINUE
   GO TO 89
  88   FRTM = FRFAC
  89 CONTINUE.
   J3 = J * 3
   DB11 = (D1*SHP(1,J)) * FRTM
   DB12 = (D2*SHP(2,J)) * FRTM
   DB13 = (D2*SHP(3,J)) * FRTM
   DB21 = (D2*SHP(1,J)) * FRTM
   DB22 = (D1*SHP(2,J)) * FRTM
DB2 = DB13
DB31 = DB21
DB32 = DB12
DB33 = (D3*SHP(3,J))*FRTM
DB41 = (D3*SHP(2,J))*FRTM
DB42 = (D3*SHP(1,J))*FRTM
DB43 = 0.0
DB51 = 0.0
DB52 = (D3*SHP(3,J))*FRTM
DB53 = DB41
DB61 = DB52
DB62 = 0.0
DB63 = DB42
90 CONTINUE
C
C............FOR EACH I NODE COMPUTE S = BT*DB
C
DO 100 I = 1,J
   I3 = I*3
   S(I3-2,J3-2) = S(I3-2,J3-2) + SHP(1,I)*DB11 + SHP(2,I)*DB41
   S(I3-2,J3-1) = S(I3-2,J3-1) + SHP(3,I)*DB61
   S(I3-1,J3-1) = S(I3-1,J3-1) + SHP(2,I)*DB22 + SHP(3,I)*DB52
   S(I3-1,J3 ) = S(I3-1,J3 ) + SHP(1,I)*DB13 + SHP(3,I)*DB63
   S(I3 ,J3-2) = S(I3 ,J3-2) + SHP(3,I)*DB31 + SHP(1,I)*DB61
   S(I3 ,J3-1) = S(I3 ,J3-1) + SHP(3,I)*DB32 + SHP(2,I)*DB52
100 CONTINUE
100  
  
\[ S(I3 \cdot J3) = S(I3 \cdot J3) + \text{SHP}(3 \cdot I) \times \text{DB33} + \text{SHP}(2 \cdot I) \times \text{DB53} + \text{SHP}(I \cdot I) \times \text{DB63} \]

C CONVERT FOR LOCALLY ORIENTED AXIS

C IF(LOFL) CALL STCNV(NLOC,IX,TRANF,SNST,NEL,NDM)

C COMPUTE LOWER TRIANGULAR PART BY SYMMETRY

C NL = NEL \times 3
DO 200 I = 2, NL
DO 200 J = 1, I
200  S(I,J) = S(J,I)
RETURN

C

4 L = D(5)
IF(ISW.EQ.4) L = D(6)
IF(L.LT.L.INT) CALL GAUSS3(L,L.INT,SG,TG,UG,WG)

C COMPUTE ELEMENT STRESSES, STRAINS AND FORCES.

C DO 600 L = 1, L.INT
CALL SHAPE3(SG(L),TG(L),UG(L),XL,SHP,XSJ,NDM,NEL,IX,.FALSE.)

C CHECK UNLOAD AND PRINT STRESS

C IF(ISW.EQ.4.AND.UNFL) GO TO 618
IF(ISW.EQ.4.AND.(NOT.ELAS)) GO TO 618

C COMPUTE STRAINS AND COORDINATES.

C DO 410 I = 1, 6
410  EPS(I) = 0.0
XX = 0.0
YY = 0.0
ZZ = 0.0
DO 415 J = 1, NEL
    XX = XX + SHP(4,J) * XL(1,J)
    YY = YY + SHP(4,J) * XL(2,J)
    ZZ = ZZ + SHP(4,J) * XL(3,J)
415 CONTINUE

C FOR UNLOAD AND REACTION PHASE NO NEED TO CALCULATE
C STRESS AND STRAIN AGAIN. THE TOTAL STRESS IS USED
C INSTEAD. (REF. "RESID").
C
IF(UNFL .OR. NDFL) GO TO 421
C
DO 420 J = 1, NEL
    EPS(1) = EPS(1) + SHP(1,J) * UL(1,J)
    EPS(2) = EPS(2) + SHP(2,J) * UL(2,J)
    EPS(3) = EPS(3) + SHP(3,J) * UL(3,J)
    EPS(4) = EPS(4) + SHP(2,J) * UL(1,J) + SHP(1,J) * UL(2,J)
    EPS(5) = EPS(5) + SHP(3,J) * UL(2,J) + SHP(2,J) * UL(3,J)
    EPS(6) = EPS(6) + SHP(3,J) * UL(1,J) + SHP(1,J) * UL(3,J)
420 CONTINUE

C IF(ELAS) GO TO 422
C
IF(ISW .EQ. 7) GO TO 422

421 CALL RESID(D(1), D(2), EPS(1:6), EPS(1:6), TSIG(1:6), RATIO(1:6), NDM2)
    GO TO 620
C
C............. COMPUTE STRESSES.
C
422 CONTINUE
    SIG(1) = D(1) * EPS(1) + D(2) * (EPS(2) + EPS(3))
    SIG(2) = D(1) * EPS(2) + D(2) * (EPS(1) + EPS(3))
    SIG(3) = D(1) * EPS(3) + D(2) * (EPS(1) + EPS(2))
    SIG(4) = D(3) * EPS(4)
    SIG(5) = D(3) * EPS(5)
    SIG(6) = D(3) * EPS(6)
C
IF(ISW.EQ.7) GO TO 615
IF(ISW.EQ.6) GO TO 620

C........OUTPUT STRESSES AND STRAINS.
C
MCT = MCT - 2
IF(MCT.GT.0) GO TO 430
WRITE(6,2001) 0,HEAD
MCT = 50
430 WRITE(6,2002) N,MA,XX,YY,ZZ,(SIG(JP),JP=1,6),
1 (EPS(JP),JP=1,6)
GO TO 600

C
C........COMPUTE INTERNAL FORCES.
C
620 DV=X5J*WG(L)
Jl=1
DO 610 J = 1,NEL
C
CHECK ANY FRICTION EFFECT
C
IF(.NOT.FRFL) GO TO 606
DO 602 JK = 1,NTOTAL
1 IF(NLOC(l.JK).EQ.IX(J).AND.NLOC(2.JK).NE.0)
1 GO TO 604
602 CONTINUE
GO TO 606

C
MODIFY THE NODE FOR FRICTION EFFECT IN ROLLING
C
604 P(J1 )=P(J1 )-(SHP(1.J)*ROSIG(1)+SHP(2.J)*ROSIG(4)
1 +SHP(3.J)*ROSIG(6))**DV
P(J1+1)=P(J1+1)-(SHP(2.J)*ROSIG(2)+SHP(1.J)*ROSIG(4)
1 +SHP(3.J)*ROSIG(5))**DV
P(J1+2)=P(J1+2)-(SHP(3.J)*ROSIG(3)+SHP(2.J)*ROSIG(5)
1 +SHP(1.J)*ROSIG(6))**DV
GO TO 608
CONTINUE

P(J1 ) = P(J1 ) -(SHP(1,J)*SIG(1) + SHP(2,J)*SIG(4) + SHP(3,J)*SIG(6)) * DV
1

P(J1 +1) = P(J1 +1 ) -(SHP(2,J)*SIG(2) + SHP(1,J)*SIG(4) + SHP(3,J)*SIG(5)) * DV
1

P(J1 +2) = P(J1 +2 ) -(SHP(3,J)*SIG(3) + SHP(2,J)*SIG(5) + SHP(1,J)*SIG(6)) * DV
1

CONTINUE

J1 = J1 + NDF

CONTINUE
GO TO 600

CALL PRATIO(EPS, SIG, D, TSIG(1,L), RATIO(L), EQSIG(L),
1
UNSIGY(L), NDM2)
GO TO 600

C

C

C

C

C

C

C

C

NONLINEAR STRESS/STRAIN OUTPUT

XX = 0.0
YY = 0.0
ZZ = 0.0

DO 6180 J = 1, NEL
  XX = XX + SHP(4,J)*XL(1,J)
  YY = YY + SHP(4,J)*XL(2,J)
  ZZ = ZZ + SHP(4,J)*XL(3,J)

6180 CONTINUE

IF(PDFL) GO TO 5310
MCT = MCT - 2
IF(MCT .GT. 0) GO TO 5300
WRITE(6,2001) , HEAD
MCT = 50

5300 WRITE(6,2002) , MA, XX, YY, ZZ, (TSIG(JP,L), JP=1,6),
1 (TEPS(JP,L), JP=1,6)
GO TO 600

C
C...........STRESS STRAIN OUTPUT FOR POSTPROCESSOR
C
5310 WRITE(NTAPE1) XX,YY,ZZ,(TSIG(JP,L),JP=1,6),(TEPS(JP,L),JP=1,6)
C
600 CONTINUE
   RETURN
5 RETURN
C
C...........FORMATS FOR INPUT AND OUTPUT.
C
1000 FORMAT(3F10.0,3I5,F5.0,2F10.0)
2000 FORMAT(/5X,*3-D SOLID ELEMENT */10X,*MODULUS**,E18.5/
   1 10X,*POISSON RATIO**,F8.5/10X,*DENSITY**,E18.5/10X,
   2 *GAUSS POINTS **,13/10X,*STRESS PTS**,16)
2001 FORMAT(A1,20A4//5X,*ELEMENT STRESSES**/* ELEMENT MATERIAL**,
   1 2X,*1-CORD**,6X,*2-CORD**,6X,*3-CORD**,6X,*11-STRS**,
   1 4X,*22-STRS**,4X,*33-STRS**,4X,*12-STRS**,4X,
   1 *23-STRS**,4X,*31-STRS**/,
   1 4X,*23-STRN**,4X,*31-STRN**)
2002 FORMAT(2I10,3G12.4,4G12.4/5X,**ELEMENT MATERIAL**//10X,
   1 *HARDENING PARAMETER**,F16.5/10X,*YIELD STRESS **,
   2 F16.5//)
END
SUBROUTINE EPMAT(NDM2,DV,D,EQSIG,TSIG,EP)

******************************************************************************

PROGRAM NAME : EPMAT (ELASTO-PLASTIC MATRIX)

PURPOSE : CALCULATE DEP = D - DP FOR 2-D AND 3-D ELEMENTS.

CALLED BY : 1. "RESID" ("PFORM", MACRO 'FORM')
2. "INSTS" ("PLASTIC", MACRO 'PLAS')
3. "TANG" ("PFORM", MACRO 'TANG', ISW = 3)

OUTPUT : EP(NDM2,NDM2)

******************************************************************************

LOGICAL PMFL
COMMON /PMDATA/PMFL
DIMENSION D(1),TSIG(1),EP(6,6),S(6)

CALCULATE MEAN AND DEVIATORIC STRESSES.

SM = (TSIG(1)+TSIG(2)+TSIG(3))/3.0
S1 = TSIG(1) - SM
S2 = TSIG(2) - SM
S3 = -(S1+S2)
S4 = TSIG(4)

CALCULATE CONSTANT FOR PLASTIC MODULOUS.

H = D(9)
G = D(3)
D1 = D(1)*DV
D2 = D(2)*DV
D3 = D(3)*DV
\[
SS = 3.0*G/(EQ*2)/(H/3.0/G + 1.0)*DV
\]

C............CALCULATE UPPER HALF OF EPHAT.

C

EP(1,1) = D1 -S1*S1*SS
EP(1,2) = D2 -S1*S2*SS
EP(1,3) = D2 -S1*S3*SS
EP(1,4) = -S1*S4*SS
EP(2,2) = D1 -S2*S2*SS
EP(2,3) = D2 -S2*S3*SS
EP(2,4) = -S2*S4*SS
EP(3,3) = D1 -S3*S3*SS
EP(3,4) = -S3*S4*SS
EP(4,4) = D3 -S4*S4*SS

C............FOR PLANE STRESS, THE THIRD COLUMN AND ROW IS ZERO.

C

IF(D(7).GE.0 .OR.NDM2 .EQ. 6) GO TO 5
DO 1 I = 1,3
   EP(I,3) = 0.0
1 CONTINUE
   EP(3,4) = 0.0

C 5 CONTINUE

C.............CHECK 2-D FOR 3-D.

C

IF(NDM2.EQ.4) GO TO 10
SS = TSIG(5)
S6 = TSIG(6)
EP(1,5) = -S1*S5*SS
EP(1,6) = -S1*S6*SS
EP(2,5) = -S2*S5*SS
EP(2,6) = -S2*S6*SS
EP(3,5) = -S3*S5*SS
EP(3,6) = -S3*S6*SS
EP(4,5) = -S4*S5*SS
EP(4,6) = -S4*S6*SS

GO TO 10
\[
\begin{align*}
EP(4,6) &= -S4 \times S6 \times SS \\
EP(5,5) &= D3 -S5 \times S5 \times SS \\
EP(5,6) &= -S5 \times S6 \times SS \\
EP(6,6) &= D3 -S6 \times S6 \times SS \\
\end{align*}
\]

C
\[\text{FILL IN THE LOWER HALF BY SYMMETRY}\]

C
\[
10 \text{ DO } 20 \ I = 1, NDM2 \\
\text{IM1} = I - 1 \\
\text{DO } 20 \ J = 1, \text{IM1} \\
\text{EP}(I, J) = \text{EP}(J, I) \\
20 \text{ CONTINUE}
\]

C
\[
S(1) = S1 \\
S(2) = S2 \\
S(3) = S3 \\
S(4) = S4 \\
\text{IF(NDM2, EQ, 4) GO TO 30} \\
S(5) = S5 \\
S(6) = S6
\]

30 \text{ CONTINUE} \\
\text{IF(PMFL) CALL .MVMAT(NDM2, DV, D, EQSIG, TSG, SIG, SM, S, EP)}

C
\[
\text{RETURN} \\
\text{END}
\]
FUNCTION EQSTS(TSIG,NDM2,D)
LOGICAL PMFL
COMMON/PMDATA/PMFL
DIMENSION D(I)

C
C A FUNCTION TO CALCULATE "EQUIVALENT STRESS" FOR VON- MISES
C YIELD CRITERION. (J2)
C
DIMENSION TSIG(I)
S1M2 = (TSIG(1)-TSIG(2)) ** 2
S2M3 = (TSIG(2)-TSIG(3)) ** 2
S3M1 = (TSIG(3)-TSIG(1)) ** 2
S12 = 6.0*TSIG(4) ** 2
S = S1M2+S2M3+S3M1+S12

C PRESSURE MODIFIED VON MISES YIELD CRITERION
C
IF(NDM2.EQ.0.4) GO TO 10
S = S+6.0*(TSIG(5)**2 + TSIG(6)**2)
10 EQSTS = 1.0/SQR(2.0)*SQR(S)
RETURN
END
SUBROUTINE EUPDAT(DR,U,V,A,XM,DT,NEQ)
DIMENSION U(1),V(1),A(1),DR(1),XM(1)

C........UPDATE SOLUTION USING EXPLICIT CENTRAL DIFFERENCES.
DATA DTHP/0.0/
DTH = DT/2.
DTAV = DTH + DTHP
DTHP = DTH
DO 100 N = 1,NEQ
   A(N) = DR(N)/XM(N)
   V(N) = V(N) + DTAV*A(N)
100 U(N) = U(N) + DT*V(N)
RETURN
END
SUBROUTINE GAUSS3(L,LINT,R,Z,U,W)
C
C..........GAUSS POINTS & WEIGHTS FOR THREE DIMENSIONS.
C
DIMENSION LR(8),LZ(8),LU(8),R(I),Z(I),U(I),W(I)
DATA LR/-1.1.1.1.1.1.1.1./,
        LZ/-1.1.1.1.1.1.1.1./,
        LU/1.1.1.1.1.1.1.1./,
LINT = L*L*L
GO TO (1,2),L
C..........1X1X1 INTEGRATION.
   1 R(I) = 0.
   Z(I) = 0.0
   U(I) = 0.0
   W(I) = 8.0
RETURN
C..........2X2X2 INTEGRATION.
   2 G = 1./SORT(3.)
   DO 21 I = 1,8
      R(I) = G*LR(I)
      Z(I) = G*LZ(I)
      U(I) = G*LU(I)
   21 W(I) = 1.
RETURN
END
SUBROUTINE GENVEC (NDM,X,CD,PRT,ERR)

C
C...........GENERATE REAL DATA ARRAYS BY LINEAR INTERPOLATION
C

LOGICAL PRT,ERR,PCOMP
COMMON/CDATA/0.HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
DIMENSION X(NDM,1),XL(7),CD(2)
DATA BL/4HBLAN/
N = 0
NG = 0
102 L = N
LG = NG
READ(5,1000) N,NG,XL
IF(N.LE.0.OR.N.GT.NUMNP) GO TO 108
   DO 103 I = 1,NDM
   X(I,N) = XL(I)
   IF(LG) 104,102,104
   LG = ISIGN(LG,N-L)
   LI = (IABS(N-L*LG)-1)/IABS(LG)
   DO 105 I = 1,NDM
   XL(I) = (X(I,N)-X(I,L))/LI
   105 XL(I) = (X(I,N)-X(I,L))/LI
   L = L + LG
   IF((N-L)*LG.LE.0) GO TO 102
   IF(L*LE.0.OR.L.GT.NUMNP) GO TO 110
   DO 107 I = 1,NDM
   X(I,L) = X(I,L-LG) + XL(I)
   GO TO 106
106 WRITE(6,3000) L,(CD(I),I=1,3)
   FRR = .TRUE.
   GO TO 102
108 DO 109 I = 1,NUMNP,50
   IF(PRT) WRITE(6,2000) 0,HEAD,(CD(L),L=1,3),(L,CD(1),CD(2),L=1,NDM)
   N = MIN0(NUMNP,1+49)
   DO 109 J = 1,N
   IF(PCOMP(X(I,J),BL).AND.PRT) WRITE(6,2008) N
109 IF(.NOT.PCOMP(X(I,J),BL).AND.PRT) WRITE(6,2009) J,(X(L,J),L=1,NDM)
RETURN
1000 FORMAT(2I5.7F10.0)
2000 FORMAT(A1,20A4//5X,5HNODEAL,3A4//6X,4HNODE,9(I7,A4,A2))
2008 FORMAT(I10,* HAS NOT BEEN INPUT OR GENERATED*)
2009 FORMAT(I10,9F13.4)
3000 FORMAT(5X,* FATAL ERROR 02* ATTEMPT TO GENERATE NODE*,15*,
1 I IN*,3A4)
END
SUBROUTINE GEORF (DE, J1, SHP, DV, P, RRINV)
C
C NAME : GEOMETRICAL FORCE
C
C CALLED BY : ELMTO1 BY MACRO FORM WHEN UPLRFL = .TRUE.
C
C CALCULATE UNBALANCED FORCE FOR GEOMETRICALLY
C NONLINEAR PROBLEMS.
C
C USE VALUES CALCULATED FROM WRTBUD, DSTARB
C
C OPERATE AT THE LEVEL OF EACH NODE
C
C DIMENSION DE(3,3), SHP(1), P(1)
C
P(J1) = P(J1) - (DE(1,1)*SHP(1)+DE(1,2)*SHP(2)
1 +DE(3,3)*SHP(3)*RRINV) * DV
P(J1+1) = P(J1+1) - (DE(2,1)*SHP(1) + DE(2,2)*SHP(2)*DV
C
RETURN
END
SUBROUTINE GET3DU(EPGR)

C

PURPOSE : DERIVE B*D from TAPE 18

C

NAME : GET B*D

C

NOTE : OPERATE AT THE LEVEL OF GAUSS POINT

C

CALLED BY : INSTS (PLASTI) BY MACRO 'PLAS'

C

ELMT01 BY MACRO 'FORM'

C

TAPE USED : 18

C

DIMENSION EPGR(3,3)

C

DO 1 I = 1,2

READ(18,ERR=100) (EPGR(I,K), K = 1,2)

1 CONTINUE

READ(18) EPGR(3,1)

C

RETURN

C

RETURN

C

FIRST TIME THRU THE LOOP FILE 18 HAS NOT BEEN WRITTEN YET

C

100 RETURN

END
SUBROUTINE INSTS(TSIG, EPS, EDOSIG, D, RATIO, UNSIGY, NGAUP, NDM2, 1 RMIN, JUNLD)

C
C---------------------------------------------------------------
C
C PROGRAM NAME : INSTS (INITIAL STRESS)
C
C PURPOSE : 1. ACCUMULATING TOTAL STRESSES AND STRAINS
C FOR BOTH TANGENTIAL AND INITIAL STRESS
C METHOD.
C 2. ADJUSTING STRESS AND STRAIN INCREMENT IF
C NECESSARY DUE TO TRANSITION ZONE.
C 3. CALCULATE EQUIVALENT STRESS.
C 4. DECIDE WHETHER THE CURRENT INTGT. PT. IS--
C (A). ELASTIC
C (B). PLASTIC ; RATIO = -ABS(RATIO)
C (C). ELASTIC OF POST-YIELD PT. (UNLOAD)
C RATIO = ABS(RATIO)
C
C CALLED BY : PLASTIC (MACRO "PLAS")
C
C MAJOR INPUT : (1). RATIO -- IS CURRENT FROM "PRATIO".
C (2). EOSIG -- IS FROM THE LAST INCREMENT
C (ITERATION), TO BE UPDATED IN THIS ROUTINE
C (3). EPS AND SIG ARE CURRENT ELASTIC VALUES FROM
C "ELMT01" BY MACRO "PLAS".
C
C THEORY : INITIAL STRESS METHOD AND INCREMENTAL METHOD.
C
C MODULE CALLED : "EPMAT".
C
C LOGICAL UNIT : (1) 11 FOR STORING PLASTIC STRESS INCRE.
C (2) 12 FOR STORING STRAIN AND STRESS INCRE.
C (3) 15 FOR STORING AND RETRIEVING YIELD
C STRESS (FOR UNLOADED POINT WHICH HAS
PREVIOUSLY YIELDED IT IS THE PREVIOUS
EQUIVALENT STRESS. FOR ELASTIC MATERIAL
IT IS 0.0. OTHERWISE INITIAL YIELD POINT.

TOLERANCES : YTOL = FOP CHECKING ELASTIC/PLASTIC
UNTOL = FOP CHECKING UNLOAD.

DIMENSION TSIG(NDM2,1), TEPS(NDM2,1), E0SIG(I), D(1), RATIO(I),
1 EP(6,6), EPS(6), SIG(6), EPSG(6), UNSIGY(I), ACCSIG(6),*
2 EPGR(3,3)
LOGICAL ACFL, ELAS, INCFL, ISFL, NDFL, TRANS, UNFL, PMFL, UPLRFL
COMMON/NLDATA/ACFL, ELAS, INCFL, ISFL, NDFL, TRANS, UNFL
COMMON/PMDATA/PMFL
COMMON/UPDATE/UPLRFL
COMMON/ACDATA/S
DATA YTOL/-0.001/, UNTOL/-0.015/, ONTL/-0.001/, TRTOL/0.05/
DATA RUNLD/12345.6789/

DO 250 L = 1, NGAUP

....RESTORE FROM AUX. STORAGE STORED IN "PRATIO" CALLED BY
"ELMTON", "PFORM" WITH ISW = 7, MACRO "PLAS".

READ(12) (EPS(J), J=1, NDM2), (SIG(J), J=1, NDM2)

IF(.NOT.TRANS,.OR. (.NOT.INCFL)) GO TO 20

INCREMENTAL METHOD ONLY.

DO 10 I = 1, NDM2
EPS(I) = EPS(I)*RMIN
SIG(I) = SIG(I)*RMIN
CONTINUE

20 DO 30 I = 1, NDM2
   TSIG(I,L) = TSIG(I,L) + SIG(I)
   TEPSE(I,L) = TEPSE(I,L) + EPS(I)
30 CONTINUE

IF(D(8).GE.0) GO TO 200
   TEMP = EQSIG(L)
   EQSIG(L) = EQSTS(TSIG(I,L), NDM2, D)
C
SIGY = UNSIGY(L)
   IF(PMFL) SIGY = PMSIGY(D, TSIG(I,L))
C FIRST CHECK WHETHER AN ELASTIC POINT
C
IF(((EQSIG(L)-SIGY)/SIGY .LT. YTOL .AND. RATIO(L).GE.0.0)
   GO TO 35
   GO TO 45
C
C EVEN IF IT IS ELASTIC, WE STILL LIKE TO SEE IF WE
C COULD YIELD SEVERAL POINTS TOGETHER TO SAFE TIME.
C VALID ONLY FOR TANGENTIAL STIFF. METHOD
C
35 IF(INCFL. AND. RATIO(L).EQ. RUNLD) GO TO 200
   IF(INCFL. AND. (RATIO(L)-RMIN)/RMIN.LE.TRTOL) GO TO 45
   GO TO 200
C
45 CONTINUE
   IF(INCFL. AND. RATIO(L).EQ. RUNLD) GO TO 300
C
C ONLY PLASTIC POINTS ARE PROCESSED HERE.
C /** NOTE -- TRANSITION ZONE CORRECTION OF RMIN IS
C ALREADY FINISHED IN "PLASTIC" AND THE ABOVE PART OF
C THIS SUBROUTINE. NOW THE CORRECTION IS TO BE PERFORMED
C FOR INITIAL STRESS METHOD ONLY. **/
SUBNOTE -- FOR FIRST TIME YIELD, DO NOT CALCULATE
ELASTIC-PLASTIC STRESSES, SINCE THERE IS NO PLASTIC
STRESS ON THE INITIAL YIELD SURFACE. CONDITION IS
SIGNALED BY RATIO GREATER THAN 0.0 (NOT YET PLASTIC),
HOWEVER COULD BE LARGER THAN ONE BECAUSE OF POSSIBLE
SIMULTANEOUSLY YIELDING OF SEVERAL POINTS TOGETHER.

THIS APPLY TO INCREMENTAL METHOD ONLY, FOR INITIAL
STRESS METHOD THIS IS TAKEN CARE OF AUTOMATICALLY.

IF(RATIO(L).GE.0.0.AND.INCFL) GO TO 190
C
RTEMP = RATIO(L)
IF(RTEMP.LT.0.0.OR..NOT.TRANS) GO TO 50
    IF(.NOT.INCFL.AND. RTEMP.LT.1.0) GO TO 60
    RTEMP = 0.0
    CONTINUE
    RTEMP = 1.0 - RTEMP
    CALL EPMAT(NDM2,1.0,DEOSIG(L),TSIG(1,L),EP)
    DO 80 I = 1,NDM2
        EPSG(I) = 0.0
        DO 70 J = 1,NDM2
            EPSG(I) = EPSG(I)+EP(I,J)*EP5(J)
        CONTINUE
    CONTINUE
ACCSIG IS (I + S) SIGE
    DO 85 I = 1,NDM2
        ACCSIS(I) = RTEMP » SIG(I)
    CONTINUE
STORE PLASTIC STRESS IN AUX. STORAGE SPACE.
    DO 90 I = 1,NDM2
        SIG(I) = RTEMP »(SIG(I)-EPSG(I))
    CONTINUE
DO 100 I = 1,NDM2
    TSIG(I,L) = TSIG(I,L) - SIG(I)
100    CONTINUE

C..................ADJUSTING OF ACCELERATION STRESS (PLASTIC STRESS).
C
    IF(.NOT.ACFL) GO TO 107
C
    SDSP1 = S / (S + 1.0)
    DO 105 I = 1,NDM2
       SIG(I) = SIG(I) - SDSP1 * ACCSIG(I)
105    CONTINUE

107    WRITE(11) (SIG(I),I = 1,NDM2)
C
    EOSIG(L) = EQSTS(TSIG(I,L),NDM2,D)
    IF(PMFL.AND.RATIO(L).GT.0.0) UNSIGY(L) = SIGY
C
    FOR PLASTIC POINTS CHECK IF UNLOADING HAS OCCURED.
C
    IF(TEMP.EQ.0.0) GO TO 190
    IF((EOSIG(L)-TEMP)/TEMP.LE.UNTOL) GO TO 110
    GO TO 190
C
C..................UNLOAD OF POST-YIELD ELEMENT.
C
110    RATIO(L) = ABS(RATIO(L))
    JUNLD= JUNLD + 1
C
    CHANGE CURRENT YIELD STRESS HERE STRAIN HARDENING
C
    FOR POINTS RIGHT ON THE YIELD SURFACE DO NOT TREAT
C
    IT AS UNLOADING OF POST YIELD POINT RATHER AS
C
    ELASTIC UNLOADING - DO NOT CHANGE THE YIELD STRESS.
C
    IF(TEMP.LE.UNSIGY(L)) GO TO 200
    UNSIGY(L) = TEMP
    GO TO 200
C
C REMEMBERS THAT THIS POINT HAS ALREADY YIELDED
C     BY ASSIGNING A NEGATIVE NUMBER.
C
190  RATIO(L) = -ABS(RATIO(L))
200 IF(.NOT.UPLRFL) GO TO 250
     CALL GETBDU(EPGR)
     CALL JAUMAN (TSIG(1),L,EPGR)
250  CONTINUE
     RETURN
C
C ERROR EXIT
C
300  WRITE(6,1000) SIGY,EOSIG(L)
1000 FORMAT(/5X,'COLLAPSE OF AN UNLOADED POINTED'/10X,'PREVIOUS',
          1     ' YIELD STRESS IS',G10.3/10X,'CURRENT YIELD STRESS',
          2     ' IS ',G10.3)
     CALL EXIT
     END
SUBROUTINE JAUMAN (TSIG, EPGR)

C
C PURPOSE: CONVERT JAUMAN RATE OF KIRCHHOFF STRESS TO
C INTRINSIC RATE OF EULER STRESS
C
C NOTE: ONLY WORKS NOW FOR TWO DIMENSIONAL CASES
C
C DIMENSION EPGR(3,3), TSIG(1)
C
C TEMP1 = TSIG(1)
TEMP2 = TSIG(2)
W12 = 0.5*(EPGR(1,2) - EPGR(2,1))
TSIG(1) = TSIG(1) - 2.0 * W12 * TSIG(4)
TSIG(2) = TSIG(2) + 2.0 * W12 * TSIG(4)
TSIG(4) = TSIG(4) + W12 * (TEMP1 - TEMP2)
C
RETURN
END
SUBROUTINE LOCAL (NLOC, TRANF, NPR)

PURPOSE: INPUT NUMBER OF NODE, NUMBER OF TRANSFORMATION

MAJOR VARIABLES:

NLOC = ARRAY OF LENGTH NTLC*2. NLOC(1, J) STORES THE
NODE NUMBER, NLOC(2, J) STORES THE ASSOCIATED
NUMBER OF TRANSFORMATION MATRIX

NTLC = MAXIMUM NUMBER OF NODES THAT CAN BE LOCALLY
ORIENTED

NPR = 0 DO NOT PRINT INPUT INFORMATION.
    .NE.0 PRINT

TRANF = ARRAY OF LENGTH NDME2*NTLC. TRANF(NDME2, NTLC)
CONTAINS THE DIRECTION COSINE OF THE LOCAL
AXIS. E.G. U = L * U'. WHERE ' (PRIME) IS THE
LOCAL COORDINATE

INPUT IS IN THE ORDER OF 1ST ROW, 2ND ROW ETC.

NTR = ACTUAL NUMBER OF TRANSFORMATION MATRIX INPUT

NDME = NDM * NDM (E.G. FOR 2D, NDME = 4, 3D = 9)

NOTE: TERMINATING THE INPUT OF TRANSFORMATION MATRIX WITH
A BLANK CARD.

ROUTINES CALLED: "BCINPU".

CALLED BY: "PMACR" OF MACRO "LOCA".

COMMON /LOCA/ LOFL,NOME2,NTLC,NTOTAL,NTR
LOGICAL LOFL
DIMENSION NLOC(2,1),TRANF(NDME2,1)
DATA BL/4H/

INPUT NODE AND ITS TRANSFORMATION MATRIX NUMBER.

CALL BCINPU(NLOC,2,1,NTOTAL)

INPUT THE COEFFICIENTS OF THE TRANSFORMATION MATRIX

NTR = 0
10 READ(5,1001) NI
   IF(NI.EQ.0 OR NI.EQ.BL) GO TO 20
      READ(5,1002) (TRANSF(J,NI),J = 1,NDME2)
      NTR = NTR + 1
   GO TO 10

20 IF(NTR.EQ.0) GO TO 50
25 IF(NPR.EQ.0) RETURN

PRINT INPUT DATA

WRITE(6,1003)
   DO 30 I = 1,NTOTAL
      WRITE(6,1004) (NLOC(J,I),J = 1,2)
   30 CONTINUE
   IF(NTR.EQ.0) RETURN
C PRINT TRANSFORMATION MATRIX

WRITE(6,1005)
DO 40 I = 1,NTR
   40 WRITE(6,1006) I,(TRANF(J,I), J = 1,NDME2)
RETURN

C 50 WRITE(6,1007)
GO TO 25

C FORMAT GROUP

C

1001 FORMAT(16I5)
1002 FORMAT(9F8.0)
1003 FORMAT(/5X,"LOCALLY ORIENTED NODES"/10X,4HNODE.6X,
   1 15HTRANSF. MAT.NO.)
1004 FORMAT(10X,15)
1005 FORMAT(/10X,15HTRANSF. MAT.NO.,10X,"COEFFICIENTS")
1006 FORMAT(20X,15.9G9.6)
1007 FORMAT(/5X,"*** WARNING *** ERROR IN LOCALLY ORIENTED NODE INPUT*
   1 "*/15X,"ZERO TRANSFORMATION MATRIX*/")

END
SUBROUTINE MVMAT(NDM2,DV,D,EOSIG,TSIG,SM,S,EP)
C
C PROGRAM NAME : MVMAT (MODIFIED VON MISES MATRIX)
C PURPOSE : DEP = (D-EP) - MVMAT
C CALLED BY : EPMAT
C FUNCTION CALLED : PMSIGY(D,TSIG)
C OUTPUT : EP(NDM2,NDM2)
C
DIMENSION S(l),ED(6,6),D(1),TSIG(n)

EOSIGP = PMSIGY(D,TSIG)
Z = D(9) + 3.0*D(3)
SST = 3.0/2.0*D(3)/EOSIGP*D(12)/ED(1)+2.0*D(2))/Z*DV

DO 20 J = 1,NDM2
   DO 10 I = 1,3
      EP(J,I) = EP(J,I) - S(J)*SST
   10 CONTINUE
20 CONTINUE

3RD ROW AND COLUMN FOR PLANE STRESS IS ZERO
IF(D(7),GE.0.0 OR NDM2.EQ.6) RETURN

DO 30 I = 1,4
   EP(I,3) = 0.0
30 CONTINUE

RETURN
END
SUBROUTINE NEWOC(NDM, XR, X, TRANF)
C
C
C NAME : NEW DIRECTION COSINE
C
C INPUT : CURRENT COORD. AND COORD. OF CENTER OF ROLL.
C
C OUTPUT : DIRECTION COSINES IN THE TRANSFORMATION ARRAY.
C
C CALLED BY : ROLBC.
C
C
DIMENSION XR(I), X(I), TRANF(1)
C
ALPHA = ATAN2((XR(2)-X(2)),(XR(1)-X(1)))
ALPHA = ALPHA - 1.5707963
TRANF(1) = COS(ALPHA)
TRANF(2) = -SIN(ALPHA)
IF(NDM.EQ.3) GO TO 1
TRANF(3) = -TRANF(2)
TRANF(4) = TRANF(1)
RETURN
C
1 TRANF(4) = -TRANF(2)
TRANF(5) = TRANF(1)
RETURN
END
SUBROUTINE NORM(X,Y,N)
C
C NORMALIZE VECTOR Y TO UNIT VECTOR X
C
DIMENSION X(1), Y(I)
SCALE = SQRT(DOT(Y,Y,N))
DO 100 I = 1*N
100 X(I) = Y(I)/SCALE
RETURN
END
LOGICAL FUNCTION PCOMP(A, B)
PCOMP = .FALSE.
C................IT MAY BE NECESSARY TO REPLACE THE FOLLOWING ALPHANUMERIC
C................COMPARISON STATEMENT IF COMPUTER PRODUCES AN OVERFLOW.
   IF(A.EQ.B) PCOMP = .TRUE.
RETURN
END
SUBROUTINE PCONTQ
C
LOGICAL PCOMP
COMMON/COAT A/ 0.HEAD(20) .NUMNP .NUMEL .NUMMAT .NEN .NEQ .IPR
COMMON/LABEL/ PDIS(6) .A(6) .BC(2) .DI(6) .CD(3) .TE(3) .TD(3)
COMMON M(I)
DIMENSION TITL(20) .WD(3)
DATA WD/4HFEAP,4HMACR,4HSTOP/
C
C..........READ A CARD AND COMPARE FIRST 4 COLUMNS WITH MACRO LIST.
C
1 READ(5,1000) TITL
IF(PCOMP(TITL(1),WD(1))) GO TO 100
IF(PCOMP(TITL(1),WD(2))) GO TO 200
IF(PCOMP(TITL(1),WD(3))) RETURN
GO TO 1
C
C..........READ AND PRINT CONTROL INFORMATION
C
100 DO 101 I = 1,20
101 HEAD(I) = TITL(I)
READ(5,1001) NUMNP, NUMEL, NUMMAT, NDM, NDF, NEN, NAD
WRITE(6,2000) HEAD, NUMNP, NUMEL, NUMMAT, NDM, NDF, NEN, NAD
C
C..........SET POINTERS FOR ALLOCATION OF DATA ARRAYS.
C
PDIS(2) = A(NDM)
NEN1 = NEN + 1
NST = NEN*NDF + NAD
N0 = 1 + NST*IPR
N1 = N0 + NEN*NDM*IPR
N2 = N1 + NEN*IPR
N3 = N2 + NST
N4 = N3 + NST*IPR
N5 = N4 + NST*NST*IPR
N6 = N5 + NUMMAT
N7 = N6 + 12*NUMMAT*IPP
N8 = N7 + NDF*NUMNP
N9 = N8 + NDM*NUMNP*IPR
N10 = N9 + NEN1*NUMEL
N11 = N10 + NDF*NUMNP*IPR
N12 = N11 + NUMNP*IPR
N13 = N12 + NDF*NUMNP

C
C.........CHECK THAT SUFFICIENT MEMORY EXISTS.
C
CALL SETMEM(N13)
CALL PZERO(M,N12)
C
C.........CALL MESH INPUT SUBROUTINE TO READ AND PRINT MESH DATA.
C
III = 0
CALL PMESH(M(N2),M(N5),M(N6),M(N7),M(N8),M(N9),M(N10),M(N11),NDF,
1
NDM,NEN1,NST,III)
C
C.........STORE ID AND F VALUE OF TAPE16 FOR POSSIBLE B.C. CHANGE
C
CALL BCSTOR(M(M71),M(N10),NDF*NUMNP,IPR,1)
C
C.........ESTABLISH PROFILE OF RESULTING EQUATIONS FOR STIFFNESS,MASS.
C
CALL PROFIL(M(N12),M(N7),M(N9),NDF,NEN1,NAD)
C
C.........SET POINTERS FOR SOLUTION ARRAYS * CHECK FOR SUFF. MEMORY.
C
N13 = N12 + NEO
C
THE ABOVE STATEMENT IS DELETED, BECAUSE WE WANT TO GIVE
C
MAXIMUM SPACE FOR THE DISPLACEMENT ARRAY (B).
C
DO THE SAME FOR THE NEXT STATEMENT.
C
C
N14 = N13 + NEO*IPR
C
N14 = N13 + NDF*NUMNP*IPR
NE = N14 + NUMNP*NDF*IPR
CALL SETMEM(NE)
GO TO 1
C
C..................CALL MACRO SOLUTION MODULE FOR ESTABLISHING SOL. ALGORITHM.
C
200 CALL PMACR(M,M(N0),M(N1),M(N2),M(N3),M(N4),M(N5),M(N6),M(N7),M(N8)
1       *M(N9),M(N10),M(N11),M(N12),M(N13),M(N14),M(NE),NDF,
2       NDM,NEN1,NST,NE)
C
GO TO 1
C
C............INPUT/OUTPUT FORMATS
C
1000 FORMAT(20A4)
1001 FORMAT(16I5)
2000 FORMAT(1H1,20A4//5X,'NUMBER OF NODAL POINTS =•.I6/5X,'NUMBER
1 OF ELEMENTS =•.I6/5X,'NUMBER OF MATERIAL SETS =•,
2 I6/5X,'DIMENSION OF COORDINATE SPACE =•.I6/5X,'DEGREE OF FREEDOM
3S/NODE =•.I6/5X,'NODES PER ELEMENT (MAXIMUM) =•.I6/5X,'EXTRA
4D.O.F. TO ELEMENT =•.I6)
END
SUBROUTINE PFOM(UL,XL,TL,LD,P,S,IE,D,X,I,F,TDIAG,B,A,C,NDF,
1       NDM,NEN1,NST,ISW,T5G,TEPS,RATIO,EOSIG,FT,STK,UNSIG,YNLOC,TRANF,
2       NDM2,NGAUP,N101)
C
C..................COMPONENT ELEMENT ARRAYS AND ASSEMBLE GLOBAL ARRAYS.
C
LOGICAL AFL,BFL,CFL,DFL,
1       ACFL,ELAS,INCFL,ISFL,NDFL,TRANS,UNFL,LDFL
COMMON/NLDATA/ACFL,ELAS,INCFL,ISFL,NDFL,TRANS,UNFL,LDFL
COMMON /CDA TA/0.HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
COMMON /CLOCA/LDFL,NDME2,NTLC,NTOTAL,NTR
COMMON /ELDATA/ DM,N,M,A,CT,EEL,NEL
COMMON/PPLOD/ TPROP,PROP
DIMENSION XL(NDM),LD(NDF),P(S(NST)),IE(1),D(12,1),
2C(1).UL(NDF.1).TL(1).I(1).U(1).UD(NDF.1).
3TSIG(NTOT.1).TEPS(NTOT.1).RATIO(NGAUP.1).EOSIG(NGAUP.1).
4FT(1).UAC(1).UNSIG(NGAUP.1).NLOC(1).TRANF(1).TEMPU(6)

C
LOOP ON ELEMENTS
C
IEL = 0
REWIND 11
REWIND 18
DO 110 N = 1.NUMEL
C SET UP LOCAL ARRAYS.
DO 108 I = 1.MEN
II = IX(I,N)
IF(II.NE.0)GO TO 105
TL(I) = 0.
DO 103 J = 1.NDM
XL(J.I) = 0.
DO 104 J = 1.NDF
UL(J.I) = 0.
UL(J.1+II) = 0.
104
LD(J.I) = 0
GO TO 108
105
IID = II*NDF-NDF
NEL = I
TL(I) = T(II)
DO 106 J = 1.NDM
XL(J.I) = X(J.II)
DO 107 J = 1.NDF
K = IABS(ID(J.II))
UL(J.I) = F(J.II)**PROP
UL(J.1+II) = UD(J.II)
107
C
C USING TOTAL DISPLACEMENT (THE PART OTHER THAN THE
C INPUT) IS NOT USED FOR NONLINEAR ANALYSIS. WE
C USE STRESS DIRECTLY, I.E. INSTEAD OF BT*DOB*U
C THE EQUATION BT*SIG IS USED. WHEREIN UNLOADING IS
A SPECIAL PROCESS EVEN IT IS SET TO BE ELASTIC
THE TOTAL STRESS IS STILL TO BE USED. */

/* FORCE INPUT IS TO BE SUPPRESSED (K.GT.0) -- SET
TO ZERO. ORIGINALLY TOTAL DISPLACEMENT IS USED */

INSTED OF THE TOTAL DISPLACEMENT, INCREMENTAL
DISPLACEMENT IS USED WHICH ARE NEEDED TO CALCULATE
THE ELASTIC STRAINS AND STRESSES FOR VARIOUS OPERA
TION . (REF. "RESID" -- ONLY FOR ISW = 7 )

............

IF(K.GT.0) UL(J,I) = 0.0
IF(K.GT.0 .AND. ISW.EQ.7) UL(J,I) = B(K)
IF(K.GT.0 .AND. UNFL) UL(J,I) = B(K)
   IF(K.GT.0 .AND. ELAS .AND. (.NOT. UNFL))
      UL(J,I) = U(K)
      IF(DFL) K = IID + J
107    LD(J,I) = K

CHECK IF ANY LOCAL NODE(D)

IF(.NOT. LOFL .OR. (ISW.EQ.3)) GO TO 108

CALL DISLO(I,I,NTLC,NDF,UL,NLOC,TRANF,NDE2+1)

108    CONTINUE

FORM ELEMENT ARRAY.
MA = IX(NEW1*N)
IF(IE(MA).NE.IEL) MCT = 0
IEL = IE(MA)
CALL EL4LIB(D(I,MA),UL,XX(I,N),TL,SP,NDF,NDM,NST,ISW,
1 TSIG(I,N),TEPS(I,N),RATIO(I,N),EOSIG(I,N),
2 FT,UA,UNSIGY(I,N),NLOC,TRANF,NDM2,NGAPS,NTOT)

C TRANSFORM THE BT*SIG TERM (FOR DISLP. INPUT)

IF(.NOT.LOFL) GO TO 109
J1 = 1
DO 1080 I = 1,NEN
II = IX(I,N)
IF(II.EQ.0) GO TO 1080
CALL DISLO(II,1,NTLC,NDF,P(J1),NLOC,TRANF,NDME2,2)
J1 = J1 + NDF
1080 CONTINUE

C 109 CONTINUE
C
C ADD TO TOTAL ARRAY.
C
IF(AFL.OR.BFL.OR.CFL) CALL ADDSTF(A,B,C,SP,
1 JDIA,LD,NST,NEL,NDF,AFL,BFL,CFL)
110 CONTINUE
RETURN
END
SUBROUTINE PEIGS(A,B,F,X,Y,Z,ID,IX,JDIAG,NDF,NDM,NENI,DFL)
C
C...............COMPUTE DOMINANT EIGENVALUE BY INVERSE ITERATION
C
LOGICAL DFL
COMMON/CDATA/O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
COMMON/ENGYS/ AENGY
DIMENSION A(1),B(1),F(1),X(1),Y(1),Z(1),ID(1),IX(1),JDIAG(1)
DATA ITS/100/,TOL/1.E-9/
C GET START VECTOR FROM DIAGONAL OF MASS MATRIX
DO 100 I = 1,NEQ
   J = JDIAG(I)
   IF(DFL) J = I
100 Y(I) = B(J)
EIGP = 0.
CALL ACTCOL(A,Y,JDIAG,NEQ,.TRUE.,.FALSE.)
DO 200 I = 1,ITS
   CALL PZERO(Z,NEQ)
   CALL PROMUL(B,Y,Z,JDIAG,NEQ)
C RAYLEIGH QUOTIENT
   EIG = AENGY/DOT(Y,Z,NEQ)
   IF(ABS(EIG-EIGP).LT.TOL*ABS(EIG)) GO TO 300
   CALL NORM(Y,Z,NEQ)
   EIGP = EIG
C INVERSE ITERATION
200 CALL ACTCOL(A,Y,JDIAG,NEQ,.FALSE.,.TRUE.)
WRITE(6,2001) ITS
RETURN
C
300 WRITE(6,2000) O,HEAD,EIG,I
   CALL NORM(Z,Y,NEQ)
   CALL PRTDISX(ID,X,Z,F,NDM,NDF)
RETURN
2000 FORMAT(A1,20A4//5X,*EIGENVALUE =','G13.4/5X,*ITERATIONS = ','1
   I 19/)
2001 FORMAT(5X,*FATAL ERROR 0900 NO CONVERGENCE IN EIGENVALUES*.
   1 *ITS = '*,15)
END
SUBROUTINE PGAUS5(L,LINT,R,Z,W)

C
C
DIMENSION LR(9),LZ(9),LW(9),R(1),Z(1),W(1)
DATA LR/-1.1.1.-1.0.1.0.-1.0./
DATA LZ/-1.1.1.-1.0.1.0.-1.0/0.
DATA LW/4*25.4*40.64/
LINT = L*L
GO TO (1,2,3)*L
C**********1X1 INTEGRATION.
  1 R(1) = 0.
  Z(1) = 0.
  W(1) = 4.
  RETURN
C**********2X2 INTEGRATION.
  2 G = 1./SQRT(3.)
  DO 21 I = 1,4
    R(I) = G*LR(I)
    Z(I) = G*LZ(I)
  21 W(I) = 1.
  RETURN
C**********3X3 INTEGRATION.
  3 G = SQRT(0.6)
  H = 1./81.
  DO 31 I = 1,9
    R(I) = G*LR(I)
    Z(I) = G*LZ(I)
  31 W(I) = H*LW(I)
  RETURN
END
SUBROUTINE PLASTI (D,IX,DR,B,NEN1,TSIG,TEPS,RATIO,EQSIG,FT,UAC,
                UNSIGY,NDM2,NGAUP,NTOT,JUNLD)

C

C PROGRAM NAME : PLASTI(C)

C PURPOSE : (1) FIND THE MINIMUM VALUE OF RATIO FOR TRANSITION
            ZONE IF APPLICABLE.
            (2) UPDATE STRESS AND STRAIN LOAD INCREMENT DUE TO
            ** NOTE --- FOR INITIAL STRESS METHOD
            THE SCALING IS DONE SEPARATELY FOR
            EACH INTEG. POINT. (REF. "INSTS") **/
            TRANSITION. (THIS AND ABOVE APPLY TO INCREMENTAL
            METHOD ONLY).
            (3) UPDATE TOTAL DISPLACEMENTS AND LOADS.
            (4) INVOKE "INSTS" FOR STRESS OPERATION(REF. INSTS)

C IMPORTANT INDICATORS:

C (1) JR : FLAG TO INDICATE WHETHER ANY NEW
            INTEG. PT. ENTER THE YIELD ZONE OR NOT.

C (2) NENP: WHETHER ANY INTEG. PT. IS STILL NEEDED
            FOR PROCESSING IN TRANSITION ZONE.

C (3) JUNLD: IF ANY NEW POINT IS UNLOADED, JUNLD
            WILL BE .GT. 0, SIGNAL OUT TO 'PLAS'.

C

C******************************************************************************
C
LOGICAL ACFL,ELAS,INCFR,ISFL,NDFL,TRANS,UNFL
COMMON/NLDATA/ ACFL,ELAS,INCFR,ISFL,NDFL,TRANS,UNFL
COMMON/CODATA/ 0,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEO,IPR
COMMON/FLDATA/ DM,N,MA,MCT,IEL,NEL

371
COMMON/DEBUG/IDEBUG
COMMON/PRLOD/TPROP,PROP
DIMENSION D(12,1),IX(NEN1,1),TSIG(NTOT,1),TEPS(NTOT,1),
1 RATIO(NGAUP,1),EQSIG(NGAUP,1),FT(1),UAC(1),DR(1),B(1)
2 UNSIGY(NGAUP,1)
C
IF(.NOT.TRANS) GO TO 50
C.................FIND RMIN IF APPLICABLE.
C
C...........*/WHY JR AND NENP ??*/
C............(1) JR : IT IS POSSIBLE THAT THE LOAD INCREMENT IS NOT
C............ LARGE ENOUGH TO CAUSE "ADDITIONAL" YIELDING
C............ (NO AUTOMATIC SCALING IS PERFORMED IN THE
C............ CURRENT PROGRAM.))
C............ (2) NENP : ALSO POSSIBLE THAT ALL THAT ARE
C............ POSSIBLE TO GO THRU TRANSITION ZONE HAS PASSED.
C............ IN OTHER WORDS, IF ALL ARE NONLINEAR MATERIAL,
C............ THIS MEANS "ALL NONLINEAR ELEMENTS HAVE YIELD-
C............ ED".
C
JR = 0
RMIN = 1.0
NENP = 0
DO 20 N = 1,NUMEL
   MA = IX(NEN1,N)
   IF(D(B,MA),GE,0) GO TO 20
   DO 10 L = 1,NGAUP
      IF(RATIO(L,N),LT,0.0) GO TO 10
      NENP = NENP + 1
   C
   IF(RATIO(L,N),GT,1.0) GO TO 10
   JR = JR + 1
   IF(RATIO(L,N),GE,RMIN) GO TO 10
   RMIN = RATIO(L,N)
10 CONTINUE
20 CONTINUE
CHECK WHETHER NO ADDITIONAL ELEMENT YIELDED.
AND WHETHER ALL ELEMENTS HAVE PASSED TRANS. ZONE.
   IF(JR.NE.0) GO TO 30
C
C NO ADDITIONAL POINTS IN YIELDING. (JR = 0; ACTION : RMIN = 1.0 I.E. NO SCALING )
C
RMIN = 1.0
   IF(NENP.NE.0) GO TO 30
C
C ALL PASSED TRANSITION ZONE. (NENP = 0 )
C
TRANS = .FALSE.
30 CONTINUE
C
IF(.NOT.TRANS) GO TO 50
C TRANSITION ZONE.
   IF(.NOT.INCFL) GO TO 50
C INCREMENTAL METHOD.
   DO 40 I = 1,NEQ
      DR(I) = DR(I)*RMIN
   40 CONTINUE
C
PROP = PROP*RMIN
C
C UPDATE TOTAL DISPLACEMENT. /* NOTE -- IN THE ORIGINAL
C CODE THE DISPLACEMENTS ARE UPDATED RIGHT AFTER SOLUTION
C OF DISPLA. (REF. MACRO 'SOLV'). */
50 DO 60 I = 1,NEQ
     B(I) = B(I) + DR(I)
   60 CONTINUE
C
80 CONTINUE
C
IPROP = IPROP + PROP
100 CONTINUE
C
    REWIND 12
    REWIND 11
    REWIND 18
    JUNLD = 0
    DO 300 N = 1, NUMEL
        MA = IX(NEN1, N)
        CALL INSTS(TSIG(1:N), TEPS(1:N), EDSIG(1:N), D(1:MA),
                  1         RATI0(1:N), UNSIGY(1:N), NGAUP, NDM2, RMIN, JUNLD)

300 CONTINUE
RETURN
END
SUBROUTINE PLOAD(ID,F,B,NN,P)
C
C FORM LOAD VECTOR IN COMPACT FORM.
C
DIMENSION ID(I),F(I),B(I)
DO 100 N = 1,NN
   J = ID(N)
100 IF(J.GT.0) B(J) = F(N)*P
RETURN
END
SUBROUTINE PMAC3(UL, XL, TL, LD, P, S, IE, ID, IX, F, T, JDIAG, B, DR, C
  +NDF, NDM, NEN1, NST, NEND)

C
C............MACRO INSTRUCTION SUBPROGRAM.
C
C............CONTROLS PROBLEM SOLUTION AND OUTPUT ALGORITHMS BY
C............ORDER OF SPECIFYING MACRO COMMANDS IN ARRAY WD.
C
LOGICAL AFR, BFR, CFR, AFL, BFL, CFL, DFL, EFL, FFL, GFL, PCOMP, LOFL
  1          +ACFL, ELAS, INCFL, ISFL, NDFL, TRANS, UNFL, ACEFL, POFL, BCFL
  2          +FRFL, CUVCH, FFL, PMFL, UPLRF
COMMON/NLOATA/ACFL, ELAS, INCFL, ISFL, NDFL, TRANS, UNFL
COMMON/CLLOCA/ LOFL, NDME2, NTLC, NTOTAL, NTR
COMMON/UPDATE/UPLRF
COMMON/CROLL/XR(3), ROLLRA, FRFAC, OMEGA, CUVCH, FRFL
COMMON/PODATA/POFL, FMFL
COMMON/PMDATA/PMFL
COMMON M(1)
COMMON/CDATA/0, HEAD(20), NUMP, NUMEL, NUMMAT, NEN, NEQ, IPR
COMMON/LABEL/PDIS(6), Z(6), BC(2), DI(6), CD(3), TE(3), FD(3)
COMMON /PRLOD/ TPROP, PPROP
COMMON /DATA/ TIME, DT, C1, C2, C3, C4, C5
DIMENSION WD(35), CT(4, 100), CTL(4), LVS(9), LVE(9), JDIAG(1),
  1          +UL(1), XL(1), TL(1), LD(1), P(1), S(1), IE(1), ID(1),
  2          +ID(1), IX(1), F(1), T(1), B(1), DR(1)
DATA WD/4HT01, 4HD0, 4HSTRE, 4HDISP, 4HTANG, 4HFORM, 4HLOOP, 4HNEXT,
  1        +4HPROP, 4HDATA, 4HTIME, 4HCONV, 4HSOLV, 4HMAS, 4HCAS, 4HMESH,
  2        +4HEIGE, 4HEXCD, 4HUNLD, 4HPLAS, 4HACCE, 4HTNCR, 4HFNV
  3        +4HPOST, 4HBCNG, 4HLOCA, 4HROLL, 4HUPCO,
  4        +4HRBC, 4HYLD, 4HULP/
DATA NW0/35/, ENDW/4HEND, /NY, NC/1, 1/
  1        +ISG, IEP, IRA, IEQ, IFT, IAC, NGAUP, NDM2, NTOT, I5Y, IBB, IBN
  2        +NTND, NTRN/1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
DATA NTAPE1/55/, STRESS, 4HSTRE, 4HDISP, 4HSTRE, 4HDISP, 4HSTRE, 4HDISP,
  1        +NTAPE2/31/

C
C............SET INITIAL VALUES OF PARAMETERS
DT = 0.0
PROP = 1.0
RNMAX = 0.0
TIME = 0.0
TOL = 1.0E-9
NDME2 = 1
UN = 0.0
AFR = .TRUE.
AFR = .FALSE.
BFL = .TRUE.
BFR = .FALSE.
CFL = .TRUE.
CFR = .FALSE.
DFL = .TRUE.
EFL = .TRUE.
FFL = .FALSE.
GFL = .TRUE.
NE = NEND
NNEQ = NDFNUMP
NPLD = 0

C............INITIALIZING FOR PLASTICITY ANALYSIS.
ELAS = .TRUE.
TRANS = .FALSE.
BCFL = .FALSE.
ISFL = .FALSE.
LDFL = .FALSE.
ACFL = .FALSE.
NDFL = .FALSE.
UNFL = .FALSE.
INCFL = .FALSE.
P0FL = .FALSE.
FMFL = .FALSE.
ACEFL = .TRUE.
FRFL = .FALSE.
UPLRFL = .FALSE.
CUVCH = .FALSE.
TPROP = 0.0
KTANG = -1
WRITE(6,2001) 0,HEAD
C ********** READ MACRO CARDS.
   LL = 1
   LMAX = 16
   CALL SETMEM(NE+LMAX*4*IPR)
   CT(1,1) = WD(7)
   CT(3,1) = 1.0
100 LL = LL + 1
   IF(LL.LT.LMAX) GO TO 110
   LMAX = LMAX + 16
   CALL SETMEM(NE+LMAX*4*IPR)
110 READ(5,1000) (CT(J,LL),J = 1,4)
   WRITE(6,2000) (CT(J,LL),J = 1,4)
   IF(.NOT.PCOMP(CT(1,LL),ENDM)) GO TO 100
   200 CT(1,LL) = WD(8)
C ********** SET LOOP MARKERS.
   NE = NE + LMAX*4*IPR
   LX = LL - 1
   DO 230 L = 1,LX
      IF(.NOT.PCOMP(CT(1,L),WD(7))) GO TO 230
         J = 1
         K = L + 1
      DO 210 I = K,LL
         IF(PCOMP(CT(1,I),WD(7))) J = J + 1
         IF(J.GT.9) GO TO 401
         IF(PCOMP(CT(1,I),WD(8))) J = J-1
         IF(J.EQ.0) GO TO 220
      210 CONTINUE
      GO TO 400
   C
   220 CT(4,I) = L
   CT(4,L) = I
   230 CONTINUE
C
J = 0
DO 240 L = 1, LL
   IF(PCOMP(CT(I,L), WD(7)))) J = J + 1
   IF(PCOMP(CT(I,L), WD(8))) J = J - 1
   IF(J NE 0) GO TO 400
C...........EXECUTE MACRO INSTRUCTION PROGRAM.
   LV = 0
   L = 1
299 DO 300 J = 1, NWD
   IF(PCOMP(CT(I,L), WD(J))) GO TO 310
300 CONTINUE
GO TO 330
C
310 I = L-1
   IF(L NE 1 .AND. L NE LL)
      WRITE(6,2010) I, (CT(K,L), K = 1, 4)
      GO TO (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
      1 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35), J
C
C...........SET SOLUTION TOLERANCE.
C
   1 TOL = CT(3,L)
   GO TO 330
C
C...........SET TIME INCREMENT.
C
   2 DT = CT(3,L)
   GO TO 330
C
C...........PRINT STRESS VALUES
   3 LX = LVE(LV)
C
C...........OVERRIDE PROP. SET PROP = 0.0, WILL USE TOTAL
C          STRESS INSTEAD OF RECALCULATE USING DISP INPUT.
IF(.NOT.ELAS)
1   PROP = 0.0
IF(AMOD(CT(3,LX))AMAX1(CT(3,L))EQ.0.0) GO TO 3000
GO TO 330
C
C........POSTPROCESSOR OUTPUT
C
3000 IF(POFL.AND..NOT.FMFL) WRITE(NTAPE1) STRESS,TIME
IF(POFL.AND.FMFL) WRITE(NTAPE2,4011) STRESS,TIME
CALL PFORM(UL,TL,LD,PS,IE,ID,IX,IF,TD,JDIAG,DR,DR,DR,
2   NDF,NDM,NEN1,NST,4,B,M(NV),FALSE,FALSE,0,
3   FALSE,FALSE,M(ISG),M(IEP),M(IRA),M(IEQ),
4   M(IE),M(IAC),M(ISY),M(NTRN),M(NTRN),NDM2,NGAUP,NTOT)
GO TO 330
C
C.........PRINT DISPLACEMENTS.
C
4   LX = LVE(LV)
IF(AMOD(CT(3,LX))AMAX1(CT(3,L))NE.0.0) GO TO 330
IF(.NOT.POFL) WRITE(6,2003) O.HEAD,TIME,PROP
IF(POFL.AND..NOT.FMFL) WRITE(NTAPE1) DISPLA,TIME
IF(POFL.AND.FMFL) WRITE(NTAPE2,4011) DISPLA,TIME
C
CALL PRDIS (ID,IX,8,F,NDM,NDF)
GO TO 330
C
C.........FORM TANGENT STIFFNESS.
C
19 IF(CFL) CALL PSETM(NC,NE,JDIAG(NEO)*IPR,CFL)
   CALL PZERO(M(NC),JDIAG(NEO))
   CFR = .TRUE.
   IF(J,EQ,5) CFR = .FALSE.
   IF(GFL) CALL PSETM(NA,NE,JDIAG(NEO)*IPR,GFL)
   IF(NPLD.GT.0) PROP = PROPLD(TIME,0)
   CALL PZERO(M(NA),JDIAG(NEO))
   IF(.NOT.ELAS) KTANG = KTANG + 1
   CALL PFORM(UL,TL,LD,PS,IE,ID,IX,IF,TD,JDIAG,DR,4,NA,M(NA),M(NC),
C FORM OUT DF BALANCE FORCE FOR TIME STEP/ITERATION.
C
6 IF(NPLD.GT.0) PROP = PROPLD(TIME,0)
C
C FOR ITERATION WITHIN LOOP (INITI. STRESS. METH.) NO NEED
C TO USE ANY INPUT LOAD/DISP -- USE THE INCREMENTAL PLASTIC
C STRESS INSTEAD. */
C
IF(ISFL) PROP = 0.0
DPROP = PROP
   IF(.NOT.ELAS.AND. (.NOT.ISFL)) DPROP = PROP + TPROP
   CALL PLOAD( ID,F,DR,NNEQ,DPROP)
   CALL PFORM(UL,XL,TL,LD,P,IE,D,ID,X,T,JOAG,DR,DR,DR,
1   NDF,NDM,NEN1,NST,6,B,M(NV),.FALSE.,TRUE.,.FALSE.,
2   .FALSE.,M(ISG),M(IEP),M(IRA),M(IEQ),M(IFT),M(IAC),
3   M(ISY),M(NTND),M(NTRN),NDM2,NGAUP,NTOT)
   BFR = .TRUE.
C
C STORAGE THE INCREMENTAL LOAD (DR) FOR TWO REASONS:
C
C (1). SCALING FOR INCREMENTAL METHOD FOR TRANSITION ZONE.
C (2). OBTAINING TOTAL LOAD FOR BOTH METHOD FOR
C OVERALL EQUILIB. CHECK FOR EACH INCREMENT.
C */ NO LONGER NECESSARY TPROP CAN INDICATE THE CURRENT
C VALUE*/
C
REWRITE 13
GO TO 330
C
C...........SET LOOP START INDICATORS
C
7 LV = LV + 1
LX = CT(4,L)
LVS(LV) = L
LVE(LV) = LX
CT(3,LX) = 1.
GO TO 330

C...........LOOP TERMINATOR CONTROL
8 N = CT(4,L)
   CT(3,L) = CT(3,L) + 1.0
   IF(CT(3,L).GT.CT(3,N)) LV = LV - 1
   IF(CT(3,L).LE.CT(3,N)) L = N
GO TO 330

C...........INPUT PROPORTIONAL LOAD TABLE
9 NPLD = CT(3,L)
PROP = PROPLD(0.,NPLD)
GO TO 330

C...........READ COMMAND.
10 READ(5,1000) (CTL(I),I=1,4)
   IF(.NOT.PCOMP(C1(2,L),C1(1))) GO TO 402
   IF(PCOMP(C1(1),W1(1))) TOL = C1(3)
   IF(PCOMP(C1(1),W2(1))) DT = C1(3)
GO TO 330

C...........INCREMENT TIME
11 TIME = TIME + DT
RNMAX = 0.0
UN = 0.0

C
C.............(1). RESET INITIAL STRESS FLAG TO FALSE SO THE
C              FIRST TIME THROUGH THE PATH IS THE SAME
C              FOR BOTH METHODS. IT IS TO BE SET TRUE
C              FOR INITIAL STRESS NOT CONVERGED CASE ONLY
C              IN MACRO "FCNV".
C
C               (2). INCREMENTAL FLAG IS SET TO FALSE FOR FUTURE
POSSIBLE DEVELOPMENT ONLY (SAFE PROTECTION).
IT IS SET TRUE ONLY WHEN INCREMENTAL METHOD
IS TO BE USED IN MACRP 'INCR'.

ISFL = .FALSE.
INCFL = .FALSE.
IF(.NOT.ELAS) CALL ULSTOR(B,DI,IX,M(ISG),M(IEP),M(IRA),M(IEQ),
M(ISY),NEN1,NGAUP,NTOT, 2)
   IF(.NOT.ELAS.AND.ACFL) CALL PZERO(M(IAC),IACDIS)
            GO TO 330

C...........COMPUTE CONVERGENCE TEST
C
12 RN = 0.0
   DO 121 N =1,NEQ
      UN = UN + B(N)**2
   121 RN = RN + DR(N)**2
   UN = AMAX1(UN,RN)
   CN = SORT(UN)
   RN = SORT(RN)
   WRITE(6,2002) CN,RN,TOL
   LX = LVE(LV)
   LO = LVS(LV)
   IF(RN.LT.CN*TOL) CT(3,LX) = CTO.LO)
            GO TO 330

C............SOLVE THE EQUATIONS
C
13 IF(CFR) GO TO 131
   CALL ACTCOL(M(NA),DR,JDIAG,NEQ,AFR,BFR)
         GO TO 132
C
131 CALL UAICTOL(M(NA),M(NC),DR,JDIAG,NEQ,AFR,BFR)
132 AFR = .FALSE.
   IF(.NOT.BFR) GO TO 330
   BFR = .FALSE.
THE TOTAL DISPLACEMENT IS TO BE ADDED LATER IN MODULE "PLASTIC" BECAUSE OF POSSIBLE TRANSITION (FOR INCREM. METHOD) OR ACCELERATION (INITIAL STRESS METHOD MACRO 'ACCE') ADJUSTMENT.

IF(.NOT.ELAS) 60 TO 330
DO 133 N = L.NEQ
133 B(N) = B(N) + DR(N)
GO TO 330

FORM A LUMPED MASS APPROXIMATION.

AFL = .FALSE.
BFL = .TRUE.
IF(EFL) CALL PSETM(NN,NEQ,IPR,EFL)
139 CALL PZERO(M(NN),NEQ)
GO TO 140

FORM A CONSISTENT MASS APPROXIMATION.

AFL = .TRUE.
BFL = .FALSE.
IF(DFL) CALL PSETM(NM,NEQ,JDIAG(NEQ),IPR,DFL)
152 CALL PZERO(M(NM),JDIAG(NEQ))
140 CALL PF0PM(UL,TL,LD,P,S,IE,D,ID,X,F,T,JDIAG,M(NN),M(NM),
1 M(NM),NDF,NDM,NEN1,NS1,5,B,M(NV),AFL,BFL,.FALSE,.FALSE,.FALSE.
2 *M(ISG),M(IEP),M(IRA),M(IEO),M(IFT),M(IAC),M(ISY),
3 M(NTRN),M(NTRN),NDM2,NGAUP,NTOT)
GO TO 300

I = -1
16 CALL PMESH(LD,IE,D,ID,X,F,T,NDF,NDM,NEN1,NS1,I)
IF(I.GT.0) GO TO 404
GO TO 330

J = NM
IF(DFL) J = NN
CALL PEIGS(M(NA),M(J),F*X*B,DR,IX,JDIAG,NDF,NEM,NEN1,DFL)
GO TO 330
C
18 IF(FFL) GO TO 181
C
C MACRO \*EXCD\* EXPLICIT INTEGRATION OF EQUATIONS OF MOTION.
C
NO = NE
NV = NO + NE*IPR
NR = NV + NDF*NUMNP*IPR
NE = NR + NE*IPR
CALL SETMEM(NE+1)
CALL PZERO(M(NO),NE-NO)
FFL = .TRUE.
GO TO 330
C
181 IF(.NOT.BFR.OR.EPL) GO TO 403
CALL EUPDAT(DR,B,M(NQ),M(NR),M(NN),DT,NEO)
GO TO 330
C
C COMPUTE REACTIONS AND PRINT.
C
20 CALL PZERO(DR,NNEO)
C
C /REACTIONS IN NONLINEAR ANALYSIS WILL USE THE
C TOTAL STRESS ONLY BT*SIG*DV. NO NEED TO USE
C AND LOAD/DISP INPUT OR CONVERSION /\nC
C IF(.NOT.FLAS) NDFL = .TRUE.
IF(NDFL) PROP = 0.0
CALL PFORM(UL,XL,TL,LD,P,S,IE,D,IX,F,T,JDIAG,DR,DR,DR,
1 NDF,NDM,NEN1,NST,B,M(NV),.FALSE..TRUE.,
2 .FALSE..TRUE..M(ISG),M(IIEP),M(IRA),M(IEO),M(IIF),
3 M(IAC),M(ISY),M(NTND),M(NTRN),NDAUP,NTOT)
CALL PRTREA(DR,NDF)
C
C  \*NDFL HAS NO MORE USE NOW, SO RESET TO FALSE. \*/
C
C
NDFL = .FALSE.
GO TO 330
C
C  \*CALL PF\*FORM\(UL, XL, TL, LD, P, S, IE, ID, IX, IF, T, JDIAG, DR, DR, DR, DR\)
1
C NDF, NDM, NEN1, NST, 2, B, F, .FALSE., .FALSE., .FALSE., .FALSE., .FALSE.
2
C .FALSE., M(ISG), M(IEP), M(IRA), M(IEQ), M(IF), M(IAC),
3
C M(ISY), M(NTD), M(NTRN), NDM, NGAUP, NTOT)
GO TO 330
C
C  \*NON-LINEAR PLASTICITY.
C
C
C  \*ELAS = .FALSE.
C TRANS = .TRUE.
C
C
C  \*SET POINTERS FOR TOTAL STRESS, STRAIN, RATIO, EOSIG.
C
C
NGAUP = D(6)*NDM
NDM2 = NDM*2
NTOT = NGAUP*NDM2
NTOT1 = NTOT*IPR
INDN = NGAUP*NUMEL
ISGEP = NTOT1*NUMEL
IFT = NDF*NUMN*IPR
ISG = NE
IEP = ISG + ISGEP
IRA = IEP + ISGEP
IEQ = IRA + INON
ISY = IEQ + INON
IFT = ISY + INON
NE = IFT + IFT
CALL SETMEM(NE)
CALL PZERO(M(ISG), M(ISG), M(IEP), M(IRA), M(IEQ), M(ISY), NEN1, NGAUP, 1
1
NTOT1)
CALL ULSTOR(B, 0, IX, M(ISG), M(IEP), M(IRA), M(IEQ), M(ISY), NEN1, NGAUP, 1
1
NTOT1)
C................CHECK IF POSTPROCESSING
C
IF(POFL.AND..NOT.FMFL) WRITE(NTAPE1) NGAUP,NDM2,NTOT,INON
IF(POFL.AND.FMFL) WRITE(NTAPE2,4009) NGAUP,NDM2,NTOT,INON
NEN1MN = NEN1*NUMEL
NDMPNU = NDM*NUMNP
IF(POFL.AND..NOT.FMFL) WRITE(NTAPE1)(IX(K),K=1,NEN1MN),(X(K),K=1,NDMPNU)
        IF(POFL.AND.FMFL) GO TO 221
        GO TO 330
221 WRITE(NTAPE2,4009) (IX(K),K=1,NEN1MN)
        WRITE(NTAPE2,4010) (X(K),K=1,NDMPNU)
        GO TO 330
C................THE PART THAT REALLY DO THE WORK ON PLASTICITY.
C
23 IF(ACFL) CALL ACCELE(DR,M(IAC),NEQ)
        IF(UPLRFL) CALL XCOOR(NUMNP,NDM,M(NTND),M(NTRN),X,DR,ID,F)
            REWIND 12
            CALL PFORM(UL,TL,LD,P,S,IE,D,X,F,T,JDIA,DR,DR
            DR,NDF,NDM,NEN1,NST,7,UD,.FALSE..FALSE,,FALSE.,
            2 .FALSE..FALSE.,M(ISG),M(IEP),M(IRA),M(IEQ),
            3 M(IF),M(IAC),M(ISY),M(NTND),M(NTRN),NDM2,NGAUP,NTOT)
C
        CALL PLASTIC(X,DR,B,NEN1,M(ISG),M(IEP),M(IRA),M(IEQ),
1 M(IF),M(IAC),M(ISY),NDM2,NGAUP,NTOT,JUNLD)
C
C...............CHECK WHETHER UNLOADING OCCURED AND UPDATING STIFFNESS IS
C NECESSARY.
C
        IF(JUNLD.EQ.0) GO TO 330
        WRITE(6,4005) JUNLD
        TRANS = .TRUE.
        IF(KTANG.LE.D) GO TO 330
STIFFNESS NEEDS TO BE REFORMED.

CALL ULST0R(B,DX*M(ISG),M(IEP),M(IRA),M(IEO),M(ISY),NENI,
  1
  NGAUP,NTOT,3)

RNMAX = 0.0
UN = 0.0
ISFL = .FALSE.
IF(INCFL) GO TO 330
INCFL = .FALSE.
IF(AUFL) CALL PZERO(M(IAC),IACDIS)
  GO TO 5

UNLOADING IS ASSUMED ELASTIC.

24 UNFL = .TRUE.
  IF(BCFL) CALL BCHNG(ID,IX,F,JO,DF,NENI,NAD,NA,NE,NDF,NUMNP,
  1
  NBN,NDF+1,M(IBN),B,M(IBB))
  ELAS = .TRUE.

SET PROP = 0.0 (ISFL = .TRUE.)

ISFL = .TRUE.
ACFL = .FALSE.
GO TO 330

ACCELERATING OF INITIAL STRESS METHOD.

25 ACFL = .TRUE.
  IF(.NOT.ACFL) GO TO 2500
  ACFL = .FALSE.
    IACDIS = NEQ*IPR
    CALL PSETM(IAC,NE,IACDIS,ACFL)
2500 CALL PZERO(M(IAC),IACDIS)
  GO TO 330
C.............FLAG FOR INCREMENTAL ANALYSIS ONLY.
C
26 INCFL = .TRUE.
   GO TO 330
C
C.............FORCE CONVERGENCE ONLY APPLY TO INITIAL STRESS METHOD.
C
27 RN = 0.0
   DO 2700 N = 1,NEO
2700   RN = RN+DR(N)**2
   RN = SORT(RN)
   RNMAX = AMAX1(RNMAX,RN)
   WRITE(6,2005) RNMAX,RN,TOL
C
C............NOT CONVERGED, RESET INITIAL STRESS FLAG (ISFL) TO TRUE,
C AND ITERATE.
C
   ISFL = .TRUE.
   IF(RN.GE.RNMAX*TOL) GO TO 330
C
C............IF LESS THAN TOLERANCE THEN CONVERGED, STEP OUT OF LOOP.
C
   ISFL = .FALSE.
   LX = LVE(LV)
   L0 = LVS(LV)
   CT(3*LX) = CT(3,L0)
   L = LX-1
   GO TO 330
C
C............POSTPROCESSING FLAG CHANGE
C
28 POFL = .TRUE.
   REWIND NTAPE1
   REWIND NTAPE2
   IF(.NOT.FMFL)
      1 WRITE(NTAPE1) (HEAD(K),K=1,20),NUMNP,NUMFL,NE,N

IF (FMFL) GO TO 281
GO TO 330

281 WRITE(NTAPE2,4009) (HEAD(K),K=1,20)
WRITE(NTAPE2,4009) NUMNP,NUMEL,NEN,NDM
GO TO 330

C............SET POINTERS AND INPUT FOR B.C. CHANGE

29 IF(BCFL) GO TO 330
READ(5,2020) NBN
IBN = NE
IBB = IBN + NBN*(NDF+1)
NF = IBB + NUMNP * NDM * IPR
CALL SETMEM(NE)
CALL PZERO(M(IBN),NE-IBN)
CALL BCINPU(M(IBN),NDF+1,NDF,NCOTA)
BCFL = .TRUE.
GO TO 330

C............DEFINING LOCAL NODES

30 LOFL = .TRUE.
READ(5,2020) NTLC
NDME2 = NDM * 2
NTRANF = NTLC * NDME2
NTND = NE
NTRN = NTND + NTLC*2
NE = NTRN + NTRANF
CALL SETMEM(NE)
CALL PZERO(M(NTND),NE-NTND)
CALL LOCAL(M(NTND),M(NTRN),CT(3,L))
GO TO 330

C

ROLLING COMMAND,SET FRICTION FLAG,CURVE CORRECTION FLAG.

C

31 FRFL = .TRUE.
31 CUVCH = .TRUE.
     READ(5,4006) ROLLRA, OMEGA, FRFAC, (XR(K), K = 1, 3)
     WRITE(6,4007) ROLLRA, OMEGA, FRFAC, (XR(K), K = 1, 3)
     GO TO 330
C
C UPDATE COORDINATE
C
32 CALL XCOORD(NUMNP, NDM, M(NTND), M(NTRN), X, DR, ID, F)
     GO TO 330
C
B.C. CHANGE FOR ROLLING.
C
33 CALL ROLBC(NBN, NDF, NDF+1, M(NTND), M(NTRN), M(IBN), X, IROLBC)
     IF(IROLBC.NE.0) CALL BCHNG(ID, IX, F, JDIAG, NDF, NEN1, NAD, NA, 1)
                     NE, NDF & NUMNP, NBN, NDF+1, M(IBN), B, M(IBB))
     GO TO 330
C
......... POSTPROCESS OUTPUT FORMATTED (OPT).
C
34 FNL = .TRUE.
     GO TO 330
C
......... UPDATED LAGRANGIAN FLAG SET
C
35 UPLRFL = .TRUE.
C
330 L = L + 1
     IF(L.GT.LL) RETURN
     GO TO 299
400 WRITE(6,4000)
     RETURN
401 WRITE(6,4001)
     RETURN
402 WRITE(6,4002)
RETURN
403 WRITE(5,4003)
404 WRITE(5,4004)
RETURN
1000 FORMAT(A4,1X,A4,1X,2F5.0)
2000 FORMAT(10X,A4,1X,A4,1X,2G15.5)
2001 FORMAT(A1,2A4,/*MACRO INSTRUCTIONS*//5X,/*MACRO STATEMENT*//5X
      1 /*VARIABLE 1*/5X,/*VARIABLE 2*/)
2002 FORMAT(5X,/*DISPLACEMENT CONVERGENCE TEST*//10X,/*UNMAX = */G15.5,
      1 5X,/*UN = */G15.5,5X,/*TOL = */G15.5)
2003 FORMAT(A1,2A4,/*TIME*/G15.5/5X,/*PROPORTIONAL LOAD*/G13.5)
2004 FORMAT(5X,/*CN = */G12.5,5X,/*DN = */G12.5,5X,/*UN = */G12.5,5X,
      1 /*AG = */G12.5,5X,/*AC = */G12.5)
2005 FORMAT(5X,/*FORCE CONVERGENCE TEST*//10X,/*RNM = */G15.5,5X,
      1 /*RN = */G15.5,5X,/*TOL = */G15.5)
2010 FORMAT(2X,/*MACRO INSTRUCTION*/I4,/* EXECUTED*/ I2(A4,2X),
      1 'V1 = ',G13.4,' V2 = ',G13.4)
2020 FORMAT(16I5)
4000 FORMAT(5X,/*FATAL ERROR 10*/UNBALANCED LOOP/NEXT MACROS*)
4001 FORMAT(5X,/*FATAL ERROR 11*/LOOPS NESTED DEEPER THAN 8*)
4002 FORMAT(5X,/*FATAL ERROR 12*/MACRO LABEL MISMATCH ON A READ*)
      1 /*COMMAND*/
4003 FORMAT(5X,/*FATAL ERROR 13*/MACRO EXCD MUST BE PRECEDED BY*
      1 /*LMA'S AND FORM*]
4004 FORMAT(5X,/*FATAL ERROR 14*/ATTEMPT TO CHANGE EXECUTION*)
      1 /*RAINT CODS DURING MACRO EXECUTION*/)
4005 FORMAT(5X, //LOOK OUT // '110, /* POINTS HAVE BEEN UNLOADED*/)  
4006 FORMAT(6F10.3)
4007 FORMAT(//4X,/*SIMULATION OF ROLLING *//10X,/* RADIUS OF ROLL = */
      1 F10.3/10X,*ANGULAR VELOCITY = */
      1 F10.3/10X,*FRICCTION FACTOR (M/(1-M)) = */F10.3/
      2 10X,*COORD. OF CENTER OF ROLL = */F10.3/)
4008 FORMAT(20A4)
4009 FORMAT(20A4)
4010 FORMAT(8G10.3)
4011 FORMAT(A4,41O.3)
FND
SUBROUTINE PMESH(IDL, IIE, IE, D, ID, X, IX, I, NDF, NDM, NEN, NST, III)
C
C DATA INPUT ROUTINE FOR MESH DESCRIPTION.
C
LOGICAL PRT, ERR, PCCMP
LOGICAL PMFL
COMMON /CDATA/O, HEAD(20), NUMNP, NUMEL, NUMMAT, NEN, NEO, IPR
COMMON M(I)
COMMON /ELDATA/ DM, N, MA, MCT, IEL, NEL
COMMON /PMDATA/ PMFL
COMMON /LABEL/ PDIS(6), A(6), SC(2), DI(6), CD(3), TE(3), FD(3)
DIMENSION IE(1), D(12, 1), ID(NDF, 1), X(NDM, 1), IX(NEN1, 1), XHED(7)
1 , IDL(6), XL(3), F(NDF, 1), FL(6), T(1), WD(10), VA(2)
DATA WD'/HC00R, 4HELEM, 4HMATE, 4HBOUN, 4HFORC, 4HTEMP, 4HEND /
1.4HPRIN, 4HNDPR, 4HPAGE/BL/4HBLAN/, VA/4HVAL, 2HUE/,
2 LIST/10/PRT/P.TURE/.
3 ISG, IEP, IRA, IEO, IFT, IAC, NDM2, NGAUP, NTO, ISY, NTD, NTR
4 /1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1/
C
C /*NOTE THE I5G, IEP, M(I) ARE INSERTED TO SATISFY THE CALL
C OF "ELMTLIB" FOR PLASTICITY ANALYSIS. THEY ARE DUMMY*/
C
C INITIALIZE ARRAYS
C
ERR = .FALSE.
IF (III.LT.0) GO TO 10
DO 100 I = 1, NUMNP
DO 100 T(I) = 0.0
100 F(I,N) = 0.0
101 T(N) = 0.0
10 READ (5, 1000) CC
DO 20 I = 1, LIST
IF (PCOMP(CC, WD(I))) GO TO 30
20 CONTINUE
GO TO 10
30 GO TO (1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 1)
C******** NODAL COORDINATE DATA INPUT.
  1 DO 102 N = 1,NUMNP
     X(1,N) = AL
     CALL GENVEC(NDM,X,CD,PRT,ERR)
   10  CONTINUE
C******** ELEMENT DATA INPUT.
  2 L = 0
     DO 206 I = 1,NUMEL+50
        IF(PRT) WRITE*(6,2001) 0,HEAD,(K,K=1,NEN)
        J = MINO(NUMEL,I+49)
        DO 206 N = I,J
           IF(L-N) 200,202,203
           READ(5,1001) L,LK,(IDL(K),K=1,NEN),LX
           IF(L.EQ.0) L = NUMEL + 1
           IF(LX.EQ.0) LX = 1
           IF(L-N) 201,202,203
           WRITE(6,3001) L,N
           ERR = .TRUE.
           GO TO 206
           201 NX = LX
           DO 207 K = 1,NEN
              IX(K,L) = IDL(K)
              IX(NEN1,L) = LK
           GO TO 205
           207 IX(NEN1,N) = IX(NEN1,N-1)
           DO 204 K = 1,NEN
              IX(K,N) = IX(K,N-1) + NX
           IF(IX(K,N-1).EQ.0) IX(K,N) = 0
           IF(PRT) WRITE(6,2002) N,IX(NEN1,N),(IX(K,N),K=1,NEN)
        204 CONTINUE
     206 CONTINUE
C******** MATERIAL DATA INPUT
  3 WRITE(6,2004) 0,HEAD
     PMFL = .FALSE.
     DO 300 N = 1,NUMMAT
        READ(5,1002) MA,IEL,XHED
        WRITE(6,2003) MA,IEL,XHED
        IF(MA) = IEL
300 CALL EMLIB(D1,MA,DUM,X,IX,T,S,P,NDF,NDM,NST,1)
        M(ISG),M(IEP),M(IRA),M(IEO),M(IFT),M(IAC),M(ISY),
        M(NTND),M(NTRN),NDM2,NGAUP,NTOT)
        GO TO 10
C...........READ IN THE RESTRAINT CONDITIONS FOR EACH NODE
1 IF(PRT) WRITE(6,2000) 0,HEAD,(I,BC,I=1,NDF)
      III = 1
      N = 0
      NG = 0
502 L = N
      LG = NG
      READ(5,1001) N,NS,IDL
      IF(N.LE.0.OR,N.GT.NUMNP) GO TO 60
         DO 51 I = 1,NDF
            ID(I,N) = IDL(I)
51 IF(L.NE.0.AND.IDL(I).EQ.0.AND.ID(I-L).LT.0) ID(I,N) = -1
            LG = ISIGN(LG,N-L)
52 L = L + LG
      IF((N-L).LT.LG.LE.0) GO TO 502
         DO 53 I = 1,NDF
53 IF(ID(I-L-LG).LT.0) ID(I,L) = -1
            GO TO 52
50 DO 58 N = 1,NUMNP
      DO 56 I = 1,NDF
         IF(ID(I,N).NE.0) GO TO 57
      CONTINUE
      GO TO 58
57 IF(PRT) WRITE(6,2007) N,(ID(I,N),I=1,NDF)
58 CONTINUE
      GO TO 10
C..............FORCE/DISPL DATA INPUT.
5 CALL GENVEC(NDF,F,FD,PRT,ERR)
      GO TO 10
C..............TEMPERATURE DATA INPUT.
6 CALL GENVEC(T,TE,PRT,ERR)
      GO TO 10
7 IF(ERR) STOP
   RETURN
8  PRT = .TRUE.
   GO TO 10
9  PRT = .FALSE.
   GO TO 10
11 READ(5,1000) 0
   GO TO 10
1000 FORMAT(A4,75X,A1)
1001 FORMAT(16I5)
1002 FORMAT(15,4X,11,17A4)
2000 FORMAT(A1,20A4//5X,*NODAL B.C.*//6X,*NODE*,9(I7,A4,A2)/1X)
2001 FORMAT(A1,20A4//5X,*ELEMENTS*//3X,*ELEMENT*,2X,*MATERIAL*.
   1     14(I3,* NODE*)/(20X,14(I3,* NODE*)))
2002 FORMAT(2110,1418/(20X,1418))
2003 FORMAT(//5X,*MATERIAL SET*,13,* FOR ELEMENT TYPE*,12,5X,17A4/1X)
2004 FORMAT(A1,20A4//5X,*MATERIAL PROPERTIES*)
2005 FORMAT(A1,20A4//5X,*NODAL FORCE/DISPL*//6X,*NODE*,9(I7,A4,A2))
2006 FORMAT(110,9E13.3)
2007 FORMAT(110,9I13)
3001 FORMAT(5X,* **ERROR 03** ELEMENT*,15,*APPEARS AFTER ELEMENT*,15)
END
FUNCTION PMSIGY(D, TSIG)

MODIFIED YIELD STRESS DUE TO PRESSURE EFFECT

DIMENSION D(1), TSIG(1)

PMSIGY = SQRT(D(10) * D(10) - (TSIG(1) + TSIG(2) + TSIG(3)) * D(12))

RETURN

END
SUBROUTINE PRATID(EPS, SIG, SIG, RATIO, EOSIG, UNSIG, NDM2)

C*********************************************
C
C PROGRAM NAME: (P)RATIO OF YIELD STRESS.
C
C PURPOSE : CALCULATE RATIO FOR INTEGRATION POINT IN TRANSITION
C REGION ACCORDING TO FORMULA OF YAMADA ET. AL.
C
C CALLED BY : EL4TON((PFORM WITH ISW = 7), PLASTIC, MACRO 'PLAS')
C
C FUNCTION : EOSIG, PMSIGY
C
C LOGIC_UNIT : 12 FOR STORING THE STRESS AND STRAIN INCREMENTS.
C
C NOTE : (1): ONLY THOSE ARE STILL POSSIBLE TO GO INTO
C TRANSITION ZONES ARE PROCESSED, THEY ARE:
C (A): NONLINEAR MATERIAL STILL ELASTIC.
C (B): UNLOADED OF POST-YIELD POINT.
C
C (2): (A): RATIO'S ARE PRESET TO 0.0 FOR ALL NON-
C LINEAR AND LINEAR MATERIAL FOR NONLINEAR
C ANALYSIS.
C (B): THE CALCULATION OF THIS ROUTINE WILL GIVE
C RATIO AS ::
C (B.1) RATIO.LT. 1.0 --- YIELDED.
C (B.2) RATIO.GT. 1.0 --- STILL ELASTIC.
C
C (3): RATIO IS TO BE MODIFIED AS INDICATER OF YIE-
C LDING LATER IN SUBROUTINE "INSTS".
C
C (4): YIELD STRESS IS ALWAYS FROM TAPE 15.
C BECAUSE OF THE POSSIBILITY OF UNLOADING.
C
C*********************************************
C
LOGICAL ACFL, ELAS, INCFL, ISFL, NDFL, TRANS, UNFL, PMFL
COMMON/NLDA/PACFL, ELAS, INCFL, ISFL, NDFL, TRANS, UNFL
COMMON/PMDA/PMFL
DIMENSION EPS(10), SIG(10), TSIG(10), D(10), SGT(6)
DATA RUNLD/12345.6789/

C ********** STORE STRESS AND STRAIN INCREMENTS IN AUX. STORAGE PLACE.
C
WRITE(12) (EPS(I), I=1, NDM2), (SIG(I), I=1, NDM2)
IF(D(8) .GE. 0) RETURN
IF(.NOT. TRANS) RETURN
IF(RATIO .LT. 0.0) RETURN
C
C ********** NON/ANAL/MATE. POSSIBLE TRANSITION INTGT. POINT.
C
SIGY = D(10)
DO 10 I = 1, NDM2
   SGT(I) = TSIG(I) + SIG(I)
10 CONTINUE
C
EOS = EOSTS(SGT, NDM2, D)
DEQS = EOS - EQSIG
EQDSIG = EQSIG(SIG, NDM2, D)
C
C ********** UNLOADED ELEMENT (AFTER PAST YIELDING).
C
C ********** /* RELOADING OF AN UNLOADED INTEGRATION POINT ASSUMES
C ********** THE PREVIOUS EQUIVALENT STRESS AS ITS NEW YIELD
C ********** STRESS -- THE CHANGE OF SSY -- AND THIS IS THE BASIS
C ********** OF "STRAIN-HARDENING" ISN'T IS ? */
C
SSY = UNSIGY
IF(PMFL) SSY = PMSIGY(D, SGT)
GAMMA = (E0SIG**2 - 2.0*EQSIG*DEQS - DE0SIG**2)
TEMP = GAMMA**2*GAMMA + 4.0*EQSIG**2*(SSY**2 - EQSIG)**2
IF(TEMP .GT. 0.0) GO TO 20
IF(PMFL) GO TO 15
C
C IF(PMFL) GO TO 15
C
C ********** ERROR EXIT, PROBABLY COLLAPSE INDICATED.
C
WRITE(6,100) RATIO,SSY,EQS,EQDSIG
CALL EXIT

C SPECIAL TREATMENT
C
15  RATIO = SSY/EQS
    WRITE(6,100) RATIO,SSY,EQS,EQDSIG
    RETURN
C
C
20  TEMP = SQRT(TEMP)
    RATIO = (GAMMA+TEMP)/(2.0*EQDSIG**2)
    IF(RATIO.EQ.0.0) RATIO = RUNLD
    RETURN

100 FORMAT(//5X,**FATAL ERROR #### FOUND IN ROUTINE "PRATIO"*/
1      5X,**ORIGINAL RATIO IS **.G10.3/5X,**PREVIOUS YIELD STRESS**
2      ** IS **.G10.3/5X,**CURRENT EQUIV. STRESS IS **.G10.3/
3      5X,** CURRENT EQUIV. STRESS INCREMENT IS **.G10.3/
4      5X,** POSSIBLE COLLAPSE OF WHATEVER YOU HAVE , STAY AWAY **
5   */
   END
SUBROUTINE PROFIL (JDIAG, ID, IX, NDF, NEN1, NAD)

C

C............COMPUTE PROFILE OF GLOBAL ARRAYS

C

COMMON/CDATA/ O.HEAD(20), NUMNP, NUMEL, NUMMAT, NEN, NEQ, IPR

DIMENSION JDIAG(1), ID(NDF+1), IX(NEN+1)

C............SET UP THE EQUATION NUMBERS

NEQ = 0
DO 50 N = 1, NUMNP
   DO 40 I = 1, NDF
      J = ID(I+N)
      IF(J) 30, 20, 30
   20   NEQ = NEQ + 1
       ID(I+N) = NEQ
       JDIAG(NEQ) = 0
   GO TO 40
   30   ID(I+N) = 0
40   CONTINUE
50 CONTINUE
C.............COMPUTE COLUMN HEIGHTS.
   DO 500 N = 1,NUMEL
      DO 400 I = 1,NELEN
         II = IX(I,N)
         IF(II.EQ.0) GO TO 400
         DO 300 K = 1,NDF
            KK = ID(K,II)
            IF(KK.EQ.0) GO TO 300
            DO 200 J = 1,NELEN
               JJ = IX(J,N)
               IF(JJ.EQ.0) GO TO 200
               DO 100 L = 1,NDF
                  LL = ID(L,JJ)
                  IF(LL.EQ.0) GO TO 100
                  M = MAX0(KK,LL)
                  JDIAG(M) = MAX0(JDIAG(M),IABS(KK-LL))
   100 CONTINUE
   200 CONTINUE
   300 CONTINUE
   400 CONTINUE
   500 CONTINUE
C.............COMPUTE DIAGONAL POINTERS FOR PROFILE.
   NAD = 1
   JDIAG(1) = 1
   IF(NEQ.EQ.1) RETURN
   DO 600 N = 2,NEQ
      JDIAG(N) = JDIAG(N)+JDIAG(N-1)+1
      NAD = JDIAG(NEQ)
   600 CONTINUE
   RETURN
END
SUBROUTINE PROMUL(A,B,C,JDIAG,NEQ)
DIMENSION A(1),B(1),C(1),JDIAG(1)

C ROUTINE TO FORM C = C + A*B WHERE A IS A SYMMETRIC SQUARE
C MATRIX STORED IN PROFILE FORM, B,C ARE VECTORS AND JDIAG
C LOCATES THE DIAGONALS IN A.

JS = 1
DO 200 J = 1,NEQ
  JD = JDIAG(J)
  IF(JS.GT.JD) GO TO 200
  BJ = B(J)
  AB = A(JD)*BJ
  IF(JS.EQ.JD) GO TO 150
    JB = J - JD
    JE = JD + 1
    DO 100 JJ = JS,JE
      AB = AB + A(JJ)*B(JJ+JB)
    100
  C(JJ+JB) = C(JJ+JB) + A(JJ)*BJ
  C(J) = C(J) + AB
    JS = JD + 1
200 JS = JD + 1
RETURN
END
FUNCTION PROPLD(T,J)
C
C********PROPORTIONAL LOAD TABLE (ONE LOAD CARD ONLY).
C
DIMENSION A(5)
IF(J.GT.0) GO TO 200
C********COMPUTE VALUE AT TIME T.
C
C********PROPLD = 0.0 ORIGINALLY, CHANGED TO 1.0 SINCE
C
IN NONLINEAR ANALYSIS THE FIRST TIME IS A FULL LOAD
C
NEAR THE YIELD POINT.
C
PROPLD = 1.0
IF(T.LT.TMIN.OR.T.GT.TMAX) RETURN
L = MAX0(L,1)
PROPLD = A(1) + A(2)*T + A(3)*SIN(A(4)*T+A(5))*L
RETURN
C********INPUT TABLE OF PROPORTIONAL LOADS.
200 I = 1
READ(5,1000) K,L,TMIN,TMAX,A
WRITE(6,2000) I,K,L,TMIN,TMAX,A
RETURN
1000 FORMAT(215,7F10.0)
2000 FORMAT(5X,'PROPORTIONAL LOAD TABLE'//' NUMBER TYPE ')
2 *A2* M, A3* M, A4* M, A5* M/(318,7G15.5))
END
SUBROUTINE PRTDIS(ID,X,B,F,NDM,NDF)

C
C-----------OUTPUT NODAL VALUES.
C
LOGICAL PCOMP,ACFL,ELAS,INCFL,ISFL,NDFL,TRANS,UNFL,POFL,FMFL
COMMON/NLDA TA/ACFL,ELAS,INCFL,ISFL,NDFL,TRANS,UNFL
COMMON/PRLDP/ TPROP,PROP
COMMON/CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEP,NEQ,IPR
COMMON/LABEL/ PDIS(6),A(6),BC(2),DI(6),CD(3),TE(3),FD(3)
COMMON/TDATA/ TIME,DT,C1,C2,C3,C4,C5
DIMENSION X(NDM,1),B(1),UL(6),ID(NDF,1),F(NDF,1)
COMMON/PODATA/POFL,FMFL
DATA BL/4H8LAN/,NTAPE1/30/,NTAPE2/3I/

Cooting since PROP could be changed by transition zone
C adjustment, and since the load is input incrementally
C tally the current total value of input is determined
C by TPROP which is updated for each increment for
C nonlinear analysis in module "PLASTIC".

DPROP = PROP
IF(.NOT.ELAS) DPROP = TPROP
IF(POFL) GO TO 103

DO 102 II = 1,NUMNP,50
WRITE(6,2000) 0,HEAD,TIME,(I.CD(1),CD(2),I=1,NDM),(I,.DI(1)
JJ = MINO(NUMNP,II+49)
DO 102 N = II,JJ
IF(PCOMP(X(I,N)<BL)) GO TO 101
DO 100 I = 1,NDF
UL(I) = F(I,N)*DPROP
K = IABS(ID(I,N))
100 IF(K.GT.0) UL(I) = B(K)
WRITE(6,PDIS) N,(X(I,N),I=1,NDM), (UL(I),I = 1,NDF)
101 CONTINUE
102 CONTINUE
C
C........PRINT FOR POSTPROCESSOR.
C
103 DO 106 II = 1, NUMNP
   IF(PCOMP(K(1,II),BL)) GO TO 105
   DO 104 I = 1, NDF
      UL(I) = F(I,II)*DPROP
      K = IABS(ID(I,II))
      IF(K.GT.0) UL(I) = B(K)
   104 CONTINUE
   IF(.NOT.FMFL) WRITE(NTAPE1) (UL(I), I=1,NDF)
   IF(FMFL) WRITE(NTAPE2,2001) (UL(I), I=1,NDF)
  105 CONTINUE
  106 CONTINUE
RETURN
END

2000 FORMAT(A1,20A4,/*NODAL DISPLACEMENTS*/,5X,/*TIME*/,E13.5/
       1 6X,/*NODF*/,9(I7,A4,A2))
2001 FORMAT(8G10.3)
END
SUBROUTINE PRTREA(R,NDF)

C
C............PRINT NODAL REACTIONS
C

DIMENSION R(NDF,1),RSUM(6),ASUM(6)
COMMON /CDATA/ 0,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
DO 50 K = 1,NDF
   RSUM(K) = 0.
50  ASUM(K) = 0.
DO 100 N = 1,NUMNP,50
   J = MIN0(NUMNP,N+49)
   WRITE(6,2000) O,HEAD,(K,K=1,NDF)
DO 100 I = N,J
   DO 75 K = 1,NDF
      R(K,I) = -R(K,I)
      RSUM(K) = RSUM(K) + R(K,I)
      ASUM(K) = ASUM(K) + ABS(R(K,I))
75  WRITE(6,2001) I,(R(K,I),K = 1,NDF)
C.............PRINT STATICS CHECK.
   WRITE(6,2002) (RSUM(K),K=1,NDF)
   WRITE(6,2003) (ASUM(K),K=1,NDF)
RETURN
2000 FORMAT(A1,20A4/*5X,'NODAL REACTIONS'/*6X,'NODE',6(I9,' DOF'))
2001 FORMAT(I10,6E13.4)
2002 FORMAT(/7X,'SUM',6E13.4)
2003 FORMAT(/3X,'ABS SUM',6E13.4)
END
SUBROUTINE PSETM(NA, NE, NJ, AFL)
C
C............SET POINTER FOR ARRAYS.
C
LOGICAL AFL
NA = NE
NE = NE + NJ
AFL = .FALSE.
CALL SETMEM(NE)
RETURN
END
SUBROUTINE PSTRES(SIG,P1,P2,P3,NSW)
C
C............COMPUTE PRINCIPAL STRESSES(2 DIMENSIONS). AND STRAINS.
C
DIMENSION SIG(4).
C......STRESSES MUST BE STORED IN ARRAY SIG(4) IN THE ORDER
C...... SIG-11, SIG-22, SIG-33, SIG-12
C (1). PLANE STRESS, PLANE STRAIN -- 1-X, 2-Y, 3-Z
C (2). AXISYMMETRIC -- 1- RADIAL, 2- AXIAL, 3- THETA
C
NSW = 1 -- PRINCIPAL STRAINS (USE 0.5*SIG(4) FOR SHEAR STRAIN.
C = 0 -- PRINCIPAL STRESSES (NO CHANGE --FACTOR = 1.0 )
C
FACTOR = 1.0
IF(NSW.EQ.1) FACTOR = 0.5
XII = (SIG(1)+SIG(2))/2.
XI2 = (SIG(1)-SIG(2))/2.
RHO = SQRT(XI2*XI2+SIG(4)*FACT0R*SIG(4)*FACT0R)
P1 = XII + RHO
P2 = XII - RHO
P3 = 45.0
IF(XI2.NE.0.0) P3=22.5*ATAN2(SIG(4)*FACT0R,XI2)/ATAN(1.0)
RETURN
END
SUBROUTINE PZERO(V, NN)

C
C*####*##««ZERO REAL ARRAY#
DIMENSION V(NN)
DO 100 N = 1, NN
  100 V(N) = 0.0
RETURN
END
SUBROUTINE RESIO (D,DV,SIG,EPS,EOSIG,TSIG,RATIO,NDM2)
C
C***************************************************************************
C
C PROGRAM NAME : RESIO (UAL FORCE)
C
C PURPOSE : FIND APPROPRIATE STRESS COMPONENTS TO BE
C PROCESSED AS -BT*SIG*DV
C
C LOGICAL FLOW : 1. IF UNLOAD PHASE -TOTAL STRESSES.
C 2. IF LIN/NONLIN MATER'L BUT STILL ELASTIC
C -- ELASTIC STRESS (FOR DISP. INPUT )
C 3. IF ITERATION PHASE - RESTORE PLASTIC
C STRESS FROM AUX. STORAGE -- DSIG=-DSIGP
C 4. IF NOT ABOVE (SEQUENTIALLY) THEN ::
C 4.1 : CALCULATE "EPMAT".
C 4.2 : DSIG = DSIG*DEP (FOR DISP. INPUT)
C 5. IF( NOT.ITERATION) - DSIG=DSIG+TOTAL STRESS
C (EQUILIBRIUM CHECK).
C CALLED BY : "PFORM" * MACRO *FORM* WITH ISW = 6.
C
C NOTE : MAINLY FOR DISPLACEMENT INPUT CONVERSION (TO
C EQUIVALENT LOAD ). EXCEPT FOR UNLOAD AND ITERA-
C TION PHASE.
C
C OUTPUT : APPROPRIATE STRESS FOR PROCESSING.
C
C MODULE CALLED: "EPMAT"
C
C LOGICAL UNIT : 11 FOR STORING PLASTIC STRESS INCRE.
C
C***************************************************************************
C
C LOGICAL ACFL ,ELAS,INCFI,ISFL,NDFL,TRANS,UNFL
C
C 1 ,CUVCH,FRFL
C COMMON/NL DATA/ ACFL ,ELAS,INCFI,ISFL,NDFL,TRANS,UNFL
COMMON /CR0LL/XR(3),ROLLRA,FRFAC,UVCH,FRFL
COMMON/EPRS/ROSIG(6)
DIMENSION  D(1),TSIG(1),EPS(1),EP(6,6),SIG(1)

IF(UNFL.OR.NDFL) GO TO 70
  IF(D(8).GE.0) GO TO 20
    IF(PAT10.GE.0.0) GO TO 20
    IF(ISFL) GO TO 30
C
C................NONLINEAR MAT/ANAL. POST-YIELD ELEMENT.
C
C    CALL EPMAT( NDM2,DV,D,EQSIG,TSIG,EP)
C
C................CALCULATE ELASTO-PLASTIC STRESS : SIG = DEP * EPS
C
DO 10  I = 1,NDM2
  SIG(I) = 0.0
  DO 10  J = 1,NDM2
    SIG(I) = SIG(I) + EP(I,J)*EPS(J)
  CONTINUE
10 CONTINUE
GO TO 50

C
C................CALCULATE ELASTIC STRESS.
C
20 SIG(1) = D(1)*EPS(1) + D(2)*(EPS(3)+EPS(2))
SIG(2) = D(1)*EPS(2) + D(2)*(EPS(1)+EPS(3))
SIG(3) = D(1)*EPS(3) + D(2)*(EPS(1)+EPS(2))
SIG(4) = D(3) * EPS(4).

C
C................FOR PLANE STRESS, SIG-Z IS ZERO.
C
IF(D(7).LT.0 .AND.NDM2.EQ.4) SIG(3) = 0.0
IF(NDM2.EQ.4) GO TO 50
  SIG(5) = D(3)*EPS(5)
  SIG(6) = D(3)*EPS(6)
GO TO 50
C.............ITERATION WITHIN LOOP. PLASTIC STRESSES.
C
30      READ(11) (SIG(I),I=1,NDM2)
  DO 40 I = 1,NDM2
      SIG(I) = -SIG(I)
  40      CONTINUE
C
50      IF(ISFL) RETURN
      IF(.NOT.FRFL) GO TO 56
C
C        FRICTION EFFECT IN ROLLING
C
  DO 55 JJ = 1,NDM2
      ROSIG(JJ) = FRAC*SIG(JJ) + TSIG(JJ)
  55      CONTINUE
C
56      CONTINUE
C
C.............OVERALL EQUIL. CHECK (ANOTHER PART IN "PFORM" AFTER
C............."ADDSTF" IS CALLED )
C
  DO 60 I = 1,NDM2
      SIG(I) = SIG(I) + TSIG(I)
  60      CONTINUE
C
C.............UNLOAD OF PRINT REACTIONS. /TOTAL STRESS*/
C
  DO 80 I = 1,NDM2
      SIG(I) = TSIG(I)
  80      CONTINUE
C
RETURN
END
SUBROUTINE ROLBC (NBN,NDF,NDF1,NLOC,TRANF,IPBCN,X,ICOUNT)

NAME: BOUNDARY CONDITION CHANGE CHECK FOR ROLLING PROCESS

LOGIC: BASED ON GEOMETRY CHECK WHETHER BOUNDARY CONDITION CHANGE
IS NECESSARY. IF NEW NODES ARE IN CONTACT WITH THE ROLL,
OR NEW POSITIONS ARE FOUND, THEN CALCULATE THE NEW OR
CHANGED DIRECTION COSINES AND STORE THEM.

SUBROUTINES CALLED: NEWDC

CALLED BY: PMACR BY MACRO 'ROBC'.

INPUT: NBN = MAXIMUM NUMBER OF BOUNDARY NODES
NLOC(2,J) = LOCAL NODE AND THEIR TRANF. MATRIX No.
IPBCN(NDF+1,J) = BOUNDARY NODE No. AND FLAG.
CUVCH = LOGICAL FLAG . TRUE = NEEDS CURVATURE CORREC.
XRC3) = CENTER OF THE ROLL.

OUTPUT: IPBCN(NDF+1,J), TRANF(NDME2,J)
ICOUNT = IF ZERO MEANS NO CHANGE OF BOUNDARY CONDITION.

NOTE: THIS ROUTINE ONLY DO CHECKING. ACTION IS TAKEN IN ROUTINE
PMACR.

COMMON/CROLL/ XQ(3),ROLL,FRFAC,OMEGA,CUVCH,FRFL
COMMON/CLOC5/LOFL,NDME2,NTLC,NIDIAL,NTR
DIMENSION IPBCN(NDME2,1), NLOC(2,1), X(NDME2,1), TRANS(NDME2,1)
LOGICAL LOFL, CUVCH, FRFL

C ICOUNT = 0
DO 400 I = 1, NBN
C
C DO 100 J = 1, NTLC
    IF(NLOC(1, J).EQ.IPBCN(1, I)) GO TO 150
100 CONTINUE
150 IF(NLOC(2, I).EQ.0) GO TO 300
C
C NODE IN CONTACT WITH THE ROLL AT THE START OF INCRE.
C
200 IF(X(1, IPBCN(1, I)) .LT.XR(1)) GO TO 250
    ICOUNT = ICOUNT + 1
    NLOC(2, J) = 0
    IPBCN(3, I) = 1
    GO TO 400
C
C NEED NEW CURVATURE
C
250 IF(.NOT.CUVCH) GO TO 400
    IPBCN(3, I) = 0
    JTR = NLOC(2, J)
    CALL NEWDC(NDME2, XR, X(1, IPBCN(1, I)), TRANS(1, JTR))
    GO TO 400
C
C NODF NOT IN CONTACT WITH THE ROLL AT START OF INCRE.
C
300 DIST = SORT((X(1, IPBCN(1, I)) - XR(1))**2 + 1
            (X(2, IPBCN(1, I)) - XR(2))**2)
    IF(DIST.GT.ROLLRA) GO TO 400
    ICOUNT = ICOUNT + 1
    IPBCN(3, I) = 1
C NEED NEW CURVATURE

NTR = NTR + 1
NLOC(2*J) = NTR
CALL NEWOC(NDF,XR,X(1*IPBCN(1*I)),TRANF(1*NTR))

400 CONTINUE
RETURN
END
SUBROUTINE ROLLCU(ROLLRA,NDM,K,JTR,NDME2,F,DR,TRANF,X,XR)

C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
NAME: ROLLING CURVATURE CORRECTION
C
LOGIC: USING DERIVED FORMULA TO ENFORCE THE CONTACT
OF BILLET TO ROLL.
C
OUTPUT: UPDATED COORDINATES.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
DIMENSION DR(1),TRANF(NDME2,1), X(1), XR(1)
C
DTHETA = 1./ROLLRA * F
DX = F * TRANF(1,JTR)
DY = DR(K) * TRANF(2,JTR)
XTEMP = X(1) + DX
YTEMP = X(2) + DY
C
CHECK IF IT IS ABOUT LEAVING THE ROLL.
C
IF(XR(1).NE.XTEMP) GO TO 5
ALPHA = 0.0
GO TO 10
5 ALPHA = ATAN2((XR(2)-YTEMP),(XR(1)-XTEMP))
10 DX = ROLLRA *(COS(ALPHA-DTHETA) - COS(ALPHA))
DY = ROLLRA *(SIN(ALPHA-DTHETA) - SIN(ALPHA))
X(1) = X(1) + DX
X(2) = X(2) + DY
IF(NDM.EQ.2) RETURN
K = K + 1
X(3) = X(3) + DR(K)
RETURN
END
SUBROUTINE SETMEM(J)
C............MONITOR AVAILABLE MEMORY IN BLANK COMMON
C
    COMMON M(1)
    COMMON /PSIZE/ MAX
    K = J
    IF(K.LE.MAX) RETURN
    WRITE(6,1000) K,MAX
    STOP
1000 FORMAT(5X,'**ERROR 0108 INSUFFICIENT STORAGE IN BLAND COMMON*/
          1    17X,'REQUIRED =',18/17X,'AVAILABLE =',18/)
END
SUBROUTINE SHAPE (SS, TT, X, SHP, XS, NDM, NEL, IX, FLG)

C SHAPE FUNCTION ROUTINE FOR TWO DIMENSIONAL ELEMENTS.
C
LOGICAL FLG
DIMENSION SHP(3,1), X(NDM,1), S(4), T(4), XS(2,2), SX(2,2), IX(1)
DATA S/-0.5, 0.5, 0.5, -0.5/, T/-0.5, -0.5, 0.5, 0.5/
C FORM 4-NODE QUADRILATERAL SHAPE FUNCTIONS.
DO 100 I = 1, 4
   SHP(3,I) = (0.5 + S(I) * SS) * (0.5 + T(I) * TT)
   SHP(1,I) = S(I) * (0.5 + T(I) * TT)
100 SHP(2,I) = T(I) * (0.5 + S(I) * SS)
   IF(NEL.GE.4) GO TO 120
C FORM TRIANGLE BY ADDING THIRD AND FOURTH TOGETHER.
DO 110 I = 1, 3
   SHP(I,3) = SHP(I,3) + SHP(I,4)
110 SHP(I,3) = SHP(I,3)
C ADD QUADRATIC TERMS IF NECESSARY.
   IF(NEL.GT.4) CALL SHAP2(SS, TT, SHP, IX, NEL)
C CONSTRUCT JACOBIAN AND ITS INVERSE.
   DO 130 I = 1, NDM
      DO 130 J = 1, 2
         XS(I,J) = 0.0
      130 XS(I,J) = XS(I,J) + X(I,K) * SHP(J,K)
   XSJ = XS(1,1) * XS(2,2) - XS(1,2) * XS(2,1)
   IF(FLG) RETURN
      SX(1,1) = XS(2,2) / XSJ
      SX(2,2) = XS(1,1) / XSJ
      SX(1,2) = -XS(1,2) / XSJ
      SX(2,1) = -XS(2,1) / XSJ
C FORM GLOBAL DERIVATIVES.
   DO 140 I = 1, NEL
      TP = SHP(I,1) * SX(1,1) + SHP(I,2) * SX(2,1)
      SHP(I,2) = SHP(I,1) * SX(1,2) + SHP(I,2) * SX(2,2)
140 SHP(I,1) = TP
   RETURN
END
SUBROUTINE SHAP2(S.T,SHP,IX,NEL)

C ADD QUADRATIC FUNCTIONS AS NECESSARY.

DIMENSION IX(1),SHP(3,1)
S2 = (1.-S*S)/2.
T2 = (1.-T*T)/2.
DO 100 I = 5,NEL
    DO 100 J = 1,3
        SHP(J,I) = 0.0
100    C MIDSIDE NODES (SERENDIPITY).
    IF(IX(5).EQ.0) GO TO 101
        SHP(1,5) = -S*(1.-T)
        SHP(2,5) = -S2
        SHP(3,5) = S2*(1.-T)
101 IF(NEL.LT.6) GO TO 107
    IF(IX(6).EQ.0) GO TO 102
        SHP(1,6) = T2
        SHP(2,6) = -T*(1.+S)
        SHP(3,6) = T2*(1.+S)
102 IF(NEL.LT.7) GO TO 107
    IF(IX(7).EQ.0) GO TO 103
        SHP(1,7) = -S*(1.+T)
        SHP(2,7) = S2
        SHP(3,7) = S2*(1.+T)
103 IF(NEL.LT.8) GO TO 107
    IF(IX(8).EQ.0) GO TO 104
        SHP(1,8) = -T2
        SHP(2,8) = -T*(1.-S)
        SHP(3,8) = T2*(1.-S)
104 IF(NEL.LT.9) GO TO 107
    IF(IX(9).EQ.0) GO TO 107
        SHP(1,9) = -S*T2
        SHP(2,9) = -T*S2
        SHP(3,9) = 4.*S*S*T2
C............CORRECT EDGE NODES FOR INTERIOR NODE (LAGRANGIAN)
   DO 106 J = 1,3
      DO 105 I = 1,4
105   SHP(J,I) = SHP(J,I) - 0.25 * SHP(J,9)
      DO 106 I = 5,8
106   IF(I*X(I) .NE. 0) SHP(J,I) = SHP(J,I) - 0.5 * SHP(J,9)
C............CORRECT CORNER NODES FOR PRESENCE OF MIDSIDE NODES.
   107 K = 8
   DO 109 I = 1,4
      L = I + 4
      DO 108 J = 1,3
108   SHP(J,I) = SHP(J,I) - 0.5 * (SHP(J,K) + SHP(J,L))
   109 K = L
   RETURN
END
TRANSPose OF THE COFACTOR MATRIX

X(I) = X(I)+X(I)*X(I)*X(I)*X(I)

100 CONTINUE

DIMENSION SHP(4*4*DMP*2),UX(U),UX(I),UX(I),UX(I),UX(I),UX(I)

SUBROUTINE SHAPE3(SS,UX,UX,UX,UX,UX,UX,UX,UX)
XSJ = XS(1,1) * SX(1,1) + XS(2,1) * SX(2,1) + XS(3,1) * SX(3,1)

IF(XSJ.EQ.0.0) GO TO 200
IF(FLG) RETURN
DO 135 I = 1,3
DO 135 J = 1,3
135 SX(I,J) = SX(I,J) / XSJ
C............FORM GLOBAL DERIVATIVFS(ELEMENTS OF THE B MATRIX).
C
DO 140 I = 1, NEL
    TP1 = SHP(1,I) * SX(1,1) + SHP(2,I) * SX(1,2) + SHP(3,I) * SX(1,3)
    TP2 = SHP(1,I) * SX(2,1) + SHP(2,I) * SX(2,2) + SHP(3,I) * SX(2,3)
    SHP(3,I) = SHP(1,I) * SX(3,1) + SHP(2,I) * SX(3,2) + SHP(3,I) * SX(3,3)
    SHP(1,I) = TP1
    SHP(2,I) = TP2
140 CONTINUE
RETURN
C............ERROR EXIT : JACOBIAN ZERO.
200 WRITE(6,1000)
1000 FORMAT(/1X,'*FATAL ERROR JACOBIAN ZERO *')
STOP
END
SUBROUTINE STCNV (NLOC, IX, TRANF, S, NST, NEL, NDM)

NAME : STIFFNESS CONVERSION ROUTINE

PURPOSE : FIND THE NODE THAT NEEDS TO BE TRANSFORMED DUE TO
          LOCALLY ORIENTED NODES. FIND THE SUBMATRIX AND
          TRANSFORM BEFORE THE LOWER HALF OF THE STIFFNESS
          MATRIX IS FILLED UP.

MAJOR VARIABLES :

NLOC : ARRAY (2,NTLC) CONTAIN NODE AND TRANS. MAT. NO.
IX : ARRAY (NEL) HAS THE CONNECTIVITY DATA OF THE ELE.
NTOTAL: TOTAL NUMBER OF ACTUAL LOCAL NODES.
NTR : TOTAL NUMBER OF TRANSFORMATION MAT. (ACTUAL)
NEL : MAX. NO. OF NODES /ELEMENT.
TRANF : TRANSFORMATION MATRIX (NDME2,NTLC)
TEMP : ARRAY OF LENGTH NDM * NDM FOR TEMPORARY STORAGE
       OF SUB STIFFNESS MATRIX IN TRANSFORMATION.

OUTPUT S : STIFFNESS MATRIX OF AN ELEMENT (NST * NST)
ROUTINE CALLED: "STMULT".
CALLED BY : "ELMTON".
DIMENSION NLOC(2,1), IX(1), TRANF(NDME2,1), S(NST,1), TEMP(9)
COMMON/CLOCA/ LDFL, NDME2, NTLC, NTOTAL, NTR
LOGICAL LOFL
C
DO 50 I = 1, NEL
   DO 10 II = 1, NTOTAL
      IF(NLOC(I,II) .EQ. IX(I)) GO TO 20
   CONTINUE
   GO TO 50
C
A LOCAL NODE IS FOUND. FIRST GET THE TRANF. MAT. NO.
C
20 ITR = NLOC(2,II)
   DO 40 J = 1, NEL
      IROW = (I-1)*NDM + 1
      ICOL = (J-1)*NDM + 1
C
      CALCULATE THE ADDRESS OF THE FIRST ELEMENT OF THE SUBMAT.
C
      IF(J .LT. I) GO TO 30
      S = LT*S (AND LT*S*L FOR DIAGONAL TERM (J=I))
      CALL STMULT(S,TRANF(1,ITR), TEMP,NST,NDM,ICOL,ICOL,1)
      IF(J.EQ.1) CALL STMULT(S,TRANF(1,ITR), TEMP,NST,NDM,
                                 ICOL,ICOL,2)
      GO TO 40
C
S = S * L
C
30 CALL STMULT(S,TRANF(1,ITR), TEMP,NST,NDM,ICOL,IROW,2)
40 CONTINUE
50 CONTINUE
RETURN
END
SUBROUTINE STMULT(S,TRANF,TEMP,NST,NDM,ISTART,JSTART,NTSW)

NAME : STIFFNESS MULTIPLICATION ROUTINE
PURPOSE : PERFORM EITHER LT * S OR S * L FOR LOCAL NODES.

MAJOR VARIABLES :

NTSW = TRANSFORMATION SWITCH.
= 1 LT * S
= 2 S * L

TRANF = ARRAY OF DIMENSION (NDM,NDM) STORES THE COEFF.
OF DIRECTION COSINE. (= LT )

TEMP = WORKING ARRAY .

S = STIFFNESS MAT. OF AN ELEMENT.

ISTART = STARTING ADDRESS OF ROW OF THE 1ST ELM. IN SUB MAT.
JSTART = " "... COLUMN........

CALLED BY : "STCNV" BY "ELMTON".

DIMENSION TEMP(NDM,1), TRANF(NDM,1), S(NST,1)

IEND = ISTART + NDM - 1
JEND = JSTART + NDM - 1
GO TO (1,2), NTSW

LT*S (NOTE THAT THE STORAGE OF L IS REALLY ITS TRANSPOSE DUE
TO CONVENIENCE OF INPUT AND MEMORY MANGEMENT)
1 DO 30 I = ISTART, IEND
   T1 = I - ISTART + 1
   DO 20 J = JSTART, JEND
      T2 = J - JSTART + 1
      TEMP(T1, T2) = 0.0
      DO 10 K = ISTART, IEND
         K1 = K - ISTART + 1
         TEMP(T1, T2) = TEMP(T1, T2) + TRANF(T1, K1) * S(K, J)
      10 CONTINUE
   20 CONTINUE
30 CONTINUE
GO TO 70
C
C S* L
C
2 DO 60 I = ISTART, IEND
   T1 = I - ISTART + 1
   DO 50 J = JSTART, JEND
      T2 = J - JSTART + 1
      TEMP(T1, T2) = 0.0
      DO 40 K = JSTART, JEND
         K1 = K - JSTART + 1
         TEMP(T1, T2) = TEMP(T1, T2) + S(I, K) * TRANF(T2, K1)
      40 CONTINUE
   50 CONTINUE
60 CONTINUE
C
C UPDATE S
C
70 DO 90 I = ISTART, IEND
   T1 = I - ISTART + 1
   DO 80 J = JSTART, JEND
      T2 = J - JSTART + 1
      S(I, J) = TEMP(T1, T2)
   80 CONTINUE
90 CONTINUE
RETURN
END
SUBROUTINE UACTOL(A,C,B,JDIAG,NEQ,AFAC,BACK)
LOGICAL AFAC,BACK
DIMENSION A(1),C(1),JDIAG(1),B(1)

C
C............UNSYYMETRIC ACTIVE COLUMN PROFILE EQUATION SOLVER.
C
C............FACTOR A TO UT*DU, REDUCE B TO Y.
JR = 0
DO 300 J = 1,NEQ
   JD = JDIAG(J)
   JH = JD - JR
   IF(JH.LE.1) GO TO 300
   IS = J + 1 - JH
   IE = J - 1
   IF(.NOT.AFAC) GO TO 250
   K = JR + 1
   ID = 0
   DO 200 I = IS,IE
      IR = ID
      ID = JDIAG(I)
      IH = MIN0(13-IR-1,I-IS)
      IF(IH.EQ.0) GO TO 150
      A(K) = A(K) - DOT(A(K-IH)*C(ID-IH),IH)
      C(K) = C(K) - DOT(C(K-IH)*A(ID-IH),IH)
150     IF(A(ID).NE.0.0) C(K) = C(K)/A(ID)
   K = K + 1
   A(JD) = A(JD) - DOT(A(JR+I)*C(JR+1),JH-1)
C FORWARD REDUCE THE RIGHT HAND SIDE.
200   IF(BACK) B(J) = B(J) - DOT(C(JR+1)*B(IS),JH-1)
300   JR = JD
IF(.NOT.BACK) RETURN
C BACK SUBSTITUTION.
J = NEQ
JD = JDIAG(J)
500   IF(A(JD).NE.0.0) B(J) = B(J)/A(JD)
D = B(J)
J = J-1
IF(J.LE.0) RETURN
JR = JDIAG(J)
IF(JD-JR.LE.1) GO TO 700
IS = J - JD + JR + 2
K = JR - IS + 1
600 DO 600 I = IS,J
     B(I) = 3(I) - A(I+K)*D
700 JD = JR
GO TO 500
END
SUBROUTINE ULSTOR(B, IX, TSIG, TEPS, RATIO, EOSIG, UNSIGY, NEN, NGAUP, NTOT, NSW)

PROGRAM NAME : ULSTOR (UNLOAD STORAGE)

PURPOSE : STORING AND RETRIEVING DATA FOR INVOLUNTARY UNLOADING.

LOGICAL UNIT USED :

14-- STORING DATA FOR EACH INCREMENT ONLY, NOT TO BE OPERATED ON.

CALLING MODULE:

(1): "PMACR", MACRO 'NONL' NSW = 1
(2): "PMACR", MACRO 'TIME' NSW = 2
(3): "PMACR", MACRO 'PLAS' NSW = 3 (RETRIEVING)

COMMON/CDATA/ O,HEAD(20),NUMNP,NUMMAT,NEN,NEQ,IPR
COMMON/ELDATA/ 3M,N,MA,MCT,IEL,NEL
COMMON/PROLD/ TPROP,PROP
DIMENSION TSIG(NTOT,1), TEPS(NTOT,1), RATIO(NGAUP,1),
1 EOSIG(NGAUP,1), R(1), IX(NEN,1), D(12,1), UNSIGY(NGAUP,1)
2 *TEMP(64)

REWIND 14
GO TO (100, 200, 300), NSW
C............ PRESETTING YIELD STRESS ARRAY FOR CHECKING UNLOADING.
C
  100 DO 120 N = 1,NUMEL
      MA = IX(NEN1,N)
      DO 110 L = 1,NGAUP
          UNSIGY(L,N) = D(IO,MA)
      110 CONTINUE
C
  120 CONTINUE
  RETURN
C
C............STORING ON TAPE 14.
C
  200 WRITE(14) TPROP
     DO 210 N = 1,NUMEL
          WRITE(14) (TSIG(J,N), J = 1,NTOT)
          WRITE(14) (TEPS(J,N), J = 1,NTOT)
          WRITE(14) (RATIO(L,N), L = 1,NGAUP)
          WRITE(14) (EOSIG(L,N), L = 1,NGAUP)
     210 CONTINUE
     WRITE(14) (B(J),J=1,NE0)
     REWIND 14
     RETURN
C
C............RETRIEVING TAPE 14.
C
  300 READ(14) TPROP
     DO 310 N = 1,NUMEL
          READ(14) (TSIG(J,N), J = 1,NTOT)
          READ(14) (TEPS(J,N), J = 1,NTOT)
C
  FOR UNLOADED POINT NEED PRESERVE THE POSITIVE SIGN (CHANGED
  IN ROUTINE "INSTS") FOR RECALCULATING STIFFNESS IN "ELMTON".
C
  305 DO 305 I = 1,NGAUP
      TEMP(I) = RATIO(I,N)
      READ(14) (RATIO(L,N), L = 1,NGAUP)
      305
C
C......A PREVIOUSLY YIELDED POINT AFTER MODIFICATION WILL BE
C POSITIVE (ELASTIC) AND LESS THAN ONE (YIELDED).
C
DO 306 I = 1,NGAUP
306 IF(TEMP(I).GT.0.0.AND.TEMP(I).LT.1.0) RATIO(I,N) = TEMP(I)
C
READ(14) (EQSIG(L,N), L = 1,NGAUP)
310 CONTINUE
READ(14) (B(J),J = 1,NEQ)
REWIND 14
RETURN
END
SUBROUTINE WRTBDU (NDF, NEL, RRINV, UL, SHP, EPGR)

C
C PURPOSE : CALCULATE B * DU AT THE GAUSS POINT LEVEL
C
C NAME : WRITE B * DU
C
C CALLED BY : ELMT01 BY MACRO PLAST
C
C TAPE USED : 18
C
C
DIMENSION EPGR(3,3), SHP(3,1), UL(NDF,1)

C
DO 100 I = 1,2
DO 90 K = 1,2
EPGR(I*K) = 0.0
DO 80 J = 1, NEL
EPGR(I*K) = EPGR(I*K) + SHP(K,J) * UL(I,J)
80 CONTINUE
90 CONTINUE
100 CONTINUE

C
EPGR(3,1) = 0.0
DO 200 J = 1, NEL
EPGR(3,1) = EPGR(3,1) + SHP(3,J) * UL(1,J) * RRINV
200 CONTINUE

C
C STORE ON TAPE 18
C
DO 300 I = 1,2
WRITE(18) (EPGR(I*K), K = 1,2)
300 CONTINUE
WRITE(18) EPGR(3,1)

C
RETURN
END
SUBROUTINE XCOOR(NUMNP, NDM, NLOC, TRANF, X, DR, ID, F)

NAME : UPDATE COORDINATE

LOGIC : UPDATE THE COORDINATES ALSO CHECK

(1) ANY RESTRAINT?
(2) ANY LOCAL DEFINED AXIS?
(3) ANY CURVATURE CORRECTION?

IN ORDER TO EXTRACT THE CORRECT DISPLACEMENT COMPONENT.

SUPPORTING ROUTINE : ROLLCU (CURVATURE CORRECTION )

CALLED BY : ROUTINE PMACR OF MACRO 'UPCO'.

COMMON/PRLOD/TPROP..PROP
COMMON/CROLL/XR(3),ROLLRA,FRFAC,OMEGA,CUVCH,FRFL
COMMON/LOCA/LOFL,NDME2,NTLC,NTOTAL,NTR
LOGICAL LOFL, CUOCH, FRFL
DIMENSION NLOC(2,1),TRANF(NDME2,1),X(NDM,1),DR(1),
1 ID(NDM,1),F(NDM,1)

K = 0
DO 700 I = 1,NUMNP
  IF (.NOT. LOFL) GO TO 400

NEED TO CHECK LOCAL AXIS

DO 100 J = 1,NTLC
  IF (NLOC(I,J).EQ.1) GO TO 200
100 CONTINUE
GO TO 400
THE NODE HAS LOCAL DEFINED AXIS

200  JTR = NLOC(2,J)
     IF(JTR.EQ.0) GO TO 400
     IF(.NOT.CUVCH) GO TO 300

CURVATURE CORRECTION IS NEEDED

CALL ROLLCU(ROLLRA,NDM,K,JTR,NDME2,PROP*F(1,I),DR,TRANF,1,
X(1,I),XR)
GO TO 700

NO NEED FOR CURVATURE CORRECTION, BUT LOCAL AXIS.

300  K = K + 1
     DX = DR(K)*TRANF(1,JTR)
     DY = DR(K)*TRANF(2,JTR)
     X(1,I) = X(1,I) + DX
     X(2,I) = X(2,I) + DY
     IF(NDM.EQ.2) GO TO 700
     K = K + 1
     X(3,I) = X(3,I) + DR(K)
GO TO 700

IF NOT LOCALLY ORIENTED, CHECK IF ANY RESTRAINT.

400  DO 600 JJ = 1,NDM
     IF(ID(JJ,I).EQ.0) GO TO 500

NOT RESTRAINED

K = K + 1
X(JJ,I) = X(JJ,I) + DR(K)
GO TO 500

RESTRANDED
500      X(JJ,1) = X(JJ,1) + PROP*F(JJ,1)
600      CONTINUE
700      CONTINUE
    RETURN
    END
B. Listing of the Finite Element Postprocessor
SUBROUTINE ADD (A,B,C,N)

C
C---------- ADD TWO VECTORS WITH N ELEMENTS---------------------
C
C---------- C = A + B-------------------------------------
C
C-------------------------------------------------------------
C
DIMENSION A(I),B(I),C(I)
DO 100 I = 1,N
   C(I) = A(I) + B(I)
100 CONTINUE
RETURN
END
SUBROUTINE CONTOUR (NT, IPT, X, Y, Z, N, NBL, IPL, TERM)

PURPOSE: DRAW CONTOUR ON A TRIANGULIZED 2-D REGION.

THE OPTIONS OF THIS PROGRAM ARE AS FOLLOWS

(1) CAN SPECIFY MAXIMUM AND MINIMUM CONTOUR VALUES
   TO BE CUT OFF. USE OF THIS LIKE IN AVOIDING
   SINGULARITY, OR TO CREATE A SUBREGION OF INTEREST
   FOR DETAIL ED CONTOURING.

(2) THE DEFAULT CONTOUR LINES ARE 11, THAT IS THERE
   ARE 10 EQUIVALENT INTERVALS. INTERVALS OTHER THAN
   THE DEFAULT VALUES CAN BE OBTAINED BY INPUTTING
   INC TO GIVE INC + 1 CONTOUR LINES. NOTE THAT USUALLY
   THE FIRST (MINIMUM) AND THE LAST (MAXIMUM) CONTOUR
   LINES ARE NOT PLOTTED SINCE THEY ARE USUALLY SINGLE
   POINTS ONLY. HENCE, THE FIRST CONTOUR LINE (IF
   EVER PLOTTED) SHOULD CORRESPOND WITH THE 0 LEVEL,
   THE SECOND CONTOUR LINE SHOULD CORRESPOND WITH THE
   FIRST LEVEL AND THE LAST CONTOUR LINE (IF EVER PLOTTED
   ) SHOULD CORRESPOND WITH THE LAST LEVEL (MAXIMUM).

(3) NONEQUAL CONTOUR INTERVALS CAN BE OBTAINED BY INPUTTING
   SPECIFIC CONTOUR VALUES CORRESPONDING TO THE NUMBER
   OF CONTOUR LINES IN (2) ABOVE. THE USE OF THIS CAN
   BE LIKE TO COMPARE WITH OTHER GRAPHS WHICH HAVE THEIR
   OWN CONVENTIONS.

(4) DASHED LINES CAN BE DRAWN INSTEAD OF SOLID LINES.
   THIS HAS THE ADVANTAGE IN IDENTIFYING THE SPECIFIC
   LEVELS OF THE CONTOUR LINES. SINCE IT IS USUALLY
   VERY FAST IN DRAWING ON A GRAPHIC TERMINAL. HOWEVER
   IT IS RECOMMENDED THAT FOR PRODUCTION/DUPLICATION'S
   PURPOSE THAT SOLID LINES BE DRAWN. SINCE THE PARTICULAR
ALGORITHM USED HERE WILL RENDER THE DASHED LINE EVEN
MORE 'BROKEN' HENCE NOT ESTHETICALLY PLEASING.

(5) NO ATTEMPT HAS BEEN MADE TO MINIMIZE 'PEN' MOVEMENT.
IF THIS PROGRAM IS TO BE ADAPTED TO AN INCREMENTAL
PLOTTER SUCH AS A CALCOMP TYPE PLOTTIER THAN THIS
MINIMIZATION MUST BE DONE TO SAVE PLOTTING TIME AND
COST.

(6) NOTE THAT POSSIBLE JAGGED CONTOUR LINES WILL APPEAR
DEPENDS ON THE DISTRIBUTION OF THE DATA POINTS
HENCE THE RESULTANT TRIANGLES GENERATED. THIS IS
MAINLY DUE TO THE 'NO-SMOOTHING' AND 'RAW' PROCESSING
PHILOSOPHY UNDERLYING THE DESIGN OF THIS PROGRAM.
TO MAKE SURE THAT THE CONTOURS ARE NOT TO BE DISTORTED
BY ANY NUMERICAL PROCEDURES HENCE PROVIDE A 'FIRST
HAND' LOOK ON THE RESULTS.

THE PROGRAM FIRST OBTAIN SCALING AND TRANSLATING FACTORS
FROM ROUTINE JSCTR. IT THEN QUERIES FOR OPTIONS.
COORDINATES OF A SPECIFIC CONTOUR VALUE ARE FOUND BY
INVERSE INTERPOLATION OVER A TRIANGULAR CELL. FINALLY,
THE BOUNDARY LINES ARE DRAWN TO FORM A FRAME.

-----------------------------

INPUT : NT = NO. OF TRIANGLES.
 X,Y = 1ST AND 2ND COORDINATES OF THE VERTEX OF
       TRIANGLE.
 Z  = VALUE TO BE CONTOURED
 N  = TOTAL NO. OF NODES OF THE DOMAIN
 NBL = TOTAL NO. OF BOUNDARY SEGMEMENTS.
 IPL = (3,NBL) CONTAINS 1ST,2ND NODE OF THE
       BOUNDARY LINES AND THE TRIANGLE NO.
 IPT = GLOBAL NUMBERS OF TRIANGLES
 LEVEL = FLAG OF SPECIFIC DESIGNATED CONTOUR VALUES.
DASHFL = FLAG FOR DASHED PATTERN.
TERM = FLAG FOR TERMINAL CALLING.
FIRST = FLAG FOR FIRST TIME GOING INTO ANY TRIANGLE.
INC = NO OF INCREMENT OF CONTOUR.

SUBROUTINES/FUNCTIONS CALLED:
MAXMIN, JSCTR (USER SUPPLIED)
GRSTRT, BAUDRT, VIEWT, SCALE, TRANSL, DASHPT,
NEWPAG, MOVE, DRAW, GRSTOP. (IGL SYSTEM SUPPLIED)

JX, JY = 1 AND 2 IF X IS HORIZONTAL, Y IS VERTICAL.
2 AND 1 IF THE OPPOSITE, CHANGED BY JSCTR.

DIMENSION IPT(3,1), X(1), Y(1), Z(1), XL(3), YL(3), ZL(3), XP(2)
I, NL(3), IPL(3,1), ZLEVEL(100)
LOGICAL FIRST, TERM, DASHFL, LEVEL

CHECK EXTREME VALUES TO BE USED FOR SCALING

CALL MAXMIN(X,N,XMAX,XMIN,DX)
CALL MAXMIN(Y,N,YMAX,YMIN,DY)
CALL MAXMIN(Z,N,ZMAX,ZMIN,DZ)
IF(DX.EQ.0.0 .OR. DY.EQ.0.0 .OR. DZ.EQ.0.0)
1 GO TO 100

PRFSET LOGICAL FLAGS AND DIRECTIONS OF AXIS

LEVEL = .FALSE.
DASHFL = .FALSE.
JX = 1
JY = 2
C----- SCALING -----
C
C
CALL JSCTR(DX, DY, XMIN, YMIN, HTRSL, VTRSL,
        1 HSCALE, VSCALE, JX, JY)
C
C----- GET TERMINAL CONVENTION -----
C
CALL TERTMIN(ITEM, IBAUD, IOPT, TERM)
C
C----- INITIALIZE GRAPHIC SYSTEM -----
C
CALL GRTSTRT(ITEM, IOPT)
CALL BAUDRT(IBAUD)
C
C----- TRANSFORMATION ON VIEWING SCREEN ONLY -----
C
CALL VIEWT
CALL SCALE(HSCALE, VSCALE)
CALL TRANSL(HTRSL, VTRSL)
C
C
C----- QUERY FOR OPTIONS -----
C
C
C
C
C----- DASH PATTERN DESIRED ? -----
C
1 WRITE(6, 1100)
   READ(5, 5, ERR=1) IDUM
   IF(IDUM.EQ.1) DASHFL = .TRUE.
C
C----- MAX AND MIN CONTOUR VALUES CHANGED ? -----
C
INC = 10
? WRITE(6, 1000) ZMAX, ZMIN
READ(5,*,ERR=2) NMAI
IF(NMAI.EQ.0) GO TO 10
3 WRITE(6,1001)
   READ(5,*,ERR=3) ZMAX, ZMIN
   DZ = ZMAX - ZMIN
C----- NUMBER OF CONTOUR LEVELS CHANGED FROM DEFAULT ? -------------------
C
10 WRITE(6,1002)
   READ(5,*,ERR=10) NINC
   IF(NINC.NE.0) READ(5,*,ERR=10) INC
   ZDIS = DZ/FLOAT(INC)
C----- CAN INPUT SPECIFIC CONTOUR LEVEL --------------------------------
C
20 WRITE(6,1110)
   READ(5,*,ERR=20) IDUM
   IF(IDUM.NE.1) GO TO 25
   INC1 = INC + 1
   LEVEL = .TRUE.
24 WRITE(6,1120) INC1
   READ(5,*,ERR=24) (ZLEVEL(J), J = 1, INC1)
C----- CONTOURING STARTS -----------------------------------------------
C
25 WRITE(6,1008)
   CALL NEWPAG
C
   ZINT = ZMIN
C----- FOR EACH CONTOUR LEVEL, LOOP OVER ALL TRIANGLES -----------------
C
   DO 80 J = 1, INC1
      IF(LEVEL) ZINT = ZLEVEL(J)
      IF(DASHFL) CALL DASHPT(J)
      LEVEL = .FALSE.
DO 70 I = 1, MT
C
C-------- EXTRACT NODE NUMBER AND THEIR COORDINATES OF A TRIANGLE --------
C
DO 30 K = 1, 3
NL(K) = IPT(K, I)
XL(K) = X(NL(K))
YL(K) = Y(NL(K))
ZL(K) = Z(NL(K))
30 CONTINUE
C
FIRST = .TRUE.
C
C---------LOOP OVER EACH SIDE OF THE TRIANGLE-----------------------------
C
DO 60 L = 1, 3
LP1 = L + 1
IF(LP1.GT.3) LP1 = 1
C
C--------ABOVE IS CYCLIC PERMUTATION-- NOW CUT OFF PEAK IF DESIRED------
C
IF(44MAI.NE.0.AND.ZINT.LT.ZMIN.OR.ZINT.GT.ZMAX)
1
C
C--------EQUAL VALUES OF TWO VERTICES OF A SIDE MEANS NO CONTOUR-------
C
CC
C
GO TO 60
C
CC
C
GO TO 60
C
C
BEGIN
C
C----------ORDER THE TWO VALUES ON VERTICES FOR COMPARISION-------------
C
ZP = ZL(L)
ZM = ZL(LP1)
C
IF(ZL(LP1).LT.ZP) GO TO 40
ZP = ZL(LP1)
ZM = ZL(L)
C
40
C
IF(ZINT.LT.ZP.AND.ZINT.GT.ZM) GO TO 50
GO TO 60
C
END;
C
INVERSE LINEAR INTERPOLATION TO LOCATE THE CONTOUR COORD----
C
BEGIN
50
ZDUM = (ZINT-ZL(L))/(ZL(LP1)-ZL(L))
XP(1) = XL(L)+ZDUM*(XL(LP1)-XL(L))
XP(2) = YL(L)+ZDUM*(YL(LP1)-YL(L))
IF(.NOT.FIRST) GO TO 55
CALL MOVE(XP(JX),XP(JY))
FIRST = .FALSE.
GO TO 60
C
55
CALL DRAW(XP(JX),XP(JY))
C
60 CONTINUE
70 CONTINUE
C
OBTAIN NEW CONTOUR VALUES -----------------------------------------------
C
IF(.NOT.LEVEL) ZINT = ZINT + ZDIS
80 CONTINUE
C
CONTOUR COMPLETED. NOW PROCEED TO DRAW BOUNDARY LINES -----------------
C
MOVE PEN TO THE FIRST POINT OF THE BOUNDARY LINE.
C
NL(1) = IPL(1,1)
XP(1) = X(NL(1))
XP(2) = Y(NL(1))
CALL MOVE(XP(JX),XP(JY))
C
NOW DRAW THE REST LINES
C
DO 90 I = 2,NBL
90
NL(1) = IPL(1,1)
XP(1) = X(NL(1))
XP(2) = Y(NL(1))
CALL DRAW(XP(JX),XP(JY))
90 CONTINUE

C
C NOW DRAW THE LAST SEGMENT
C
NL(1) = IPL(2,N3L)
XP(1) = X(NL(1))
XP(2) = Y(NL(1))
CALL DRAW(XP(JX),XP(JY))

C----- DISCONNECT COMMUNICATION WITH IGL --------------------------
C
CALL GRSTOP
C
RETURN
C
100 WRITE(6,1003)
RETURN
C
1000 FORMAT(/5X,'ZMAX = *., G10.3,' ZMIN = *.,G10.3/
1 5X,' DO YOU WANT TO CHANGE THEM? _RET* = NO*)
1001 FORMAT(/5X,' ENTER ZMAX AND ZMIN */)
1002 FORMAT(/5X,' DEFAULT CONTOUR LEVEL IS 11, DO YOU WANT TO CHANGE?*)
C
1003 FORMAT(/5X,' ERROR EXIT. DISTANCE IS ZERO. NO GRAPH. */)
C1004 FORMAT(/5X,' ENTER TERMINAL *BAUDRATE*TERMINAL OPTION*/)
C1005 FORMAT(/5X,' ERROR IN DEVICE CODE*5X,* TEXTRONICS IS 4051*,
C 1 5X, * ADM IS 4010, BOTH HAVE OPTION 1 */)
C1006 FORMAT(/5X,'TERM = *.,15,* BAUDRATE = *.,15,* OPTION = *.,15/
C 1 5X, * ARE THESE O.K. _RET* MEANS O.K.*)
1007 FORMAT(F10.3)
1008 FORMAT(/5X,* PRESS RETURN AFTER GRAPH IS DONE TO CONTINUE*)
C1009 FORMAT(/5X,*DEFAULT AXIS SWITCHED, 1ST AXIS = VERT, 2ND AXIS*,
C 1 * = HORIZ.*)
C1010 FORMAT(/5X,*DO YOU WANT TO CHANGE THE SCALE IN THE VER*,
C   1   'TICAL DIRECTION ?*/5X,*ENTER 0 = NO */)
1011 FORMAT(20I4)
1100 FORMAT(1X,*ENTER <1> IF DASH PATTERN IS DESIRED *)
1110 FORMAT(1X,*ENTER <1> IF WANT TO PLOT SPECIFIC CONTOUR VALUE*)
1120 FORMAT(1X,* THE NUMBER OF LEVELS ARE *,14,/
1    1X,* NOW ENTER THE VALUES *)
   END
SUBROUTINE FRINGE (CONS, TEPS, FRNG, N1, N2, NDM2, NGAUP)
C
C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C C PURPOSE : CALCULATE FRINGE ORDERS.
C C MAJOR VARIABLES :
C C
C NI, N2 : INDICATE WHICH COMPONENTS OF THE STRAIN
C ARE TO BE USED IN THE FORMULA.
C C CONS : OPTICAL CONSTANT/THICKNESS
C C TEPS : TOTAL STRAIN (1 - 4 FOR 2 -D)
C C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C C DIMENSION TEPS(NDM2,1), FRNG(1)
C C
C DO 1 II = 1, NGAUP
C C FRNG(II) = (TEPS(N1,II) - TEPS(N2,II))/CONS
C C 1 CONTINUE
C C
C RETURN
C END
SUBROUTINE IDTANG(NDP,XD,YD,NT,IPT,NL,IPL,IWL,IWP,WK)
C THIS SUBROUTINE PERFORMS TRIANGULATION
C DECLARATION STATEMENTS
DIMENSION XD(1),YD(1),IPT(1),IPL(1),
1 IWL(1),IWP(1),WK(1)
DIMENSION ITF(2)
DATA RATIO/1.0E-6/,NREP/100/, LUN/6/
C STATEMENT FUNCTIONS
DSOF(U1,V1,U2,V2)=(U2-U1)**2+(V2-V1)**2
SIDE(U1,V1,U2,V2,U3,V3)=(V3-V1)*(U2-U1)-(U3-U1)*(V2-V1)
C PRELIMINARY PROCESSING
10 NDP0=NDP
NDPM1=NDP0-1
IF(NDP0.LE.4) GO TO 90
C DETERMINES THE CLOSEST PAIR OF DATA POINTS AND THEIR MIDPOINT.
20 DSQMN=DSOF(XD(1),YD(1),XD(2),YD(2))
IPMN1=1
IPMN2=2
DO 22 IP1=1,NDPM1
  X1=XD(IP1)
  Y1=YD(IP1)
  IP1P1=IP1+1
  DO 21 IP2=IP1P1,NDP0
    DSQI=DSOF(X1,Y1,XD(IP2),YD(IP2))
    IF(DSQI.EQ.0.0) GO TO 91
    IF(DSQI.GE.DSQMN) GO TO 21
    DSQMN=DSQI
    IPMN1=IP1
    IPMN2=IP2
21 CONTINUE
22 CONTINUE
DSQ12=DSQMN
XDMP=(XD(IPMN1)+XD(IPMN2))/2.0
YDMP=(YD(IPMN1)+YD(IPMN2))/2.0
C SORTS THE OTHER (NDP-2) DATA POINTS IN ASCENDING ORDER OF
C DISTANCE FROM THE MIDPOINT AND STORES THE SORTED DATA POINT
C NUMBERS IN THE IWP ARRAY
30  JP1=2
   DO 31  IP1=1,NDP0
      IF(IP1.EQ.IPMN1.OR.IP1.EQ.IPMN2) GO TO 31
      JPI=JP1+1
      IWP(JPI)=IP1
      WK(JPI)=DSQF(XDMP*YDMP*XD(IP1)*YD(IP1))
   31  CONTINUE
   DO 33  JP1=3,NDPO
      DSQMN=WK(JP1)
      JPMN=JP1
      DO 32  JP2=JP1,NDPO
         IF(WK(JP2).GE.DSQMN) GO TO 32
         DSQMN=WK(JP2)
         JPMN=JP2
      32  CONTINUE
      ITS=IWP(JP1)
      IWP(JP1)=IWP(JPMN)
      IWP(JPMN)=ITS
      WK(JPMN)=WK(JP1)
   33  CONTINUE
C IF NECESSARY, MODIFIES THE ORDERING IN SUCH A WAY THAT THE
C FIRST THREE DATA POINTS ARE NOT COLLINEAR.
   35  AR=DSQ12*RATIO
      X1=XD(IPMN1)
      Y1=YD(IPMN1)
      DX21=XD(IPMN2)-X1
      DY21=YD(IPMN2)-Y1
      DO 36  JP=3,NDPO
         IP=IWP(JP)
         IF(ABS((YD(IP)-Y1)*DX21-(XD(IP)-X1)*DY21).GT.AR) 1
      1      GO TO 37
   36  CONTINUE
GO TO 92
   37  IF(JP.EQ.3) GO TO 40
      JPMX=JP
      JP=JPMX+1
   38  JPC=4,JPW
      DO 38
JP=JP-1
IWP(JP)=IWP(JP-1)
38 CONTINUE
IWP(3)=IP
C FORMS THE FIRST TRIANGLE. STORES POINT NUMBERS OF THE VER-
C TERXES OF THE TRIANGLE IN THE IPT ARRAY, AND STORES POINT NUM-
C BERS OF THE BORDER LINE SEGMENTS AND THE TRIANGLE NUMBER IN
C THE IPL ARRAY.
40 IP1=IPMN1
IP2=IPMN2
IP3=IWP(3)
IF(SIDE(XD(IP1),YD(IP1),XD(IP2),YD(IP2),XD(IP3),YD(IP3)))
1           .GE.0.0) GO TO 41
IP1=IPMN2
IP2=IPMN1
41 NT0=1
NTT3=3
IPT(1)=IP1
IPT(2)=IP2
IPT(3)=IP3
NL0=3
NLT3=9
IPL(1)=IP1
IPL(2)=IP2
IPL(3)=1
IPL(4)=IP2
IPL(5)=IP3
IPL(6)=1
IPL(7)=IP3
IPL(8)=IP1
IPL(9)=1
C ADDS THE REMAINING (NDP-3) DATA POINTS, ONE BY ONE.
50 DO 79 JP1=4,NDP0
   IP1=IWP(JP1)
   X1=XD(IP1)
   Y1=YD(IP1)
C -DETERMINES THE VISIBLE BORDER LINE SEGMENTS.
IP2=IPL(1)
JP MN=1
DXMN=XD(IP2)-X1
DYMN= YD(IP2)-Y1
DSQMN=DXMN**2+DYMN**2
ARMN=DSQMN*RATIO
JP MX=1
DXMX=DXMN
DYMX=DYMN
DSQMX=DSQMN
ARMX=ARMN
DO 52 JP2=2,NL0
IP2=IPL(3*JP2-2)
DX= XD(IP2)-X1
DY=YD(IP2)-Y1
AR=DY*DXMN- DX*DYMN
IF(AR.GT.ARMN)GO TO 51
DSQI=DX**2+DY**2
IF(AR.GE.(-ARMX).AND.DSQI.GE.DSQMN)GO TO 51
JP MN=JP2
DXMN=DX
DY MN=DY
DSQ MN=DSQI
ARMN=DSQ MN*RATIO
51 AR=DY*DXMX- DX*DYMX
IF(AR.LT.(-ARMX))GO TO 52
DSQI=DX**2+DY**2
IF(AR.LE.ARMX. AND.DSQI.GE.DSQMX)GO TO 52
JP MX=JP2
DXMX=DX
DYMX=DY
DSQMX=DSQI
ARMX=DSQMX*RATIO
52 CONTINUE
IF(JP MX.LT.JPMN)JPMX=JPMX+NL0
NSH=JPMN-1
IF(NSH.LE.0)GO TO 60
C - SHIFTS(ROTATES) THE IPL ARRAY TO HAVE THE INVISIBLE BORDER
C - LINE SEGMENTS CONTAINED IN THE FIRST PART OF THE IPL ARRAY.
NSHT3=NSHT3*3
DO 53 JP2T3=3,NSHT3,3
   JP3T3=JP2T3+NLT3
   IPL(JP3T3-2)=IPL(JP2T3-2)
   IPL(JP3T3-1)=IPL(JP2T3-1)
   IPL(JP3T3) =IPL(JP2T3)
53 CONTINUE
DO 54 JP2T3=3,NLT3,3
   JP3T3=JP2T3+NSHT3
   IPL(JP2T3-2)=IPL(JP3T3-2)
   IPL(JP2T3-1)=IPL(JP3T3-1)
   IPL(JP2T3) =IPL(JP3T3)
54 CONTINUE
JPMX=JPMX-NSH
C - ADDS TRIANGLES TO THE IPT ARRAY, UPDATES BORDER LINE
C - SEGMENTS IN THE IPL ARRAY, AND SETS FLAGS FOR THE BORDER
C - LINE SEGMENTS TO BE REEXAMINED IN THE IWL ARRAY.
60 JWL=0
DO 64 JP2=JPMX,NL0
   JP2T3=JP2+3
   IPT1=IPL(JP2T3-2)
   IPT2=IPL(JP2T3-1)
   IPT =IPL(JP2T3)
   NT0=NT0+1
   NTT3=NTT3+3
   IPT(NTT3-2)=IPT2
   IPT(NTT3-1)=IPT1
   IPT(NTT3) =IPT
C - - ADDS A TRIANGLE TO THE IPT ARRAY.
C - - UPDATES BORDER LINE SEGMENTS IN THE IPL ARRAY.
   IF(JP2.NE.JPMX) GO TO 61
   IPL(JP2T3-1)=IPT1
   IPL(JP2T3) =NT0
61 IF(JP2.NE.NL0) GO TO 62
   NLTN=JPMX+1
NLNT3=NLN*3
IPL(NLNT3-2)=IPL
IPL(NLNT3-1)=IPL(1)
IPL(NLNT3) =NT0
C - DETERMINES THE VERTEX THAT DOES NOT LIE ON THE BORDER
C - LINE SEGMENTS.
62  ITT3=IT3
   IPT1=IPT(ITT3-2)
   IF(IPT1.NE.IPL1.AND.IPT1.NE.IPL2) GO TO 63
   IPT1=IPT(ITT3-1)
   IF(IPT1.NE.IPL1.AND.IPT1.NE.IPL2) GO TO 63
   IPT1=IPT(ITT3)
C - CHECKS IF THE EXCHANGE IS NECESSARY.
63  IF(IDXCHG(XD,YD,IPT1,IPT2,IPL1,IPL2).EQ.0) GO TO 64
C - MODIFIES THE IPT ARRAY WHEN NECESSARY
   IPT(1TT3-2)=IPT1
   IPT(1TT3-1)=IPL1
   IPT(1TT3) =IPL
   IPT(1TT3-1)=IPT1
   IF(JP2.EQ.JPMX.AND.IPL(JP2T3)=IT)
      IPL(JP2T3)=IT
   IF(JP2.EQ.NL0.AND.IPL(3).EQ.IT)
      IPL(3)=NT0
C - SETS FLAGS IN THE IWL ARRAY.
   JWL=JWL+4
   IWL(JWL-3)=IPL1
   IWL(JWL-2)=IPT1
   IWL(JWL-1)=IPT1
   IWL(JWL) =IPL2
64  CONTINUE
   NLO=NLN
   NLT3=NLNT3
   NLF=JWL/2
   IF(NLF.EQ.0) GO TO 79
C - IMPROVES TRIANGULATION.
70  NTT3P3=1TT3+3
    DO 78 IREP=1,NREP
       DO 76 ILF=1,NLF
          ILFT2=ILF+2
IPL1 = IWL(ILFT2 - 1)
IPL2 = IWL(ILFT2)

C - - LOCATES IN THE IPT ARRAY TWO TRIANGLES ON BOTH SIDES OF
C - - THE FLAGGED LINE SEGMENT.

NTF = 0
DO 71 ITT3R = 3 * NTT3, 3
ITT3 = NTT3P3 - ITT3R
IPT1 = IPT(ITT3 - 2)
IPT2 = IPT(ITT3 - 1)
IPT3 = IPT(ITT3)
IF(IPL1.NE.IPT1 .AND. IPL1.NE.IPT2 .AND.
IPL1.NE.IPT3) GO TO 71
IF(IPL2.NE.IPT1 .AND. IPL2.NE.IPT2 .AND.
IPL2.NE.IPT3) GO TO 71
NTF = NTF + 1
ITF(NTF) = ITT3 / 3
IF(NTF.EQ.2) GO TO 72
71 CONTINUE
IF(NTF.LT.2) GO TO 76

C - - DETERMINES THE VERTEXES OF THE TRIANGLES THAT DO NOT LIE
C - - ON THE LINE SEGMENT.

72 ITT3 = ITF(1) * 3
IPT11 = IPT(ITT3 - 2)
IF(IPT11.NE.IPL1 .AND. IPT11.NE.IPL2) GO TO 73
IPT11 = IPT(ITT3 - 1)
IF(IPT11.NE.IPL1 .AND. IPT11.NE.IPL2) GO TO 73
IPT11 = IPT(ITT3)
73 ITT3 = ITF(2) * 3
IPT12 = IPT(ITT3 - 2)
IF(IPT12.NE.IPL1 .AND. IPT12.NE.IPL2) GO TO 74
IPT12 = IPT(ITT3 - 1)
IF(IPT12.NE.IPL1 .AND. IPT12.NE.IPL2) GO TO 74
IPT12 = IPT(ITT3)

C - - CHECKS IF THE EXCHANGE IS NECESSARY

74 IF(IDXCHG(XD, YD, IPT11, IPT12, IPL1, IPL2), EQ.0) 1
GO TO 76
C - MODIFIES THE IPT ARRAY WHEN NECESSARY.
  IPT(I1T3-2)=IPT11
  IPT(I1T3-1)=IPT12
  IPT(I1T3) =IPT11
  IPT(I2T3-2)=IPT12
  IPT(I2T3-1)=IPT11
  IPT(I2T3) =IPT12

C - SETS NEW FLAGS.
  JWL=JWL+8
  IWL(JWL-7)=IPT11
  IWL(JWL-6)=IPT11
  IWL(JWL-5)=IPT11
  IWL(JWL-4)=IPT12
  IWL(JWL-3)=IPT12
  IWL(JWL-2)=IPT12
  IWL(JWL-1)=IPT12
  IWL(JWL) =IPT11
  DO 75 JLT3=3,NLT3,3
     IPLJ1=IPL(JLT3-2)
     IPLJ2=IPL(JLT3-1)
     IF((IPLJ1.EQ.IPL1.AND.IPLJ2.EQ.IPT12).OR.
     (IPLJ2.EQ.IPL1.AND.IPLJ1.EQ.IPT12))
     1 IPL(JLT3)=ITF(1)
     IF((IPLJ1.EQ.IPL2.AND.IPLJ2.EQ.IPT11).OR.
     (IPLJ2.EQ.IPL2.AND.IPLJ1.EQ.IPT11))
     2 IPL(JLT3)=ITF(2)
  75 CONTINUE

76 CONTINUE
  NLFC=NLF
  NLF=JWL/2
  IF(NLF.EQ.NLFC) GO TO 79

C - - RESETS THE IWL ARRAY FOR THE NEXT ROUND.
  JWL=0
  JWL1MN=(NLFC+1)*2
  NLFT2=NLF*2
  DO 77 JWL1=JWL1MN,NLFT2,2
     JWL=JWL+2
IWL(JWL-1) = IWL(JWL1-1)
IWL(JWL) = IWL(JWL1)
77 CONTINUE
NLF = JWL/2
78 CONTINUE
79 CONTINUE
C REARRANGES THE IPT ARRAY SO THAT THE VERTEXES OF EACH TRIANGLE
C ARE LISTED COUNTER-CLOCKWISE.
80 DO 81 ITT3 = 3, NTT3, 3
IP1 = IPT(ITT3-2)
IP2 = IPT(ITT3-1)
IP3 = IPT(ITT3)
IF(SIDE(XD(IP1), YD(IP1), XD(IP2), YD(IP2), XD(IP3), YD(IP3))
1 .GE. 0.0) GO TO 81
IPT(ITT3-2) = IP2
IPT(ITT3-1) = IP1
81 CONTINUE
NT = NTO
NL = NLO
RETURN
C ERROR EXIT
90 WRITE(LUN, 2090) NDP0
GO TO 93
91 WRITE(LUN, 2091) NDP0, IP1, IP2, X1, Y1
GO TO 93
92 WRITE(LUN, 2092) NDP0
93 WRITE(LUN, 2093)
NT = 0
RETURN
C FORMAT STATEMENTS
2090 FORMAT(IX, '*** NDP LESS THAN 4.**/1.2X, 'NDP = ', I5)
2091 FORMAT(IX, '*** IDENTICAL DATA POINTS.**/
1   'NDP = ', I5, 5X, 'IP1 = ', I5, 5X, 'IP2 = ', I5,
2   5X, 'XD = ', E12.4, 5X, 'YD = ', E12.4)
2092 FORMAT(IX, '*** ALL COLLINEAR DATA POINTS.**/
I   *2X, 'NDP = ', I5)
2093 FORMAT(IX, 'ERROR DETECTED IN ROUTINE IDTANG*/
END
FUNCTION IDXCHG(X, Y, I1, I2, I3, I4)
C THIS FUNCTION DETERMINES WHETHER OR NOT THE EXCHANGE OF TWO
C TRIANGLES IS NECESSARY ON THE BASIS OF MAX-MIN-ANGLE CRITERION
C BY C. L. LAWSON
C DECLARATIVE STATEMENTS
DIMENSION X(1), Y(1)
EQUIVALENCE (C250, C150), (A350, B250), (B350, A150),
(A450, B150), (B450, A250), (C450, C350)
C PRELIMINARY PROCESSING
10 X1=X(I1)
   Y1=Y(I1)
   X2=X(I2)
   Y2=Y(I2)
   X3=X(I3)
   Y3=Y(I3)
   X4=X(I4)
   Y4=Y(I4)
C CALCULATION
20 IDX=0
   U3=(Y2-Y3)*(X1-X3)-(X2-X3)*(Y1-Y3)
   U4=(Y1-Y4)*(X2-X4)-(X1-X4)*(Y2-Y4)
   IF(U3*U4.LE.0.0) GO TO 30
   U1=(Y3-Y1)*(X4-X1)-(X3-X1)*(Y4-Y1)
   U2=(Y4-Y2)*(X3-X2)-(X4-X2)*(Y3-Y2)
   A1SQ=(X1-X3)**2+(Y1-Y3)**2
   B1SQ=(X4-X1)**2+(Y4-Y1)**2
   C1SQ=(X3-X4)**2+(Y3-Y4)**2
   A2SQ=(X2-X4)**2+(Y2-Y4)**2
   B2SQ=(X3-X2)**2+(Y3-Y2)**2
   C3SQ=(X2-X1)**2+(Y2-Y1)**2
   S1SQ=U1*U1/(C1SQ*AMAX1(A1SQ, B1SQ))
   S2SQ=U2*U2/(C2SQ*AMAX1(A2SQ, B2SQ))
   S3SQ=U3*U3/(C3SQ*AMAX1(A3SQ, B3SQ))
   S4SQ=U4*U4/(C4SQ*AMAX1(A4SQ, B4SQ))
   IF(AMIN1(S1SQ, S2SQ).LT. AMIN1(S3SQ, S4SQ)) IDX=1
30 IDXCHG=IDX
RETURN
END
SUBROUTINE JSCTR ( DH, DV, HMIN, VMIN, HTRSL, VTRSL, 
    HSCALE, VSCALE, JX, JY )

C-----------------------------------------------
C
C NAME = SCALING AND TRANSLATION
C
C THIS PROGRAM UPON GIVEN THE MAXIMUM DISTANCE IN 
C HORIZONTAL AND VERTICAL DISTANCE OF A GRAPH AND 
C RETURNS THE SCALING AND TRANSLATING FACTORS FOR 
C A SCREEN OF SIZE XSIZE X YSIZE.
C DEFAULT SCREEN SIZE IS 120 X 90 IN GRAPHICAL 
C UNITS. USER CAN DEFINE HIS OWN XSIZE AND YSIZE 
C TO OBTAIN DIFFERENT PROPORTION OF THE GRAPH.
C HORIZONTAL AND VERTICAL AXES CAN ALSO BE 
C SWITCHED FOR BETTER VIEWING.
C
C DH = MAXIMUM HORIZONTAL DISTANCE OF THE GRAPH 
C DV = MAXIMUM VERTICAL DISTANCE OF THE GRAPH 
C HMIN = MINIMUM HORIZONTAL COORDINATE 
C VMIN = MINIMUM VERTICAL COORDINATE 
C HUNIT = HORIZONTAL UNIT OF THE SCREEN 
C VUNIT = VERTICAL UNIT OF THE SCREEN 
C XSIZE = HORIZONTAL SIZE OF THE SCREEN FOR VIEWING 
C YSIZE = VERTICAL SIZE OF THE SCREEN FOR VIEWING 
C JX = COORDINATE INDICATOR 
C JY = COORDINATE INDICATOR 
C HSCALE = SCALING FACTOR IN HORIZONTAL DIRECTION 
C VSCALE = SCALING FACTOR IN VERTICAL DIRECTION 
C HTRSL = TRANSLATING FACTOR IN HORIZONTAL DIRECTION 
C VTRSL = TRANSLATING FACTOR IN VERTICAL DIRECTION 
C SLFSIZ = LOGICAL FLAG INDICATING SELF DEFINED AXIS
C
C PROGRAMMER = JONAH H.Y. LEE
C DATE = OCT 31, 1982
C
C-----------------------------------------------
DATA HUNIT/120.0/ VUNIT/90.0/
LOGICAL SLFSIZ

C
HSIZE = HUNIT
VSIZE = VUNIT
SLFSIZ = .FALSE.

C----- AXIS LENGTH CAN BE CHOSEN BY USER -------------------------------

C 20 WRITE(6,1000) HSIZE,VSIZE
   READ(5,*,ERR=20) IDUM
   IF(IDUM.EQ.1) GO TO 45

C----- USER CHOSEN AXIS LENGTH -------------------------------------------

C 30 WRITE(6,1004)
   READ(5,*,ERR=30) IDUM
   IF(IDUM.EQ.1) GO TO 45

C----- REDEFINE AXIS -----------------------------------------------------

C 40 WRITE(6,1005)
   READ(5,*,ERR=40) HUNIT,VUNIT
   HSIZE = AMIN1(120.0,AMAX1(HUNIT,0.0))
   VSIZE = AMIN1(90.0,AMAX1(VUNIT,0.0))
   HUNIT = HSIZE
   VUNIT = VSIZE

C----- CAN DISTORT GRAPH -------------------------------------------------

C 45 WRITE(6,1008)
   READ(5,*,ERR=45) IDUM
   IF(IDUM.EQ.1) SLFSIZ = .TRUE.

C 50 WRITE(6,1006) HSIZE,VSIZE
PROCESSING BEGINS

IF(DH.LT.DV) GO TO 80

IF(SLFSIZ) GO TO 60

GRAPH NOT DISTORTED

HSCALE = HSIZE/DH
VSCALE = VSCALE
GO TO 70

GRAPH TO BE DISTORTED

60 HSCALE = HSIZE/DH
VSCALE = VSIZE/DV

TRANSLATION

70 HTRSL = -HMIN * HSCALE
VTRSL = -VMIN * VSCALE
GO TO 200

DH < DV


80 WRITE(6,1007)
READ(5,*,ERR=80) IDUM
IF(IDUM.NE.1) GO TO 110

C-------- AXIS NOT TO BE SWITCHED --------
C
IF(SLFSIZ) GO TO 90
C
C-------- GRAPH NOT DISTORTED --------------
VSCALE = VSIZE/DV
HSCALE = VSCLAE
GO TO 100
C
C-------- GRAPH IS DISTORTED ----------------
C
90 HSCALE = HSIZE/DH
VSCALE = VSIZE/DV
C----- TRANSLATION ----- 
100 HTRSL = -HMIN*HSCALE
VTRSL = -VMIN*VSCALE
GO TO 200
C
C-------- AXIS IS SWITCHED -----------------
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
110 JX = 2
JY = 1
IF(SLFSIZ) GO TO 120

C------ GRAPH NOT DISTORTED -----------------------------------------------
C
HSIZE = HSIZE/DV
VSCALE = HSCALE
GO TO 130

C------ GRAPH IS DISTORTED -----------------------------------------------
C
120 HSCALE = HSIZE/DV
VSCALE = VSIZE/DH

C------ TRANSLATION ------
130 HTRSL = -VMIN*HSCALE
VTRSL = -HMIN*VSCALE

C

200 CONTINUE

C------ FORMAT GROUP ------

C
1000 FORMAT(/1X,'FULL SCREEN SIZE IS HUNIT = 120.0 VUNIT = 90.0/.
1 1X,'CURRENT SCREENSIZE IS HSIZE = ',G8.2/,
2 3X,'VSIZE = ',G8.2/.
3 1X,'DO YOU WANT TO CHOOSE YOUR OWN SCREEN SIZE ?
4 <1> = NO/)
1004 FORMAT(/1X,'DO YOU WANT TO USE SAME SCREEN SIZE AS LAST ?'
1 ' TIME ? <1> = YES '/)
1005 FORMAT(/1X,'ENTER HORIZONTAL AND VERTICAL SIZES '/)
1006 FORMAT(/1X,'CURRENT SCREENSIZE ARE'/,
1 5X,'HSIZE = ',G8.2,'VSIZE = ',G8.2)
1007 FORMAT(/1X,'HORIZONTAL DISTANCE IS SMALLER THAN VERTICAL',
1 ' DISTANCE, SINCE THE SCREEN IS LONGER IN THE'/,
2 'HORIZONTAL AXIS, YOU CAN SWITCH THE AXIS '/,
2 'FOR BEST VIEWING '/,
3
  * ENTER <1> IF YOU DO NOT WANT SWITCH AXIS */
1008 FORMAT(/IX, * ENTER <0> IF YOU DO NOT WANT TO */,
  * DISTORT THE AXIS */)
  RETURN
END
SUBROUTINE MAXMIN (X, N, XMAX, XMIN, DX)

C

C PURPOSE: FIND THE MAXIMUM AND MINIMUM OF A VECTOR OF LENGTH N
C
INPUT: VECTOR X(N)
C
OUTPUT: MIN, MAX AND DISTANCE BETWEEN MIN AND MAX.
C
NO SUBROUTINE CALLED.
C
DIMENSION X(N)

XMAX = X(1)
XMIN = X(1)
DO 20 I = 2, N
  BEGIN
    IF(X(I) .GT. XMAX) GO TO 10
    IF(X(I) .LT. XMIN) XMIN = X(I)
  10    XMAX = X(I)
  END;
  20 CONTINUE

DX = XMAX - XMIN
RETURN
END
SUBROUTINE MESH (NUMEL, NEN, IX, X, JX, JY, NEN1, NDM, DASH)

C
C---------------------------------------------------------------------
C ROUTINE DRIVEN BY MESHPLT TO PLOT MESH -----------------------------
C---------------------------------------------------------------------
C
C NUMEL = NUMBER OF TOTAL ELEMENTS
C NEN = MAXIMUM NUMBER OF NODES OF AN ELEMENT
C IX = CONNECTIVITY DATA FOR AN ELEMENT
C X = COORDINATES OF THE NODES
C JX, JY = 1, 2 IF X AXIS IS HORIZONTAL AND Y AXIS VERTICAL.
C = 2, 1 IF THE OPPOSITE. TO BE CHANGED BY ROUTINE JSCTR
C NEN1 = NEN + 1
C NDM = MAXIMUM NUMBER OF DIMENSIONS
C NBNODE = SEQUENCE NO OF NODES IN AN 8-NODE ELEMENT.
C
C

<table>
<thead>
<tr>
<th>4</th>
<th>7</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

FIRST = FLAG FOR GOING INTO ANY ELEMENT THE FIRST TIME.
DASH = FLAG TO DRAW DASHED PATTERNS

---------------------------------------------------------------------

THE MESH IS DRAWN ELEMENT BY ELEMENT GOING IN THE COUNTER-
CLOCKWISE DIRECTION AS DEFINED BY THE FINITE ELEMENT PROGRAM.

---------------------------------------------------------------------

SUBROUTINES CALLED :
MOVE, DRAW, DASHPT (IGL SYSTEM ROUTINES)
DIMENSION IX(NEN1,1), X(NDM,1), N8NODE(8)
LOGICAL FIRST,DASH
DATA N8NODE /1,5,2,6,3,7,4,8/

C--------- LOOP OVER EACH ELEMENT -------------------------------
C
IF(DASH) CALL DASHPT(3)
C BEGIN
DO 40 I = 1,NUMEL
FIRST = .TRUE.
DO 30 J = 1,NEN
JJ = J
IF(NEN.GT.4) JJ = N8NODE(J)
C BEGIN
K = IX(JJ,I)
C
C-------------POSSIBLE NONEXISTENT NODE OF 'VARIABLE NODE ELEMENT'.
C IF(K.EQ.0) GO TO 30
X1 = X(JX,K)
X2 = X(JY,K)
C
C----------------MOVES PEN TO NEW POSITION FOR 1ST EXISTING NODE.
C IF(FIRST) GO TO 20
CALL DRAW (X1,X2)
GO TO 30
20 CALL MOVE (X1,X2)
FX1 = X1
FX2 = X2
FIRST = .FALSE.
C END;
30 CONTINUE
CALL DRAW (FX1,FX2)
40 CONTINUE
C END;
RETURN
END
SUBROUTINE MESHPLT(NUMNP,NUMEL,NEN,IX,X,SC,
1 DM,DFMESH,NDM,TERM)

PURPOSE: PLOT TWO DIMENSIONAL FINITE ELEMENT MESH

THIS PROGRAM IS A DRIVER FOR PLOTTING TWO DIMENSIONAL FINITE
ELEMENT MESH. DEFORMED AND UNDEFORMED MESH CAN BE PLOTTED
TOGETHER OR SEPARATELY. DEFORMED MESH CAN BE EXAGGERATED
TO EMHASIZE DEFORMATION. IT IS ESPECIALLY USEFUL FOR
VISUALIZING SMALL DEFORMATIONS.

IT FIRST QUERIES FOR OPTIONS, THEN SCALES THE MESH BY CALLING
ROUTINE JSCTR, IT THEN DRIVES ROUTINE MESH WHICH IS THE SLAVE
(DRIVEN ONE) IN DOING THE DIRTY WORK.

NUMNP = TOTAL NO. OF NODAL POINTS.
NUMEL = TOTAL NO. OF ELEMENTS.
IX = CONNECTIVITY DATA FOR EACH ELEMENT(NEN,NUMEL)
NEN = MAXIMUM NO. OF NODES PER EACH ELEMENT.
X = COORDINATES OF NODAL POINTS (NDM,NUMNP)
SC = SCRATCH ARRAY OF DIMENSION NUMNP.
DM = SAME AS X EXCEPT THE COORDINATES ARE NOW
THOSE OF THE DEFORMED MESH.
JX,JY = (1) = 1 HORIZONTAL AXIS
(2) = 2 VERTICAL AXIS

DFMESH = FLAG FOR PLOTTING DEFORMED MESH

TERM = FLAG FOR FETCHING TERMINAL.
DASH = FOP DRAWING DASHED PATTERN.
SUBROUTINES CALLED:

MAXMIN, SCRATCH, JSCTR, TERMIN, MESH (USER SUPPLIED)
VIEWT, GRSTART, BAUDRT, SCALE, TRANSL, NEWS, NEWPA6,
GRASTOP. (IGL SYSTEM ROUTINES)


DIMENSION SC(1), IX(NEN1,1), X(NDM,1), DM(NDM,1)
DATA GUNIT/130.0/
LOGICAL DFMESH, BOTH, TERM, DASH

JX = 1
JY = 2

IF(.NOT.DFMESH) GO TO 210

DEFORMED MESH NEEDS TO BE PLOTTED

DASH = .FALSE.
BOTH = .FALSE.

WRITE(6,103)
READ(5,*,ERR=200) IDUM
IF(IDUM.EQ.1) BOTH = .TRUE.

SCALING FOR DEFORMED MESH

CALL SCRATCH(1,NUMNP,DM,SC,NDM)
CALL MAXMIN(SC,YUMNP,YMAXD,YMIND,DXD)
CALL SCRATCH(2,YUMNP,DM,SC,NDM)
CALL MAXMIN(SC,NMNP,YMAXD,YMIND,DYD)

IF(DXD.EQ.0.0 .OR. DYD.EQ.0.0) GO TO 50
CALL JSCTR(DXD,DYD,XMIND,YMIND,HTRSLD,VTRSLD)
CALL JSCTR(DXD,DYD,XMIND,YMIND,HTRSLD,VTRSLD)

C IF(.NOT.BOTH) GO TO 220
C
C------ SCALE UNDEFORMED MESH ------------------------------------------
C
210 CALL SCRATCH (1,NUMNP, X,SC,NDM )
CALL MAXMIN (SC,NUMNP,XMAX,XMIN,DX )
CALL SCRATCH (2,NUMNP, X,SC,NDM )
CALL MAXMIN (SC,NUMNP,YMAX,YMIN,DX )
C
IF(DX .EQ. 0.0 .OR. DY .EQ. 0.0) GO TO 50
C
CALL JSCTR (DX,DY,XMIN,YMIN,HTRSL,VTRSL,1
HSCALD,VSCALD,JX,JY)
IF(.NOT.DFMESH) GO TO 230
C
C------ DEFORMED AND UNDEFORMED MESH PLOTTED TOGETHER -----------------
C
C IF(HSCALE .LT. HSCALD) GO TO 230
C
C------ CHOOSE SMALLER ONE OF DEFORMED AND UNDEFORMED SCALING -------
C
220 HSCALE = HSCALD
VSCALE = VSCALD
HTRSL = HTRSLD
VTRSL = VTRSLD
C
230 CONTINUE
C
CALL TERMIN(ITERM,IBAUD,IOPT,TERM)
CALL GRSTRT(ITERM,IOPT)
CALL BAUDRT(IBAUD)
CALL VIEWT
CALL SCALE(HSCALE,VSCALE)
CALL TRANSL(HTRSL,VTRSL)
CALL NEWPAG
C
C----- PLOT MESH --------------------------------------------
C IF(.NOT. DFMESH .OR. BOTH) CALL MESH (NUMEL,NEN,IX,X,JX,JY,
1 NEN1,NDM,DASH)
IF(BOTH) DASH = .TRUE.
IF(DFMESH) CALL MESH (NUMEL, NEN, IX, DM, JX, JY, NEN1, NDM, DASH)
C
C----- TERMINATE COMMUNICATION FROM IGL ENVIRONMENT ------------
C
CALL GRSTOP
RETURN
C
C
50 WRITE(6,100)
103 FORMAT(IX,"ENTER <1> IF BOTH DEFORMED AND DEFORMED*.
1 /", "ARE TO BE PLOTTED ")
100 FORMAT(5X,"*** ERROR EXIT *** DISTANCE IS ZERO * NO GRAPH*/)
C 101 FORMAT(5X,"ENTER TERMINAL, BAUDRATE, TERMINAL OPT.")
C 1 "4051, ADM IS 4010, OPTIONS FOR BOTH ARE 1 */
C
RETURN
END
C
LOGICAL FUNCTION PCOMP(A,B)
PCOMP = .FALSE.
C IT MAY BE NECESSARY TO REPLACE THE FOLLOWING ALPHANUMERIC
C COMPARISION STATEMENT IF COMPUTER PRODUCES AN OVERFLOW.
C IF(A.EQ.B) PCOMP = .TRUE.
RETURN
END
SUBROUTINE PGO ( IX, X, TSIG, TEPS, EQSIG, B, FRNG, SC, XG, NE, NDGA, 
1 NENL)

C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C PURPOSE : ROUTINE COLLECTS DATA TO BE SENT TO CONTOUR
C AND MESH PLOTTING PROGRAMS.
C
C POSSIBLE PLOT PARAMETERS :
C
C : PRINCIPLE STRESSES, STRAINS, FRINGE ORDERS, MESH
C
C SUBROUTINES OR FUNCTIONS CALLED :
C
C PCOMP, RESTOR, SETMEM, PZERO, SC2, FRINGE,
C CONTOUR, ADD, MESHPLT, IDTANG.
C
C CALLED BY : PWORK
C
C -------------------------------------------------------------
C
C FLAGS FOR PREVENTING MEMORY BE SET TWICE :
C
C TRFL = FOR TRIANGULATION.
C DEFL = FOR CHANGING THE TOTAL NUMBER OF ELEMENTS TO BE
C PROCESSED.
C THRFE = FOR THREE DIMENSIONAL PLOTTING.
C DFMFRY = FOR PLOTTING DEFORMED MESH.
C
C OTHER FLAGS :
C
C TERM = FOR FETCHING TERMINAL.
C FMFL = FOR FIXED FORMAT INPUT.
C
C NOTE : NUMELD = WILL NEVER BE CHANGED. ONLY OPERATIONAL
C VARIABLE NUMEL CAN BE CHANGED ONLY
C ONCF. HENCN, IN CREATING THE FINITE
ELEMENT MESH, THOSE ELEMENTS NOT TO BE PLOTTED HAS TO COME AT THE END OF THE ELEMENT LIST. E.G. TOTAL NO. IS 100 ELEMENTS, NUMEL = 5, THEN ONLY 1 TO 95 ELEMENTS ARE PLOTTED.

COMMON /CDATA/HEAO(20),NUMNP,NUMELD,NEN,NDM,NGAUP,NDM2,NTOT,INDN
COMMON H(1)
LOGICAL PCOMP,TRFL,GAING,DEFL,THREE,FMFL
1 DFMESH,DFMERY,TERM
INTEGER CNTR
COMMON/PODATA/FMFL
DIMENSION WD(4), CNWD(8), IX(1), X(1), TSIG(NTOT,1),
1 TEPS(NTOT,1), EQSIG(NGAUP,1), B(NDM,1), FRNG(NGAUP,1),
2 SC(1), XG(NDGAP,1), CNTR(4), RNMIN(3), RNMAX(3)
DATA STRESS/4HSTRE/, WD/4HC0NT,4HHEMT,4HST0P,4HAGAI/,
1 CNWD/4HTRIA,4HSIGM,4HEP5I,4HFIRN,4HFL0T,4HEND,4HDISP, 4HEQSG/,
2 NWD/4/, NCTWD/8/, NTAPE1/30/, I3DST/1/, NTAPE2/3I/

C INITIALIZATION

NZRAN = 1
NDFMSH = 1
TRFL = "FALSE"
DEFL = "FALSE"
THREE = "FALSE"
DFMERY = "FALSE"
TERM = "FALSE"

C CHECK IF THREE DIMENSIONAL

C------- CHECK IF THREE DIMENSIONAL -------
IF(NDM.NE.3.OR.THREE) GO TO 10

C ALLOCATION OF MEMORY FOR STORING POINTERS IN 3-D CLIPPING.

I3DST = NE
NE = I3DST + NGAUP*NUMELD
CALL SETMEM(NE)
THREE = .TRUE.
10 CONTINUE

C-----------------------------------------------
C---------------------------------------------
C RESTORE GEOMETRY AND ELEMENT CONNECTIVITY DATA-
C---------------------------------------------
C-----------------------------------------------
C IF(FMFL) GO TO 90
C----- INPUT IS IN FREE FORMAT ----- C
READ(NTAPE1) (IX(K),K=1,NEN1*NUMELD), (X(K),K = 1,NDM*NUMNP)
GO TO 91
C----- INPUT IS IN FIXED FORMAT ----- C
90 READ(NTAPE2,4009) (IX(K),K=1,NEN1*NUMELD)
READ(NTAPE2,4010) (X(K),K = 1,NDM*NUMNP)
91 CONTINUE
C-----------------------------------------------
C----- CHECK IF STRESS/STRAIN OR DISPLACEMENT INPUT IS FOUND ----- C
C
100 IF(.NOT.FMFL)READ(NTAPE1,END=3) PARA, TIME
   IF(FMFL) READ(NTAPE2,4011,END=3) PARA, TIME
   IF(PCOMP(PARA,STRESS)) GO TO 102
C-----------------------------------------------
C-----------------------------------------------
C-----------------------------------------------
IF (FMFL) GO TO 92

C FREE FORMAT INPUT

DO 101 II = 1, NUMNP
    READ(NTAPE1) (B(J,II), J = 1, NDM)
101 CONTINUE
GO TO 104

C FIXED FORMAT INPUT

92 DO 93 II = 1, NUMNP
    READ(NTAPE2,4010) (B(J,II), J=1, NDM)
93 CONTINUE
GO TO 104

C STRESS/STRAIN FILE FOUND

102 DO 103 II = 1, NUMELD
    CALL RESTOR(FG(1,II), TSIG(1,II), TEPS(1,II), EOSIG(1,II),
    NGAUP, NDM, NDM2)
103 CONTINUE

C CHECK IF WANT TO CONTINUE READ IN FILE

104 WRITE(6,1100) PARA, TIME
    READ(5,*) I AGAIN
    IF (I AGAIN) GO TO 100

C DATA INPUT COMPLETE, NOW ASK FOR OPTIONS IN PROCESSING

C OPTION TO ALLOW THE FIRST NUMEL ELEMENTS TO BE PLOTTED ONLY

IF (DEFL) GO TO 105
    NUMEL = NUMFLD
    WRITE(6,1109)
READ(5,*) DECRE
IF(DECRE.EQ.0.0) GO TO 105
WRITE(6,1110) NUMELD
READ(5,*) NUMEL
DEFL = .TRUE.

C---------------------------------------------------------------------
C----->> SECOND LEVEL OF OPERATION -----------------------------------
C------- MACROS ARE = CONT, MESH, STOP, AGAI -------------------------
C---------------------------------------------------------------------

C 105 WRITE(6,1102)
   READ(5,1107,ERR=107) CT
   DO 106 L = 1,NWD
       IF(PCOMPCT,WDL(L))) GO TO 108
106 CONTINUE
C------ ERROR PROCESSING ----- 
 107 WRITE(6,1101)
      GO TO 105

C ...........GO TO APPROPRIATE PROCESSOR 
C
108 GO TO (1,2,3,4), L
C
C-------- CONT ------------------------------------------------------
C-------- CONT contour plot desired ---------------------------------
C
1 WRITE(6,1103)
C-------- commands for contour plot ------
C-------- tria, (sigm.n), (psi,n), (frin,n1,n2), plot, end, ------
C-------- disp, eqsg -----------------------------------------------
C
109 WRITE(6,1104)
   READ(5,1108,ERR=111) (CNTR(J),J=1,4)
   DO 110 LL = 1,NCTWD
       IF(PCOMP(CNTR(1),CNWD(LL))) GO TO 112
110 CONTINUE
C
C...........INPUT ERROR.
C
111 WRITE(6,1101)
       GO TO 109
C
C...........GO TO APPROPRIATE PROCESSORS INSIDE 'CONT' MACRO.
C
112 GO TO (51,52,53,54,55,56,57,58), LL
C
MACRO = TRIA
C-------------------TRIANGULIZE A 2-D DOMAIN-------------------------
C
51 IF(TRFL) GO TO 1001
C
ALLOCATE MEMORY -------
C
NDP = NGAUP*N NUMEL
IF(NDP+LT,NUMNP) NDP = NUMNP
   NPT = NE
   NPL = NPT + 6*NDP - 15
   IWL = NPL + 6*NDP
   IWP = IWL + 18*NDP
   IWK = IWP + NDP
   NE = IWK + NDP
   CALL SETMEM(NE)
   TRFL = .TRUE.
C
EXTRACT 1ST AND 2ND COORDINATES-----------------------------
C
1001 CALL PZERO(SC,NGAUP*NUMEL,2)
       CALL PZERO(M(NPT),NE-NPT)
       NDATA = NGAUP*NUMEL
       IF(NDM.NE.3) GO TO 999
CLIPPING PROGRAM FOR PLOTTING 3-D VALUES.

CALL RZERO(M(I3DST),NGAUP*NUMEL)
WRITE(6,1120)
READ(5,0) IXDOF,RNMIN(IXDOF),RNMAX(IXDOF)

CYCLIC PERMUTATION - IF ONE DIRECTION IS FIXED, THEN
THE TWO AND THREE DIRECTIONS ARE EXTRACTED, ETC.

IXDF1 = IXDOF+1
IXDF2 = IXDOF+2
IF (IXDF1.GT.3) IXDF1 = IXDF1 - 3
IF (IXDF2.GT.3) IXDF2 = IXDF2 - 3
N3DTOT = 0

DO 9010 II = 1,NUMEL
   CALL SC3(II,IXDOF,IXDF1,NGAUP,XG(1,II),NDM,SC(1),M(I3DST),
     1 RNMIN(IXDOF),RNMAX(IXDOF),N3DTOT)
9010 CONTINUE

N3DTOT = 0
DO 9020 II = 1,NUMEL
   CALL SC3(II,IXDOF,IXDF2,NGAUP,XG(1,II),NDM,SC(1+NGAUP*NUMEL),
     1 M(I3DST),RNMIN(IXDOF),RNMAX(IXDOF),N3DTOT)
9020 CONTINUE

IF (N3DTOT.GT.4) GO TO 998
WRITE(6,1130)
GO TO 109
998 NDATA = N3DTOT
GO TO 1004

TWO DIMENSIONAL PLOT

DO 1002 II = 1,NUMEL
   CALL SC2(II,1,NGAUP,XG(1,II),NDM,SC(1),M(I3DST),
     1 ILOC)
1002 CONTINUE
DO 1003 II = 1, NUMEL
   CALL SC2(II*2+NGAUP, XG(1+II), NDM, SC(1+NGAUP*NUMEL),
   1 M(I3DST), ILOC)
1003 CONTINUE

C
1004 CONTINUE
C
C TRIANGULIZATION -------
C
CALL IDTANG(NDATA, SC(1), SC(1+NGAUP*NUMEL), NLT, M(NPT),
1 NL*M(NPL), M(IWL), M(IWP), M(IWK))
WRITE(6,1105) NT
C
C------- ALLOCATING POINTERS ------
C
NCOR1 = 1
NCOR2 = 1 + NGAUP*NUMEL
NZRAN = 1 + NGAUP * NUMEL * 2
GO TO 109
C
C MACRO = EPSIG*N
C---------------- STRESS CONTOUR PLOT ----------------
C
52 NN = CNTR(2)
CALL PZERO(SC(NZRAN), NGAUP*NUMEL)
ILOC = 0
DO 2001 II = 1, NUMEL
   CALL SC2(II*NN, NGAUP, TSIG(1+II), NDM2, SC(NZRAN), M(I3DST), ILOC)
2001 CONTINUE
GO TO 109
C
C---------------- MACRO = EPSI*N ----------------
C---------------- STRAIN CONTOUR PLOT ----------------
C
C
53 NN = CNTR(2)
    CALL PZERO(SC(NZRAN), NGAUP*NUMEL)
    ILOC = 0
    DO 3001 II = 1,NUMEL
        CALL SC2(II,NN,NGAUP,TEPS(1,II),NDM2,SC(NZRAN),M(I3DST),ILOC)
    3001 CONTINUE
    GO TO 109

C---------------------------------------------------------------------
C------ MACRO = FRIN,N1,N2 -----
C------ FRINGE CONTOUR PLOT ------------------------------------------
C---------------------------------------------------------------------
C
54 FRCN = 1.0
    THICK = 1.0
    N1 = CNTR(2)
    N2 = CNTR(3)

C------ FRINGE CONSTANT AND SLICE THICKNESS ARE OPTIONAL INPUT ------
C
3999 WRITE(6,1106)
    READ(5,*) IFRCN
    IF(FRCN*NE.0) READ(5,*) FRCN,THICK
        IF(THICK*EQ.0.0) GO TO 3999
    CONS = FRCN/THICK
C
C------ FRINGE CAN BE COMPUTED FROM STRESS OR STRAIN DIFFERENCE ------
C
WRITE(6,4020)
    READ(5,*) IFRST
    IF(IFRST*NE.0) GO TO 4001
C
C------ FRINGE CALCULATED BY PRINCIPAL STRAIN DIFFERENCE -------
C
    DO 4000 I1 = 1,NJHME
        CALL FRINGE(CONS,TEPS(1,I1),FRNG(1,I1),N1,N2,NDM2,NGAUP)
    4000 CONTINUE
GO TO 4003

C
C------FRINGE BASED ON PRINCIPAL STRESS DIFFERENCE------
C
4001 DO 4002 II = 1,NUMFL
    CALL FRINGE(C3NS,TSIG(1,II),FRNG(1,II),N1,N2,NDM2,NGAUP)
  4002 CONTINUE
C
C------ MOVE FRINGE VALUES TO AREAS FOR PLOTTING -------
C
4003 CALL PZERO(SC(NZ3AN),NGAUP*NUMEL)
   DO 4005 II = 1,NUMFL
       NFR = NGAUP*(II-1)
       DO 4004 JJ = 1,NGAUP
           SC(NZRAN+JJ+NFR-1) = FRNG(JJ,II)
  4004 CONTINUE
  4005 CONTINUE
GO TO 109
C
C------ MACRO = PLOT ------
C------ PLOT ACTION IS HERE -------------------------------------
C
  55 CALL CONTOUR(NT,W(NPT),SC(NCOORD1),SC(NCOORD2),SC(NZ3AN),NDATA, 1    NL,W(NPL),TERM)
      READ(5,4010,ERR=109) ANYT
      GO TO 109
C
  56 GO TO 105
C
C------ MACRO = DISP ------
C------ PRINT SPECIFIC RANGE OF DISPLACEMENT-------------------
C
  57 WRITE(6,1140)
      READ(5,4009,ERR=111) NDIS1,NDIS2
      NDINC = NDIS2 - NDIS1
      IF(NDINC.LE.0) GO TO 5702
DO 5701 II=NDIS1:NDIS2
   WRITE(6,1150) II,(B(JJ,II)*JJ=1:NDM)
5701  CONTINUE
   GO TO 109
5702 WRITE(6,1160)
   GO TO 109
C
C------ MACRO = EQS6 ------
C------ EQUIVALENT STRESS CONTOUR -----------------------------------
C
58 CALL PZERO(SC(NZAN),NGAUP*NUMEL)
   ILOC = 0
   DO 5801 II = 1:NUMEL
      CALL SC4(II,NGAUP,EQSIG(1,II)*NDM2,SC(NZAN),
               M(I3ST),ILOC)
5801  CONTINUE
   GO TO 109
C
C----------------------------------------------
C
C 2 CONTINUE
   DFMESH = .FALSE.
200 WRITE(6,4021)
   READ(5,**,ERR=200) IDUM
   IF(IDUM.NE.0) DFMESH = .TRUE.
C
   IF(.NOT.DFMESH) GO TO 230
C
C----- DEFORMED MESH PLOT NEEDED -------------------
C
   IF(DFMERY) GO TO 201
C
C----- ALLOCATING MEMORY FOR DEFORMED MESH ONCE FOR ALL ---------
C
NDFMSH = NE
NE = NDFMSH + NDM*NUMNP
CALL SETMEM(NE)
DFMERY = .TRUE.
C
C QUERY FOR AMPLIFYING THE DISPLACEMENT ----------------------------
C
201 DISCA = 1.0
203 WRITE(6,4022)
   READ(5,*II,ERR=203) IDUM
   IF(IDUM.EQ.1) READ(5,*II,ERR=203) DISCA
C
C----- CONVERT DISPLACEMENT MATRIX TO VECTOR ------------------------
C
205 DO 220 II = 1,NUMNP
   DO 210 JJ = 1,NDM
      SC(NZRAN+(II-1)*NDM+JJ-I) = B(JJ,II)*DISCA
   210 CONTINUE
220 CONTINUE
C
C----- SEND DATA TO MESH DRIVER -------
C
   CALL PZERO(M(NDFMSH),NDM*NUMNP)
   CALL ADD(X,SC(NZRAN),M(NDFMSH),NDM*NUMNP)
230 CALL MESHPLT(NUMNP,NUMEL,NEN,NEN1,IX,SC(NZRAN),
   1 M(NDFMSH),DFMESH,NDM,TERM)
C
   READ(5,4010,ERR=109) ANYT
   GO TO 105
C
C----- EXIT FROM PLOTTING ------------------------------------------
C
3 RETURN
C
C----- CONTINUE READING MORE DATA ---------------------------------
C
4 GO TO 100
C FORMAT GROUP

1100 FORMAT (/5X,A4,': FILE OF ".IN", ". TIM: STEP RESTORED ".
1 5X,': TO USE "/5X, ". DO YOU WANT TO CONTINUE READING ?")
1101 FORMAT (/5X,'**INPUT ERROR **, NO SUCH COMMAND "/")
1102 FORMAT (/5X,'AVAILABLE COMMANDS ARE ":/5X,"CONT.MESH,STOP,AGAIN"
1 5X,">>>")
1103 FORMAT (/5X,'CONTOUR PLOT REQUESTED'
1104 FORMAT (/5X,'AVAILABLE COMMANDS ARE ":/5X,"TRIANGLES CREATION'SIGM,N)"
1 5X,">>")
1105 FORMAT (/5X,110,': TRIANGLES CREATED ":/5X,">>>")
1107 FORMAT (A4)
1108 FORMAT (A4,3I3)
1106 FORMAT (/5X,'DO YOU WANT TO INPUT THE OPTICAL COEFFICIENT".
1 5X,'AND THICKNESS "/5X,"DEFAULT ARE 1 FOR BOTH")
1109 FORMAT (/5X,'DO YOU WANT TO DECREASE THE NO. OF ELEMENTS TO".
1 5X,'BE PLOTTED ? 0 = NO. "")
1110 FORMAT (/5X,'ORIGINAL NO. OF ELEMENTS IS ",I5," ENTER ".
1 5X,'NO. OF ELEMENTS (LESS THAN ORIGINAL) ?? "")
1120 FORMAT (/5X,'3-D CLIPPING',4X,'INPUT =',/5X,'THE D.O.F. TO BE ".
1 5X,'FIXED), XMIN, XMAX WHERE X IS THE D.O.F. IN FREE".
2 5X,'FORMAT")
1130 FORMAT (/5X,'ACTUAL NUMBER OF DATA POINTS IS "/I3," NOT"
1 5X,'ENOUGH , CHOOSE OTHER RANGE "")
1140 FORMAT (/5X,'DISPLACEMENT OUTPUT DESIRED".
1 8X,'ENTER STARTING AND ENDING NODE IN 2I4 FORMAT")
1150 FORMAT (I4,14,3X,6G10.3)
1160 FORMAT (5X,'** INPUT ERROR ** NODAL DIFFERENCE ".
1 5X,'MUST BE GREATER OR EQUAL TO 1")
4008 FORMAT (20A4)
4009 FORMAT (20A4)
4010 FORMAT (8G10.3)
4011 FORMAT (A4,6G10.3)
4020 FORMAT (5X,'STRESS COMPUTED FROM STRESS OR STRAIN ?")
1 5X,'ENTER 0 FOR STRAIN=DEFAULT")
4021 FORMAT(1X,'DO YOU WANT TO PLOT DEFORMED MESH ? ',
1       /1X, ' <0> = NO ')

4022 FORMAT(1X,'ENTER <1> IF YOU WANT TO AMPLIFY DISPLACEMENT ',
2       /', THEN TYPE THE AMPLIFYING FACTOR ')
       END
PROGRAM POSTPR
C
C.............A POSTPROCESSOR FOR FINITE ELEMENT ANALYSIS
C
COMMON M(20000)
COMMON/PSIZE/ MAX
MAX = 20000
CALL PWORK
STOP
END
SUBROUTINE PWORK

C
--------------------
C
C THIS PROGRAM ALLOCATES INITIAL MEMORY BY READING
C NECESSARY CONTROL INFORMATION.
C
--------------------
C
C THE MAIN PROGRAM OF THIS PACKAGE <POSTPR> IS A DUMMY ROUTINE
C TO INITIALIZE ALL THE DYNAMIC MEMORY NEEDED.
C
C ALL SOURCE ROUTINES (EXCEPT <POSTPR>) ARE CONTAINED IN THE
C LIBRARY <PROC.TLB> AND ALL OBJECT ROUTINES (EXCEPT <POSTPR>)
C ARE CONTAINED IN LIBRARY <AFTER.OLB>. ALL GRAPHIC PRIMITIVES
C ARE THEN LINKED FROM THE IGL LIBRARIES.
C
--------------------
C
C NTAPE1 = 30, FREE FORMAT INPUT FILE
C NTAPE2 = 31, FIXED FORMAT INPUT FILE
C
C POSTFL = FLAG TO PREVENT MEMORY TO BE SET TWICE.
C FMFL = FLAG FOR FIXED FORMAT INPUT.
C
C POST> = MACRO FOR INITIALIZING THE SYSTEM.
C GO > = MACRO FOR PROCESSING.
C STOP> = MACRO FOR EXITING FROM THE SYSTEM.
C
C THE SEQUENCE OF COMMANDS TO BE ISSUED IS
C POST>, GO>, STOP>.
C
--------------------
C
C FOR NEW OR MODIFIED ROUTINES, COMPILE LIKE THIS:
C
C FORT/NOOPT/DE8  PWORK.TXT
WHICH SAYS NO COMPILER OPTIMIZING AND DEBUGGER IS INVOKED.
AND TO LINK LIKE THIS:

`@POST`

WHERE POST IS DEFINED IN THE LOGIN.COM FILE AS

```$ POST:==LIN POSTPR,AFTER/LIB,VAXBOI:GRAPHICS,IGL,IGL/LIB/DEB```

AND THEN USE THE LIBRAIEN TO REPLACE THE SOURCE AND OBJ
FILE FOR THE ROUTINE ADDED OR CHANGED (SEE DCL MANUAL).
FINALLY THE TWO LIBRARIES PROC.TLB AND AFTER.OLB NEEDS TO
BE COMPRESSED BY USING COMMAND <COMP> (DEFINED IN LOGIN.COM
FILE) TO SAVE SPACE ON DISK. ALSO USE <SS> COMMAND
TO PROTECT THE LIBRARIES FROM INADVERTENT DELETION.

NO PAGE MARKS CAN BE EMBEDDED IN ANY OF THE DATA FILE.
IT IS RECOMMENDED THAT ALL DATA FILES BEFORE USING BE
EDITED USING TEXT EDITOR BY (1) FIRST ENTERING THE
EDITOR (2) THEN DO NOTHING (3) EXIT USING THE COMMAND
<ET> TO GET RID OF ALL LINE NUMBERS AND PAGE MARKS.
SEE VAX TEXT EDITORS IF YOU NEED ANY HELP.

BEFORE RUNNING, SHOULD ASSIGN LIKE:

```$ ASSIG CASE###.DAT FOR030```
```$ ASSIG CASE###.DAT FOR031```

FOR FREE AND FIXED FORMAT INPUT RESPECTIVELY.
FREE FORMAT INPUT IS USEFUL IF SAME COMPUTER IS USED
FOR BOTH FINITE ELEMENT ANALYSIS AND POSTPROCESSING.
FIXED FORMATINPUT IS MANDATORY WHEN DIFFERENT COMPUTERS
C ARE USED FOR EXAMPLE RUNNING THE FINITE ELEMENT PROGRAM ON
C IBM OPERATING SYSTEM (AS/6) AND PROCESSING DATA ON VAX
C BASED SYSTEM --- THE INTERNAL REPRESENTATION OF NUMBERS
C ARE NOT THE SAME ........

C---------------------------------------------------------------
COMMON H(I)
LOGICAL PCOMP, POSTFL, FMFL
COMMON/PODATA/FMFL
DIMENSION WD(3), HEAD(20)
DATA NWD/3/, WD/4HPOST, 4HGD, 4HSTOP/, NTAPE1/30/, NTAPE2/31/
COMMON/CDATA/ HEAD, NUMP, NUMEL, NEN, NDM, NGAUP, NDM2, NTOT, INON
C---------------------------------------------------------------
C------ INITIALIZING PARAMETERS -------------------------------
C---------------------------------------------------------------
REWIND NTAPE1
REWIND NTAPE2
POSTFL = .TRUE.
FMFL = .FALSE.
WRITE(6,999)

C------ QUERY FOR FIXED OR FREE FORMAT ? -----------------------
C
READ(5,4010,ERR=100) DUM
   IF(DUM.EQ.0.0) GO TO 150
100 FMFL = .TRUE.
150 CONTINUE
C
C------ WAITING FOR COMMANDS -----------------------------------
C
WRITE(6,1000)
200 READ(5,1004,ERR=400) CT
   DO 300 L = 1, NWD
      IF(PCOMP(CT,WD(L))) GO TO 500
300 CONTINUE
C
C------ ERROR INPUT IN MACRO COMMAND--
400 WRITE(*,1001)
   WRITE(*,1000)
   GO TO 200
C
C-------------------GO TO APPROPRIATE PROCESSORS---
C
500 GO TO (1,2,3) *L
C
C-----------------RESTORE CONTROL INFORMATION AND ALLOCATING MEMORY-
C
1 IF(.NOT.POSTFL) GO TO 700
   IF(FMFL) GO TO 510
C
C-------- FREE FORMAT INPUT FROM TAPE NTAPE1 ------------------------
C
   READ(NTAPE1) (HEAD(K), K = 1,20), NUMNP, NUMEL, NEN, NDM
   READ(NTAPE1) NGAUP,NDM2, NTOT, INON
   GO TO 520
C
C-------- FIXED FORMAT INPUT FROM TAPE NTAPE2 ------------------------
C
510 READ(NTAPE2,4008) (HEAD(K),K=1,20)
   READ(NTAPE2,4009) NUMNP,NUMEL,NEN,NDM
   READ(NTAPE2,4009)NGAUP,NDM2,NTOT,INON
520 CONTINUE
C
C-------- ESTABLISH POINTERS FOR DYNAMIC MEMORY ALLOCATION ----------
C
   NEN1 = NEN + 1
   ISGEP = NTOT + NUMEL
   ICONN = 1
   ICOORD = ICONN + NEN1 + NUMEL
   ISG = ICOORD + NDM+NUMNP
IEP = ISG + ISGFP
IEQ = IEP + ISGEP
IDIS = IEO + INDN
IFR = IDIS + NDM * NUMNP
IGAUS = IFR + NGAUP * NUMEL
ISC = IGAUS + NUMEL * NGAUP * NDM
NE = ISC + NGAUP * NDM * NUMEL * 3

CALL SETMEM( NE )
CALL PZERO ( M(1), NE)
POSTFL = .FALSE.
GO TO 600

C------ GO> MACRO ISSUED -----------------------------------------------
C
C------ EXECUTE PROCESSOR
C
2 CALL PGD(M(ICONN), M(ICORD), M(ISG), M(IEP), M(IEO), M(IDIS),
1   M(IFR), M(ISC), M(IGAUS), NE, NDM, NGAUP, NENI)
GO TO 600
C
3 RETURN
C
C------ STOP> MACRO ISSUED ----------------------------------------------
C
600 WRITE(6,1002) CT
   GO TO 200
C
700 WRITE(6,1003)
   GO TO 200
C
1000 FORMAT(/5X,'AVAILABLE COMMAND : POST, GO, STOP' )
1001 FORMAT(/5X,'** INPUT ERROR ** NO SUCH COMMAND')
1002 FORMAT(/5X,'COMMAND ' A4, ' EXECUTED ' )
1003 FORMAT(/5X,'** INPUT ERROR ** "POST" MACRO CAN ONLY BE ISSUED'
1  ' ONCE. ' )
1004 FORMAT(A4)
999 FORMAT(/5X,'DATA FORMATTED ? RET = YES ' )
4008 FORMAT(20A4)
4009 FORMAT(20I4)
4010 FORMAT(8G10.3)
END
SUBROUTINE PZERO(V, NN)

C
C ZEOR REAL ARRAY.
C
DIMENSION V(NN)
DO 100 N = 1, NN
100 V(N) = 0.0
RETURN
END
SUBROUTINE RESTOR(XG, TSIG, TEPS, EQSIG, NGAUP, NDM, NDM2)
C FOR STRESS AND STRAIN:
C PLANESTRESS AND STRAIN: 1-1ST PRINCIPAL STRESS
C 2-2ND PRINCIPAL STRESS
C 3-OUT OF PLANE PRINCIPAL STRESS
C 4-ANGLE (IN PLANE)
C AXISYMMETRIC: 3 IS THETA DIRECTION. (OTHERS SAME).
DIMENSION XG(NDM,1), TSIG(NDM2,1), TEPS(NDM2,1), EQSIG(1)
LOGICAL FMFL
COMMON/PDATA/FMFL
DATA NTAPE1/30/, NTAPE2/31/
C IF(FMFL) GO TO 10
DO 1 I = 1, NGAUP
   READ(NTAPE1) (XG(J,I), J=1, NDM), (TSIG(J,I), J=1, NDM2),
   (TEPS(J,I), J=1, NDM2), EQSIG(I)
1 CONTINUE
RETURN
C 10 CONTINUE
DO 20 I = 1, NGAUP
   READ(NTAPE2,4010) (XG(J,I), J=1, NDM), (TSIG(J,I), J=1, NDM2),
   (TEPS(J,I), J=1, NDM2), EQSIG(I)
20 CONTINUE
RETURN
4010 FORMAT(8G10.3)
END
SUBROUTINE SC2(I, JK, NGAUP, XG, NDM, SC, NSCT, ILOC)

C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C PURPOSE : EXTRACT A VECTOR OUT OF THE ARRAY XG (SPECIAL ONE).
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
DIMENSION XG(NDM,1), SC(1), NSCT(NGAUP,1)
NN = NGAUP*(II-1)
DO 1 J = 1, NGAUP
C
1 IF(NDM.EQ.6) GO TO 2
C
SC(J+NN) = XG(JK•J)
GO TO 1
C
CHECK RANGE IN 3-D CLIPPING:
C
2 IF(NSCT(J,II).EQ.0) GO TO 1
  ILOC = ILOC + 1
  SC(ILOC) = XG(JK•J)
C
1 CONTINUE
RETURN
END
SUBROUTINE SC3(IXDDF,EXDDF,NGAUP,XG,NDM,SC,NSCT,RNMIN,RNMAX,
N3DTOT)

C PURPOSE : SET POINTER ARRAY NSCT TO ONE IF A POINT
C IS IN THE RANGE AND COUNT THE TOTAL NUMBER
C OF DATA POINTS IN THE RANGE FOR 3-D CLIPPING
C TO BE SENT TO A 2-D CONTOUR PLOT PROGRAM.
C
C NSCT = 1 ON ( IN THE RANGE )
C
C NSCT = 0 OFF ( NOT IN THE RANGE )
C
C DIMENSION XG(NDM,1),SC(1),NSCT(NGAUP,1)
C
DO 10 I = 1,NGAUP
  IF(XG(IXDDF,I).GT.RNMAX.OR.XG(IXDDF,I).LT.RNMIN) GO TO 10
C FOUND COORDINATE IN THE RANGE
C
  N3DTOT = N3DTOT + 1
  SC(N3DTOT) = XG(IXDDF,I)
  NSCT(I,1) = 1

10 CONTINUE
RETURN
END
SUBROUTINE SC4(II,NGAUP,XG,NDM,SC,NSCT,ILOC)

C
C PURPOSE: EXTRACT A VECTOR OUT OF THE ARRAY XG (SPECIAL ONE).
C
DIMENSION XG(1), SC(1), NSCT(NGAUP,1)

NN = NGAUP*(II-I)
DO 1 J = 1,NGAUP
C
C IF(NDM.EQ.6) GO TO 2
C
SC(J+NN) = XG(J)
GO TO 1
C
CHECK RANGE IN 3-D CLIPPING:
C
2 IF(NSCT(J,II).EQ.0) GO TO 1
ILOC = ILOC + 1
SC(ILOC) = XG(J)
C
1 CONTINUE
RETURN
END
SUBROUTINE SCRATCH(K,N,X,SC,NDM)

C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C PURPOSE : CONVERT A COLUMN WISE ARRAY TO A SINGLE VECTOR.
C
C K = THE ORDER OF DIMENSION
C N = NUMBER OF ELEMENTS IN THE VECTOR
C NDM = DIMENSION OF THE ARARY
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
DIMENSION SC(1), X(NDM,1)
DO 10 I = 1,N
   SC(I) = X(K,I)
10 CONTINUE
RETURN
END
SUBROUTINE SFTMEM(J)

C................MONITOR AVAILABLE MEMORY IN BLANK COMMON
C
COMMON M(1)
COMMON /PSIZE/ MAX
K = J
IF(K.LE.MAX) RETURN
   WRITE(6,1000) K,MAX
STOP
1000 FORMAT(5X,'**ERROR 01** INSUFFICIENT STORAGE IN BLANK COMMON**/
       1 17X,'REQUIRED =',I8,'/17X,'AVAILABLE =',I8)
END
SUBROUTINE TERMTM ( ITERM, IBAUD, IOPT, TERM )

C
C
C
C
C PROGRAM NAME = TERMINAL
C
C THIS PROGRAM FETCHES THE TERMINAL IDENTIFICATION
C CODE FOR LOCALLY SUPPORTED GRAPHICS HARDWARE.
C
C ITERM = TERMINAL TYPE
C IBAUD = TERMINAL BAUD RATE
C IOPT = OPTION
C
C TERM = LOGICAL FLAG INDICATING FETCHED TERMINAL
C STATUS
C
C PROGRAMMER = JONAH H-Y LEE
C DATE = OCT 31, 1982
C
C
C -----------------------------
C
C LOGICAL TERM
C -----------------------------
C TERM = .FALSE. PRESET IN ROUTINE PWORK
C
C IF(TERM) RETURN
C
C 50 WRITE(6,100)
C 100 FORMAT(/1X,'ENTER TERMINAL CODE,BAUDRATE,TERMINAL OPTION*/
C 1      5X,'TEXTRONICS = 4051,1200,1 */
C 2      5X,'ADM3A = 4010,1200,1 */)
C READ(5,*,ERR=50) ITERM, IBAUD, IOPT
C RETURN
C END
C. Listing of the Strain Separation Program
NAME : STRAIN

PURPOSE: CALCULATE FRINGE ORDER FROM STRAIN

THIS PROGRAM RECEIVES INPUT FROM TWO FILES
IUNT,IUNZ CONTAINING THE TANGENTIAL AND AXIAL
FRINGE ORDERS AND THEIR COORDINATES IN STANDARD
AGRAPH FORMAT, THAT IS

NX,NY,KXDAT,KYDAT
XMIN,........,XMAX
YMIN,........,YMAX
(XMIN,YMIN),(...,YMIN),...(XMAX,YMIN)
(XMIN,...),(...........(XMAX,...)

(XMIN,YMAX), (XMAX,YMAX)

WHERE KXDAT AND KYDAT ARE IGNORED BY THIS PROGRAM.

AND OUTPUT THREE FILES OUEI,OUEJ,OUEZ THE STRAINS
ALSO IN STANDARD AGRAPH FORMAT.

WHERE FRINGE CONSTANTS FEP,THICKNESSES TT,TZ
ARE ALSO NEEDED FROM ONLINE INPUT.

FINITE DEFORMATION IS OPTIONAL.

ALSO OUTPUT RADIAL DISPLACEMENT AS OPTION TO FILE
DESIGNATED.
C
C
NT = FRINGE IN THETA DIRECTION
NZ = FRINGE IN AXIAL DIRECTION
X  = RADIAL COORDINATE
Y  = AXIAL COORDINATE
NX = NUM OF DATA IN RADIAL DIRECTION
NY = NUM OF DATA IN AXIAL DIRECTION
FEP = STRAIN FRINGE VALUE
TT = THICKNESS OF SLICE IN THETA DIRECTION
TZ = THICKNESS OF SLICE IN AXIAL DIRECTION
ER = RADIAL STRAIN
ET = TANGENTIAL STRAIN
EZ = AXIAL STRAIN
UR = RADIAL DISPLACEMENT
DUDR = RADIAL DISPLACEMENT GRADIENT
FINITE = FLAG OF FINITE DEFORMATION
DISPLA = FLAG TO CALCULATE DISPLACEMENT
ETGTE = FLAG TO CHOOSE WHETHER ET > ER
IUNT = LOGICAL UNIT OF INPUT NT FILE
IUNZ = LOGICAL UNIT OF INPUT NZ FILE
IOUER = LOGICAL UNIT OF OUTPUT ER FILE
IOUET = LOGICAL UNIT OF OUTPUT ET FILE
IOUEZ = LOGICAL UNIT OF OUTPUT EZ FILE
IOUDIS = LOGICAL UNIT OF OUTPUT UR FILE

C
C
REAL NT(50,50),NZ(50,50)
INTEGER IOUER,IOUET,IOUEZ
LOGICAL FINITE,DISPLA,ETGTER

DIMENSION X(50),Y(50),ER(50,50),ET(50,50),EZ(50,50),
       UR(50,50),DUDR(50,50)

TYPE *," INPUT NT, NZ FILE NUMBER ",IUNT, IUNZ

INPUT NT FILE

READ(IUNT,*) NY,DUM,DUM
READ(IUNT,*) (X(I),I=1,NX)
READ(IUNT,*) (Y(I),I=1,NY)
DO 100 I = 1,NY
     READ(IUNT,*) (NT(I,J),J=1,NX)
100 CONTINUE

TYPE *," NT FILE READ COMPLETE FROM UNIT ",IUNT

INPUT NZ FILE

READ(IUNZ,*) NY,DUM,DUM
READ(IUNZ,*) (X(I),I=1,NX)
READ(IUNZ,*) (Y(I),I=1,NY)
DO 110 I = 1,NY
     READ(IUNZ,*) (NZ(I,J),J=1,NX)
110 CONTINUE

TYPE *," NZ FILE READ COMPLETE FROM UNIT ",IUNZ

TT = 1.0
TZ = 1.0
FEP = 1.0
FINITE = .FALSE.
DISPLA = .FALSE.
ETGTER = .FALSE.
TYPE *, ' TYPE OUTPUT FILE NUMBER FOR ER,ET,EZ * READ(5,*) OIER,OUFT,OUEZ
TYPE *, ' INPUT THICKNESS TT,TZ AND FRINGE CONSTANT * READ(5,*,ERR=120) TT,TZ,FEP
120 TYPE *, ' FINITE DEFORMATION ? <1> = NO * READ(5,*,ERR=130) IDUM IF (IDUM.NE.1) FINITE = .TRUE.
130 TYPE *, ' DISPLACEMENT OUTPUT ? <1> = NO * READ (5,*) IDUM IF (IDUM.NE.1) DISPLA = .TRUE.
TYPE *, ' ET > ER ? <1> = NO * READ (5,*) IDUM IF (IDUM.NE.1) ETGTER = .TRUE.

C-----CALCULATION BEGINS-----------------------------------------------

CT = FEP/TT
CZ = FEP/TZ
IF (FINITE) GO TO 250

C------SMALL DEFORMATION---------------------------------------------

IF (ETGTER) GO TO 215

C------ WHEN ER > ET

DO 200 I = 1,NY
   DO 210 J = 1,NX
      ER(I,J) = (CT*NT(I,J)+CZ*NZ(I,J))/3.0
      EZ(I,J) = (CZ*NZ(I,J)-2.0*CT*NT(I,J))/3.0
      ET(I,J) = (CT*NT(I,J)-2.0*CZ*NZ(I,J))/3.0
   210 CONTINUE
200 CONTINUE
GO TO 300
(DE (I'2) - CZ share) - (E) share
ER (I'2) = (sqr (E) - 120 C) - 6.0
C = CZ share (I'2) - CZ share (I'2) - CZ share (I'2)
B = 3.0 - 2.0 CZ share (I'2) - 2.0 CZ share (I'2)

DO 270 J = 1, NX
260 DO 280 I = I, NY

-------- ER > EL --------

GO TO 300
230 CONTINUE

-------- ER > EL --------

GO TO 260

-------- FINITE DEFORMATION --------

GO TO 300
210 CONTINUE

-------- WHEN --------

GO TO 300
200 CONTINUE

-------- WHEN --------
EZ(I, J) = ER(I, J) - CT*NT(I, J)

270 CONTINUE
280 CONTINUE
GO TO 300

C-----------------------------------------------
C
C OUTPUT IN AGRAPH FORMAT
C-----------------------------------------------
C
300 ONE = 1.0
WRITE(OUER, *) NX, NY, ONE, ONE
WRITE(OUER, *) (X(I), I=1, NX)
WRITE(OUER, *) (Y(I), I=1, NY)
DO 310 I = 1, NY
   WRITE(OUER, *) (ER(I,J), J=1, NX)
310 CONTINUE

C WRITE(OUET, *) NX, NY, ONE, ONE
WRITE(OUET, *) (X(I), I=1, NX)
WRITE(OUET, *) (Y(I), I=1, NY)
DO 320 I = 1, NY
   WRITE(OUET, *) (ET(I,J), J=1, NX)
320 CONTINUE

C WRITE(OUEZ, *) NX, NY, ONE, ONE
WRITE(OUEZ, *) (X(I), I=1, NX)
WRITE(OUEZ, *) (Y(I), I=1, NY)
DO 330 I = 1, NY
   WRITE(OUEZ, *) (EZ(I,J), J=1, NX)
330 CONTINUE

C IF (.NOT. DISPIA) GO TO 600
OUTPUT DISPLACEMENT

TYPE *, 'WHAT IS THE OUTPUT FILE NUMBER FOR DISPLACEMENT?'
READ(5,*) OUDIS

DO 500 I = 1,NY
  DO 400 J = 1,NX
    UR(I,J) = X(J)*SQRT(1.0-2.0*ET(I,J))
    DUDR(I,J) = 1.0 - SQRT(1.0-2.0*ER(I,J))
  CONTINUE

WRITE(OUDIS,1000) Y(I)
  DO 410 JJ = 1,NX
    WRITE(OUDIS,1001) X(JJ), NT(I, JJ), NZ(I, JJ), ER(I, JJ),
                       ET(I, JJ), EZ(I, JJ), UR(I, JJ), DUDR(I, JJ)
  CONTINUE

500 CONTINUE

1000 FORMAT (/IX, '------------AT Z = ',F8.4,'------------/')

1001 FORMAT (1X,10(F8.4,2X))

STOP
END