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
Development of a strain softening constitutive model for rock

Claborn, Forrest Jay, M.S.

Rice University, 1988
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. Dissertation contains pages with print at a slant, filmed as received 
16. Other ____________________________________________________________

________________________________________________________

________________________________________________________

UMI
RICE UNIVERSITY

DEVELOPMENT OF A STRAIN SOFTENING
CONSTITUTIVE MODEL FOR ROCK

by

F. JAY CLABORN

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

MASTER OF SCIENCE

APPROVED, THESIS COMMITTEE:

Dr. J. B. Cheatham, Jr.,
Professor of Mechanical Engineering,
Director

Dr. J. E. Akin,
Professor of Mechanical Engineering

Dr. A. J. Duran,
Associate Professor of Civil Engineering

Houston, Texas
August, 1987
ABSTRACT

DEVELOPMENT OF A STRAIN SOFTENING
CONSTITUTIVE MODEL FOR ROCK

by
F. Jay Claborn

A plasticity based constitutive model with two yield surfaces is developed to model the nonlinear axial and volumetric behavior observed for rocks in triaxial compression. The failure surface is allowed to soften at low pressures to simulate fracturing. A method for treating the corner where the yield surfaces intersect is developed.

In comparisons with experimental data, the model accurately predicts the axial response in triaxial tests. The model predicts too much dilation for low confining pressure triaxial tests.

The constitutive model was used in conjunction with a finite element code to simulate axisymmetric indentation of rock. Because of the existence of the corner in the yield surface, the simulation was computationally too slow to be practical.
ACKNOWLEDGEMENTS

I would like to express my appreciation to the Hughes Tool Company and its management, specifically Messrs. E. M. Galle, H. S. Word, Jr. and J. L. Kelly, Jr. for the complete funding and patient support of this research.

I have benefited greatly from the wealth of practical rock mechanics experience that Dr. J. B. Cheatham has brought to this project.

I thank Georgia Andrews for assistance with production of the figures.

Most of all, I must thank my wife, Tish, for her longsuffering and constant encouragement.
# Table of Contents

Abstract .................................................................................................................. ii
Acknowledgments .................................................................................................... iii
Table of Contents ..................................................................................................... iv
List of Figures ......................................................................................................... v
Nomenclature .......................................................................................................... vi
I. Introduction ......................................................................................................... 1
II. The Mechanical Behavior of Rock ..................................................................... 3
III. Review of Previous Work .................................................................................. 15
    3.1 Constitutive Modeling of Geomaterials ....................................................... 15
    3.1.1 Purpose of a Constitutive Model ........................................................... 15
    3.1.2 Description of Various Constitutive Models .......................................... 17
    3.2 Finite Element Modeling of Rock Indentation ........................................... 25
IV. Development of Relevant Plasticity Theory ..................................................... 30
V. Numerical Implementation of the Constitutive Model ....................................... 39
    5.1 Form of the Constitutive Model ................................................................. 39
    5.2 Elastic Strain Increments .......................................................................... 40
    5.3 Failure Surface Strain Increments .............................................................. 43
    5.4 Cap Surface Strain Increments .................................................................. 49
    5.5 Corner Treatment ....................................................................................... 50
    5.6 Computer Programs .................................................................................... 55
    5.6.1 Material Model Subroutines ................................................................... 55
    5.6.2 Triaxial Driver Subroutines .................................................................... 56
    5.7 Demonstration of Model Performance ...................................................... 57
VI. Implementation of the Model into NIKE2D ..................................................... 69
    6.1 Finite Element Procedure .......................................................................... 69
    6.2 Elastic Constitutive Matrix ......................................................................... 73
    6.3 Elastic-Plastic Tangent Constitutive Matrix ............................................... 74
    6.3.1 Failure Surface Constitutive Matrix ....................................................... 75
    6.3.2 Cap Surface Constitutive Matrix ............................................................ 75
    6.4 Formation of Tangent Constitutive Matrix for Corner Increment ............... 76
    6.5 Revised NIKE2D Computer Code ............................................................... 78
VII. Indentation Simulations .................................................................................... 80
    7.1 Problem Description .................................................................................... 80
    7.2 Results and Discussion .............................................................................. 81
VIII. Conclusions and Recommendations for Further Work .................................. 93
IX. References ........................................................................................................ 95
Appendix ................................................................................................................ 100
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Influence of Diameter-to-Height Ratio on Stress-Strain Behavior</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Tensile Stress Contours from McClintock-Walsh Equation</td>
<td>5</td>
</tr>
<tr>
<td>2.3</td>
<td>Axial Strain versus Differential Stress for Oya Tuff</td>
<td>7</td>
</tr>
<tr>
<td>2.4</td>
<td>Volumetric Strain Response of Oya Tuff</td>
<td>7</td>
</tr>
<tr>
<td>2.5</td>
<td>Description of Stain Softening Rock</td>
<td>10</td>
</tr>
<tr>
<td>2.6</td>
<td>Unloading Behavior of Sandstone in Unaxial Compression</td>
<td>12</td>
</tr>
<tr>
<td>2.7</td>
<td>Response of Porous Rock to Hydrostatic Pressure</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>Interrelationships of Variables in Solid Mechanics Problem</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>Evolution of Yield Surfaces for the Farque-Desai Model</td>
<td>22</td>
</tr>
<tr>
<td>5.1</td>
<td>Angles Used to Determine which Yield Surface is Violated</td>
<td>42</td>
</tr>
<tr>
<td>5.2</td>
<td>Geometry of the Cap</td>
<td>51</td>
</tr>
<tr>
<td>5.3</td>
<td>Types of Corner Stress Increments</td>
<td>51</td>
</tr>
<tr>
<td>5.4</td>
<td>Effect of Corner Treatment on Volumetric Strain Response</td>
<td>54</td>
</tr>
<tr>
<td>5.5</td>
<td>Failure Surface at Initial Yield</td>
<td>61</td>
</tr>
<tr>
<td>5.6</td>
<td>Failure Surface at Residual Strength</td>
<td>62</td>
</tr>
<tr>
<td>5.7</td>
<td>Failure Surface at Peak Strength</td>
<td>63</td>
</tr>
<tr>
<td>5.8</td>
<td>The Extreme Positions of the Failure Surface</td>
<td>64</td>
</tr>
<tr>
<td>5.9</td>
<td>Axial Response Predicted by the Model</td>
<td>65</td>
</tr>
<tr>
<td>5.10</td>
<td>Experimental and Model Axial Behavior</td>
<td>66</td>
</tr>
<tr>
<td>5.11</td>
<td>Volumetric Response Predicted by the Model</td>
<td>67</td>
</tr>
<tr>
<td>5.12</td>
<td>Experimental and Model Volumetric Behavior</td>
<td>68</td>
</tr>
<tr>
<td>7.1</td>
<td>Mesh Geometry</td>
<td>83</td>
</tr>
<tr>
<td>7.2</td>
<td>Slide-Line Nodes</td>
<td>84</td>
</tr>
<tr>
<td>7.3</td>
<td>Deformed Configuration for Original Model</td>
<td>85</td>
</tr>
<tr>
<td>7.4</td>
<td>Effective Plastic Strain for Original Model</td>
<td>86</td>
</tr>
<tr>
<td>7.5</td>
<td>Volumetric Strain for Original Model</td>
<td>87</td>
</tr>
<tr>
<td>7.6</td>
<td>Maximum Principal Stress for Original Model</td>
<td>88</td>
</tr>
<tr>
<td>7.7</td>
<td>Deformed Configuration for Capless Model</td>
<td>89</td>
</tr>
<tr>
<td>7.8</td>
<td>Effective Plastic Strain for Capless Model</td>
<td>90</td>
</tr>
<tr>
<td>7.9</td>
<td>Volumetric Strain for Capless Model</td>
<td>91</td>
</tr>
<tr>
<td>7.10</td>
<td>Maximum Principal Stress for Capless Model</td>
<td>92</td>
</tr>
</tbody>
</table>
Nomenclature

\( F \) \hspace{1cm} \text{Cap surface function}
\( G \) \hspace{1cm} \text{Elastic shear modulus}
\( G_1, G_2 \) \hspace{1cm} \text{Shear modulus parameters}
\( I_1, I_2, I_3 \) \hspace{1cm} \text{Invariants of the stress tensor, } \sigma_{ij}
\( I_i^n, s_{ij}, etc. \) \hspace{1cm} \text{Values of the appropriate quantities at the end of the } n \text{th loading increment}
\( I_i^n, s_{ij}, etc. \) \hspace{1cm} \text{Elastic trial values of the appropriate quantities}
\( J_1, J_2, J_3 \) \hspace{1cm} \text{Invariants of the deviatoric stress tensor, } s_{ij}
\( K \) \hspace{1cm} \text{Elastic bulk modulus}
\( K_1, K_2 \) \hspace{1cm} \text{Bulk modulus parameters}
\( p \) \hspace{1cm} \text{Pressure}
\( Q \) \hspace{1cm} \text{Failure surface function}
\( X(e_{lk}^P) \) \hspace{1cm} \text{Cap surface hardening function}
\( X_{beg}, c_8, c_9 \) \hspace{1cm} \text{Parameters for cap surface hardening function}
\( a \) \hspace{1cm} \text{Partial of the yield surface function with respect to } I_1
\( c_1, c_2, ..., c_7 \) \hspace{1cm} \text{Failure surface constants}
\( \det \) \hspace{1cm} \text{Determinant of a tensor or matrix}
\( e_{ij} \) \hspace{1cm} \text{Components of the deviatoric strain tensor}
\( d \lambda \) \hspace{1cm} \text{Scalar plastic multiplier}
\( d\sigma_{ij}, d\varepsilon_{ij}, etc. \) \hspace{1cm} \text{Incremental values of the appropriate quantities}
\( f \) \hspace{1cm} \text{A loading function or yield surface equation}
\( r_{cap} \) \hspace{1cm} \text{Ratio of semi-minor to semi-major axes of cap}
\( r_1, r_2 \) \hspace{1cm} \text{Strain splitting ratios}
\( s_{ij} \) \hspace{1cm} \text{Components of deviatoric stress tensor}
[B] Strain-displacement matrix
[C] Tangent constitutive matrix
[D] Elastic constitutive matrix
[f] External loads vector
[K] Tangent stiffness matrix
[N] Shape function matrix
{ψ} Residual load vector for jth iteration
{σ} Finite element stress vector
δ_{ij} Kronecker delta
ε_{ij} Components of the strain tensor
ε_{ij}^e Components of the elastic strain tensor
ε_{ij}^p Components of the plastic strain tensor
ε_{kk} Plastic volumetric strain
\bar{\varepsilon}^p Equivalent plastic strain
κ Hardening parameter
κ_a Failure surface hardening parameter
κ_H Cap surface hardening parameter
σ_{ij} Components of Cauchy stress tensor
I. INTRODUCTION

The goal of this work is to develop a constitutive model that can reproduce many of the non-linear stress - strain characteristics that are commonly observed for rock. In conventional triaxial tests two unusual non-linear stress - strain responses that can be observed for many rocks are strain softening and volumetric expansion. Strain softening describes the low confining pressure phenomenon in which the axial stress has a distinct maximum such that axial straining beyond the maximum axial strength causes reduced axial stress or softening. Also at low confining pressures the volumetric response of the rock displays a reversal at some critical axial strain and becomes dilative leading to volumetric expansion. At higher confining pressures neither of these behaviors is observed.

Recent research in the area of constitutive models for geomaterials has produced a multitude of models each of which is aimed at capturing a particular range of non-linear behaviors. It appeared to the author, though, that a fairly general model which was capable of simulating both the low and high confining pressure response of rock was not available in the literature. The model described in this thesis was developed as an attempt to fill this apparent gap in existing constitutive models.

The model that was developed falls within the description of classical plasticity theory. It employs a two surface elastic - plastic boundary or yield surface and uses an associated flow rule to define the plastic strains.

In order to use the model to solve a boundary value solid mechanics problem, the finite element method is used. A quite comprehensive solid mechanics finite element system is available to the public from Lawrence Livermore National Laboratories. The material model was implemented into this finite element code. A boundary value prob-
lem of particular interest to the author is the indentation of rock. Finite element simulations of rock indentation were performed, and conclusions as to the applicability of this model in conjunction with the finite element technique to the indentation problem have been made.
II. THE MECHANICAL BEHAVIOR OF ROCK

The purpose of this section is to describe and classify some of the pertinent nonlinear mechanical behaviors common to most rocks. It is by no means a complete treatment of the subject. In particular, only those mechanisms which relate to the proposed constitutive model will be dealt with in detail. Other behaviors for which the model has no predictive capabilities will only be mentioned in passing. Excellent references solely devoted to the mechanical behavior of rock include the works by Farmer [1], Pincus and Hoskins [2], Baidyuk [3], and Chong and Smith [4].

Most mechanical data on rocks is obtained from conventional compressive triaxial tests. In order to understand the mechanisms of deformation involved in this rather simple test, it is worthwhile to consider the test in some detail. Of particular importance are the test specimen geometry and the method by which the test apparatus applies the load. The rock test specimen is a right circular cylinder having a length-to-diameter ratio of 2.0 - 2.5. The sides of the specimen are to be smooth and free from abrupt irregularities, and the specimen ends are to be parallel to each other, normal to the longitudinal axis, and ground and end lapped. In a test, an axial compressive load is applied to the ends of the specimen by the platens of the test apparatus. A radial load is independently applied to the specimen by pressurized oil around the circumference of the specimen. The circumference of the specimen is jacketed with a flexible, impermeable membrane to prevent intrusion of oil into the specimen. One loading platen is spherically seated, while the other is rigid.

The choice of cylindrical specimen geometry is to minimize edge effects. Ideally, the test would induce a known uniform stress field throughout the specimen; however, it is not possible to avoid the end effects which disturb the stress distribution. The large influence of the end effect can be seen in experimental axial stress-strain curves in which
the specimen length is varied. Curves of this type have been developed by Hudson, Brown and Fairhurst [5] for Georgia Cherokee Marble and by Bieniawski [6] for a fine-grained sandstone. Bieniawski's curves are reproduced in Figure 2.1. It is interesting to note that for the uniaxial case (i.e. a test with zero lateral or confining pressure), Hawkes and Mellor [7] have predicted the existence of tensile zones at the corners and in the middle of the specimen based on the McClintock-Walsh failure criterion (see Figure 2.2). Hence, it can be concluded that the standard triaxial test is not an "ideal" test since it induces a non-uniform stress field, and, in fact, the test may be capable of disguising the expected failure mode by inducing tensile zones and, consequently, tensile failure mechanisms.

Prior to about 1965, almost all of the triaxial testing performed on rock measured only the prefailure behavior of the specimen. The construction of the testing machines during that era was such that when the maximum load bearing capacity of the specimen was attained, the release of elastic energy from the test apparatus resulted in sudden and violent disintegration of the specimen. It was discovered that by making the test machine sufficiently stiff and by controlling the machine displacement rather than the applied load, it was possible to measure the post "failure" strength of the rock. In this sense "failure" occurs when the specimen is deformed beyond the maximum axial strength. Indeed, beyond the failure point the rock as a material has fractured; however, the specimen still possesses load bearing capacity as a structure. In the post failure region the specimen strength decreases with increasing axial deformation. The slope of the axial stress-strain curve is negative in this region, and the behavior is referred to as strain-softening. For a complete characterization of a rock specimen's deformational response, it is, therefore, important to have either a "stiff" testing machine in which the displacement can be controlled or a fast acting closed-loop servo-controlled testing system. (See Hudson, Crouch, and Fairhurst [8] for a complete discussion of machines for measuring
FIGURE 2.1 The Influence of Diameter-to-Height Ratio on Stress-Strain Behavior of Sandstone in Uniaxial Compression (After Bieniawski [5])

FIGURE 2.2 Tensile Stress Contours Calculated from the McClintock-Walsh Equation for an Unconfined Cylindrical Specimen Subjected to Unit Axial Load. Areas of Peak Tensile Stress are Stippled. (After Hawkes and Mellor [7])
the post-failure behavior of rock).

At low confining pressures, "strong" rocks will exhibit a distinct maximum strength. "Weak" rocks, on the other hand, may display increasing strength with increasing axial strain even at low confining pressures. (No attempt is made here to distinguish between "weak" rocks and soils.) For many rocks at sufficiently high confining pressures, the distinct maximum strength peak disappears and the axial stress-strain curve either continues to rise or approaches zero slope. Experimental axial stress-strain curves from Ichikawa [9] for Oya tuff showing strain softening behavior at low confining pressures and nearly perfectly plastic behavior at high confining pressures are shown in Figure 2.3. The elimination of the strain softening region with increasing confining pressure is referred to as the "brittle to ductile" transition. Figure 2.4 shows the volumetric strain results from Ichikawa's tests. Again the brittle to ductile transition with increasing pressure is quite evident. Associated with the strain softening parts of the low pressure curves is volumetric expansion or dilatancy. At pressures above the brittle to ductile transition volumetric compaction occurs. Obviously, the confining pressure has a dramatic influence on the deformational behavior of rock. Farmer [1] makes the following further observations as to the effects of increasing confining pressure.

(a) Confining pressure increases the strength of the rock and the degree of post-yield axial strain hardening. These effects diminish with increasing confining pressure.

(b) At low confining pressures there is increasing dilation, which reduces at at higher confining pressures until at the highest there is little or no dilation.

(c) If Poisson's ratio, axial stress and the slope of the stress-axial strain and stress-volumetric strain curves are plotted, it can be seen that they approach a limiting value with increasing confining pressure, the limiting value of Poisson's ratio being 0.5.

In 1968 Scholz [10,11] investigated the mechanism which gives rise to dilative behavior at low pressures. For six types of rocks, he used a stiff testing machine to meas-
FIGURE 2.3 Experimentally Determined Curves of Axial Strain versus Differential Stress for Oya Tuff (after Ichikawa, et al [9] )

FIGURE 2.4 Experimentally Determined Volumetric Strain Response of Oya Tuff (after Ichikawa, et al [9] )
ure the volumetric strain during deformation. He also measured the corresponding microseismic emissions. He found that there was an initial flurry of microseismic activity at low stresses which he attributed to the closing of existing cracks and pores. Following this initial activity, there was only a low level of activity up to about 40% of the peak stress. Above 40%, the activity steadily increased until just before peak strength at which point there was a rapid increase in activity. In seeking a correlation between the microseismic activity and the volumetric strains, Scholz found that for all six rocks tested the accumulated frequencies of microseismic events correlated very well with the volumetric strain over the dilatant range up to 95% of peak stress. Both the dilative behavior and the corresponding microseismic emissions are evidence of microfracturing or microcracking in the rock. It would seem reasonable that in order to gain insight into the process of rock degradation or failure, one should consider the factors affecting the propagation of microfractures. Farmer [1] makes the following two observations concerning the mechanical description of microfracturing.

(a) Microfractures are initiated at a stress level related to the peak stress or 'strength' of the rock.

(b) The dilation in a direction normal to the major (compressive) principal stress indicates that the microfractures spread in a direction parallel to the major principal stress - in other words they are probably a result of tensile failure.

This description of microfracture is consistent with the classical elastic fracture mechanics criterion for crack initiation as proposed by Griffith. Jaeger and Cook [12] have used Griffith's criterion to derive a fracture criterion for the spreading of a randomly oriented elliptical crack in a two-dimensional stress field. Further theoretical development was done by McClintock and Walsh [13] who modified Griffith's criterion to allow for friction along the surface of closing cracks. Unfortunately, this fracture mechanics approach has not yielded an accurate means for describing the total rock behavior. Farmer explains the short comings of the fracture approach.
It should be stated that, whereas the Griffith criterion illustrates the mechanism of rock failure, it cannot describe or predict it accurately. This is because the inhomogeneity intrinsic in most rocks will produce fluctuations in the stress field which will both limit the propagation of cracks after initiation and reduce the expected fracture strength of the specimen as a whole. Nevertheless it does provide a tool with which to examine experimental data and to develop an understanding of the processes which lead to rock fracture.

Price [14] has devised a useful classification of the progression of deformation in terms of the dominant mechanism of deformation. Figure 2.5 illustrates the stages of strain-softening rock deformation as described by Price.

Stage I - When the rock is initially stressed any pre-existing microcracks or pore space orientated at suitable angles to the applied stress will close. This causes, in weaker and more porous rocks, an initial nonlinearity of the axial stress - strain curve.

Stage II - The rock has a near-linear axial and lateral stress-strain curve which in largely recoverable. This is accompanied by compaction - again recoverable - which has distinct similarities with elastic compaction, although the Poisson's ratio, particularly in stiffer unconfined rocks, tends to be low. It is quite reasonable to describe the deformation characteristics as elastic in this stage. There is a minimum of seismic activity during this stage and it is also reasonable to argue that microcrack propagation only starts at the upper boundary of about 35-40% peak stress.

Stage III - This stage is characterized by the onset of dilation and by a near-linear increase in volume, which is offset against continuing compaction. There is also a near-linear axial stress-strain curve which like Stage II in nearly fully recoverable. In can be proposed that microcrack propagation occurs in a stable manner during this stage and that microcracking events occur independently of each other and are distributed throughout the specimen. The upper boundary of the stage is the point of maximum compaction and zero volume change. It occurs at about 80% peak stress and has occasionally been used as a reference point for critical state descriptions of rock deformation.

Stage IV - This stage is characterized by rapid acceleration of microcracking events and of volume increase. The spreading of microcracks is no longer independent and clusters of cracks in the zones of highest tensile stress tend to coalesce and start to form tensile fractures or shear planes - depending on the strength (and degree of confinement) of the rock.
FIGURE 2.5 Description of Strain Softening Rock Deformation
(after Price [14])
Stage V - This is the stage where the rock has passed peak stress, but is still intact, even though the internal structure is highly disrupted. In this stage the crack arrays fork and coalesce into macrocracks or faults. It is possible to talk about failure at this point; more sensible to talk about strain softening deformation. In this description, at peak stress the test specimen starts to become weaker with increasing strain. Thus further strain will be concentrated on weaker elements of the rock which have already been subjected to strain. This in turn will lead to zones of concentrated strain or shear planes.

Stage VI - In this stage the rock has essentially parted to form a series of blocks rather than an intact structure. These blocks slide across each other and the predominant deformation mechanism is friction between the sliding blocks. Secondary fractures may occur due to differential shearing. The axial stress or force acting on the specimen tends to fall to a constant residual strength value, equivalent to the frictional resistance of the sliding blocks.

Another important phenomenon associated with strain-softening is the unloading behavior. Figure 2.6 from Bieniawski [6] shows unloading "loops" in the strain-softening region for sandstone in uniaxial compression. The unloading and reloading curves are quite close except for the small amount of hysteresis. Also, both curves have a nearly constant slope or "modulus of elasticity". It is easily seen, though, that as the maximum axial strain increases the slope of the unloading/reloading curves decreases so that the resistance of the structure to deformation is decreased due to the increased deformation.

Strain-hardening deformation is somewhat easier to understand. In the hardening case, the strain does not localize as in the softening case. A stain-hardening rock becomes stronger as it deforms. Consequently, the strain tends toward homogeneity throughout a confined specimen, since those elements of the rock which have strained the most will be stronger than those which have strained less. If the specimen continues to deform homogeneously (i.e. provided a flaw does not cause strain localization), a state will be reached at which large distortions occur without volume change. Upon unloading of such a specimen the inelastic or irrecoverable strain that has occurred in the still intact
FIGURE 2.6 Unloading Behavior of Sandstone in Uniaxial Compression (after Bieniawski [6])
specimen is easily seen. Because there is no obvious maximum strength for a hardening specimen, failure is identified as large axial strain.

It is important to realize that even in the absence of shear stresses, rock can be made to deform inelastically. In an isotropic consolidation test, the test sample is loaded in hydrostatic pressure only so that there is effectively no shear stress. Typical response of a porous rock under isotropic consolidation is shown in Figure 2.7. From the figure three behaviors associated with isotropic consolidation can be observed.

(a) Rocks exhibit a nonlinear compacting behavior under hydrostatic states of stress.
(b) The compressive volumetric strain is bounded.
(c) After isotropic unloading, there remains some permanent compaction.

The nonlinear compacting and resulting permanent compaction are attributed to the collapse of pore space within the rock matrix.

The description of the mechanical response of rock given in this chapter has not been complete. Several important behaviors have not been mentioned and will not be treated in this thesis. The author believes that trying to simulate the characteristics anteriorly described is quite enough of a complex task without introducing further complications. Rock behaviors that are knowingly being ignored are listed below.

(1) Rate of loading effects.
(2) Temperature effects.
(3) Effect of fluid in the pore space.
(4) Anisotropy of material properties.
FIGURE 2.7 Typical Inelastic Response of a Porous Rock to Hydrostatic Pressure (after Chen and Baladi [28])
III. REVIEW OF PREVIOUS WORK

This chapter will review research which has been done in two areas. Firstly, the different approaches to the constitutive modeling of geomaterials will be classified and examined. Secondly, previous attempts at finite element modeling of the process of rock indentation will be reviewed.

3.1 Constitutive Modeling of Geomaterials

There is a vast amount of literature describing constitutive models for geomaterials. The bulk of this work has been published since 1978, and there remains much research activity in the field. Much of this recent work has been compiled into six books [15,16,17,18,19,20]. The formulation of a coherent synthesis of the work in this area is a very difficult task. The researchers attack the problem from quite varied viewpoints and, hence, the character and basis of their implementations are likewise varied. The small subset of constitutive models that is considered herein is restricted to those which describe time independent behavior of a drained geomaterial.

3.1.1 Purpose of a Constitutive Model

A constitutive model comprises a systematic, consistent set of mathematical laws which relates the stress at a point in a solid to the strain at that point. The purpose of establishing such a mathematical model is for use in solving complex boundary value problems. The usefulness of a constitutive model in such a solution depends on the extent to which the actual physical phenomenon can be understood and simulated. The role of the constitutive model in the solution of a boundary value problem is explained by
Desai and Siriwardane [18].

A solution to a boundary value problem in continuum mechanics requires constitutive equations in addition to the governing field equations. The basic principles governing Newtonian mechanics are (a) conservation of mass, (b) conservation of momentum, (c) conservation of moment of momentum (or angular momentum), (d) conservation of energy, and (e) laws of thermodynamics; these principles are considered to be valid for all materials irrespective of their internal constitution. Therefore, a unique solution to a boundary value problem in continuum mechanics cannot be obtained only with the applications of governing field equations. Hence a unique determination of the response requires additional considerations that account for the nature of different materials. The equations that model the behavior of a material are called "constitutive equations" or "constitutive laws" or "constitutive models".

Consider a static solid mechanics problem. The governing field equations are simply the equations of equilibrium. For points in the interior of the body, equilibrium is given by

\[ \sigma_{ji,j} + F_i = 0 \]  \hspace{1cm} (3.1)

For points on the surface

\[ T_i = \sigma_{ji} n_j \]  \hspace{1cm} (3.2)

where \( \sigma_{ij} \) is the Cauchy stress tensor, and the comma in the term \( \sigma_{ji,j} \) denotes partial differentiation with respect to the coordinate axes, \( x_j \). Following the summation convention, repeated indices in a term denote summation over the applicable range. \( F_i \) represents the components of the body force per unit volume, \( T_i \) are the surface tractions, and \( n_j \) is the unit outward normal to the surface.

The conditions of geometric compatibility require the integrability of the strain field to yield the displacement field. For small deformations this condition leads to the strain-displacement relations.

\[ \varepsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}) \]  \hspace{1cm} (3.3)
where $\varepsilon_{ij}$ are the components of the small strain tensor and $u_i$ are displacement components. The interrelationships of equilibrium, compatibility, and the constitutive laws are illustrated for a static solid mechanics problem in Figure 3.1.

### 3.1.2 Description of Various Constitutive Models

Chen [21,22] has developed classifications for many of the existing constitutive models. A summary of the advantages and limitations of the various models he has classified appears in Table 3.1. It is not practical to discuss all of these models in this thesis, so the interested reader is referred to Chen [21,22]. It is fruitful, though, to examine several of the more advanced models that have been proposed.

A major step in obtaining a more realistic model that was based on the principles of work-hardening plasticity, was the introduction of cap models. Drucker, Gibson and Henkel [23] were the first to propose that successive yield surfaces might resemble Drucker-Prager cones with convex end spherical caps. There are two important innovations in this work. The first is the introduction of the idea of a spherical cap fitted to the cone. The second is the use of the plastic compaction as the strain-hardening parameter to determine successive hardened positions of the cap. Applying these ideas, Roscoe and his co-workers at Cambridge University developed a family of strain-hardening cap type models. Two of these Cambridge models, the Cam-clay model [24,25] and the modified Cam-clay model [26] have been widely used. Both of these models use an associated flow rule. An important concept in all of the Cambridge models is that of the critical state. The critical state line is the locus of the failure points of all shear tests under both drained and undrained conditions. At the critical state, large shear deformations occur with no change in stress or plastic volumetric strain.
FIGURE 3.1 Interrelationships of Variables in the Solution of a Static Solid Mechanics Problem (after Chen [21])
### Table 3.1
(After Chen [21])

<table>
<thead>
<tr>
<th></th>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cauchy Elastic: Modifications of Linear Elastic Models</strong></td>
<td>Conceptually and mathematically simple. Easy to determine the constants and wide data base is established for many parameters.</td>
<td>Path-independent, reversible. No coupling between volumetric and deviatoric responses. For arbitrary functions of the moduli, energy generation may occur for certain stress cycles.</td>
</tr>
<tr>
<td><strong>Hyperelastic Models</strong></td>
<td>Satisfies stability and uniqueness. Shear-dilatancy and effect of all stress invariants may be included. Attractive from programming and computer economy points of view.</td>
<td>Path-independent, reversible. Difficult to fit and requires large number of tests. Most models confined to small regions of applications.</td>
</tr>
<tr>
<td><strong>Hypoeelastic: Modifications of Linear Elastic Models</strong></td>
<td>Conceptually and mathematically simple. Ideal for finite element implementation. Easy to fit. Many of the parameters have wide data base. Has been used successfully in many practical applications.</td>
<td>Incrementally reversible. No coupling between volumetric and deviatoric responses. Behavior near failure cannot be described adequately. Possible energy generation for certain stress cycles if arbitrary functions for the moduli are used.</td>
</tr>
<tr>
<td><strong>First-Order Hypoeelastic Models</strong></td>
<td>Stress-path dependency. Stress-induced anisotropy.</td>
<td>Incrementally reversible. Tangent stiffness matrix is generally unsymmetric; thus requires increased storage and computation. Difficult to fit and requires large numbers of tests. Possible energy generation for certain stress cycles. No uniqueness proof in general.</td>
</tr>
<tr>
<td><strong>Deformation Theory of Plasticity Models</strong></td>
<td>Simple formulation. Allow hysteretic behavior.</td>
<td>Continuity problem at or near neutral loading. With the exception of unloading behavior is still path-independent.</td>
</tr>
</tbody>
</table>


### Table 3.1
(continued)

<table>
<thead>
<tr>
<th>Comparison of Modeling Techniques</th>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drucker-Prager Perfectly Plastic Models</td>
<td>Simple to use. Can be matched with Mohr-Coulomb by a proper selection of constants. Computer codes available. Limit analysis techniques can be used. Satisfies uniqueness requirements with associated flow rule.</td>
<td>Excessive plastic dilatancy at yielding. Cannot reproduce the hysteretic behavior within the failure surface.</td>
</tr>
<tr>
<td>Cap Type of Isotropic Hardening Models</td>
<td>Satisfy all the theoretical requirements of stability, uniqueness, and continuity. Give a proper control on plastic dilatancy. Allow hysteretic compaction during hydrostatic load-unload cycles.</td>
<td>Trial-and-error procedure needed to fit the data. Relatively complicated.</td>
</tr>
</tbody>
</table>
DiMaggio and Sandler [27] have proposed a generalized cap model. In their formulation the yield surface consists of a perfectly-plastic (failure) surface to which a strain-hardening elliptical cap is fitted. An associated flow rule was used on both yield surfaces. In this model, the functional forms of the failure and cap surfaces may be quite general thus allowing for the fitting of the model to a wide range of material responses. Chen [28] has detailed the implementation of a model of this type and developed computer code to perform the calculations. Sandler and Baron [29] have adapted this model to rocks by allowing only expansion of the cap. In this variation of the model, the cap movement is assumed to depend only on the maximum previous value of the plastic volumetric strain, hence the end cap expands but does not contract. This type of cap movement allows representation of a relatively large amount of dilatancy such as that which is often observed during failure of rocks at low pressures. Since the cap does not contract, low pressure stress states are able to reach the failure surface where the associated flow rule causes plastic dilation. When the cap is allow to contract, these low pressure states are captured on the cap surface which results in plastic compactional strain. Data from both hydrostatic compression and triaxial tests are used to determine the model parameters. Unfortunately, the fitting procedure is a trial-and-error process.

Faruque and Desai [30] have proposed a single, smooth yield surface model based on classical plasticity which alleviates many of the computational difficulties associated with cap models (i.e. the mathematical treatment of the intersection of the failure and cap surfaces). The growth of their yield surface due to hardening is illustrated in Figure 3.2 where the successive yield surfaces are plotted in a two-dimensional representation of a reduced stress space that is obtained by using the first invariant of the stress tensor, I₁, and the second invariant of the deviatoric stress tensor, J₂, as axes. Their model employs an associated flow rule. Implementation of the associated flow is straight-forward since the normal to the yield surface is everywhere uniquely defined. A clever definition of the
FIGURE 3.2 Evolution of Yield Surfaces for the Faruque-Desai [29] Model
hardening function allows for a realistic representation of both hydrostatic and deviatoric plastic behaviors. The hardening function, $\kappa$, is expressed in terms of the trajectory of total plastic strains and the ratio of the trajectory of deviatoric to total plastic strains.

$$\kappa = \kappa(\xi, r_D)$$  \hspace{1cm} (3.4)

where

$$\xi = \int (de_{ij}^p, de_{ij}^p)^{1/2}$$  \hspace{1cm} (3.5)

$$r_D = \int \frac{(de_{ij}^p, de_{ij}^p)^{1/2}}{\xi}$$  \hspace{1cm} (3.6)

and $de_{ij}^p$ is the incremental plastic strain tensor and $de_{ij}^p$ is the incremental deviatoric plastic strain tensor. A fitting procedure including a minimum of one hydrostatic and three conventional triaxial tests is described for determining the nine material constants of the model. As described the model is restricted to hardening behavior, and it is not known how one could devise a modification to allow for softening on only a portion of the surface.

Dougill [31,32] has developed the theory of "progressively fracturing solids" to treat materials which soften due to degradation or damage. In the development of this theory he exploits the so called "dual relationships" of hardening plasticity. The dual relationships refer to the fact that at any stage of deformation the yield surface in stress space can be mapped to provide a corresponding surface in strain space. In a strain space formulation, many of the important equations can be derived by analogy with hardening plasticity by simple exchange of the kinematic and static variables. Hence, instead of the plastic strain increment, $de_{ij}^p$, of hardening plasticity, the stain space description uses the inelastic stress decrement, $d\sigma_{ij}''$. In hardening plasticity the associated flow for $de_{ij}^p$ is derived from Drucker's postulate of a stable work hardening material. Softening
behavior violates Drucker's postulate so that conventional plasticity is theoretically constrained to hardening behavior. To derive the "flow" rule for $\sigma_{ij}^{''}$, Il'iuishin's postulate of plasticity replaces Drucker's definition of work hardening, hence the theory is able to represent softening behavior without violation of basic principles. The strain space "yield" surface is called the "fracture surface" and is a function of the strain tensor, $\varepsilon_{ij}$, and a damage parameter, H.

$$F(\varepsilon_{ij}, H) = 0$$  (3.7)

The measure of damage is taken to be the energy dissipated per unit volume of material. Clearly this model provides a consistent way to represent softening; however, it requires the use of many loading (fracture) functions to adequately describe real materials thus rendering it computationally unattractive.

In 1971 Valanis [33] proposed endochronic plasticity as an alternative theory to classical plasticity for the description of rate independent yet history dependent response of materials. The two theories differ in certain important respects in that

(a) the endochronic theory does not require the concept of a yield surface for its development,

(b) the physical assumptions that underlie the theory have their origin in irreversible thermodynamics of internal variables,

(c) the material memory is defined in terms of an intrinsic time scale which is a material property, and

(d) whereas for incremental plasticity the relationship between stress and strain increments is linear, for endochronic plasticity it is nonlinear.

The endochronic theory is attractive in that it provides comprehensive representation of the various aspects of inelastic behavior especially regarding the strain-softening and lateral and volumetric strain behavior in triaxial tests. The incremental nonlinearity,
though, is a computational roadblock.

Bazant and Kim [34] have developed an advanced model which uses an incrementally linear form of the endochronic theory in conjunction with the fracturing theory of Dougill. They give the following description of their "Plastic-Fracturing Theory".

Incremental plasticity and fracturing (microcracking) material theory are combined to obtain a nonlinear triaxial constitutive relation that is incrementally linear. A new hardening rule, called jump-kinematic hardening, is used for unloading, reloading, and cyclic loading. The theory combines the plastic stress decrements with the fracturing stress decrements, which reflect microcracking, and accounts for internal friction, pressure sensitivity, inelastic dilatancy due to microcracking, strain-softening, degradation of elastic moduli due to macrocracking, and hydrostatic nonlinearity due to pore collapse. Failure envelopes are obtained from the constitutive law as a collection of the peak points of the stress-strain response curves. The jump-kinematic hardening allows for inelastic response during unloading, reloading, and cyclic loading and, at the same time, it does not itself cause violation of Drucker's postulate. As a consequence of the incremental linearity, the plastic strain increments vanish for loading that is parallel to the loading surface; this response may be too stiff and questionable for material instability predictions. Six scalar material functions are needed to fully define the monotonic response.

Although the theory is quite complete in its ability to model various inelastic behaviors, the complexity of the theory itself and the difficulty in determining the six material functions detracts from the attractiveness of this model.

3.2 Finite Element Modeling of Rock Indentation

J. K. Wang [35] has developed a plane strain finite element program for simulation of the complete process of indentation, from indenter contact to chip formation. In summarizing this work, it is logical to first examine the constitutive model used and then to examine the implementation of the model in the finite element method.

Wang uses a Mohr type of failure envelope to identify the onset of inelastic defor-
motions. For stress states inside the failure envelope, the rock is treated as a linear elastic, isotropic, and homogeneous material. The inelastic behavior of his model is explained below.

1. The simplified failure criterion for an intact rock is assumed to be a linear Mohr envelope. Tensile rupture occurs when the minor principal stress equals the uniaxial tensile strength of the material. When the normal stress on the potential shear surface is compressive, i.e., where the modified Griffith theory applies, a linear envelope is chosen. The transition between tensile and compressive failures can be approximated by the relationship between intrinsic shear strength $\tau_0$ and uniaxial strength $\sigma_i$ as:

$$\sigma = \frac{1}{2\tau_0\mu} \tau^2 + \sigma_i$$

where $\mu$ is the slope of the Mohr envelope.

2. After tensile fracture, rock loses its cohesion on the newly created surface and still retains its strength in the direction parallel to the fracture surface.

3. After compressive failure, rock strength and stiffness decrease gradually along with the displacements until they finally reach the residual values. Degrees of failure are represented by dividing the space between two extreme envelopes, intact and residual, into many levels. On each level, i.e., the same failure envelope, the degree of failure and material properties are assumed the same.

The approach used to model inelastic deformation is developed purely from a behavioral viewpoint. No unified theory is proposed to describe the behavior. After failure, the material stiffness and Poisson's ratio are modified in a rather ad hoc fashion by employing mathematical relations which mimic, in form at least, observed behaviors. Essentially this constitutive model falls into the category of a variable moduli elasticity model.

The finite element code uses an iteration process to achieve equilibrium at each quasi-static load increment. An anisotropic element is used to represent a tensile fractured element. This procedure assumes that the crack plane is a principal plane for the anisotropic element; hence, in the direction normal to the plane, Young's modulus and Poisson's ratio are reduced to very small values. As desired, such an element retains the capability to withstand stress parallel to the crack plane. Another finite element "trick"
used is that of stress release. This process converts excessive stress that an element cannot bear to nodal loads and reapply these nodal loads to the element nodes and thereby to the system. In order to determine what procedure is to be used in a failing element, the failing elements are classified according to the complex scheme described below.

**Class 1:** Tensile fracture with an open crack. An anisotropic element is used for this class. Incremental stresses generated on the crack surface are released. Tensile or compressive stresses in the crack direction are checked for possible further failure. If the stress in that direction reaches the tensile or compressive strength, then the failed element becomes class 4 or class 3, respectively.

**Class 2:** Tensile fracture with closed crack. If the crack was closed, then the normal stress on the crack surface becomes compressive. Therefore, anisotropic material properties are abandoned. However, the crack direction is recorded in case of crack reopening. Compressive failure is checked for a possible change of failure state. Crack opening is determined by comparing the current element volume with a testing volume which is obtained by applying the current major principal stress on the element. If the volume of the current element is greater, then the crack is open.

**Class 3:** Compressive failure. Elements in this class follow the progressive strength failure. When the sum of the principal stresses of an element is in tension, the element is classified as class 4.

**Class 4:** Loose fragments. All the stresses in the element are released. Young’s modulus and Poisson’s ratio are assigned small values. The element volume at the beginning of this class is recorded. If the current volume is smaller than the recorded value, the element becomes class 3.

These classifications do allow one to gain a sense of physical insight into the indentation process; however, the fact that these classifications are not grounded in some sort of theory casts a shadow of doubt concerning the reliability of such physical interpretations. Wang recognized this deficiency and indicated in his recommendations that a better understanding of the post-failure behavior was essential to improve his analysis. Despite these shortcomings, he reported obtaining reasonable agreement between experiment and his simulations. It should be noted, though, that Wang’s implementation lacks the capability to simulate the behaviors of low pressure dilation and of continued hardening.
(without softening) at high confining pressures.

D. V. Swenson [36,37], working at Sandia National Laboratories, has used the HONDO [38] explicit finite element code to simulate rock indentation. The main purpose of his work was to model the drag bit cutting process; however, along the way he performed plane strain analysis of flat punch indentation to verify his constitutive models. Swenson asserts that a geologic constitutive model "sufficient for engineering use" must incorporate three behaviors: (1) brittle cracking in tension, (2) a pressure-dependent failure criterion in compression, and (3) a postfailure material description. Two models which satisfy these criteria have been implemented into HONDO. One of the models, the so called tensile fracture/plasticity model, uses plasticity to predict inelastic compressive behavior. A perfectly plastic parabolic yield or failure surface is employed along with a planar end cap which is normal to the hydrostat. A non-associated flow rule is used on the failure surface such that plastic strains arising from yielding on this surface produce no volume change, i.e. only deviatoric plastic strains result. On the planar end cap an associated flow rule and a tabulated pressure/compaction curve are used so that only plastic volumetric stain results from yielding of the cap. The second model, the tensile fracture/Coulomb model, uses a Mohr-Coulomb criterion to predict the formation of shear cracks. For this model, shear failure results in the simultaneous formation of two cracks which are equally inclined on either side of the most compressive stress direction at an angle which is a function the material internal friction angle. Swenson argues that the different post failure behaviors of the two models provides a realistic upper (plasticity model) and lower (Coulomb model) bound.

Both models exhibit the same cracking behavior in tension. Tensile failure is assumed to occur if the maximum principal stress exceeds a specified tensile strength. If failure occurs, a crack plane normal to the maximum principal stress direction is formed
by setting the shear stress on the crack face equal to zero and the normal stress to a specified pressure. In addition to modifying the stresses, it is necessary to calculate and accumulate the strain normal to a crack. This strain is used to indicate whether a crack is open or closed. Closed cracks are treated differently by the two models. When a crack closes in the plasticity model, all memory of the crack is lost and the material "heals". In the Coulomb model when a crack closes the shear stress is limited by a Coulomb friction criterion for post-failure material.

Swenson has recognized and discussed the problems associated with the use of a stress criterion for determining tensile fracture in finite element applications. He reports the following three limitations.

(1) Crack propagation is mesh-size dependent.
(2) Cracks tend to follow mesh lines.
(3) Spurious cracks may be introduced if the input tensile failure strength is relatively low.

Swenson and Ingraffea [39,40] have subsequently used linear elasto-dynamic fracture mechanics concepts to develop a finite element procedure for modeling dynamic crack propagation. This new formulation alleviates the problems listed above.

For both a confined and an unconfined test with a flat indenter on Berea sandstone, Swenson's finite element models, using a coefficient of friction of 0.6 between the indenter and the rock, predicted fractured patterns similar to those observed experimentally. As expected the two models bounded the postfailure response. The Coulomb model yielded a failure load below that which was measured experimentally, while the plasticity model continued to carry load after fracture and overestimated the failure load.
IV. DEVELOPMENT OF RELEVANT PLASTICITY THEORY

In this chapter the theory of plasticity will be used to derive pertinent equations needed for the development of a cap type constitutive model. The stress notation to be used is described first, followed by an introduction to the concept of plastic loading functions. The hardening parameters for the two loading functions of the model are defined, and the mathematical criterion used to determine the type of loading is given. Next, the strain response is considered. The elastic strain response is developed using variable elastic moduli. An associated flow rule is used to define the incremental plastic strains, and the consistency condition is used to find the scalar plastic multiplier. Finally, constitutive forms are derived.

The stress tensor, which has components denoted by $\sigma_{ij}$, can be decomposed into hydrostatic and deviatoric components.

$$\sigma_{ij} = \frac{1}{3} \sigma_{kk} \delta_{ij} + s_{ij}$$

(4.1)

Where $s_{ij}$ represents the components of the deviatoric stress tensor and $\delta_{ij}$ is the Kronecker delta. (The notational convenience of the summation convention is assumed throughout unless explicitly stated otherwise). Adopting the usual convention for describing geologic materials, the compressive stresses are taken as positive in sign. Hence, compacting strains are likewise positive in sign. The hydrostatic pressure, $p$, is defined as

$$p = \frac{1}{3} \sigma_{kk}$$

(4.2)

The invariants of the stress tensor, $\sigma_{ij}$, are denoted as

$$I_1 = \sigma_{kk} = 3p$$

(4.3)
\[ I_2 = \frac{1}{2} [tr(\sigma^2) - (tr \sigma)^2] = \frac{1}{2} (\sigma_{ij} \sigma_{ij} - \sigma_{kk}^2) \quad (4.4) \]

\[ I_3 = \det \sigma \quad (4.5) \]

The invariants of the deviatoric stress tensor, \( s_{ij} \), are given by

\[ J_1 = s_{kk} = 0 \quad (4.6) \]

\[ J_2 = \frac{1}{2} (s_{ij} s_{ij} - s_{kk}^2) = \frac{1}{2} s_{ij} s_{ij} \quad (4.7) \]

\[ J_3 = \det s \quad (4.8) \]

The yield surface(s) in stress space are defined mathematically by a loading function(s). The general form of a loading function is

\[ f(\sigma_{ij}, \kappa) = 0 \quad (4.9) \]

where the hardening parameter, \( \kappa \), is generally taken to be a function of the plastic strain, \( \varepsilon_{ij}^p \). For isotropic materials the loading function can be written in terms of the stress invariants and the hardening parameter.

\[ f(I_1, \sqrt{I_2}, I_3, \kappa) = 0 \quad (4.10) \]

By further restricting the yield surface(s) to have circular crossections in deviatoric planes, the loading function can be written simply as

\[ f(I_1, \sqrt{I_2}, \kappa) = 0 \quad (4.11) \]

For the cap model, the loading function or loading surface will consist of two parts: (1) a work hardening/softening "failure" surface and (2) a strain hardening end cap. The "failure" envelope portion of the loading function is given by
\[ f = h(I_1, \sqrt{J_2}, \kappa_h) = \sqrt{J_2} - Q(I_1, \kappa_h) = 0 \] (4.12)

The strain hardening cap is given by
\[ f = H(I_1, \sqrt{J_2}, \kappa_H) = \sqrt{J_2} - F(I_1, \kappa_H) = 0 \] (4.13)

The hardening parameters, \( \kappa_h \) and \( \kappa_H \), are different so that the hardening on the two surfaces is governed by different criteria. Following Owen and Hinton [41], the failure surface hardening parameter is assumed to be a function of the equivalent plastic strain, \( \bar{\varepsilon}^p \).

\[ \kappa_h = \kappa_h(\bar{\varepsilon}^p) \] (4.14)

where
\[ d\bar{\varepsilon}^p = \sqrt{\frac{2}{3}} (d\varepsilon_{ij}^p d\varepsilon_{ij}^p)^{1/2} \] (4.15)

As suggested by Chen and Baladi [28], the hardening parameter for the cap is taken to be a function of the plastic volumetric strain, \( \varepsilon_{kk}^p \).

\[ \kappa_H = \kappa_H(\varepsilon_{kk}^p) \] (4.16)

Starting from an initially elastic stress state (i.e., one for which \( f < 0 \)), if a stress increment results in a new stress state such that \( f < 0 \) then the behavior is elastic for that increment. If a stress state is on the current yield surface (i.e., \( f = 0 \)), then the behavior is determined by
\[ df = \frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} \begin{cases} > 0 & \text{plastic loading} \\ = 0 & \text{neutral loading} \\ < 0 & \text{elastic unloading} \end{cases} \] (4.17)

It is assumed that the total strain increment, \( d\varepsilon_{ij} \), can be divided into elastic and plastic components.

\[ d\varepsilon_{ij} = d\varepsilon_{ij}^e + d\varepsilon_{ij}^p \] (4.18)
The elastic strain increment is given by

\[ \varepsilon_{ij}^e = \frac{dI_1}{9K} \delta_{ij} + \frac{ds_{ij}}{2G} \]  \hspace{1cm} (4.19)

The variable elastic moduli, K and G, are restricted to the functional forms

\[ K = K(I_1, e_{ij}^p) \]  \hspace{1cm} (4.20)

\[ G = G(\sqrt{2}, e_{ij}^p) \]  \hspace{1cm} (4.21)

in order to assure path independency in the elastic range as shown by Chen and Saleeb [15]. The elastic strain increment can be decomposed into its spherical and deviatoric parts.

\[ \varepsilon_{ij}^e = \frac{1}{3} \varepsilon_{kk}^e \delta_{ij} + \varepsilon_{ij}^e \]  \hspace{1cm} (4.22)

From (4.19) it is seen that the spherical elastic strain increment is

\[ \varepsilon_{kk}^e = \frac{dI_1}{3K} \]  \hspace{1cm} (4.23)

and that the deviatoric elastic strain increment is

\[ \varepsilon_{ij}^e = \frac{ds_{ij}}{2G} \]  \hspace{1cm} (4.24)

By adopting an associated flow rule, the plastic strain increment tensor is given by

\[ \varepsilon_{ij}^p = \begin{cases} 
    d\lambda \frac{\partial f}{\partial \sigma_{ij}} & \text{if } f = 0 \text{ and } \frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} > 0 \\
    0 & \text{if } f < 0 \text{ or if } f = 0 \text{ and } \frac{\partial f}{\partial \sigma_{ij}} d\sigma_{ij} \leq 0 
\end{cases} \]  \hspace{1cm} (4.25)

where \( d\lambda \) is a scalar plastic multiplier. The plastic strain increment can also be decomposed.
\[ de_{ij}^p = \frac{1}{3} de_{kk}^p \delta_{ij} + de_{ij}^p \]  \hspace{1cm} (4.26)

To solve for the spherical and deviatoric plastic strain increments, use (4.1) to write (4.9) as

\[ f = f(\sigma_{kk}, s_{ij}, \kappa) \]  \hspace{1cm} (4.27)

Using the form of the loading function given in (4.27) and using the chain rule to differentiate (4.25) yields

\[ de_{ij}^p = d\lambda [ \frac{\partial f}{\partial \sigma_{kk}} \frac{\partial \sigma_{mn}}{\partial \sigma_{ij}} + \frac{\partial f}{\partial s_{mn}} \frac{\partial s_{mn}}{\partial \sigma_{ij}} + \frac{\partial f}{\partial \kappa} \frac{\partial \kappa}{\partial \sigma_{ij}} ] \]  \hspace{1cm} (4.28)

Observe that

\[ \frac{\partial \kappa}{\partial \sigma_{ij}} = 0 \]  \hspace{1cm} (4.29)

and that

\[ \frac{\partial \sigma_{mn}}{\partial \sigma_{ij}} = \delta_{im} \delta_{jn} = \delta_{ij} \]  \hspace{1cm} (4.30)

and that in light of (4.30) one can write

\[ \frac{\partial s_{mn}}{\partial \sigma_{ij}} = \partial \left[ \frac{\sigma_{mn} - \frac{1}{3} \sigma_{kk} \delta_{mn}}{\sigma_{ij}} \right] = \delta_{im} \delta_{jn} - \frac{1}{3} \delta_{ij} \delta_{mn} \]  \hspace{1cm} (4.31)

Using (4.29)-(4.31) and recalling that \( s_{mn} = 0 \), equation (4.28) becomes

\[ de_{ij}^p = d\lambda \left[ \frac{\partial f}{\partial \sigma_{kk}} \delta_{ij} + \frac{\partial f}{\partial s_{mn}} (\delta_{im} \delta_{jn} - \frac{1}{3} \delta_{ij} \delta_{mn}) \right] \]

\[ = d\lambda \left[ \frac{\partial f}{\partial \sigma_{kk}} \delta_{ij} + \frac{\partial f}{\partial s_{ij}} \frac{1}{3} \frac{\partial f}{\partial s_{mn}} \delta_{ij} \right] \]

\[ = d\lambda \left[ \frac{\partial f}{\partial \sigma_{kk}} \delta_{ij} + \frac{\partial f}{\partial s_{ij}} \right] \]  \hspace{1cm} (4.32)
Comparing (4.26) and (4.32) it is easily seen that the spherical plastic strain increment is given by

\[ de^p_{kk} = 3 \lambda \frac{\partial f}{\partial \sigma_{kk}} = 3 \lambda \frac{\partial f}{\partial I_1} \]  

(4.33)

and that the deviatoric plastic strain increment is given by

\[ de^p_{ij} = d\lambda \frac{\partial f}{\partial s_{ij}} \]  

(4.34)

Using the functional form of the loading function from (4.11), one can write

\[ \frac{\partial f}{\partial s_{ij}} = \frac{\partial f( I_1, \sqrt{J_2}, \kappa) }{ \partial s_{ij}} \]

\[ = \frac{\partial f}{\partial \sqrt{J_2}} \frac{\partial \sqrt{J_2}}{\partial s_{ij}} \]

\[ = \frac{\partial f}{\partial \sqrt{J_2}} \left[ \frac{1}{2 \sqrt{J_2}} \frac{\partial J_2}{\partial s_{ij}} \right] \]  

(4.35)

Now, observe that

\[ \frac{\partial J_2}{\partial s_{ij}} = \frac{1}{2} \frac{\partial (s_{mn} s_{mn})}{\partial s_{ij}} = \delta_{im} \delta_{jn} s_{mn} = s_{ij} \]  

(4.36)

Using (4.36) in (4.35) yields

\[ \frac{\partial f}{\partial s_{ij}} = \frac{1}{2 \sqrt{J_2}} \frac{\partial f}{\partial \sqrt{J_2}} s_{ij} \]  

(4.37)

And, finally, by substituting (4.33), (4.34) and (4.37) into (4.26), the plastic strain increment can be written as

\[ de^p_{ij} = d\lambda \left[ \frac{\partial f}{\partial I_1} \delta_{ij} + \frac{1}{2 \sqrt{J_2}} \frac{\partial f}{\partial \sqrt{J_2}} s_{ij} \right] \]  

(4.38)

Equation (4.38) is not useful unless the scalar plastic multiplier, $d\lambda$, can be determined. To achieve this task, one can invoke the consistency condition. The consistency
condition states that during plastic loading the stress state can never be outside the yield surface; consequently, the yield surface must harden (soften) or enlarge (shrink) such that \( df = 0 \) during plastic loading. Using the chain rule on (4.11)

\[
df = \frac{\partial f}{\partial I_1} dI_1 + \frac{\partial f}{\partial \sqrt{J_2}} d\sqrt{J_2} + \frac{\partial f}{\partial \kappa} d\kappa = 0 \tag{4.39}
\]

Observe that by use of (4.24)

\[
d\sqrt{J_2} = \frac{\partial \sqrt{J_2}}{\partial s_{ij}} ds_{ij} = \frac{1}{2 \sqrt{J_2}} s_{ij} ds_{ij} = \frac{G \, de_{ij}^P}{\sqrt{J_2}} s_{ij} \tag{4.40}
\]

And by use of (4.38)

\[
d\kappa = \frac{\partial \kappa}{\partial e_{ij}^P} de_{ij}^P = \lambda \frac{\partial \kappa}{\partial e_{ij}^P} \left[ \frac{\partial f}{\partial I_1} \delta_{ij} + \frac{1}{2 \sqrt{J_2}} \frac{\partial f}{\partial \sqrt{J_2}} s_{ij} \right] \tag{4.41}
\]

Recalling (4.23)

\[
dI_1 = 3K \, de_{kk}^P \tag{4.42}
\]

In view of (4.40)-(4.42), (4.39) becomes

\[
df = 3K \, de_{kk}^P \frac{\partial f}{\partial I_1} + \frac{G \, de_{ij}^P}{\sqrt{J_2}} \frac{\partial f}{\partial \sqrt{J_2}} s_{ij} + \lambda \frac{\partial \kappa}{\partial e_{ij}^P} \left[ \frac{\partial f}{\partial I_1} \delta_{ij} + \frac{1}{2 \sqrt{J_2}} \frac{\partial f}{\partial \sqrt{J_2}} s_{ij} \right] \tag{4.43}
\]

Now using (4.18) in (4.43)

\[
3K \frac{\partial f}{\partial I_1} (de_{kk}^P - de_{kk}^P) + \frac{G}{\sqrt{J_2}} \frac{\partial f}{\partial \sqrt{J_2}} s_{ij} (de_{ij}^P - de_{ij}^P)
\]

\[
= -\lambda \frac{\partial \kappa}{\partial e_{ij}^P} \frac{\partial \kappa}{\partial e_{ij}^P} \left[ \frac{\partial f}{\partial I_1} \delta_{ij} + \frac{1}{2 \sqrt{J_2}} \frac{\partial f}{\partial \sqrt{J_2}} s_{ij} \right] \tag{4.44}
\]

Now substitute for \( de_{kk}^P \) using (4.33) and for \( de_{ij}^P \) using (4.34) and (4.37) so that (4.44) becomes
\[ 3K \frac{\partial f}{\partial l_1} (\varepsilon_{kk} - 3 \lambda \frac{\partial f}{\partial l_1} + \frac{G}{\sqrt{j_2}} \frac{\partial f}{\partial \sqrt{j_2}} s_{ij} (\varepsilon_{ij} - \frac{d\lambda}{2 \sqrt{j_2}} \frac{\partial f}{\partial \sqrt{j_2}} s_{ij}) \]

\[ = -d\lambda \frac{\partial f}{\partial \kappa} \frac{\partial \kappa}{\partial \varepsilon_{ij}} \left[ \frac{\partial f}{\partial l_1} \delta_{ij} + \frac{1}{2 \sqrt{j_2}} \frac{\partial f}{\partial \sqrt{j_2}} s_{ij} \right] \]

Using (4.7) and rearranging

\[ 3K \frac{\partial f}{\partial l_1} d\varepsilon_{kk} + \frac{G}{\sqrt{j_2}} \frac{\partial f}{\partial \sqrt{j_2}} s_{ij} d\varepsilon_{ij} \]

\[ = d\lambda \left[ 9K \left( \frac{\partial f}{\partial l_1} \right)^2 + G \left( \frac{\partial f}{\partial \sqrt{j_2}} \right)^2 - \frac{\partial f}{\partial \kappa} \frac{\partial \kappa}{\partial \varepsilon_{ij}} \left( \frac{\partial f}{\partial l_1} \delta_{ij} + \frac{1}{2 \sqrt{j_2}} \frac{\partial f}{\partial \sqrt{j_2}} s_{ij} \right) \right] \quad (4.45) \]

And, finally, \( d\lambda \) can be found from (4.45)

\[ d\lambda = \frac{3K \frac{\partial f}{\partial l_1} d\varepsilon_{kk} + \frac{G}{\sqrt{j_2}} \frac{\partial f}{\partial \sqrt{j_2}} s_{mn} d\varepsilon_{mn}}{9K \left( \frac{\partial f}{\partial l_1} \right)^2 + G \left( \frac{\partial f}{\partial \sqrt{j_2}} \right)^2 - \frac{\partial f}{\partial \kappa} \frac{\partial \kappa}{\partial \varepsilon_{ij}} \left( \frac{\partial f}{\partial l_1} \delta_{ij} + \frac{1}{2 \sqrt{j_2}} \frac{\partial f}{\partial \sqrt{j_2}} s_{ij} \right)} \quad (4.46) \]

Now one is in the position to write constitutive equations which relate the incremental stresses and strains. By combining (4.18), (4.19), and (4.30), the total strain increment can be written as

\[ d\varepsilon_{ij} = \frac{d l_1}{9K} \delta_{ij} + \frac{d s_{ij}}{2G} + d\lambda \left[ \frac{\partial f}{\partial l_1} \delta_{ij} + \frac{1}{2 \sqrt{j_2}} \frac{\partial f}{\partial \sqrt{j_2}} s_{ij} \right] \quad (4.47) \]

By algebraic manipulation of (4.47) the other constitutive form can be found. First, solve (4.47) for \( ds_{ij} \)

\[ ds_{ij} = 2G d\varepsilon_{ij} - 2G \frac{d l_1}{9K} \delta_{ij} - 2G d\lambda \frac{\partial f}{\partial l_1} \delta_{ij} - \frac{G}{\sqrt{j_2}} \frac{\partial f}{\partial \sqrt{j_2}} s_{ij} \quad (4.48) \]

Combining (4.48) and the incremental form of (4.1)

\[ d\sigma_{ij} = \frac{d l_1}{3} \delta_{ij} + 2G d\varepsilon_{ij} - 2G \frac{d l_1}{9K} \delta_{ij} - 2G d\lambda \frac{\partial f}{\partial l_1} \delta_{ij} - d\lambda \frac{G}{\sqrt{j_2}} \frac{\partial f}{\partial \sqrt{j_2}} s_{ij} \quad (4.49) \]
And by substituting (4.23), (4.26) and (4.33) into (4.49) the desired constitutive form is achieved.

\[
d\sigma_{ij} = K e_{kk} \delta_{ij} + \frac{2}{3} G e_{kk} \delta_{ij} + 2 G e_{ij} - \frac{2G}{3} e_{kk} \delta_{ij} \\
- \frac{2G}{3} e_{kk} \delta_{ij} - d\lambda \frac{G}{\sqrt{J_2}} \frac{\partial f}{\partial \sqrt{J_2}} s_{ij} \\
= K e_{kk} \delta_{ij} + 2 G e_{ij} - d\lambda \frac{G}{\sqrt{J_2}} \frac{\partial f}{\partial \sqrt{J_2}} s_{ij} \\
= K e_{kk} \delta_{ij} + (K e_{kk} - K e_{kk}) \delta_{ij} + 2 G e_{ij} - d\lambda \frac{G}{\sqrt{J_2}} \frac{\partial f}{\partial \sqrt{J_2}} s_{ij} \\
= K e_{kk} \delta_{ij} + 2 G e_{ij} - d\lambda \left[ 3 K \frac{\partial f}{\partial I_1} + \frac{G}{\sqrt{J_2}} \frac{\partial f}{\partial \sqrt{J_2}} s_{ij} \right]
\]
V. NUMERICAL IMPLEMENTATION OF THE
CONSTITUTIVE MODEL

This chapter details the development of the constitutive model and demonstrates the behavior of the model for a tuffaceous rock. The implementation of a constitutive model greatly depends on whether the strain increments or the stress increments are known *a priori*. In the first section of the chapter, the form of the constitutive relation to be used is explained. The next three sections discuss the treatment of elastic, plastic failure surface, and plastic cap incremental behavior, respectively. The treatment of incremental behavior during which both yield surfaces are active, i.e. at the intersection of the surfaces or corner, is explained subsequently. The computer implementation of the material constitutive model and also a program that drives the model to simulate triaxial test conditions are also presented. Finally, these programs are used to fit the model to experimental data for Oya tuff from Ichikawa, *et al* [9].

5.1 Form of the Constitutive Model

The two forms of a constitutive model are represented by equations (4.47) and (4.50). For the first form, the incremental stresses are known, and the resulting strain increments must be found. In the second form, just the opposite is true; the incremental stress is expressed in terms of the incremental strains. The constitutive model of this thesis will be developed in this latter form. There are two reasons for selecting this approach. First, softening behavior can be included in this form of model, since even when softening, each strain increment corresponds to a unique stress increment. The converse is not true, since from a stress perspective identical stress increments exist which result in either elastic unloading strain response or in softening plastic loading...
strain response. Second, for implementation into finite element codes the strain controlled form is preferred. The nonlinear finite element code in which the constitutive model will be used is formulated such that incremental displacements are the primary unknowns. At each increment in the solution process, the incremental displacements and hence the strains are iteratively varied until equilibrium is satisfied; thus, it is quite natural to have the constitutive routine constructed such that it expects incremental strains as input quantities. The task of the constitutive model can now be explicitly stated.

**GIVEN:** The stress, $\sigma_{ij}^n$, at the end of the $n$th loading increment, the values of the hardening parameters, $\kappa_h^n$ and $\kappa_h^\alpha$, at the end of the $n$th loading increment, and the $(n+1)$th increment of strain, $de_{ij}^{n+1}$.

**FIND:** The resulting (updated) stress, $\sigma_{ij}^{n+1}$ at the end of the $(n+1)$th increment and the updated hardening parameters, $\kappa_h^{n+1}$ and $\kappa_h^\alpha^{n+1}$.

### 5.2 Elastic Strain Increments

Given a strain increment, $de_{ij}$, the first task at hand is to determine if all or a portion of the strain increment causes purely elastic behavior. It is initially assumed that all of the strain increment is elastic and using equations (4.23) and (4.24) a trial elastic stress state is calculated.

$$I_1^e = I_1^n + 3K^n de_{kk} \quad (5.1)$$

$$s_{ij}^e = s_{ij}^n + 2G^n de_{ij} \quad (5.2)$$

As before, the superscript $n$ indicates the value of the superscripted quantity at the end of the $n$th increment. Observing the restrictions of (4.20) and (4.21), forms for the elastic
moduli recommended by Chen and Baladi [28] are used.

\[
K^n = \frac{K_I}{1 - K_1} \left[ 1 - K_1 e^{-K_1 I^1} \right] \tag{5.3}
\]

\[
G^n = \frac{G_I}{1 - G_1} \left[ 1 - G_1 e^{-G_1 \sqrt{J_2^n}} \right] \tag{5.4}
\]

where \( K_I, K_1, K_2, G_I, G_1 \), and \( G_2 \) are the material elastic constants. The strain increment is totally elastic if the trial stress state as defined by equations (5.1) and (5.2) lies within or on the yield surface. Since there are two yield surfaces, the failure surface and the cap, a method for determining if either of the yield surfaces has been violated by the trial stress is needed. It turns out that it is also important for subsequent calculations to determine which yield surface is violated first by the trial stress trajectory. The procedure used is to compare in \( I_1, \sqrt{J_2} \) space the slopes of the trial stress trajectory and a line connecting the beginning stress state (at the end of the \( n \)th increment) to the intersection of the yield surfaces. As illustrated in Figure 5.1, the angle \( \theta^F \) is defined to be the angle between a vertical line and the line joining the initial stress point and the trial stress point. The angle \( \theta^c \) is analogously defined for a line to the corner where the yield surfaces intersect. When \( \theta^F < \theta^c \) the cap surface is checked for penetration; otherwise, the failure surface is checked. From equation (4.13) it is seen that the cap surface is violated when

\[
\sqrt{J_2''} > F(I_1'', \kappa_{11}'') \tag{5.5}
\]

From equation (4.12) the condition for penetration of the failure surface is

\[
\sqrt{J_2''} > Q(I_1'', \kappa_{11}'') \tag{5.6}
\]

When the trial stress does not violate a yield surface, then the updated stress is the trial stress, and the hardening parameters remain unchanged.
FIGURE 5.1 Angles used to determine which yield surface has been violated.
\[ s_{ij}^{n+1} = s_{ij}^r \]  (5.7)

\[ I_1^{n+1} = I_1^r \]  (5.8)

\[ \kappa_h^{n+1} = \kappa_h^n \]  (5.9)

\[ \kappa_{ij}^{n+1} = \kappa_{ij}^n \]  (5.10)

In cases where the trial stress does violate a yield surface, the stress point, \((I_1^{ys}, \sqrt{J_2^{ys}})\), at which the penetration occurs is found, and the stress state is moved elastically to the yield surface. The new stress state on the yield surface is given by

\[ s_{ij}^{ys} = (1 - r) s_{ij}^n + r s_{ij}^r \]  (5.11)

where \( r \) denotes the fraction of the strain increment which is purely elastic.

\[ r = \frac{\sqrt{J_2^{ys}} - \sqrt{J_2^n}}{\sqrt{I_1^r} - \sqrt{I_1^n}} \]  (5.12)

The portion of the strain increment which remains will result in elastic-plastic deformation. This elastic-plastic strain increment, \( de_{ij}^{sp} \), is found by reducing the original strain increment by the strain required to move the stress onto the yield surface.

\[ de_{ij}^{sp} = (1 - r) de_{ij} \]  (5.13)

In the subsequent sections which describe elastic-plastic deformation, it will be understood that the starting stress lies on a yield surface so that in those sections the initial stress is \( \sigma_{ij}^n = \sigma_{ij}^{ys} \) and the strain increment is \( de_{ij} = de_{ij}^{sp} \).

5.3 Failure Surface Strain Increments

The implementation procedure is hardly affected by the form of the yield surfaces.
In fact, as long as the same hardening parameters are used, it is a simple task to replace
the forms proposed here with other forms. (Other forms should, of course, also satisfy
the convexity requirement). The form of the yield surface does greatly affect the models
inelastic response. The proposed failure surface has been formulated to reproduce many
of the inelastic behaviors discussed in Chapter 2. In particular, the yielding is pressure
sensitive, the material hardens and then softens to a residual strength at low pressures,
and at high pressures the material hardens to a peak strength.

In accordance with (4.14) the failure surface hardening parameter is chosen to sim-
ply be the effective plastic strain.

$$\kappa_h = \bar{\varepsilon}^p$$  \hspace{1cm} (5.14)

Using the notation of (4.12) the failure envelope is defined as

$$Q(I_1, \bar{\varepsilon}^p) = c_1 - c_2 e^{-c_3 I_1} + c_4 \left[ e^{-c_5 \bar{\varepsilon}^p} - 1 \right] + c_6 \left[ 1 - e^{-c_7 \bar{\varepsilon}^p} \right] I_1$$  \hspace{1cm} (5.15)

where $c_1, c_2, \ldots, c_7$ are constants. There are three terms in the failure surface equation:
(1) the initial failure term, $c_1 - c_2 e^{-c_3 I_1}$, (2) the softening term, $c_4 \left[ e^{-c_5 \bar{\varepsilon}^p} - 1 \right]$, and (3)
the hardening term, $c_6 \left[ 1 - e^{-c_7 \bar{\varepsilon}^p} \right] I_1$. The initial failure term represents the shape of the
failure envelope before any hardening or softening has occurred. It can be shown that the
maximum amount of hydrostatic tension or minimum hydrostatic pressure, $p_{\text{min}}$, that the
material can support is given by

$$p_{\text{min}} = \frac{I_{1\text{min}}}{3} = -\frac{1}{3 c_3} \ln \frac{c_1}{c_2}$$  \hspace{1cm} (5.16)

It is also seen that as the hydrostatic pressure becomes large the "radius" of the initial
failure surface approaches a constant value of $c_1$. The presence of $I_1$ in the hardening
term allows the amount of hardening to increase with hydrostatic pressure, thus at high
pressures the hardening dominates the softening term and only hardening behavior
results. The construction of the hardening/softening rules such that at low pressures the material hardens to a peak strength and then softens to a residual strength is a significant accomplishment. If the volumetric strain behavior is going to be modeled realistically, the low pressure hardening behavior is required. Careful examination of Figures 2.3 and 2.4 reveals that the onset of dilation occurs at stress levels below peak strength. The existence of low pressure hardening (which results in plastic dilatancy) allows the model to exhibit this same characteristic.

The constitutive equation (4.50) is applicable to elastic-plastic strain increments, but it is not in a form convenient for numerical implementation. To use (4.50) one must calculate $d\lambda$, and from (4.46) it is seen that to find $d\lambda$ requires knowledge of the incremental plastic strains. Chen and Baladi [28] have devised a clever technique to alleviate this difficulty. They have derived a scalar equation which yields a solution satisfying both the normality condition and the consistency condition. The equation can be solved by iteration of a single scalar quantity. Their method as it applies to this model is developed below.

Writing (4.38) in terms of the failure surface function, $h$,

\[ de_{ij}^p = d\lambda \left[ \frac{1}{2\sqrt{J_2}} \frac{\partial h}{\partial\sqrt{J_2}} s_{ij} + \frac{\partial h}{\partial I_1} \delta_{ij} \right] \]  

(5.17)

Using $h$ as defined by (4.12), the partial derivatives of (5.17) can be written as

\[ \frac{\partial h}{\partial \sqrt{J_2}} = 1 \]  

(5.18)

\[ \frac{\partial h}{\partial I_1} = -\frac{\partial Q}{\partial I_1} \]  

(5.19)

Using (5.18) and (5.19), (5.17) becomes

\[ de_{ij}^p = d\lambda \left[ \frac{s_{ij}}{2\sqrt{J_2}} - \frac{\partial Q}{\partial I_1} \delta_{ij} \right] \]  

(5.20)
Now define
\[
a = - \left[ \frac{\partial Q}{\partial I_1} \right]_{I_1, \bar{y}^{n+1}}
\] (5.21)

Notice that the derivative in (5.21) is taken at the end of the (n+1)th increment, rewriting (5.20) so that it also reflects conditions at the end of the (n+1)th increment
\[
de^p_{ij} = d\lambda \left[ \frac{s_{ij}^{n+1}}{2 \sqrt{J_2^{n+1}}} + a \delta_{ij} \right]
\] (5.22)

Taking the spherical part of (5.22)
\[
de^p_{kk} = \delta_{ij} de^p_{ij} = 3 \, d\lambda \, a
\] (5.23)

Or, rearranging
\[
d\lambda = \frac{de^p_{kk}}{3 \, a}
\] (5.24)

Using (5.24) and the deviatoric part of (5.22)
\[
de^p_{ij} = d\lambda \frac{s_{ij}^{n+1}}{2 \sqrt{J_2^{n+1}}} = \frac{de^p_{kk}}{6 \, a \sqrt{J_2^{n+1}}} \frac{s_{ij}^{n+1}}{} \] (5.25)

From (4.18) and (4.24) one can write
\[
de^p_{ij} = de_{ij} - de_{ij}^p = \frac{ds_{ij}}{2 \, G} = \frac{s_{ij}^{n+1} - s_{ij}^n}{2 \, G}
\] (5.26)

Rearranging (5.26) and using (5.2), one gets
\[
s_{ij}^{n+1} + 2 \, G \, de_{ij}^p = s_{ij}^n + 2 \, G \, de_{ij} = s_{ij}^r
\] (5.27)

Substituting (5.25) into (5.27)
\[
\left[ 1 + \frac{G \, de^p_{kk}}{3 \, a \sqrt{J_2^{n+1}}} \right] s_{ij}^{n+1} = s_{ij}^r
\] (5.28)
Multiplying each side of (5.28) by itself

$$\left[ 1 + \frac{G \, d e^p_{k k}}{3 \, a \sqrt{J_2^{n+1}}} \right]^2 \dot{I}_{2}^{n+1} = s_{ij} \, s_{ij}^{tr} \quad (5.29)$$

Invoking the definition of $J_2$

$$\left[ 1 + \frac{G \, d e^p_{k k}}{3 \, a \sqrt{J_2^{n+1}}} \right]^2 \dot{J}_2^{n+1} = J_2^r \quad (5.30)$$

And, finally, taking the square root of both sides

$$\sqrt{J_2^{n+1}} + \frac{G \, d e^p_{k k}}{3 \, a} = \sqrt{J_2^r} \quad (5.31)$$

Equation (5.31) is the desired scalar equation. As will be shown subsequently, choosing a value of $d \bar{\varepsilon}^p$ allows one to determine corresponding values of $a$, $d e^p_{k k}$, $I_1^{n+1}$, and, hence, $\sqrt{J_2^{n+1}}$. If these values satisfy equation (5.31), the correct value of $d \bar{\varepsilon}^p$ was chosen, and so it is seen that the solution process requires iteration with different values of $d \bar{\varepsilon}^p$. The process of obtaining values of $a$, $d e^p_{k k}$, and $I_1^{n+1}$ corresponding to $d \bar{\varepsilon}^p$ is explained below.

Begin by squaring both sides of (5.22)

$$d e^p_{i j} \, d e^p_{i j} = (\lambda)^2 \left[ \frac{s_{i j} \, s_{i j}^{n+1}}{4 \, J_2^{n+1}} + 3 \, a^2 \right] = (\lambda)^2 \left[ \frac{1}{2} + 3 \, a^2 \right] \quad (5.32)$$

Using the definition of $d \bar{\varepsilon}^p$, equation (4.15), and equation (5.32) one can write

$$d \bar{\varepsilon}^p = \sqrt{\frac{2}{3}} \, (d e^p_{i j} \, d e^p_{i j})^{1/2} = d \lambda \, \sqrt{\frac{2}{3}} \, \sqrt{\frac{1}{2} + 3 \, a^2} = d \lambda \, \sqrt{\frac{1}{3} + 2 \, a^2} \quad (5.33)$$

Solving (5.33) for $d \lambda$

$$d \lambda = \frac{d \bar{\varepsilon}^p}{\sqrt{\frac{1}{3} + 2 \, a^2}} \quad (5.34)$$
Using (5.23) and (5.34), one finds an expression for $d e_{kk}^p$ in terms of $a$ and $d \bar{e}^p$.

$$d e_{kk}^p = 3 a \, d \bar{e} = \frac{3 a \, d \bar{e}^p}{\sqrt{\frac{1}{3} + 2 \, a^2}}$$  \hspace{1cm} (5.35)

Now, rewrite (4.23) as

$$d I_1 = I_1^{n+1} - I_1^n = 3 \, K \, d e_{kk}^e$$  \hspace{1cm} (5.36)

Substituting (5.1) in (5.36)

$$I_1^{n+1} = I_1^r + 3 \, K \, (d e_{kk}^e - d e_{kk}) = I_1^r - 3 \, K \, d e_{kk}^p$$  \hspace{1cm} (5.37)

And, using (5.35) in (5.37), one gets an expression relating $I_1^{n+1}$, $a$, and $d \bar{e}^p$.

$$I_1^{n+1} = I_1^r - \frac{9 \, K \, a \, d \bar{e}^p}{\sqrt{\frac{1}{3} + 2 \, a^2}}$$  \hspace{1cm} (5.38)

Since $a$ is a function of $I_1^{n+1}$, equation (5.38) is solved by iteration of $I_1^{n+1}$. Once $I_1^{n+1}$ and $a$ are found from (5.38), $d e_{kk}^p$ is readily found from (5.35), and $\sqrt{J_2^{-n+1}}$ is given by

$$\sqrt{J_2^{-n+1}} = Q \left[ I_1^{n+1}, (\bar{e}^p)^n + d \bar{e}^p \right]$$  \hspace{1cm} (5.39)

At this point all quantities for use in (5.31) are known. Once a value of $d \bar{e}^p$ that leads to satisfaction of (5.31) has been found, the only remaining task is to find the updated stresses knowing $I_1^{n+1}$ and $\sqrt{J_2^{-n+1}}$. This chore is facilitated by defining the ratio of the updated to trial second invariants of the deviatoric stress tensor.

$$R = \frac{\sqrt{J_2^{-n+1}}}{\sqrt{J_2^{-n}}}$$  \hspace{1cm} (5.40)

The updated deviatoric stresses are given simply by

$$s_{ij}^{n+1} = R \, s_{ij}^r$$  \hspace{1cm} (5.41)
The updated failure surface hardening parameter is

$$K_{h}^{n+1} = (\bar{\epsilon}^p)^{n+1} = (\bar{\epsilon}^p)^n + d\bar{\epsilon}^p$$  \hspace{1cm} (5.42)

### 5.4 Cap Surface Strain Increments

In accordance with equation (4.16), the hardening parameter for the cap is defined as

$$K_H = X(\epsilon_{kk}^p) = X_{beg} - \frac{1}{c_8} \ln \left[ 1 - \frac{\epsilon_{kk}^p}{c_9} \right]$$  \hspace{1cm} (5.43)

where $X_{beg}$, $c_8$ and $c_9$ are constants. The form of this hardening rule is such that it can emulate the type of nonlinear hydrostatic consolidation behavior shown in Figure 2.7. The shape of the cap is chosen to be elliptical in the $I_1, \sqrt{J_2}$ space. Using the notation of equation (4.13), the cap is defined by the function $F$ as

$$F(I_1, X) = r_{cap} \sqrt{X^2 - I_1^2}$$  \hspace{1cm} (5.44)

where $r_{cap}$ is the constant ratio of the semi-minor to semi-major axes of the ellipse as illustrated in Figure 5.2. From the figure it is seen that $X_{beg}$ denotes the beginning position of the cap before any hardening.

The solution process for the cap is similar to the process used on the failure surface. By re-definition of $a$ to

$$a = - \left[ \frac{\partial F}{\partial I_1} \right]_{(I_1, X)^n}$$  \hspace{1cm} (5.45)

the previous scalar equation (5.31) applies to the cap surface. In this case, successive guesses for $d\epsilon_{kk}^p$ are made in order to satisfy (5.31). The process is straightforward. Knowing a value of $d\epsilon_{kk}^p$ one calculates the corresponding value of $X^{n+1}$ from (5.43). The
value of \( I_1^{n+1} \) is found from (5.37), then \( a \) can be calculated from (5.45). Finally, \( \sqrt{J_2^{n+1}} \) is given by

\[
\sqrt{J_2^{n+1}} = F(I_1^{n+1}, X^{n+1})
\]  

(5.46)

When equation (5.31) has been satisfied, the stresses are found using equations (5.40) and (5.41), and the updated hardening parameter is given by

\[
\kappa_H^{n+1} = X^{n+1}
\]  

(5.47)

5.5 Corner Treatment

The treatment of stress states at or near the corner, the intersection of the two yield surfaces, plays an important yet complicating role in the behavior of the model. At the corner, the yield surface has no unique normal; therefore, one cannot apply an associated flow rule. The method of corner calculation devised herein provides for a continuous response of the material for the whole range of loading increments around the corner. As discussed by Dougill [32] there three classes of stress increments which may occur at the corner: (1) elastic unloading in which neither loading surface is active, (2) loading in which only one yield surface is active, and (3) loading in which both surfaces are active. These three cases are illustrated in Figure 5.3. Case (1) provides no difficulty and is handled just like any other elastic increment. In treating a Case (2) increment, the method must assure that the correct yield surface is hardened. The surface to be hardened has already been determined using the trial elastic stress trajectory as described in Section 5.2. After either of the hardening processes as described in Sections 5.3 or 5.4 have been performed, the resulting stress state must be checked to see if the other yield surface is now violated and a Case (3) situation exists.

When Case (3) applies and both yield surfaces are to be hardened, the plastic strain
FIGURE 5.2 Geometry of the Cap

FIGURE 5.3 Types of Corner Stress Increments

\[ X - X_{\text{beg}} = -\frac{1}{c_{\text{g}}} \ln(1 - \frac{\varepsilon_p}{c_9}) \]
increment is divided into two subincrements. One subincrement of strain is used to harden the failure surface while the other subincrement hardens the cap.

\[ \text{de}^\text{p}_{ij} = (\text{de}^\text{p}_{ij})^\text{FAIL} + (\text{de}^\text{p}_{ij})^\text{CAP} \]  

(5.48)

The spherical and deviatoric parts of the plastic strain increment are independently split among the subincrements.

\[ (\text{de}^\text{p}_{ij})^\text{FAIL} = r_1 \frac{\text{de}^\text{p}_{kk}}{3} \delta_{ij} + r_2 \text{de}^\text{p}_{ij} \]  

(5.49)

\[ (\text{de}^\text{p}_{ij})^\text{CAP} = (1 - r_1) \frac{\text{de}^\text{p}_{kk}}{3} \delta_{ij} + (1 - r_2) \text{de}^\text{p}_{ij} \]  

(5.50)

where \( r_1 \) and \( r_2 \) are yet to be determined. When \((\text{de}^\text{p}_{ij})^\text{FAIL}\) is used to harden the failure surface an updated failure surface stress, \( I_1^\text{FAIL}, \sqrt{J_2^\text{FAIL}} \), is found. Likewise, the stress point resulting from hardening of the cap by \((\text{de}^\text{p}_{ij})^\text{CAP}\) is \((I_1^\text{CAP}, \sqrt{J_2^\text{CAP}})\). The strain splitting ratios, \( r_1 \) and \( r_2 \), must be chosen so that both yield surfaces predict the same stress point. This condition is written mathematically as

\[ \left[ I_1^\text{FAIL} - I_1^\text{CAP} \right]^2 + \left[ \sqrt{J_2^\text{FAIL}} - \sqrt{J_2^\text{CAP}} \right]^2 = 0 \]  

(5.51)

Satisfaction of (5.51) requires an iterative procedure that varies the values of \( r_1 \) and \( r_2 \). It is not practical to directly split the plastic strain increment as described because it is not known beforehand what proportion of the total strain increment is plastic and what proportion is elastic. An alternate splitting procedure that accomplishes the same goal is to let \( r_1' \) and \( r_2' \) be the yet unknown ratios of the spherical and deviatoric parts of the total strain increment on the failure surface to the corresponding parts of the total strain increment.

\[ (\text{de}^\text{p}_{ij})^\text{FAIL} = r_1' \frac{\text{de}^\text{p}_{kk}}{3} \delta_{ij} + r_2' \text{de}^\text{p}_{ij} \]  

(5.52)
Once the failure surface calculations have been performed, the elastic portion of $(\varepsilon_{ij}^{e_{\text{FAIL}}})$ can be found.

$$(\varepsilon_{ij}^{e_{\text{FAIL}}}) = \frac{I_1^{\text{FAIL}} - I_1^n}{9K} \delta_{ij} + \left[ \frac{\sqrt{J_2^{\text{FAIL}}}}{\sqrt{J_2^n}} - 1 \right] \frac{s_{ij}^n}{2G} \quad (5.53)$$

When equation (5.51) is satisfied the elastic strain increments for the failure and cap surfaces will be equal.

$$(\varepsilon_{ij}^{e_{\text{FAIL}}}) = (\varepsilon_{ij}^{e_{\text{CAP}}}) \quad (5.54)$$

Now the cap hardening calculations can be performed, using a cap strain increment defined by

$$(\varepsilon_{ij}^{e_{\text{CAP}}}) = (\varepsilon_{ij}^{e_{\text{FAIL}}}) + (1 - \nu_r') \frac{d\varepsilon_{kk}}{3} \delta_{ij} + (1 - \nu_r') \varepsilon_{ij} \quad (5.55)$$

When equation (5.51) is satisfied, equations (5.52) and (5.55) define strain subincrements which satisfy (5.49). The solution process now involves iteration of $\nu_r'$ and $r_2'$ in order to satisfy equation (5.51). One way to find $\nu_r'$ and $r_2'$ is to treat equation (5.51) as a minimization problem by defining the function $M(r_1', r_2')$ as

$$M(r_1', r_2') = \left[ I_1^{\text{FAIL}} - I_1^{\text{CAP}} \right]^2 + \left[ \sqrt{J_2^{\text{FAIL}}} - \sqrt{J_2^{\text{CAP}}} \right]^2 \quad (5.56)$$

and optimizing the function to its known minimum of zero.

Figure 5.4 illustrates the effects of the corner treatment on the volumetric strain behavior for a conventional triaxial loading path. The curve labeled "failure surface" shows the behavior of the model without a cap, while the "cap surface" curve gives behavior without a failure surface. As can be seen from the figure, the corner treatment has the desired effect of eliminating the reversal of the volumetric strain towards dilation that occurs when only the failure surface hardening is considered.
FIGURE 5.4 Effect of Corner Treatment on Volumetric Strain Response
5.6 Computer Programs

A computer program has been written to perform the calculations described in sections 5.2 - 5.5. The source code is written in the FORTRAN 77 programming language and was compiled and run on a Celerity 4140 L computer under the UNIX operating system. A listing of these material model subroutines is given in Appendix A1. A stress state driver program has also been written. This interactive program allows one to drive the material model subroutines through a variety of stress paths including conventional triaxial and hydrostatic loading paths. The program provides output both to the terminal screen and to a dataset for further processing. A listing of this triaxial driver program is given in Appendix A2. Both of the programs are generously documented with comments and, as such, should be easy to understand and modify.

5.6.1 Material Model Subroutines

The main subroutine of the material model is called "matlaw". This subroutine accepts as input arguments the strain increment, the current stress, and the current hardening parameters, $\vec{\varepsilon}^p$ and $X$. The updated stress state and the updated hardening parameters are returned by the subroutine. Elastic increments and elastic moves to a yield surface are also handled in this subroutine.

The failure surface hardening calculations are performed in subroutine "fails". In this subroutine equation (5.31) is solved by a procedure which alternates between bisection and the method of Regula Falsi. This procedure was employed in hopes of speeding the convergence process. Subroutine "fails" in turn calls subroutine "findi1" which uses bisection to solve equation (5.38).
Subroutine "caps" performs the cap hardening calculations. It uses bisection to solve equation (5.31). Subroutine "calff" which performs many repetitive calculations is called by "caps".

For increments at the corner, the minimization of equation (5.56) is accomplished by the simplex search method. The coding for the simplex algorithm is taken from Olsson [42]. Subroutines "simplex" and "search" perform the simplex procedure. Subroutine "fn" finds the value of the function M for given values of r_1 and r_2 by splitting the strain increment according to equations (5.52) and (5.55) and, subsequently, calling subroutines "fails" and "caps". Since the corner solution requires many calls to the hardening routines (typically about 90), it requires a measurable amount of computer time.

5.6.2 Triaxial Driver Subroutines

The idea of a program that drives a constitutive model so that the behavior of the model for simple stress-strain paths can be tested came from Chen and Baladi [28]. The structure of this program is quite similar to a driver program they have written. In order to provide flexibility, the model parameters to be used are read from a user specified dataset by subroutine "datain". An example of a typical input dataset is shown in Appendix A3. Once the material parameters have been read, the user is given the following options: (1) apply hydrostatic loading, (2) apply triaxial loading, and (3) reset all variables.

When hydrostatic loading is selected, the user is further prompted to enter the target pressure and the size of a strain increment. If the strain increment is positive then loading occurs; otherwise, unloading occurs. Driving the material in hydrostatic paths is simple. The principal strains, e_x, e_y and e_z, are incremented by the specified amount, all other
strains are set to zero, and the material model subroutines are called. When nearing the desired final target pressure, the strain increment is reduced as necessary to prevent overshoot of the target pressure.

Driving the material model through a triaxial path is more complicated. The user is prompted for an axial or z direction strain increment size and the slope of the desired stress path in $I_1, \sqrt{I_2}$ space. For a conventional triaxial test where the lateral stresses, $\sigma_x$ and $\sigma_y$, are held constant, the slope is $1/\sqrt{3}$. For a constant hydrostatic pressure triaxial test, the slope is infinity and can be approximated by a large number, say 200 or greater. Once the axial strain increment is known, the problem is to find a corresponding lateral strain increment which yields the desired lateral stress as defined the stress slope. Subroutine "lateral" uses a combination bisection/Regula Falsi method to find the correct lateral strain increment. Since this process requires iterative calls to the material model subroutines, it can be time consuming especially when the model's corner treatment is required. The triaxial phase halts when a user specified axial stress or axial strain has been achieved. The format of the output is the same for both hydrostatic and triaxial loading. For each strain increment, the resulting principal stresses, $\sigma_x, \sigma_y$ and $\sigma_z$, the total principal strains, $\varepsilon_x, \varepsilon_y$ and $\varepsilon_z$, and the updated hardening parameters, $\bar{\varepsilon}^p$ and $\chi$, are printed along with an indication of the type of treatment required for the increment (i.e. elastic, failure surface, cap or corner).

5.7 Demonstration of Model Performance

In order to demonstrate the behavior of the model, the model parameters have been fitted to the experimental data of Ichikawa, et al [9] for Oya Tuff. The determination of the failure surface parameters is facilitated by examining three cases: (1) initial yield, (2) residual strength, and (3) peak strength. In the triaxial test data, the initial yield was
identified as the point at which the volumetric strain response became nonlinear. The initial failure surface position of the model is governed by the parameters \( c_1, c_2 \) and \( c_3 \) (see equation 5.15). Figure 5.5 shows the experimentally determined initial yield points along with the initial failure surface of the model in \( I_1, \sqrt{J_2} \) space. The values of the initial failure surface parameters used were

\[
\begin{align*}
  c_1 &= 1150 \\
  c_2 &= 1000 \\
  c_3 &= 0.0006
\end{align*}
\]

The strength of the material after large axial strain is referred to as the residual strength. For large strains \( \bar{\varepsilon}^p \) will be large and, in the limit, the failure surface equation (5.15) becomes

\[
\mathcal{Q}(I_1, \infty) = c_1 - c_2 e^{-c_3 \bar{\varepsilon}^p} - c_4 + c_6 I_1
\]

Since \( c_1, c_2 \) and \( c_3 \) have already been determined, the residual strength data can be used to determine \( c_4 \) and \( c_6 \). Figure 5.6 illustrates the agreement between the residual strength data and the model for parameter values of

\[
\begin{align*}
  c_4 &= 280 \\
  c_6 &= 0.17
\end{align*}
\]

For hydrostatic pressures at which the material softens, the peak strength will be greater than the residual strength. One can differentiate the failure surface equation (5.15) with respect to \( \bar{\varepsilon}^p \) in order to find the critical value of \( \bar{\varepsilon}^p \) at which peak strength occurs.

\[
\frac{\partial \mathcal{Q}}{\partial \bar{\varepsilon}^p} = -c_4 c_5 e^{-c_3 \bar{\varepsilon}^p} + c_6 c_7 I_1 e^{-c_3 \bar{\varepsilon}^p} = 0
\]

\[
\bar{\varepsilon}^p_{critical} = \frac{1}{c_5 - c_7} \ln \left[ \frac{c_4 c_5 I_1}{c_6 c_7} \right]
\]
In theory, $c_5$ and $c_7$ can now be found by fitting the peak strength data. It has been found that there exists several combinations of $c_5$ and $c_7$ which appear to all fit the peak strength data equally well. Since these two parameters control the rate of hardening with plastic strain, they have a large effect on the magnitude and position of the maximum strength peak in an axial stress-strain plot. The values of

\[
\begin{align*}
    c_5 & = 93 \\
    c_7 & = 294
\end{align*}
\]

were chosen to provide a good fit to the axial stress-strain data. Figure 5.7 shows the shape of the yield surface at peak strength along with the peak strength data. In Figure 5.8 all three of the failure surface positions, initial, peak, and residual, are plotted.

The cap hardening rule parameters, $X_{beg}$, $c_8$ and $c_9$, can be determined from an isotropic consolidation test. The cap shape parameter, $r_{cap}$, has a large influence on the model's volumetric behavior in conventional triaxial tests and can be determined using trial-and-error to fit the volumetric data. Unfortunately, isotropic consolidation data for Oya Tuff was not available. The cap parameter values of

\[
\begin{align*}
    r_{cap} & = 0.6 \\
    X_{beg} & = 8000 \\
    c_8 & = 0.00002 \\
    c_9 & = 0.15
\end{align*}
\]

were chosen only on the basis of their ability to provide volumetric response similar to the data. Axial stress-strain curves for various confining pressures produced by the model are presented in Figure 5.9. For comparison the model and the experimental axial stress-strain curves are plotted together in Figure 5.9. The volumetric strain corresponding to the triaxial tests of the previous figure are shown in Figure 5.10. Experimental and model volumetric strain are both shown in Figure 5.11. The ability of the model to simulate the character and magnitude of the axial response should be noted. It is also
seen that the character of the volumetric response has been captured; however, the model predicts too much low pressure dilation.
FIGURE 5.5 Failure Surface at Initial Yield

Failure Surface Parameters

- \( c_1 = 1150 \)
- \( c_2 = 1000 \)
- \( c_3 = 0.0006 \)
- \( c_4 = 280 \)
- \( c_5 = 93 \)
- \( c_6 = 0.17 \)
- \( c_7 = 294 \)

Data from Ichikawa, et al [9]
Failure Surface Parameters

\[ c_1 = 1150 \]
\[ c_2 = 1000 \]
\[ c_3 = 0.0006 \]
\[ c_4 = 280 \]
\[ c_5 = 93 \]
\[ c_6 = 0.17 \]
\[ c_7 = 294 \]

\( \sqrt{I_2} \) (psi) vs. \( I_1 \) (psi)

- Data from Ichikawa, et al [9]

**FIGURE 5.7** Failure Surface at Peak Strength
Failure Surface Parameters

\[ c_1 = 1150 \]
\[ c_2 = 1000 \]
\[ c_3 = 0.0006 \]
\[ c_4 = 280 \]
\[ c_5 = 93 \]
\[ c_6 = 0.17 \]
\[ c_7 = 294 \]

**FIGURE 5.8 The Extreme Positions of the Failure Surface**
FIGURE 5.9 Axial Response Predicted by the Model
FIGURE 5.10 Experimental and Model Axial Behavior
\[ \sigma_3 = 71 \text{ psi} \]

**FIGURE 5.11 Volumetric Response Predicted by the Model**
VI. IMPLEMENTATION OF THE MODEL INTO NIKE2D

In order to use the constitutive model that has been developed to solve a particular initial/boundary value solid mechanics problem one must employ a numerical tool such as finite differences, finite elements, or boundary elements. The finite element method is the most widely used approximation technique in the solid mechanics field. This chapter describes the process of installing the developed model into a pre-existing finite element program. NIKE2D [43] is a two dimensional or axisymmetric, implicit, non-linear solid mechanics finite element program developed at Lawrence Livermore National Laboratory. The program is constructed such that the user may choose from many material constitutive models. The implementation of another material model into the program requires not only model subroutines as developed in Chapter 5 but also code to form a tangent constitutive matrix. The required constitutive matrix is referred to as "tangent" since it relates incremental rather than total stresses and strains. The form of the finite element equations and the role of the tangent constitutive matrix is explained in section 6.1. The sections following describe the processes of calculating the tangent constitutive matrix, and the final section of the chapter gives details of the modifications made to NIKE2D.

6.1 Finite Element Procedure

A complete development of the finite element method need not be presented here as there are many texts on the subject. Only that portion of the finite element procedure that illustrates the role of the constitutive model and the tangent constitutive matrix in the solution process will be considered. Following the derivation of Belytschko [44], the basic expressions needed for a finite element solution can be obtained by starting with a
statement of equilibrium.

\[ \sigma_{ij,j} = -b_i \]  \hspace{1cm} (6.1)

where \( \sigma_{ij,j} \) is the gradient of the stress tensor and \( b_i \) is the body force per unit volume. Consider a solid continuum \( \Omega \) with a boundary \( \Gamma \). The boundary conditions are specified such that

\[ u_i = u^*_i \quad \text{on } \Gamma_u \]  \hspace{1cm} (6.2)

and

\[ \sigma_{ij} n_j = t^*_i \quad \text{on } \Gamma_t \]  \hspace{1cm} (6.3)

where \( u_i \) is the displacement field and \( n_j \) is the outward normal to the boundary \( \Gamma \). Let \( v \) be a set of test functions such that \( v = 0 \) on \( \Gamma_u \), then the weak form of equation (6.1) is given by

\[ \int_{\Omega} v_{ij} \sigma_{ij} \, d\Omega = -\int_{\Omega} v_i b_i \, d\Omega + \int_{\Gamma_t} v_i t_i \, d\Gamma \]  \hspace{1cm} (6.4)

Using the usual finite element discretization procedure, the solid volume \( \Omega \) is subdivided into elements \( \Omega_e \), and the displacement field in each element is approximated by shape functions \( N_f \) such that

\[ u_i = N_f u_{il} \]  \hspace{1cm} (6.5)

where \( u_{il} \) are the nodal displacements. The strain-displacement operator is defined by

\[ B_{ji} = \frac{\partial N_f}{\partial x_j} \]  \hspace{1cm} (6.6)

where \( x_j \) are the space coordinates. The gradient of the displacement field is then given by

\[ u_{i,j} = B_{ji} u_{il} \]  \hspace{1cm} (6.7)
The test functions are approximated using the shape functions such that

\[ v_i = N_i \nu_U \]  
\[ v_{i,j} = B_{ji} \nu_U \]  

Using equations (6.8) and (6.9) in equation (6.4) and moving the nodal values of the test functions, \( \nu_U \), outside of the integrals yields

\[ \nu_U \left[ \int_{\Omega} B_{ji} \sigma_{ij} d\Omega - \int_{\Omega} N_i b_i d\Omega - \int_{\Gamma} \gamma_i d\Gamma \right] = 0 \]  

Since equation (6.10) must hold for arbitrary \( \nu_U \), it can be deduced that

\[ \int_{\Omega} B_{ji} \sigma_{ij} d\Omega - \int_{\Omega} N_i b_i d\Omega - \int_{\Gamma} \gamma_i d\Gamma = 0 \]  

Equation (6.11) is the basic integral equation of the finite element method for solid mechanics applications. Using the typical matrix representation, the equation is written as

\[ \int_{\Omega} [B]^T \{ \sigma \} d\Omega - \int_{\Omega} [N]^T \{ b \} d\Omega - \{ f \} = \{ 0 \} \]  

where \( \{ f \} \) is the external loads vector, and the first integral of equation (6.12) is called the stress divergence vector. In order to simplify the remaining equations, the body force term, \( \int_{\Omega} [N]^T \{ b \} d\Omega \), is dropped since it does not play an important role in the incremental solution scheme.

\[ \int_{\Omega} [B]^T \{ \sigma \} d\Omega - \{ f \} = \{ 0 \} \]  

If the stresses, \( \{ \sigma^n \} \), and displacements, \( \{ u^n \} \), at the end of the \( n \)th load increment are known then an incremental form of equation (6.13) can be used to approximate the displacements after the \( (n+1) \)st load increment.

\[ \int_{\Omega} [B]^T \{ d\sigma \} d\Omega = \{ f^{n+1} \} - \int_{\Omega} [B]^T \{ \sigma^n \} d\Omega \]
The tangent constitutive matrix, \([C(\sigma, \varepsilon^p)]\), which is a function of the stress tensor, \(\sigma\), and the plastic strain tensor, \(\varepsilon^p\), relates the incremental stresses and strains.

\[
\{d \sigma\} = [C(\sigma, \varepsilon^p)] \{d \varepsilon\}
\]  \hspace{1cm} (6.15)

Substituting equation (6.15) into (6.14)

\[
\int_{\Omega} [B]^T [C(\sigma, \varepsilon^p)] \{d \varepsilon\} d\Omega = \{f^{n+1}\} - \int_{\Omega} [B]^T \{\sigma^n\} d\Omega
\]  \hspace{1cm} (6.16)

Using the strain-displacement matrix to relate the incremental strains to incremental nodal displacements, i.e. \(\{d \varepsilon\} = [B] \{d u\}\), the previous equation becomes

\[
\int_{\Omega} [B]^T [C(\sigma, \varepsilon^p)] [B] d\Omega \{d u\} = \{f^{n+1}\} - \int_{\Omega} [B]^T \{\sigma^n\} d\Omega
\]  \hspace{1cm} (6.17)

The first integral in equation (6.17) is the so called tangent stiffness matrix, \([K_t(\sigma, \varepsilon^p)]\).

Inverting the tangent stiffness matrix yields the desired approximation for the incremental displacements.

\[
\{d u\} = [K_t(\sigma, \varepsilon^p)]^{-1} \left( \{f^{n+1}\} - \int_{\Omega} [B]^T \{\sigma^n\} d\Omega \right)
\]  \hspace{1cm} (6.18)

This first approximation of the incremental displacements will generally not yield stresses which satisfy the basic equilibrium statement of equation (6.13), hence equilibrium iterations are required. Denoting the iteration number with a superscript \(j\), the strain increment calculated from (6.18) is written as \(\{d u^j\}\). The incremental strains corresponding to \(\{d u^j\}\) are given by

\[
\{d \varepsilon^j\} = [B] \{d u^j\}
\]  \hspace{1cm} (6.19)

An accurate stress increment is found by inputting \(\{d \varepsilon^j\}\) to the constitutive routine to yield \(\{d \sigma^j\}\) and \(\{d \varepsilon^p^j\}\). The stress, displacement, and plastic strain updating is simply

\[
\{\sigma^j\} = \{\sigma^{j-1}\} + \{d \sigma^j\}
\]  \hspace{1cm} (6.20)
\[ \{ u^j \} = \{ u^{j-1} \} + \{ du^j \} \quad (6.21) \]
\[ \{ e^p \}^j = \{ e^p \}^{j-1} + \{ d e^p \}^j \quad (6.22) \]

To test if equation (6.12) has been satisfied to some tolerance, the residual force vector is defined as

\[ \{ \psi^j \} = \{ f^{n+1} \} - \int_{\Omega} \{ [B]^T \{ \sigma \} \} \, d\Omega \quad (6.23) \]

If the magnitude of the residual force vector, \( \sqrt{\{ \psi^j \}^T \{ \psi^j \}} \), is less than some specified tolerance, the equilibrium condition is considered to be satisfied. If equilibrium is not achieved then iterations continue with a new displacement increment given by

\[ \{ du^{j+1} \} = [K_t(\sigma, e^p)]^{-1} \{ \psi^j \} \quad (6.24) \]

If after any iteration the magnitude of the residual force vector has increased, the process is beginning to diverge. When the solution starts to diverge, the tangent stiffness matrix is reformed using the most current stress and plastic strain, i.e. \( [K_t] = [K_t(\sigma, e^p)^j] \).

### 6.2 Elastic Constitutive Matrix

For the two-dimensional or axisymmetric case the stress and strain vectors at a point each consists of four components.

\[ \{ \sigma \} = \begin{cases} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \end{cases} \quad (6.25) \]

\[ \{ \varepsilon \} = \begin{cases} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \end{cases} \quad (6.26) \]
In terms of the elastic moduli, the elastic constitutive matrix is given by [15]

\[
[D] = \begin{bmatrix}
(K + \frac{4}{3} G) & (K - \frac{2}{3} G) & (K - \frac{2}{3} G) & 0 \\
(K - \frac{2}{3} G) & (K + \frac{4}{3} G) & (K - \frac{2}{3} G) & 0 \\
(K - \frac{2}{3} G) & (K - \frac{2}{3} G) & (K + \frac{4}{3} G) & 0 \\
0 & 0 & 0 & G \\
\end{bmatrix}
\]  

(6.27)

6.3 Elastic-Plastic Tangent Constitutive Matrix

Chen and Baladi [28] have the detailed the derivation of a general elastic-plastic tangent constitutive matrix for a hardening material. The fourth order tensorial form of the matrix relates the stress and strain increment tensors.

\[
d\sigma_{ij} = C_{ijkl} d\epsilon_{kl}
\]

(6.28)

Components of the tensor are given by

\[
C_{ijkl} = D_{ijkl} - (1 - r) \frac{\partial f}{\partial \sigma_{uv}} M_{kl} \frac{\partial f}{\partial \sigma_{mn}} D_{mnqr} \frac{\partial f}{\partial \kappa} \frac{\partial f}{\partial \epsilon_{pq}^{p}} \frac{\partial f}{\partial \sigma_{mn}}
\]

(6.29)

where \( f \) is the yield function, \( \kappa \) is the hardening parameter, \( r \) is the ratio of the strain increment that is purely elastic, and \( D_{ijkl} \) is the elastic constitutive matrix given by

\[
D_{ijkl} = (K - \frac{2}{3} G) \delta_{ij} \delta_{kl} + G (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk})
\]

(6.30)

and

\[
M_{kl} = \frac{\partial f}{\partial \sigma_{mn}} D_{mnkl}
\]

(6.31)
\[
\frac{\partial f}{\partial \sigma_{ij}} = \frac{\partial f}{\partial I_1} \delta_{ij} + \frac{1}{2 \sqrt{J_2}} \frac{\partial f}{\partial J_2} s_{ij} \tag{6.32}
\]

### 6.3.1 Failure Surface Constitutive Matrix

The failure surface yield function is given by equations (4.12) and (5.15). Using these equations, the terms of equation (6.29) peculiar to failure surface strain increments are found to be

\[
\frac{\partial f}{\sigma_{ij}} = -\left\{ c_2 c_3 e^{-c_3 I_1} + c_6 \left[ 1 - e^{-c_7 \bar{e}^p} \right] \right\} \delta_{ij} + \frac{1}{2 \sqrt{J_2}} s_{ij} \tag{6.33}
\]

\[
\frac{\partial f}{\partial \kappa_H} = \frac{\partial f}{\partial \bar{e}^p} = c_4 c_5 e^{-c_4 \bar{e}^p} - c_6 c_7 e^{-c_7 \bar{e}^p} I_1 \tag{6.34}
\]

\[
\frac{\partial \kappa_H}{\partial e_{ij}^p} = \frac{\partial \bar{e}^p}{\partial e_{ij}^p} = \frac{2}{3} \bar{e}^p \tag{6.35}
\]

### 6.3.2 Cap Surface Constitutive Matrix

The cap surface yield function is given by equations (4.13) and (5.44). Using these equations, the terms of equation (6.29) peculiar to cap surface strain increments are found to be

\[
\frac{\partial f}{\partial \sigma_{ij}} = \frac{r_{\text{cap}}}{\sqrt{X^2 - I_1^2}} \delta_{ij} + \frac{1}{2 \sqrt{J_2}} s_{ij} \tag{6.36}
\]

\[
\frac{\partial f}{\partial \kappa_H} = \frac{\partial f}{\partial X} = -\frac{r_{\text{cap}} X}{\sqrt{X^2 - I_1^2}} \tag{6.37}
\]

\[
\frac{\partial \kappa_H}{\partial e_{ij}^p} = \frac{\partial X}{\partial e_{ij}^p} = \frac{\delta_{ij}}{c_8 (c_9 - e_{\kappa}^p)} \tag{6.38}
\]
By rearranging equation (5.43), the value of $\varepsilon_{kk}^p$ to be used in (6.38) is given by

$$\varepsilon_{kk}^p = c_0 \left[ 1 - e^{-c_9 (X - X_{lep})} \right]$$

(6.39)

6.4 Formation of Tangent Constitutive Matrix for Corner Increments

A special procedure is required to form the constitutive matrix when the stress state has reached the corner. Since the transition from one yield surface to another is not smooth, the derivatives needed to apply equation (6.29) are undefined. One way to obtain a reasonably accurate tangent constitutive matrix is through the use of the constitutive model itself. The corner treatment (Section 5.5) has been devised so that the stress response is continuous. By probing the stress response with four different strain increments, the components of the constitutive matrix can be determined numerically. The four strain increments are chosen to be small in magnitude and to have directions in strain space which do not vary greatly from the previous strain increment which caused the stress state to arrive at the corner. The first of the four strain increments is denoted as

$$[d\varepsilon_1] = \begin{bmatrix}
d\varepsilon_{x_1} \\
d\varepsilon_{y_1} \\
d\varepsilon_{z_1} \\
d\gamma_{x_1y_1}
\end{bmatrix}$$

(6.40)

The stress increment resulting from this strain increment is denoted as

$$[d\sigma_1] = \begin{bmatrix}
d\sigma_{x_1} \\
d\sigma_{y_1} \\
d\sigma_{z_1} \\
d\tau_{x_1y_1}
\end{bmatrix}$$

(6.41)

The other three increments are correspondingly denoted. The strain increment
components can be written as a square coefficient matrix.

\[
[E] = \begin{bmatrix}
de_{x_1} & de_{y_1} & de_{z_1} & d\tau_{x_1y_1} \\
de_{x_2} & de_{y_2} & de_{z_2} & d\tau_{x_2y_2} \\
de_{x_3} & de_{y_3} & de_{z_3} & d\tau_{x_3y_3} \\
de_{x_4} & de_{y_4} & de_{z_4} & d\tau_{x_4y_4}
\end{bmatrix}
\] (6.42)

In addition to the restrictions of magnitude and direction imposed on the strain increments, the increments are also constructed such that the matrix \([E]\) is diagonally dominant. After inversion of \([E]\), the components of the constitutive matrix are given by

\[
\begin{bmatrix}
c_{11} \\
c_{12} \\
c_{13} \\
c_{14}
\end{bmatrix} = [E]^{-1} \begin{bmatrix}
d\sigma_{x_1} \\
d\sigma_{x_2} \\
d\sigma_{x_3} \\
d\sigma_{x_4}
\end{bmatrix}
\] (6.43)

\[
\begin{bmatrix}
c_{21} \\
c_{22} \\
c_{23} \\
c_{24}
\end{bmatrix} = [E]^{-1} \begin{bmatrix}
d\sigma_{y_1} \\
d\sigma_{y_2} \\
d\sigma_{y_3} \\
d\sigma_{y_4}
\end{bmatrix}
\] (6.44)

\[
\begin{bmatrix}
c_{31} \\
c_{32} \\
c_{33} \\
c_{34}
\end{bmatrix} = [E]^{-1} \begin{bmatrix}
d\sigma_{z_1} \\
d\sigma_{z_2} \\
d\sigma_{z_3} \\
d\sigma_{z_4}
\end{bmatrix}
\] (6.45)

\[
\begin{bmatrix}
c_{41} \\
c_{42} \\
c_{43} \\
c_{44}
\end{bmatrix} = [E]^{-1} \begin{bmatrix}
d\tau_{xy_1} \\
d\tau_{xy_2} \\
d\tau_{xy_3} \\
d\tau_{xy_4}
\end{bmatrix}
\] (6.46)
The foregoing procedure does not insure the symmetry of [C]. To restore symmetry, off-diagonal components are replaced with an average of the appropriate two components, i.e., \( c_{12} \) and \( c_{21} \) are replaced with the value \( (c_{12} + c_{21})/2 \).

The slowness in terms of computer speed of the constitutive model's corner treatment has been previously mentioned. Since this procedure for finding the constitutive matrix requires four calls to the model's corner treatment, it is also quite time consuming. It should be apparent by now that the inclusion of a sharp corner in the constitutive model not only creates procedural complications but also tremendously increases the computing time required.

6.5 Revised NIKE2D Computer Code

The December 1986 version of NIKE2D which was received from Lawrence Livermore National Laboratories consisted of FORTRAN code for a Vax-750 computer. This code was altered by this thesis author to be compatible with the FORTRAN-77 / Unix operating system which resides on the Rice Mechanical Engineering Department's Celebrity computer. It is not practical to include all of the revised code in this thesis as the code is over 30,000 lines and requires 600 pages to list.

In order to incorporate the rock material model developed herein into NIKE2D several existing subroutines were modified, and several new subroutines were added. The code for these modified and new subroutines is contained in Appendix A4. Of course, the material model subroutines (Appendix A1) developed in Chapter 5 must also be included with the revised version of NIKE2D. The existing NIKE2D subroutines which have been modified are MATIN, PRINTM, BLKMAX, SSLCS, GETSTR, and PRTSTR. In the listings of these subroutines, new or modified lines are flagged with the tag "rock" in columns 74 - 77. The new subroutines include S17MAIN, PREMAT,
CMATF, CMATC, SOLSYS, and S17OUT.

Subroutine PREMAT performs an important scaling operation. If the strain increments to be input to the material model are large, the material model will not return accurate stresses. Owen and Hinton [41] have suggested subdivision of the strain increment by a factor related to the amount that the elastic trial stress violates the yield surface. The stress state is then found by sequential application of these subincrements of strain. A less elegant scaling method has been adopted in PREMAT. The strain increment is divided into equal subincrements such that the magnitude of the largest component of the subincrements is less than some specified size.

Subroutines CMATF, CMATC and SOLSYS are associated with the formation of the tangent constitutive matrix. Subroutine CMATF forms the failure surface constitutive matrix, and subroutine CMATC forms the cap surface matrix. SOLSYS solves the systems of equations indicated in equations (6.43) - (6.46) which are used to form the constitutive matrix at the corner.
VII. INDENTATION SIMULATIONS

7.1 Problem Description

In order to test the finite element implementation of the model and to determine the utility of the model in solving practical boundary problems, simulations of axisymmetric indentation have been performed. The mesh was generated and the boundary conditions specified using the INGRID [45] mesh generation program. The input dataset used by INGRID to generate the NIKE2D input dataset is given is Appendix A5. The geometry of the mesh and imposed boundary conditions are illustrated in Figure 7.1. The mesh contains 638 nodes and 576 four node elements. After application of the displacement boundary constraints, there remain 1209 degrees of freedom in the problem. The portion of the mesh which represents the cylindrical rock specimen is constrained in the radial direction along the outside diameter and along the axis of symmetry. The rock is also constrained in the axial direction along the bottom of the specimen. A pressure "preload" of 400 psi is applied to the top of the rock specimen. This "preload" which causes the entire rock sample to begin the simulation in a state of compression is used to approximate in situ stresses that exist at the bottom of a pressurized borehole in the earth. The shape of the axisymmetric indenter is typical of tungsten carbide inserts which serve as the mechanical indenters on rock bits used in drilling oil and gas wells. The indenter is constrained in the radial direction along the axis of symmetry. A linearly increasing pressure load is applied on the top the indenter forcing it into the rock sample.

This indentation problem utilizes NIKE2D's slide-line capability for modeling contact. Although NIKE2D permits specification of a contact coefficient of friction, the frictionless contact option was chosen. The slide-lines which identify the surfaces of potential contact consist of "master" nodes on one of the surfaces and "slave" nodes on the
other. There is not an association between a particular slave and master node so that unlimited sliding within the contact region is permitted. Figure 7.2 shows the designation of master and slave nodes used in the simulations.

The first simulation was run using the rock parameters for Oya Tuff given in Chapter 5. A second simulation used the same parameters except that the cap was effectively removed from the model by setting $X_{beg}$ to a very large number.

### 7.2 Results and Discussion

The time required for each simulation was much greater than anticipated. The first run with the original Oya Tuff material was designed to attain an axial load of 236 pounds on the indenter in 20 loading increments. This run was terminated in the tenth loading increment at which time it had accumulated over 2000 minutes of CPU time. In an effort to speed the solution, the problem was rerun using a total of 50 load increments. It was thought that by reducing the load increment size that the number of stiffness matrix reformations and the number of iterations per load increment could be significantly reduced. Such was not the case. This attempt ran for 3700 CPU minutes to a final axial load of 71 pounds. As was expected, the slowness of the program is due to the corner treatment. This fact was clearly illustrated by the third run in which the cap surface and, hence, the corner were removed from the model. The capless simulation completed 50 load increments to a final axial load of 236 pounds in 950 CPU minutes.

Deformed shape, contours of effective plastic strain, contours of volumetric strain, and contours of the maximum principal stress for the original Oya Tuff model at an axial load of 71 pounds are shown in Figures 7.3 - 7.6. These graphic results were produced with the TAURUS [46] interactive post-processing program. Figures 7.7 - 7.10 show corresponding results for the capless model at 236 pounds axial load. It is interesting to
note the growth of the plastic zone by comparing Figures 7.4 and 7.8. At the 71 pound load, the plastic zone has extended to a radius of about 0.124 inch and a depth of 0.195 inch. The plastic zone enlarges to a radius of 0.206 inch and a depth of 0.363 inch at the 236 pound load. The consequence of the removal of the cap can be seen by comparing Figures 7.5 and 7.9. When the cap and hence the corner are present, the model's corner treatment produces compaction. The volumetric strain contours plotted in Figure 7.5 indicate that the rock immediately under the contact region in compacting. (NIKE2D uses the conventional sign convention of positive for tensile stresses and for extensional strains.) Without the cap, the failure surface alone produces dilation. As seen in Figure 7.9, the capless model has indeed predicted volumetric expansion of the rock around the contact area.
FIGURE 7.1 Mesh Geometry
* - Master Nodes
○ - Slave Nodes

FIGURE 7.2 Slide-Line Nodes
FIGURE 7.3 Deformed Configuration for Original Model
At 71 Pounds Axial Load
FIGURE 7.4 Effective Plastic Strain for Original Model
At 71 Pounds Axial Load
CONTOUR VALUES
A = -1.00E-02
B = 7.12E-03
C = -4.25E-03
D = -1.37E-03
E = 1.50E-03

CONTOURS OF MEAN STRAIN
(GREEN-ST. VENANT)
MIN = -0.171E-01 IN ELEMENT 1
MAX = 0.281E-02 IN ELEMENT 9

FIGURE 7.5 Volumetric Strain for Original Model
At 71 Pounds Axial Load
CONTOURS OF MAXIMUM PRINC STRESS

MIN = -0.557E+04 IN ELEMENT 552
MAX = 0.134E+04 IN ELEMENT 547

CONTOUR VALUES
A = -4.80E+03
B = -4.03E+03
C = -3.07E+03
D = -2.06E+03
E = -1.73E+03
F = -9.84E+02
G = -1.97E+02
H = 5.78E+02

FIGURE 7.6 Contours of Maximum Principal Stress for the Original Model at 71 Pounds Axial Force
FIGURE 7.7  Deformed Configuration for Capless Model
At 236 Pounds Axial Load
CONTOURS OF EFF. PLASTIC STRAIN
MIN = 0.000E+00 IN ELEMENT .576
MAX = 0.927E-01 IN ELEMENT 9

CONTOUR VALUES
A = 2.00E+00
B = 1.34E+02
C = 2.67E+02
D = 5.33E+02
E = 6.67E+02
G = 8.00E-02

FIGURE 7.8 Effective Plastic Strain for Capless Model
At 236 Pounds Axial Load
FIGURE 7.9 Volumetric Strain for Capless Model
At 236 Pounds Axial Load
FIGURE 7.10 Contours of Maximum Principal Stress for Capless Model at 236 Pounds Axial Load
VIII. Conclusions and Recommendations for Further Work

1. A general two surface strain-softening constitutive model which is based on classical plasticity has been developed, and computer code to perform the model calculations has been written. The low pressure softening on the failure surface was accomplished by inclusion of special softening and hardening terms in the failure surface equation. A method for treatment of the corner where the failure and cap surfaces intersect has been developed which assures continuous stress response for the whole range of corner strain increments.

2. The model is able to emulate the axial response of experimental triaxial compression tests both qualitatively and quantitatively. The model’s volumetric strain behavior in triaxial tests has the same form as experimental measurements; however, the model predicts an excessive amount of dilation at the lower confining pressures.

3. Procedures for the implementation of the model into a finite element program including forms of the tangent constitutive matrix have been detailed. The model was installed into the two-dimensional finite element program NIKE2D.

4. Finite element simulations of an axisymmetric indentation problem revealed that the model’s corner treatment was computationally too slow for practical use.

5. In order to overcome the excessive dilation predicted by the model at low pressures one of two procedures is recommended: (1) one could reshape the low pressure end of the failure surface to resemble the traditional Drucker-Prager cone which predicts less plastic dilation and, perhaps, use a tension cut-off to prevent too much hydrostatic tension, or (2) one could adopt a non-associated flow rule for the plastic strains.

6. Because of the slowness and computational complications associated with the sharp
corner at the intersection of the two yield surfaces, it is recommended that future models employ a smooth, continuous yield surface such as that of Faruque and Desai [30].

7. The model could be generalized account for dependence of the yield functions on the third invariant of the stress tensor. Such a generalization might require new iterative solution schemes.
IX. REFERENCES


APPENDIX
A1. MATERIAL MODEL SUBROUTINES

c******************************************************************************
c
.....Material Model Subroutine
c
.....Written by:  F. J. Claborn
c.....Last Updated:  July 27, 1987
c
.....Summary:   This model attempts to simulate many of the
defformational characteristics of rock including
(1) Pressure sensitive yield
(2) Brittle to ductile transition
   i.e. pressure sensitive hardening and softening
(3) Varying unloading moduli
(4) Dilative and compacting behaviour.

c
The hardening/softening failure surface was developed
by Claborn. The strain hardening cap was adapted
from Chen's cap model in "Soil Plasticity: Theory
and Implementation". The corner treatment was
developed by Claborn.

c.....Reference:  F. J. Claborn, "Development of a Strain Softening
Constitutive Model for Rock", Rice University

c******************************************************************************

subroutine matlaw(delps, depsz, depsy, depsxy, depsyz, depsxz, epbar,
1   X, sigx, sigy, sigz, sigxy, sigyz, sigxz, ep, rr)

implicit real*8 (a-h, o-z)
real*8 il, ill, ile, ilf, ilr, ill
dimension r(2), ep(6)

common/moduli/ aki, ak1, ak2, agi, ag1, ag2
common/failc/ c1, c2, c3, c4, c5, c6, c7
common/cap/ rcap, cap1, cap2, Xbeg
common/convg/ mxiter, epsilon
common/dtype/ mtype
common/tstress/ sigx, sigy, sigz, sigxy, sigyz, sigxz
common/strmnc/ depsx, depsy, depsz, depsxy, depsyz, depsxz
common/thardp/ epbar, Xt

c.....Failure envelope functions
c
fail(il) = c1 - c2*dexp(-c3*il)
soft(epbar) = c4*(dexp(-c5*epbar)-1.0)
hard(il, epbar) = c6*(1.0-dexp(-c7*epbar))*il
Q(il,epbar) = fail(il) + soft(epbar) + hard(il,epbar)

Cap functions

F(il,X) = rcap*dsqrt(X*X - il*il)
epkkc(X) = cap2*(1.0d0 - dexp(-cap1*(X-Xbeg)))

Elastic bulk modulus

bmod(il) = aki*(1.0-ak1*dexp(-ak2*il))/(1.0-ak1)

Elastic shear modulus

smod(sj2) = agi*(1.0-ag1*dexp(-ag2*sj2))/(1.0-ag1)

pi = datan(1.0d0)*4.0d0

do 15 i=1,6
   ep(i) = 0.0
15 continue

if(dabs(depsx)+dabs(depsy)+dabs(depsz)+dabs(depsxy)+dabs(depsyz)
   +dabs(depsxz).lt.0.000001*epsilon) then
   mtype = 1
   return
endif

Save initial stresses and hardening parameters

if (X.lt.Xbeg) X = Xbeg
   call settr(sigx ,sigy ,sigz ,sigxy ,sigyz ,sigxz ,epbar ,X ,
   sigx ,sigy ,sigz ,sigxy ,sigyz ,sigxz ,epbar ,X)

Save initial strain increment

depsxi = depsx
depsiy = depsy
depsz = depsz
depsxyi = depsxy
depsyzi = depsyz
depsxzi = depsxz

Compute volumetric strain increment

and strain deviation increment tensor

dev = depsx + depsy + depsz
dev03 = dev/3.0
dexx = depsx - dev03
deyy = depsy - dev03
dezz = depsz - dev03
c.....Compute initial mean normal stress,
c.....stress deviation tensor, and stress invariants
c.....Note that szz = -(sxx+syy)
c
press = (sigx +sigy +sigz)/3.0
sxx = sigx - press
sxxi = sxx
syy = sigy - press
syyi = syy
szzi = -(sxx+syy)
sxz = sigxz
szxi = sxz
syz = sigyz
syz = syz
sxy = sigxy
sxyz = sxy
ili = 3.0 * press
sj2i = dsqrt(sxx*sxx+syy*syy+sxx*syy+sxz*szx+syz*syz+sxy*sxy)
c
c.....Elastic material properties
c
threek = 3.0*bsmod(ili)
twog = 2.0*bsmod(sj2i)
c
c......................................................
c
c.....Elastic Trial
c
ile = threek*dev+ili
sxx = sxx + twog*dsxx
syy = syy + twog*dsyy
szz = -(sxx + syy)
sxz = szx + twog*depsxz
syz = syz + twog*depsyz
sxy = sxy + twog*depsxy
sj2e = dsqrt(sxx*sxx+syy*syy+sxx*syy+sxz*szx+syz*syz+sxy*sxy*sxy)
ratio = 1.0
mtype = 1
il = ile
sj2 = sj2e
c
c......................................................
c
c.....Determine which surface should be checked for penetration
c
c...........Calculate slope of line from initial stress point to corner
c
call intsct(epbar,X,Qi)
if (ili.eq.Qi .and. sj2i.eq.Qi(epbar)) then
  slope = pi
else
slove = datan2(Q(Qi,epbar)-sj2i,Qi-ili)
if (slove.lt.0.0) slove = 2.0*pi - slove
if (slove.gt.1.5*pi) then
   slove = slove - 1.5*pi
else
   slove = slove + 0.5*pi
endif
endif

c
------Calculate slope of trail elastic stress trajectory------
c
elslp = datan2(sj2e-sj2i,i1e-i1i)
if (elslp.lt.0.0) elslp = 2.0*pi - elslp
if (elslp.gt.1.5*pi) then
   elslp = elslp - 1.5*pi
else
   elslp = elslp + 0.5*pi
endif

c
   if (elslp.lt.slove) goto 250

c
------...........................................................------
c
------Check for penetration of failure envelope------
c
   if (sj2e-Q(i1e,epbar).le.0.00001*epsilon) goto 250

c
------Perform failure surface calculations------
c
   50 mtype = 2

c
------Move elastically to failure surface------
c
   if (dabs(ili-i1e).le.0.00001*epsilon
      1.or. dabs(sj2i-Q(iili,epbar)).le.0.00001*epsilon) then
      il1f = i1i
      sj2f = Q(il1f,epbar)
      goto 70
   endif

c
------Find intersection of failure surface and elastic stress trajectory------
c
   if (ili.lt.i1e) then
      ili = ili
      ilr = i1e
   else
      ili = i1e
      ilr = ili
   endif

c
   if (dabs(i1e-ili).gt.1.0e-14) then

slopes = (s_j2 - s_j1)/(i_e - i_l)
else
i_l = i_l
s_j2 = Q(i_l, e_{obar})
goto 70
endif

d_60 icount = 1, mxiter
i_l = (i_l_{l} + i_l_{r})/2.0
s_j2 = Q(i_l, e_{obar})
if (s_j2 < 0.0) then
i_l = i_l
i_r = i_l
endif
else
if (s_j2 > 0.0) then
i_r = i_l
i_l = i_l
endif
continue
write(6,*) ' Did not converge on failure surface penetration'
write(8,*) ' Did not converge on failure surface penetration'
c.............Move stress point to failure surface
c.............Calculate ratio of purely elastic strain
c.............to total strain increment
c
70 if (dabs((s_j2 - s_j1)*(s_j2 - s_j1) + (i_l - i_l)*(i_l - i_l)).lt.1.e-007) then
r_r = 0.0
r_r = 0.0
r_r = (i_l - i_l)/(i_l - i_l)
endif
if (dabs(s_j2 - s_j1).gt.0.0001) then
r_r = (s_j2 - s_j1)/(s_j2 - s_j1)
else
r_r = (i_l - i_l)/(i_l - i_l)
endif
c
dep sx = (1.0 - r_r) * depsx
depsy = (1.0 - rr) * depsy
depsz = (1.0 - rr) * depsz
depsvy = (1.0 - rr) * depsvy
depsvz = (1.0 - rr) * depsvz
depszx = (1.0 - rr) * depszx

c
sigx = (1.0 - rr) * sigx + rr * (sxx + ile/3.0)
sigy = (1.0 - rr) * sigy + rr * (syy + ile/3.0)
sigz = (1.0 - rr) * sigz + rr * (szz + ile/3.0)
sigxy = (1.0 - rr) * sigxy + rr * sxy
sigyz = (1.0 - rr) * sigyz + rr * syz
sigzx = (1.0 - rr) * sigzx + rr * sxz
press = ilf/3.0
ili = ilf
sj2i = sj2f

c
.....Update Elastic material properties

c
threek = 3.0*mod(ili)
twog = 2.0*smod(sj2i)

c
80 if (mtype.eq.3) goto 320

c
call fails(depsx,depsy,depsz,depsvy,depsvz,depszx,epbar,
1 sigx,sigy,sigz,sigxy,sigy,zs,ili,sj2, ep)

c
.....Check if Corner Treatment is Needed

c
if (il1.lt.0.0) goto 195
if (il1-X.ge.0.00001*epsilon) goto 400
if (Q11(epbar)-F(ili,x).ge.0.00001*epsilon) goto 400

goto 195

c
...............................................................

c
.....Check for penetration of cap

c
250 if (ile-X.ge.0.00001*epsilon) goto 300
if (ile.lt.0.0) goto 195
if (F(ile,x).ge.sj2e) goto 195

c
.....Perform Cap Calculations

c
300 mtype = 3

c
.....Move elastically to cap

c
if (dabs(ili-ile).lt.0.00001*epsilon
1 .or. dabs(sj2i-F(ili,x)).lt.0.00001*epsilon) then
    ilf = ili
    sj2f = F(ilf,x)
goto 70
endif

Find intersection of cap and elastic stress trajectory

if (ili.lt.i1e) then
  ili = i1i
  if (i1e.ge.X) then
    ilr = X
  else
    ilr = ile
  endif
else
  ili = ile
  ilr = ili
endif

slo = (s2j-s2i)/(ile-i1i)
do 310 icount = 1,mxiter
  ilf = (ili+ilr)/2.0
  if (ilf.ge.X) ilf = X
  s2f = F(ilf,X)
  ff = s2f - (s2i + slo*(ilf-ili))
  if (dabs(ff).lt.0.00001) goto 70
  if (dabs(ilr-ili).lt.0.00001) goto 70
  if (ff*slo.gt.0.0) then
    ili = ilf
  else
    ilr = ilf
  endif
  continue
write(6,*),' Did not converge on cap penetration'
write(8,*),' Did not converge on cap penetration'
goto 70

call caps(depsx,depsy,depsz,depsxy,depsy,depsxz,depszy,depszxy,depszy,depsz,il,il)

Check if Corner Treatment is Needed

if (F(il,X) - Q(il,epbar).gt.0.00001) goto 400
goto 195

 Corner Treatment
400 mtype = 4
epbar = epbar
X = Xi
c

      call settr(sigx ,sigy ,sigz ,sigxy ,sigyz ,sigxz ,epbar ,X ,
        sigx,t,sigyt,sigzt,sigxty,sigyzt,sigxzt,pepbt,Xt)
      call settr(depxs ,depsy ,depsz ,depxsy ,depryz ,deprxz ,epbar ,X ,
        depxs,depsy,depsz,depxsy,depryz,deprxz,depxszt,epbar,Xt)
      call simplex(r)
      r1 = r(1)
      r2 = r(2)

c
      dev = depxs + depsy + depsz
      dev03 = dev/3.0
      dexx = depxs - dev03
      deyy = depsy - dev03
      dezz = depsz - dev03

      press = (sigx +sigy +sigz)/3.0

      epbar = epbari
      depxs = r1*dev03 + r2*dexx
      depsy = r1*dev03 + r2*deyy
      depsz = r1*dev03 + r2*dezz
      depxsy = r2*depxsy
      depsyzt = r2*depsyzt
      depxszt = r2*depxszt
      call fails(depxs,depsy,depsz,depxsy,depsyzt,depxszt,depxszt,
        epbar,sigx,sigy,sigz,sigxy,sigyzt,sigxzt,fil,fsj2,ep)

      X = Xi
      depxs = depxs - ep(1)
      depsy = depsy - ep(2)
      depsz = depsz - ep(3)
      depxsy = depxsy - ep(4)
      depsyzt = depsyzt - ep(5)
      depxszt = depxszt - ep(6)
      call caps(depxs,depsy,depsz,depxsy,depsyzt,depxszt,
        X,sigx,sigy,sigz,sigxy,sigyzt,sigxzt,fil,fsj2)

c
      ff = dsqrt((fil-il)*(fil-il)+(fsj2-sj2)*(fsj2-sj2))
      if (ff.gt.1000.0*epsilon) then
        write(6,4568) ff,r1,r2
        write(8,4568) ff,r1,r2
      endif

4568 format(' WARNING: Simplex search yielded minimum of ',d15.7,/,
      ' With R1 = ',d13.5,' and R2 = ',d13.5)

c
      call intsect(epbar,X,Qi)
      il = Qi
      sj2 = Q(il,epbar)
      if (dabs(Qi-fil).gt.epsilon*1000.0) then
        write(6,4570) fil,Qi
        write(8,4570) fil,Qi
format(' WARNING: Simplex yielded l1 = ',d13.5,/
1 ' Corner l1 = ',d13.5)

endif

c

.. Solution found, Update Stresses

c
195 if (sj2e.lt.0.00001*epsilon) then
   ratio = 0.0
   goto 200
endif
201 ratio = sj2/sj2e
200 sxz = sxz*ratio
   syy = syy*ratio
   szz = -(sxz+syy)
   sxz = sxz*ratio
   syz = syz*ratio
   sxy = sxy*ratio
   press = il/3.0

.. Restore initial strain increment

c
depsx = depsx0
   depay = depay0
   depsz = depsz0
   depsxy = depxy0
   depsyz = depyz0
   depsxz = depsxz0

c
   sj2 = dsqrt(sxx*sxx+ syy*syy+ sxz* sxz + syz* syz + sxy*sxy)
   sigx = sxx + press
   sigy = syy + press
   sigz = press - sxx - syy
   sigxz = sxz
   sigyz = syz
   sigxy = sxy

c
   return

end

.. Failure surface calculation

subroutine fails(depsx,depsy,depsz,depsxy,depsyz,depsxz,epbar,
   l
   sigx,sigy,sigz,sigxy,sigyz,sigxz,il,sj2,ep)

c
   implicit real*8 (a-h,o-z)
real*8 il,ili,ile,ilt
dimension ep(6)
c

common/moduli/ aki,ak1,ak2,agi,ag1,ag2
common/failc/ c1,c2,c3,c4,c5,c6,c7
common/convc/ mxiter,epsilon

c

....Failure envelope functions

c
fail(ii) = c1 - c2*exp(-c3*ii)
soft(epbar) = c4*(exp(-c5*epbar)-1.0)
hard(ii,epbar) = c6*(1.0-exp(-c7*epbar))*ii
Q(ii,epbar) = fail(ii) + soft(epbar) + hard(ii,epbar)

c
....Elastic bulk modulus

c
bmod(ii) = aki*(1.0-ak1*exp(-ak2*ii))/(1.0-ak1)

c
....Elastic shear modulus

c
smod(sj2) = agi*(1.0-ag1*exp(-ag2*sj2))/(1.0-ag1)

c
....Compute volumetric strain increment

c
....and strain deviation increment tensor

c
devid = depsx + depsy + depsz
devid3 = dev/3.0
dexx = depsx - dev03
deyy = depsy - dev03

c
....Compute initial mean normal stress,
c
....stress deviation tensor, and stress invariants

c
press = (sigx +sigy +siz)/3.0
sxx = sigx - press
sxxi = sxx
syy = sigy - press
syyi = syy
szz = -(sxx+syy)
szizi = szz
sxz = sigxz
sxzi = sxz
syz = sigyz
syzi = syz
sxy = sigxy
sxyi = sxy

ili = 3.0 * press
sj2i = dsqrt(sxx*sxx+syy*syy+sxx*syy+sxz*sxz+syz*syz+sxy*sxy)

c
....Elastic material properties

c
threek = 3.0*bmod(ili)
twog = 2.0*smod(sj2i)
c

c.....Elastic Trial

       i1e = threek*dev+ili
     sxx = sxx + twog*dexx
     syy = syy + twog*deyy
     sxz = sxz + twog*depsxz
     syz = syz + twog*depsyz
     sxy = sxy + twog*depsxy
     sj2e = dsqrt(sxx*sxx+syy*syy+sxx*syy+sxx*sxy+sxy*sxy)
      if (Q(i1e,epbar).ge.sj2e) then
          i1 = i1e
          sj2 = sj2e
        ep(1) = 0.0
        ep(2) = 0.0
        ep(3) = 0.0
        ep(4) = 0.0
        ep(5) = 0.0
        ep(6) = 0.0
        return
      endif

c
      c.....Calculate a trial value of the equivalent plastic strain

      depbart = dsqrt((2./3.)*(depsx*depsx + depsy*depsy + depsz*depsz
      1 + 2.0*depsxy*depsxy + 2.0*depsyz*depsyz
      2 + 2.0*depsxz*depsxz))

      c
      do 70 icount = 1,mxiter

      c

      c........Call subroutine finidil to find trial i1, trial plastic vol.
      c........strain increment and trial sqrt of j2.

      call finidil(i1t,ili,epbar,depbart,sj2i,i1e,
      1 threek,devpt,sj2t,sj2e,twog,ff)
      if (ff.le.0.0) goto 75
      depbart = depbart*2.0

    70 continue
      write(6,*) ' Failed to find depbar that made ff negative'
      write(8,*) ' Failed to find depbar that made ff negative'

      c

      c.....Set up left and right-most values of depbar for bisection process

    75 small = epsilon*10.0
      depbarl = 0.0
      ffl = sj2e - Q(i1e,epbar)
      depbarr = depbart
      ffr = ff

      c

      c.....Begin bisection process to find correct value of depbar
      c
do 80 i = 1,mxiter
   if (dabs(ff).le.small) goto 190
   if (dabs(ddepbar-ddepbar1).lt.0.001*small) goto 190
   if (mod(float(i),2.0).ne.1) then
      ddepbar = (ddepbar1*ffr-ddepbarr*ffl)/(ffr-ffl)
   else
      ddepbar = (ddepbar1+ddepbar)/2.0
   endif
   call findil(i1t,i1i,eepbar,ddepbar,sj2i,i1e,
     1
     threek,devpt,sj2t,sj2e,twog,ff)
   if (ff.gt.0.0) then
      ddepbarl = ddepbar
      ffl = ff
   else
      ddepbarr = ddepbar
      ffr = ff
   endif
80 continue

   c.....Failed to find convergent value of ddepbar
   c
   write(6,*)' Failed to find convergent ddepbar'
   write(8,*)' Failed to find convergent ddepbar'
   stop
   c.....Update EpBar, il, sj2
   c
190 epbar = epbar + ddepbar
   il = i1t
   sj2 = sj2t
   c
   if (sj2e.lt.0.00001*epsilon) then
      ratio = 0.0
   else
      ratio = sj2/sj2e
   endif
   sxx = sxx*ratio
   syy = syy*ratio
   szz = -(sxx+syy)
   sxz = sxz*ratio
   syz = syz*ratio
   sxy = sxy*ratio
   c
   sj2 = dsqrt(sxx*sxx+syy*syy+sxx*sxx+syy+sxx+syz*syz+sxy*sxy)
   c
   c.....Calculate Plastic strain increment tensor
   c
   ep(1) = depsx
   1   - (1.0/3.0*(il-il1)/threek + (sxx-sxxi)/twog)
   ep(2) = depsy
   1   - (1.0/3.0*(il-il1)/threek + (syy-syyi)/twog)
ep(3) = depsz
   l  - (1.0/3.0*(il-il)/threek + (szz-szzi)/twog)
ep(4) = depsxy - (sxy-sxyi)/twog
ep(5) = depsyz - (syz-syzi)/twog
ep(6) = depsxz - (sxx-sxxi)/twog

return
end

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

subroutine findil(il,ili,epbar,depbar,sj2i,i1e,
   l  threek,devpt,sj2t,sj2e,twog,ff)

implicit real*8 (a-h,o-z)
real*8 il,ili,i1e,il,ilr,il1,ilmin

common/moduli/ aki,ak1,ak2,agi,ag1,ag2
common/failc/ c1,c2,c3,c4,c5,c6,c7
common/convg/ mxiter,epsilon

.....Failure envelope functions

   fail(il) = c1 - c2*dexp(-c3*il)
   soft(epbar) = c4*(dexp(-c5*epbar)-1.0)
   hard(il,epbar) = c6*(1.0-dexp(-c7*epbar))*il
   Q(il,epbar) = fail(il) + soft(epbar) + hard(il,epbar)

.....Convergence test function for il

   aa(il,epbar) = -(c2*c3*dexp(-c3*il) + c6*(1.0 - dexp(-c7*epbar)))
   filt(il,epbar,depbar,i1e,threek) = (i1e-il)/(3.0*threek*depbar)
   l  = aa(il,epbar)/dsqrt(1.0+2.0)*aa(il,epbar)*aa(il,epbar))

   epbar = epbar + depbar
   call failmin(ilmin,epbar)

.....Calculate partial of Q wrt ili

   a = aa(ili,epbart)

.....Calculate test value of il based on derivative taken at ili

   ilt = i1e - (3.0*threek*a*depbar)/dsqrt(1.0+2.0*a*a)

.....Set up bisection bounds on il

   delta = dabs(i1e-ilt)
   if (delta.lt.10.0) delta = 10.0
   il1 = i1e
   ilr = i1e
c
do 100 icount = 1,mxiter
   ill = ill - delta
   if (ill.lt.ilim) ill = ilim
   if (-c3*ill.gt.200) then
      ffl = (ille-ill)/(3.0*threek*depbart)-1.0/dsqr1(2.0d0)
   else
      ffl = filt(ille,epbart,depbart,ile,threek)
   endif
   if (ffl.eq.0.0) then
      ilt = ill
      goto 200
   endif
   if (ill.eq.ilim .and. ffl.lt.0.0) then
      ilt = ilim
      goto 200
   endif
   if (illr.lt.ill) illr = ill
   illr = illr + delta
   ffr = filt(illr,epbart,depbart,ile,threek)
   if (ffr.eq.0.0) then
      ilt = illr
      goto 200
   endif
   if (ffl*ffr.lt.0.0) goto 150
   100 continue
   write(6,*) ' Could not bracket il in subroutine findil'
   write(8,*) ' Could not bracket il in subroutine findil'
c
c.....Bisection to find ilt
c
150 continue
do 190 icount=1,mxiter
   ilt = (ill+illr)/2.0
   if (-c3*ilt.gt.200) then
      ffilt = (ille-ilt)/(3.0*threek*depbart)-1.0/dsqr1(2.0d0)
   else
      ffilt = filt(ilt,epbart,depbart,ile,threek)
   endif
   if (dabs(ffilt).lt.0.1*epsilon) goto 200
   if (dabs(illr-ill).lt.epsilon) goto 200
   if (ffilt*ffr.gt.0.0) then
      ill = ilt
      ffl = ffilt
   else
      illr = ilt
      ffr = ffilt
   endif
   190 continue
   write(6,*) ' Bisection for ilt failed in subroutine findil'
   write(8,*) ' Bisection for ilt failed in subroutine findil'
c......Calculate partial of Q wrt to il1
   c
   200 a = aa(il1,epbar)
   c
   c......Calculate trial value of Plastic Volumetric Strain Increment
   c
   devpt = 3.0*a*depbart/dsqrt(1.3+2.0*a*a)
   c
   c......Calculate the trial value of the square root of J2
   c
   sj2t = Q(il1,epbar)
   c
   c......Calculate convergence function
   c
   ff = sj2e - sj2t = twog/2.0*devpt/(3.0*a)
   c
   return
   end
   c
**********
   c
   c......Subroutine to find value of il1 at which Q(il1,epbar) = 0.0
   c
   subroutine failmin(ilmin,epbar)
   c
   implicit real*8 (a-h,o-z)
   real*8 il1,ilmin,ill,ilr
   c
   common/failc/ c1,c2,c3,c4,c5,c6,c7
   common/convg/ mxiter,epsilon
   c
   c......Failure envelope functions
   c
   fail(il1) = c1 - c2*dexp(-c3*il1)
   soft(epbar) = c4*(dexp(-c5*epbar)-1.0)
   hard(il1,epbar) = c6*(1.0-dexp(-c7*epbar))*il1
   Q(il1,epbar) = fail(il1) + soft(epbar) + hard(il1,epbar)
   c
   delta = c1/10.0
   ill = -1.0/c3*dlog(c1/c2)
   ilr = ill
   c
   if (Q(ilr,epbar).eq.0.0) then
      ilmin = ilr
      return
   endif
   c
   if (Q(ilr,epbar).gt.0.0) then
      100 ill = ill - delta
      if (Q(ill, epbar).gt.0.0) goto 100
else
  ilr = ilr + delta
  if (Q(ilr,epbar).lt.0.0) goto 110
endif

100 do 200 i=1,mxiter
   ilmin = (ilr+iil)/2.0
   if (Q(ilmin,epbar).ge.0.0 .and. Q(ilmin,epbar).lt.0.001*epsilon) goto 300
   if (Q(ilmin,epbar).gt.0.0) then
     ilr = ilmin
   else
     iil = ilmin
   endif
200 continue
   write(6,*) ' Failed to find Ilmin in subroutine failmin.'
   write(8,*) ' Failed to find Ilmin in subroutine failmin.'
300 return
end

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

C.....Cap calculations

subroutine caps(depsx,depsy,depsz,depsxy,depsyz,depsxz,
                X,sigx,sigy,sigz,sigxy,sigy,z,sigxz,I,J)
implicit real*8 (a-h,o-z)
real*8 il,I,I,J
common/moduli/ aki,ak1,ak2,agi,ag1,ag2
common/cap/ rcap,capl,cap2,Xbeg
common/convg/ mxiter,epsilon

C.....Cap functions

  Xcap(I,J) = dsqrt(I*I+J*J)/(rcap*rcap)
  F(I,X) = rcap*dsqrt(X*X - I*I)
  epkkl(X) = cap2*(1.0d0 - dexp(-cap1*(X-Xbeg)))

C.....Elastic bulk modulus

  bmod(I,J) = aki*(1.0-ak1*dexp(-ak2*I))/ (1.0-ak1)

C.....Elastic shear modulus

  smod(J) = agi*(1.0-ag1*dexp(-ag2*J))/ (1.0-ag1)

C.....Compute volumetric strain increment
C.....and strain deviation increment tensor
dev = depsx + depsy + depsz
dev03 =dev/3.0
dexx = depsx - dev03
deyy = depsy - dev03
c
....Compute initial mean normal stress,
c.....stress deviation tensor, and stress invariants
c.....Note that szz = -(sxx+syy)
c
  press = (sigx +sigy +sigz)/3.0
  sxx = sigx - press
  syy = sigy - press
  szx = sigxz
  syz = sigyz
  sxy = sigxy
  ili = 3.0 * press
  sj2i = dsqrt(sxx*sxx+syy*syy+sxx*syy+sxz*sxz+syz*syz*sxy*sxy)
c
.....Elastic material properties
c
  threek = 3.0*bmod(ili)
  twog = 2.0*smod(sj2i)
c
.....Elastic Trial
c
  ile = threek*dev+ili
  sxx = sxx + twog*dexx
  syy = syy + twog*deyy
  szx = szx + twog*depsxz
  syz = syz + twog*depsyz
  sxy = sxy + twog*depsxy
  sj2e = dsqrt(sxx*sxx+syy*syy+sxx*syy+sxz*sxz+syz*syz*sxy*sxy)
c
  if (ile.1.e.X) then
    if (F(ile,X).ge.sj2e) then
      il = ile
      sj2 = sj2e
      return
    endif
  endif
c
  Xi = X
eyki = epyk(X)
c
  Xl = X
  Xr = Xi
do 100 icount = 1,mxiter
    Xr = Xr + (Xcap(ile,sj2e)-Xi)
    call calff(Xr,threek,twog,epyki,ile,sj2e,il,sj2,ffr)
    if (il.gt.0.0) goto 200
  100 continue
write(6,*) ' Xr not big enough in Cap calc.'
write(8,*) ' Xr not big enough in Cap calc.'
stop

200 do 390 icount = 1,mxiter
    X = (Xl+Xr)/2.0
    call calff(X,threkk,twog,epkki,iie,sj2e,il,sj2,ff)
    if (il.lt.0.0) then
        Xl = X
        goto 390
    endif
    if (dabs(ff).lt.epsilon*Xbeg*0.001) goto 400
    if (ff*ffr.lt.0.0) then
        Xl = X
    else
        Xr = X
    endif
390 continue

write(6,*) ' Cap calc. did not converge'
write(8,*) ' Cap calc. did not converge'

400 return
end

c**************************************************************************************************************
c......Calculate Cap Test Function
c......
c subroutine calff(X,threkk,twog,epkki,iie,sj2e,il,sj2,ff)
c implicit real*8 (a-h,o-z)
real*8 il,iie
c common/cap/ rcap,cap1,cap2,Xbeg
c common/convg/ mxiter,epsilon

c......Cap functions
c
F(il,X) = rcap*dsqrt(X*X - il*il)
epkkc(X) = cap2*(1.0d0 - dexp(-cap1*(X-Xbeg))
Xcap(epkk) = Xbeg - 1.0/cap1*dlog(1.0-epkk/cap2)

icount = 0
depkk = epkkc(X) - epkki
10 icount = icount + 1
if (icount.gt.mxiter) then
    write(6,*) ' Unable to get Il positive in subroutine calff'
    write(8,*) ' Unable to get Il positive in subroutine calff'
    return
endif
il = il* - thresk*depkk
if (il.gt.x) then
  il = -1.0
  return
endif

if (il.lt.0.0) then
  depkk = depkk/2.0
  epkk = epkki + depkk
  X = Xcap(epkk)
goto 10
endif

sj2 = F(il,x)

if (X.ne.il) then
  a = rcap*il/dsgrt(X*X-il*il)
  desp = ttwog/2.0*depkk/(3.0*a)
else
  desp = 0.0
endif
ff = sj2e - sj2 - desp

return
end

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

SUBROUTINE simplex(xmin)

-----------------------------------------------------------------------------
NELDER - MEAD SIMPLEX ALGORITHM FOR OPTIMIZATION
-----------------------------------------------------------------------------

IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER (NMAX=2, NMP1=3, MCOUNT=999, RMIN=1.0D-8)
DIMENSION START(NMAX), STEP(NMAX), XMIN(NMAX), XSEC(NMAX),
1 P(NMAX,NMP1), PSTAR(NMAX), P2STAR(NMAX),
2 PBAR(NMAX), Y(NMP1)

FN IS THE NAME OF USER INPUT PROBLEM FUNCTION
START(N) - STARTING VERTEX OF SIMPLEX
STEP(N) - STEP SIZES TO FORM P FROM START
I/O UNITS: 5 - STANDARD INPUT, 6 - STANDARD OUTPUT (TERMINAL)
            8 - HISTORY FILE COPY OF TERMINAL I/O

pi = datan(1.0d0)*4.0
icount = 600
n = 2
nplus1 = n + 1
reqmin = 0.000001
do 900 kk=1,16
angle = pi/4.0 + pi/2.0*dfloat(kk-1)
xmag = 0.5*dsqrt(2.0d0)*dfloat((kk-1)/4+1)
start(1) = xmag*dcos(angle)
start(2) = xmag*dsin(angle)
step(1) = xmag
step(2) = xmag

C 25 DO 30 I = 1,N
XMIN(I) = 0.D0
XSEC(I) = 0.D0
30 CONTINUE
YNEWLO = 0.D0
YSEC = 0.D0

C *** CALL NELDER-MEAD SUBROUTINE ***
C
CALL SEARCH ( N, NPLUS1, START, XMIN, XSEC, YNEWLO,
1 YSEC, REQMIN, STEP, ICOUNT, P, PSTAR,
2 P2STAR, PBAR, Y )

C if (ynewlo.lt.0.02) goto 1000
900 continue
1000 continue
C
RETURN
END

C***************************************************************************
C
SUBROUTINE SEARCH ( N, NPLUS1, START, XMIN, XSEC, YNEWLO,
1 YSEC, REQMIN, STEP, ICOUNT, P, PSTAR,
2 P2STAR, PBAR, Y )
C
C *** NELDER-MEAD SIMPLEX SEARCH SUBROUTINE ***
C
C---> REFERENCE: D. M. OLSSON "A SEQUENTIAL SIMPLEX PROGRAM FOR
C SOLVING MINIMIZATION PROBLEMS", J. QUALITY TECH., 6, 1, JAN74
C
C IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION START(N), STEP(N), XMIN(N), XSEC(N),
1 P(N,NPLUS1), PSTAR(N), P2STAR(N), PBAR(N),
2 Y(NPLUS1)
C
START(N) - STARTING VERTEX OF SIMPLEX
STEP(N) - STEP SIZES TO FORM P FROM START
P(N,NPLUS1) - N SIMPLEX COORDINATES AT N + 1 VERTECIES
PSTAR(N) - CENTROID OF FIXED FACE
P2STAR(N) - TEST POINT
P2STAR(N) - EXTENSION OR CONTRACTION OF PSTAR
Y(NPLUS1) - MERIT FUNCTION AT ALL VERTECIES
XMIN(N) - BEST VERTEX FOUND SO FAR
XSEC(N) - NEXT BEST VERTEX FOUND
FN

- FUNCTION SUPPLIED TO EVALUATE MERIT FUNCTION

DATA ZERO, HALF, ONE, TWO, RCOEFF, ECOEFF, CCOEFF
  1 / 0.0D0, 0.5D0, 1.0D0, 2.0D0, 1.0D0, 2.0D0, 0.5D0 /
DATA BIGNUM, DABIT, MAXLOP
  1 / 1.38, 2.04607D-35, 5 /
KCOUNT = ICOUNT
ICOUNT = 0

IF ( REQMIN .LE. ZERO ) REQMIN = DABIT + DABIT
KONVGE = MAXLOP
XN = FLOAT(N)
DN = DBLE(XN)
DN = DFLOAT(N)

*** CONSTRUCTION OF THE INITIAL SIMPLEX ***

5 DO 10 I = 1,N
10 P(I,NPLUS1) = START(I)
call fn(start,y(nplus1))
  if (y(nplus1).lt.0.002) then
    xmin(1) = start(1)
    xmin(2) = start(2)
    ynewlo = y(nplus1)
    return
  endif
ICOUNT = ICOUNT + 1
DO 20 J = 1,N
DCHK = START(J)
START(J) = DCHK + STEP(J)
DO 15 I = 1,N
15 P(I,J) = START(I)
call fn(start,y(j))
  if (y(j).lt.0.002) then
    xmin(1) = start(1)
    xmin(2) = start(2)
    ynewlo = y(j)
    return
  endif
ICOUNT = ICOUNT + 1
20 START(J) = DCHK

*** SIMPLEX CONSTRUCTION COMPLETE ***

FIND HIGHEST AND LOWEST Y VALUES
YNEWLO (Y(IHI)) INDICATES THE VERTEX OF
THE SIMPLEX TO BE REPLACED

25 YLO = Y(1)
YNEWLO = YLO
ILO = 1
IHI = 1
DO 35 I = 2,NPLUS1
   IF ( Y(I) .GE. YLO ) GO TO 30
   YLO = Y(I)
   ILO = I
30 IF ( Y(I) .LE. YNEWLO ) GO TO 35
   YNEWLO = Y(I)
   IHI = I
35 CONTINUE

C
*** PERFORM CONVERGENCE CHECKS ON FUNCTION ***
C
   DCHK = (YNEWLO + DABIT)/(YLO + DABIT) - ONE
   IF ( DABS(DCHK) .LT. REQMIN ) GO TO 135

C
   KONVGE = KONVGE - 1
   IF ( KONVGE .NE. 0 ) GO TO 55
   KONVGE = MAXLOP

C
*** CHECK CONVERGENCE OF COORDINATES ONLY ***
C
      EVERY FIVE SIMPLEXES
C
   DO 50 I = 1,N
      COORD1 = P(I,1)
      COORD2 = COORD1
   DO 45 J = 2,NPLUS1
      IF ( P(I,J) .GE. COORD1 ) GO TO 40
      COORD1 = P(I,J)
40 IF ( P(I,J) .LE. COORD2 ) GO TO 45
      COORD2 = P(I,J)
45 CONTINUE
   DCHK = (COORD2 + DABIT)/(COORD1 + DABIT) - ONE
   IF ( DABS(DCHK) .GT. REQMIN ) GO TO 55
   CONTINUE
   GO TO 135
50 CONTINUE
55 IF ( ICOUNT .GE. KCOUNT ) GO TO 135

C
*** CALCULATE PBAR, THE CENTROID OF THE SIMPLEX VERTICES ***
C
      EXCEPTING THAT WITH Y VALUE YNEWLO
C
   DO 65 I = 1,N
      Z = ZERO
   DO 60 J = 1,NPLUS1
      Z = Z + P(I,J)
   60 Z = Z - P(I,IHI)
65 PBAR(I) = Z/DN

C
*** REFLECTION THROUGH THE CENTROID ***
C
   DO 70 I = 1,N
      70 PSTAR(I) = (ONE + RCOEFF)*PBAR(I) - RCOEFF*P(I,IHI)
call fn(pstar,ystar)
if (ystar.lt.0.002) then
  xmin(1) = pstar(1)
xmin(2) = pstar(2)
ynewlo = ystar
return
endif
ICOUNT = ICOUNT + 1
if (ystar.ge.ylo) go to 90
if (ICOUNT.ge.KCOUNT) go to 125

*** SUCCESSFUL REFLECTION, SO TRY EXTENSION ***

DO 75 I = 1,N
75 P2STAR(I) = ECOEFF*PSTAR(I) + (ONE - ECOEFF)*PBAR(I)
call fn(p2star,y2star)
if (y2star.lt.0.002) then
  xmin(1) = p2star(1)
xmin(2) = p2star(2)
ynewlo = y2star
return
endif
ICOUNT = ICOUNT + 1

*** RETAIN EXTENSION OR CONTRACTION ***

IF (y2star.ge.ystar) go to 125

DO 85 I = 1,N
85 P(I,IHI) = P2STAR(I)
Y(IHI) = Y2STAR
GO TO 25

*** NO EXTENSION ***

L = 0
DO 95 I = 1,NPLUS1
95 CONTINUE
IF (L.gt.1) go to 125
IF (L.eq.0) go to 105

*** CONTRACTION ON THE REFLECTION SIDE OF THE CENTROID ***

DO 100 I = 1,N
100 P(I,IHI) = PSTAR(I)
Y(IHI) = YSTAR

*** CONTRACTION ON THE Y(IHI) SIDE OF THE CENTROID ***

IF (ICOUNT.ge.KCOUNT) go to 135
DO 110 I = 1,N
110 P2STAR(I) = CCoeff*P(I,IHI) + (ONE - CCoeff) * PBAR(I)
call fn(p2star,y2star)
  if (y2star.lt.0.002) then
    xmin(1) = p2star(1)
xmin(2) = p2star(2)
ynewlo = y2star
  return
endif
ICOUNT = ICOUNT + 1
if (Y2STAR .LT. Y(IHI)) GO TO 80
C
C *** CONTRACT THE WHOLE SIMPLEX ***
C
DO 120 J = 1,NPLUS1
DO 115 I = 1,N
  P(I,J) = (P(I,J) + P(I,ILO)) * HALF
115 XMIN(I) = P(I,J)
call fn(xmin,y(j))
  if (y(j).lt.0.002) then
    ynewlo = y(j)
  return
endif
120 continue
ICOUNT = ICOUNT + NPLUS1
if (ICOUNT .LT. KCOUNT) GO TO 25
GO TO 135
C
C *** RETAIN REFLECTION ***
C
125 DO 130 I = 1,N
130 P(I,IHI) = PSTAR(I)
Y(IHI) = YSTAR
GO TO 25
C
C *** SELECT THE TWO BEST FUNCTION VALUES (YNEWLO AND YSEC) ***
C AND THEIR COORDINATES (XMIN AND XSEC)
C
135 DO 145 J = 1,NPLUS1
DO 140 I = 1,N
140 XMIN(I) = P(I,J)
call fn(xmin,y(j))
  if (y(j).lt.0.002) then
    ynewlo = y(j)
  return
endif
145 continue
YNEWLO = BIGNUM
DO 150 J = 1,NPLUS1
  if (Y(J) .GE. YNEWLO) GO TO 150
YNEWLO = Y(J)
IBEST = J
150 CONTINUE
Y(IBEST) = BIGNUM
YSEC = BIGNUM
DO 155 J = 1,NPLUS1
IF ( Y(J) .GE. YSEC ) GO TO 155
YSEC = Y(J)
ISEC = J
155 CONTINUE
DO 160 I = 1,N
XMIN(I) = P(I,IBEST)
160 XSEC(I) = P(I,ISEC)
RETURN
END

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

subroutine fn(x,ff)

dimension r(2),ep(6)
implicit real*8 (a-h,o-z)
real*8 ii,iil
common/moduli/ aki,ak1,ak2,agi,ag1,ag2
common/tstress/ sigx,sigy,sigz,sigxy,sigy,si
common/tstrninc/ depx,depsy,depsz,depsxy,depsy,depsy
common/thardp/ epbar,X

c......Elastic bulk modulus

bmod(i1) = aki*(1.0-ak1*dexp(-ak2*i1))/(1.0-ak1)

c......Elastic shear modulus

smod(sj2) = agi*(1.0-ag1*dexp(-ag2*sj2))/(1.0-ag1)

c

c......Compute initial mean normal stress,
c......stress deviation tensor, and stress invariants

c
press = (sigx +sigy +sigz)/3.0
sxx = sigx - press
syy = sigy - press
szz = sigz - press
sxz = sigxz
syz = sigyz
sxy = sigxy
iil = 3.0 * press
sj2i = dsqrt(sxx*sxx+syy*syy+sxx*syy+sxx*sxx+syz*syz+sxy*sxy)

c......Elastic material properties

c
threek = 3.0*bmod(i1i)
twog = 2.0*smod(sj2i)

c......Save initial hardening parameters
c
epari = epbar
Xi = X
c
dev = depsx + depsy + depsz
dev03 = dev/3.0
dexx = depsx - dev03
deyy = depsy - dev03
dezz = depsz - dev03
c
r1 = r(1)
r2 = r(2)
depsxt = r1*dev03 + r2*dexx
depsyt = r1*dev03 + r2*deyy
depszt = r1*dev03 + r2*dezz
depsxyt = r2*depsxy
depsyzt = r2*depsyz
depszxt = r2*depsz

call fails(depsxt,depsyty,depszt,depsyzt,depsxzt,
        epbar,sigx,sigy,sigz,sigy,sigx,sigz,fil,fsj2,ep)

1
depsxt = depsx - ep(1)
depsyt = depsy - ep(2)
depszt = depsz - ep(3)
depsxyt = depsxy - ep(4)
depsyzt = depsyz - ep(5)
depszxt = depsxz - ep(6)
call caps(depsxt,depsyty,depszt,depsyzt,depsxzt,
        1
        X,sigx,sigy,sigz,sigy,sigx,sigz,il,sj2)

c
ff = dsqrt((fil-il)*(fil-il)+(fsj2-sj2)*(fsj2-sj2))

c
epbar = epari
X = Xi
c
return
c
end

c**********************************************************************************************************
c......Subroutine to find value of il = xil at intersection of failure surface and cap
c subroutine intsct(epbar,X,xil)
c
implicit real*8 (a-h,o-z)
real*8 il
common/moduli/ aki,ak1,ak2,agi,ag1,ag2
common/failc/ c1,c2,c3,c4,c5,c6,c7
common/cap/ rcap,cap1,cap2,Xbeg
common/convg/ mxiter,epsilon

c......Failure envelope functions

c fail(i1) = c1 - c2*dexp(-c3*i1)
soft(epbar) = c4*(dexp(-c5*epbar)-1.0)
hard(i1,epbar) = c6*(1.0-dexp(-c7*epbar))*i1
Q(i1,epbar) = fail(i1) + soft(epbar) + hard(i1,epbar)

c......Cap functions

c F(i1,X) = rcap*dsqrt(X*X + i1*i1)

c......Check values of Q and F at I1=0; if necessary change X

c if (F(0.0d0,X)-Q(0.0d0,epbar)) 100,200,300

c......Cap is not large enough to intersect failure surface, enlarge it

c......to intersect at I1=0

c 100 X = Q(0.0d0,epbar)/rcap
200 xil = 0.0d0
return

c......Bisection to find intersection

c 300 xill = 0.0d0
   fl = F(xill,X) - Q(xill,epbar)
xilr = X
   fr = -Q(xilr,epbar)

do 400 j = 1,mxiter
   xil = (xill*fr-xilr*fl)/(fr-fl)
   diff = F(xil,X)-Q(xil,epbar)
   if (dabs(diff).lt.Q(X,epbar)*epsilon*0.1) return
   if (diff.gt.0.0) then
      xill = xil
      fl = diff
   else
      xilr = xil
      fr = diff
   endif
400 continue
write(6,*)' Failed to find failure surface and cap intersection.'
write(8,*)' Failed to find failure surface and cap intersection.'

return
end

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

 subroutine settr(sigx,sigy,sigz,sigxy,sigyz,sigxz,.epbar,
 1       X, 
 2       sigxt,sigyt,sigzt,sigxyt,sigyzt,sigxzt,epbant,
 3       Xt)

 implicit real*8 (a-h,o-z)

 sigxt = sigx
 sigyt = sigy
 sigzt = sigz
 sigxyt = sigxy
 sigyzt = sigyz
 sigxzt = sigxz
 epbant = epbar
 Xt = X

 return
 end
A2. TRIAXIAL DRIVER SUBROUTINES

c******************************************************************************c
  c......Stress State Driver
  c......This driver allows one to test a constitutive model by
  c......finding the model response to a user selected loading
  c......sequence. Each loading sequence can be either
  c......hydrostatic or triaxial.
  c
    implicit real*8 (a-h,o-z)
  c
    common/stress/ sigx,sigy,sigz,sigxz,sigyz,sigxy
    common/strain/ epsx,epsy,epsz,epsxz,epsyz,epsxy
    common/cap/ rcap,cap1,cap2,Xbeg
    common/hardp/ epbar,X

    epkcc(X) = cap2*(1.0d0-dexp(-cap1*(X-Xbeg))

  c......Read Model Parameters from Specified Dataset
  c
    call datain(X)
  c
  c......Initialize all Variables
  c
    90 epsx = 0.0
    epsy = 0.0
    epsz = 0.0
    epsxz = 0.0
    epsyz = 0.0
    epsxy = 0.0
    epbar = 0.0
    sigx = 0.0
    sigy = 0.0
    sigz = 0.0
    sigxz = 0.0
    sigyz = 0.0
    sigxy = 0.0
    X = Xbeg
  c
  c......Main Menu
  c
    100 write(6,970) sigx,sigy,sigz
        write(6,980) sigxy,sigyz,sigxz
        write(6,990) epsx,epsy,epsz
        write(6,1000) epsxy,epsyz,epsxz,epbar,X,epkcc(X)
        write(6,1010)
        read(5,*) kpro
if (kpro.le.0 .or. kpro.gt.5) goto 100
  goto (110,120,130,140,150) kpro

110 call hydro
  goto 100

120 call triax
  goto 100

130 write(6,*)' Change has not yet been implemented.'
  goto 100

140 goto 90

150 stop

970 format(1a40,1a15,3f15.6,2a6,1a15,1a15,3f15.6),
  1 ' Current status:','///',' Stresses:','//,',
  2 '  SigX = ',d12.5,5x,'SigY = ',d12.5,5x,'SigZ = ',d12.5)
980 format(1a40,1a15,3f15.6,2a6,1a15,1a15,3f15.6),
  1 '  SigXY = ',d12.5,5x,'SigYZ = ',d12.5,5x,'SigXZ = ',d12.5)
990 format(1a40,1a15,3f15.6,2a6,1a15,1a15,3f15.6),
  1 '  EpX = ',d12.5,5x,' EpY = ',d12.5,5x,' EpZ = ',d12.5)
1000 format(1a40,1a15,3f15.6,2a6,1a15,1a15,3f15.6),
  1 '/',' Plastic Strain Measures:','//,',
  2 '  EpBar = ',d12.5,5x,' X = ',d12.5,5x,'Epkk = ',d12.5)
1010 format(1a40,1a15,3f15.6,2a6,1a15,1a15,3f15.6),
  1 '/','Select procedure:','//,',
  1 '  (1) Hydrostatic Loading','/,',
  2 '  (2) Triaxial Loading','/,',
  3 '  (3) Change State','/,',
  4 '  (4) Reset all Variables','/,',
  5 '  (5) Quit','/,//)
end

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

C.....Subroutine to Read Model Parameters from Dataset
C
C subroutine datain(X)
C
implicit real*8 (a-h,o-z)
character*20 dataset
logical fexist
C
common/moduli/ aki,ak1,ak2,agi,ag1,ag2
common/fails/ cl,c2,c3,c4,c5,c6,c7
common/cap/ rcap,cap1,cap2,xbeg
common/conv/ mxiter,epsilon
C
C.....Query Name of Input Dataset and Open it on Unit 7
C
write(6,100)
read(5,110) dataset
inquire (file=dataset,exist=fexist)
if (fexist) then
  open (7, file=dataset, status='OLD')
else
  write(6,*), 'Input dataset does not exist !!!'
  stop
endif

c indeh..read Bulk Modulus Parameters
c
  read(7,120) aki,ak1,ak2
c

c indeh..read Shear Modulus Parameters

c
  read(7,120) agi,ag1,ag2
c

c indeh..Read Failure Surface Parameters

c
  read(7,130) c1,c2,c3,c4,c5,c6,c7
c

c indeh..read Cap Parameters

c
  read(7,135) rcap,cap1,cap2,Xbeg
c

c indeh..Read Convergence Criterion Parameters

c
  read(7,140) mxiter,epsilon
c
c call expand(0.0,Xbeg,0.0)
c call intsct(0.0,Xbeg,Qbeg)
X = Xbeg
write(8,827) Xbeg,Qbeg
827 format(’ Xbeg = ’,f10.2,10x,’ Qbeg = ’,f10.2)
return
c 100 format(‘’/’ Enter Name of Model Parameter Input Dataset:‘’)
  1 format(a)
  120 format(30x,3e10.0)
  130 format(30x,3e10.0,/,30x,2e10.0,/,30x,2e10.0)
  135 format(30x,4e10.0)
  140 format(30x,110,e10.0)
end
c

*************************************************************************************************
c

c indeh..Subroutine to expand cap to intersect at current value of I1

c
  subroutine expand(epbar,X,I1)
c

  implicit real*8 (a-h,o-z)
  real*8  il,L

c

  common/moduli/  aki,akl,ak2,agi,agl,ag2
  common/failc/  c1,c2,c3,c4,c5,c6,c7
  common/cap/   rcap,cap1,cap2,Xbeg
  common/convg/  mxiter,epsilon

c

  data pi /3.141592654/

c

  failure envelope functions

  fail(il) = c1 - c2*dexp(-c3*il)
  soft(epbar) = c4*(dexp(-c5*epbar)-1.0)
  hard(il, epbar) = c6*(1.0-dexp(-c7*epbar)) * il
  Q(il, epbar) = fail(il) + soft(epbar) + hard(il, epbar)

  cap functions

  Xcap(L) = dsqrt(L*L+Q(L, epbar)*Q(L, epbar)/(rcap*rcap))

  check to see if cap needs to be expanded

  call intsc(epbar, X, xil)
  if (xil.ge.il) return

  calculate new X

  X = Xcap(il)
  return
  end

C******************************************************************************
C
C. Hydrostatic Loading Phase
C
  subroutine hydro
C
  implicit real*8 (a-h,o-z)
  dimension ep(6)
  logical exitf

  common/stress/  sigx,sigy,sigz,sigxz,sigy, sigxy
  common/strain/  epsx,epsy,epsz,epsxz,epsyz, epsxy
  common/hardp/  epbar, X
  common/convg/  mxiter,epsilon
  common/dtype/  mtype

  exitf = .false.
c.....Read Target Pressure and Strain Increment Size

write(6,1000)
read(5,*) pfinal
100 write(6,1010)
read(5,*) dep
if (dep.eq.0.0) then
   write(6,*) ' Strain increment must be non-zero !!!'
goto 100
endif

c.....Write type of stress path and starting state

c
if (dep.gt.0.0) write(6,1020)
if (dep.gt.0.0) write(8,1020)
if (dep.lt.0.0) write(6,1030)
if (dep.lt.0.0) write(8,1030)
write(6,1040)
write(6,1050)
write(6,1060) sigx,sigy,sigz,100.0*epsx,100.0*epsy,100.0*epsz,
1    mtype
write(8,1065) sigx,sigy,sigz,100.0*epsx,100.0*epsy,100.0*epsz,
1    mtype,epbar,X
write(10,1065) sigx,sigy,sigz,100.0*epsx,100.0*epsy,100.0*epsz,
1    mtype,epbar,X

c
c.....Initialize Strain Increments

c
depsx = dep
depsy = dep
depsz = dep
depsey = 0.0
depsyz = 0.0
depszz = 0.0

c
do 300 m = 1,100*mxiter

c
call settr(sigx,sigy,sigz,sigxy,sigyzt,sigxzt,sigxty,sigxtyt,sigxtyzt,sigxtyzt)
1   sigxt,sigyt,sigzt,sigxtyt,sigxtyzt)
c
call matlab(depsx,depsy,depsz,depsxy,depsyz,depsyzx,depszx,depszxz,depszxzy,depszxzyt,
1   Xt,sigxtyt,sigxtyzt,sigxtyzt,rr,ep)
c
c
press = (sigx + sigy + sigz)/3.0
if (dep.gt.0.0) then
   if (pfinal-press) 200,215,220
else
   if (press-pfinal) 200,215,220
endif

c
200 if (dabs(press-pfinal)-epsilon) 215,215,210
c.....Final target pressure has been overshot, Cut stain increment in
c.....half and Reset stresses to last OK state

c 210 dep = dep/2.0
depx = dep
depz = dep
depy = dep
call settr(sigx,sigy,sigz,sigy,sigy,sigz,sigz,sigx, dep, X,
1 sigx,sigx,sigz,sigy,sigy,sigy,sigy,sigz ,epbar, X)
goto 300

c.....Trial stress state is OK, so update it to current status

c 215 exitf = .true.
call settr(sigx,sigx,sigy,sigy,sigy,sigy,sigy,sigy,depbar, X, 
1 sigx ,sigy,sigy,sigy,sigy,sigy,sigy, epbar,X)
epsx = epsx + depdx
epso = epsy + depdy
epzs = epsz + depdz
epsex = epsxy + depdxy
epsey = epsyz + depdyz
epsexz = epsxz + depdz

c.....Print current state

c write(6,1060) sigx,sigy,sigz,100.*epsx,100.*epsy,100.*epsz,mtype
write(8,1065) sigx,sigy,sigz,100.*epsx,100.*epsy,100.*epsz,mtype
1 ,epbar,X
write(10,1065) sigx,sigy,sigz,100.*epsx,100.*epsy,100.*epsz,
1 mtype, epbar, X

c.....Exit if target pressure has been achieved

c if (exitf) return

c 300 continue

c write(6,*) ' Failed to reach target pressure !'
write(8,*) ' Failed to reach target pressure !'
stop

c 1000 format(//' Enter final pressure:/',//)
c 1010 format(//'  Enter stain increment:/',//
1 ' (Positive for loading, negative for unloading)\',//)
c 1020 format(//'Hydrostatic Loading Phase\',//)
c 1030 format(//'Hydrostatic Loading Phase\',//)
c 1040 format(11x,'Stress',27x,'Strain',14x,'Mtype',//
1 11x,'-----',27x,'------',14x,'-----')
c 1050 format(2x,'SigmaX  SigmaY  SigmaZ',
1 11x,'EpsX   EpsY   EpsZ',//
2 4x,3('psi',6x),2x,3(2x,'Percent'),/
3 2x,3('------',3x),4x,3(2x,'------'))
1060 format(1x,f7.1,2(2x,f7.1),7x,3f9.4,4x,i3)
1065 format(3f9.1,3f9.5,i2,2x,f9.7,f9.1)
end

c
***********************************************
c
.....Triaxial Phase

c
    subroutine triax

c
    implicit real*8 (a-h,o-z)
    real*8 ii,ilref
    logical exitf, acont

c
    common/stress/ sigx,sigy,sigz,sigxz,sigy2,sigxy
    common/strain/ epsx,epsy,epsz,epsxz,epsyz,epsxy
    common/hardp/ epbar,X
    common/moduli/ aki,ak1,ak2,agi,ag1,ag2
    common/convg/ mxiter,epsilon
    common/dtype/ mtype

c
    exitf = .false.

c
.....Read Target Axial Stress, Axial Strain Increment Size,
.....and Stress Path Slope

c
.....(Note:  acont is a logical variable indicating how the
.....final axial state is specified
.....if final axial strain is specified then acont = .true.
.....if final axial stress is specified then acont = .false.

c
90 write(6,990)
    read(5,*) kpro
    if (kpro.lt.1 .or. kpro.gt.2) goto 90
    goto (94,96) kpro
94 acont = .true.
    write(6,1000)
    read(5,*) afinal
    goto 100
96 acont = .false.
    write(6,1005)
    read(5,*) afinal
100 write(6,1010)
    read(5,*) dep
    if (dep.eq.0.0) then
        write(6,*) ' Strain increment must be non-zero !!!'
        goto 100
    endif
    write(6,1020)
read(5,*), slope

C......Compute Constants from Path Slope
C......Slope = 1/sqrt(3) for Triaxial Test
C......Slope = 200 for Constant Mean Normal Stress Test

slinv = 1.0/slope
if (dabs(slope).gt.100) slinv = 0.0
il = sigx + sigy + sigz
sj2 = dsqrt(1./6.)*(sigx-sigy)*(sigx-sigy)+(sigy-sigz)*(sigy-sigz)
     + (sigz-sigx)*(sigz-sigx) + sigxy*sigxy
     + sigyz*sigyz + sigxz*sigxz)
ilref = il - slinv*sj2

C......Calculate elastic lateral strain increment corresponding to dep
ccgk = 1.5*aki*(1.0-ak1)/(agi*(1.0-ag1))
anu = (ccgk-1.0)/(2.0*ccgk+1.0)
depl = -anu * dep

C......Write Stress path type and initial state
if (dep.gt.0.0) write(6,1030)
if (dep.gt.0.0) write(6,1030)
if (dep.lt.0.0) write(6,1040)
if (dep.lt.0.0) write(6,1040)
write(6,1050)
write(6,1060)
write(6,1070) sigx, sigy, sigz, 100.0*epsx, 100.0*epsy, 100.0*epsz,
mttype
write(8,1075) sigx, sigy, sigz, 100.0*epsx, 100.0*epsy, 100.0*epsz,
mttype, epbar, X
write(10,1075) sigx, sigy, sigz, 100.0*epsx, 100.0*epsy, 100.0*epsz,
mttype, epbar, X

C do 300 m = 1,100*mxiter

C........Don’t allow overshoot when final axial strain state is given
if (acont) then
  if (dabs(dep).gt.dabs(afinal-epsz)) dep = afinal - epsz
endif

C........Set trial stress state equal to current state
 call settr(sigx, sigy, sigz, sigxy, sigyz, sigxz, epbar, X, 
            sigxt, sigyt, sigzt, sigxyt, sigyzt, sigxzt, epbart, Xt)
C
C........Find new state with specified lateral stresses
 call lateral(dep, depl, epbart, Xt, sigxt, sigyt, sigzt,
1   sigxyt,sigyzt,sigxzt,ilref,slinv)
c
...Test to see it at specified target axial state or if overshoot...c
...has occurred
c
   if (acont) then
      if (epsz+dep.eq.afinal) exitf = .true.
   else
      if (dabs(afinal-sigzt).lt.epslion) exitf = .true.
      if ((afinal-sigz)*(afinal-sigzt).lt.0.0) goto 250
   endif

c
......Trail state is OK, make it the current state
c
   call settr(sigxt,sigyt,sigzt,sigxyt,sigyzt,sigxzt,epbart,Xt,
   1   sigx,sigy ,sigz ,sigxy ,sigyz ,sigxz ,epbar ,X )
   epsz = epsz+dep
   epsx = epsx+depl
   epsy = epsy+depl

c
......Print current state
c
   write(6,1070) sigx,sigy,sigz,100.*epsx,100.*epsy,100.*epsz,mtype
   write(8,1075) sigx,sigy,sigz,100.*epsx,100.*epsy,100.*epsz,
   1   mtype,epbar,X
   write(10,1075) sigx,sigy,sigz,100.*epsx,100.*epsy,100.*epsz,
   1   mtype,epbar,X
   if (exitf) goto 400
   goto 300

c
......Final target axial stress has been overshot, Cut strain increment
c......in half
c
   250 dep = dep/2.0

c
   300 continue
   write(6,*) ' Failed to reach target axial state !!!'
   write(8,*) ' Failed to reach target axial state !!!'
   stop

c
   400 return

c
......Format Statements
c
990 format(///////////),
   1   ' Target axial state to be defined by:',//,
   2   ' (1) Final axial strain',//,
   3   ' (2) Final axial stress',//)
1000 format(///////////),' Enter final axial strain:',//)
1005 format(///////////),' Enter final axial stress:',//)
1010 format(///),' Enter axial strain increment:',//,
1    ' (Positive for loading, negative for unloading)'/
1020 format(//,' Enter stress path slope: ',//,
    1    ' (for conventional triaxial enter 0.577350269',//,
    2    ' for constant pressure test enter 200')//)
1030 format(///19x,'Triaxial Phase (Loading)',//)
1040 format(///17x,'Triaxial Phase (Unloading)',//)
1050 format(11x,'Stress',27x,'Strain',14x,'Mtype',/
    1    11x,-------',27x,-------',14x,------')
1060 format(2x,'SigmaX' 'SigmaY' 'SigmaZ',/
    1    11x,'EpsX' 'EpsY' 'EpsZ',/
    2    4x,3('psi',6x),2x,3(2x,'Percent'),/
    3    2x,3('------',3x),4x,3(2x,'------'))
1070 format(1x,$7.1,2(2x,$7.1),7x,3f9.4,4x,i3)
1075 format(3f9.1,3f9.5,i2,2x,f9.7,f9.1)
end

C*******************************************************************************
C C.....Subroutine to find lateral strains consistent with prescribed
C.....confining pressure
C subroutine lateral(dep,depl,epbar,X,sigx,sigy,sigz,
    1 sigxy,sigy,z,sigxz,ilref,slinv)
C implicit real*8 (a-h,o-z)
real*8 ilref
C common/convrg/ mxiter,epsilon
C error = epsilon/100.
  icount = 0
C
  vh = depl
  if (dep.gt.0.0) vh = 2.0*vh
80 call settr(sigx ,sigy ,sigz ,sigxy ,sigy ,sigz ,epbar ,X ,
    1 sigx ,sigy ,sigz ,sigxy ,sigz ,epbar ,Xt)
  call findt(dep,vh,sigy,sigz,sigy,sigz,sigxz,epbar,Xt,
    1 Xt,test,ilref,slinv)
  if (dabs(test).lt.error) goto 20
  br=vh
  tr=test
C
  vh=-vh
  call settr(sigx ,sigy ,sigz ,sigxy ,sigy ,sigz ,epbar ,X ,
    1 sigx ,sigy ,sigz ,sigxy ,sigy ,sigz ,epbar ,Xt)
  call findt(dep,vh,sigy,sigz,sigy,sigz,sigxz,epbar,Xt,
    1 Xt,test,ilref,slinv)
  if (dabs(test).lt.error) goto 20
  bl=vh
  tl=test
C
if (bl.gt.bl) then
    bl=br
    br=vh
    tl=tr
    tr=test
endif

c
..... If both function values (tl and tr) have same sign,
c..... Increase bracket in both directions
c
90 if (tl*tr.lt.0.0) goto 100

c
    icount = icount + 1
    if (icount.gt.1000) goto 30
    vh = 2.0*vh
    goto 80

c
..... Bisection Phase

c
100 icount = icount + 1
    if (icount.gt.600) goto 21

c
    if (dabs(br-bl).lt.ddep*epsilon) goto 20
    if (mod(float(icount),2.0).ne.1.0) then
        vh = (bl*tr-br*tl)/(tr-tl)
    else
        vh = (bl+br)/2.0
    endif
    call settr(sigx,sigy,sigz,sigxy,sigyz,sigxz,sebar,X,
    sigxt,sigyt,sigzt,sigxty,sigzty,sigxzt,epbart,Xt)
    call findt(dep,vh,sigxt,sigyt,sigzt,sigxty,sigzty,sigxzt,epbart,
    Xt,test,ilref,slinv)
    call flush(8)
    if (dabs(test).lt.10.0*error) goto 20

c
    if (test*tl.lt.0.0) then

c
..... Test is same sign as tr, so replace br and tr with vh and test
c
    br=vh
    tr=test

c
..... Test is same sign as tl, so replace bl and tl with vh and test
c
    else
        bl=vh
        tl=test
    endif
goto 100

c
20 continue
depl = vh
    call setstr(sigxt, sigyt, sigzt, sigxzt, sigyzt, epbart, Xt,
    1      sigx, sigy, sigz, sigxy, sigyz, sigxz, epbar, X )
    return
21 write(6,*) ' Subroutine lateral did not converge'
22 write(8,*) ' Subroutine lateral did not converge'
23 goto 20
25 continue
    return
30 write(6,*) ' Bisection bracketing procedure failed'
31 write(8,*) ' Bisection bracketing procedure failed'
32 write(6,210) kount, bl, br
    return
200 format(' Kount = ',i5,5x,'Br-Bl = ','d14.6,6x,' Test = ','d14.6)
210 format(' Kount = ',i5,5x,'Bl = ','d14.6,6x,' Br = ','d14.6)
end

*********************************************************************************************************
  c
  c.....Findt
  c

  subroutine findt(dep,vh,sigx,sigy,sigz,sigxy,sigyzt, epbar,
    1      X,test, ilref,slinv)
  c
  implicit real*8 (a-h,o-z)
  real*8 il, ilref
  dimension ep(6)
  c
depsx = vh
depsy = depsx
depsz = dep
depzx = 0.0
depzy = 0.0
depxy = 0.0
  sigx0ld = sigx
  c
  call matlaw(depsx, depsy, depsz, depsxy, depsyz, depsxz, epbar,
    1      X, sigx, sigy, sigz, sigxy, sigyz, sigxz, rr, ep)
  c
  il = sigx + sigy + sigz
  sj2 = dsqrt((sigx-sigy)*(sigx-sigy)+(sigy-sigz)*(sigy-sigz)
    1       + (sigz-sigx)*(sigz-sigx) + sigx*sigxy
    2       + sigyz*sigyz + sigxz*sigxz)
  test = il - slinv*sj2 - ilref
  c
  return
  end

*********************************************************************************************************
  c
  c
  subroutine setstr
c implicit real*8 (a-h,o-z)
real*8 ii,ili,ile,L

c common/stress/ s1x,s1y,s1z,s1xz,s1yz,s1xy
common/hardp/ epbar,X
common/moduli/ aki,ak1,ak2,agi,ag1,ag2
common/convg/ mxiter,epsilon
common/failc/ c1,c2,c3,c4,c5,c6,c7
common/cap/ rcap,rcap1,rcap2,Xbeg

c.....Failure envelope functions

c fail(ii) = c1 - c2*dexp(-c3*ii)
soft(epbar) = c4*(dexp(-c5*epbar)-1.0)
hard(ii,epbar) = c6*(1.0-dexp(-c7*epbar))*ii
Q(ii,epbar) = fail(ii) + soft(epbar) + hard(ii,epbar)

c.....Cap functions

c Xcap(L) = dsqrt(L*L+Q(L,epbar)*Q(L,epbar)/(rcap*rcap))
F(ii,X) = rcap*dsqrt(X*X - ii*ii)
epk(c)(X) = cap2*(1.0d0 - dexp(-cap1*(X-Xbeg)))

c.....Elastic bulk modulus

c bmod(ii) = aki*(1.0-ak1*dexp(-ak2*ii))/(1.0-ak1)

c.....Elastic shear modulus

c smod(sj2) = aqi*(1.0-ag1*dexp(-ag2*sj2))/(1.0-ag1)

c.....Read Hydrostatic Pressure

c write(6,*) ' Enter hydrostatic pressure:'
read(5,*) press

c xilli = 3.0*press
xilr = 100000.0

do 10 icount=1,1000
ii = (xilli+xilr)/2.0
write(6,15) ii,sj2,Q(ii,0.0)
write(8,15) ii,sj2,Q(ii,0.0)
sj2 = (ii-3.0*press)*1.0/dsqt(3.0d0)
if (dabs(Q(ii,0.0)-sj2).lt.0.001) goto 13
if (sj2.lt.Q(ii,0.0)) then
  xilli = ii
else
  xilr = ii
endif

10 continue

10 continue
   write(6,*) ' did not converge '
13 write(6,15) i1,sj2,Q(i1,0.0)
15 format(' i1,sj2,Q:',3f15.5)
call expand(epbar,X,i1)
sigx = press
sigy = press
sigz = i1 - sigx - sigy
return
end
A3. MATERIAL PARAMETER INPUT DATASET
FOR TRIAXIAL DRIVER

Bulk Modulus Parameters:  270833.0  0.00  0.0000
Shear Modulus Parameters:  125000.0  0.00  0.0000
Initial Failure Surface:  1150.0  1000.0  0.0006
Softening:  280.0  93.0
Hardening:  0.17  294.0
Cap Parameters:  0.60  0.0000220  0.1500  8000.
Max. Iterations, Epsilon:  200  0.0002
A4. MODIFIED NIKE2D CODE

C******************************************************************************
C
SUBROUTINE SSICS (A, IA, MODEL)
C
COMMON/BK02/IOOFC, IPHASE, IMASS, IPAR(9)
COMMON/BK06/NPRINT, MPRINT, LOCSTR, NUMELT, JPRINT, IDUMP
COMMON/BK16/MAXINT, HGC
COMMON/BK18/NUMMAT, ITYP2D, AKO(16)
COMMON/INTGP/D(4,4), IPT, NEL, NELSUB

DIMENSION A(1), IA(1)

IF (MODEL.EQ.0) RETURN
LN=MAXINT*LEPAR(9)
K08=IGTPNT(8)
MTP=IA(K08+NELSUB-1)
MM=IGTPNT(12)+48*(MTP-1)
II=IGTPNT(14)+LN*(NEL-1)
KK=IGTPNT(6) +NEL-1
NN=II+LEPAR(9)*(IPT-1)
NE=NN+LEPAR(9)-1
NT=NE-IDUMP
NU=IGTPNT(11)+NUMMAT+MTP-1
MM=IGTPNT(2) +4*(NEL-1)
K63=IGTPNT(63)
K64=IGTPNT(64)
K81=IGTPNT(81)
K82=IGTPNT(82)

GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160, rock
1  170), MODEL

10 CALL S1MAIN (A(NM), A(NN), A(NE), A(NT), LN)
RETURN
20 CALL S2MAIN (A(NM), A(NM+29), A(NN), A(NE), A(NT), LN)
RETURN
30 IF (ITYP2D.LE.1) THEN
    CALL S3MAIN (A(NM), A(NM+29), A(NN), A(NN+4), A(NE), LN)
ELSE
    CALL S3MAINP (A(NM), A(NM+29), A(NN), A(NN+4), A(NE), A(NT), LN)
ENDIF
RETURN
40 IF (ITYP2D.LE.1) THEN
    CALL S4MAIN (A(NM), A(NM), A(NN), A(NN+4), A(NN+5), A(K81+1), A(K82+1), A
1 (KK), A(NE), LN)
ELSE
    CALL S4MAINP (A(NM), A(NM), A(NN), A(NN+4), A(NN+5), A(K81+1), A(K82+1), A
1 (KK), A(NE), A(NT), LN)
ENDIF
RETURN
50 IF (ITYP2D.LE.1) THEN
    CALL S5MAIN (A(NM), A(NN), A(NN+4), A(NE), LN)
ELSE
    WRITE (6, 180)
    WRITE (59, 180)
    CALL ADIOS(2)
ENDIF
RETURN
60 CALL S6MAIN (A(NM), A(NN), A(NN+4), A(NE), A(NT), IN)
RETURN
70 CALL S7MAIN (A(NM), A(NM+29), A(MM), A(NN), A(NN+4), A(NN+5), A(K81+1), A(K82+1), A(KK), A(NE), A(NT), IN)
RETURN
80 IF (ITYP2D.LE.1) THEN
   CALL S8MAIN (A(NN), A(MM), A(NN), A(NN+4), A(NN+6), A(K81+1), A(K82+1), A(KK), A(NE), IN)
   ELSE
      WRITE (6, 180)
      WRITE (59, 180)
   ENDIF
RETURN
90 CALL S9MAIN (A(NM), A(NN), A(NE), A(NT), A(NU), IN)
RETURN
100 IF (ITYP2D.LE.1) THEN
   CALL S10MAIN (A(NM), A(NM+29), A(NN), A(NN+4), A(NE), IN)
   ELSE
      CALL S10MAINP (A(NM), A(NM+29), A(NN), A(NN+4), A(NE), A(NT), IN)
   ENDIF
RETURN
110 IF (ITYP2D.LE.1) THEN
   CALL S11MAIN (A(NM), A(NM+29), A(NN), A(NN+4), A(NE), IN)
   ELSE
      CALL S11MAINP (A(NM), A(NM+29), A(NN), A(NN+4), A(NE), A(NT), IN)
   ENDIF
RETURN
120 IF (ITYP2D.LE.1) THEN
   CALL S12MAIN (A(NM), A(MM), A(NN), A(NN+4), A(NN+5), A(K81+1), A(K82+1), A(KK), A(NE), IN)
   ELSE
      CALL S12MAINP (A(NM), A(MM), A(NN), A(NN+4), A(NN+5), A(K81+1), A(K82+1), A(KK), A(NE), A(NT), IN)
   ENDIF
RETURN
130 IF (ITYP2D.LE.1) THEN
   CALL S13MAIN (A(NM), A(NM+29), A(NN), A(NN+4), A(NE), IN, A(K63), A(K64))
   ELSE
      CALL S13MAINP (A(NM), A(NM+29), A(NN), A(NN+4), A(NE), A(NT), IN, A(K63), A(K64))
   ENDIF
RETURN
140 IF (ITYP2D.LE.1) THEN
   CALL S14MAIN (A(NM), A(NM+29), A(NN), A(NN+4), A(NE), IN)
   ELSE
      WRITE (59, 190)
      WRITE (6, 190)
   ENDIF
RETURN
150 RETURN
160 IF (ITYP2D.LE.1) THEN
   CALL S16MAIN (A(NM), A(MM), A(NN), A(NN+4), A(NN+5), A(K81+1), A(K82+1), A(KK), A(NE), IN, A(K63), A(K64))
   ELSE
      CALL S16MAINP (A(NM), A(MM), A(NN), A(NN+4), A(NN+5), A(K81+1), A(K82+1), A(KK), A(NE), A(K63), A(K64), A(NT))
   ENDIF
RETURN
170 IF (ITYP2D.LE.1) THEN
   rock
call s17main (a(nm),a(nn),a(nn+4),a(ne),ln)

else

write(6, 200)
write(59, 200)
call adios(2)
endif
return

180 FORMAT (' MODEL 5, SOIL AND CRUSHABLE FOAM, IS NOT VALID',/
1 ' IN PLANE STRESS--ERROR TERMINATION',/
1 ' VALID IN PLANE STRAIN OR PLANE STRESS--ERROR',/
1 ' IN PLANE STRESS--ERROR TERMINATION',/
1 END

SUBROUTINE S17MAIN (PROP, SIG, EPX, ENER, LN)

COMMON/BK02/IO0FC, IPHASE, IMASS, LPAR(9)
COMMON/INTGP/
COMMON/RANGE/MLT, MLT, LFT, LTT, NFTML
COMMON/VECT3/
2 DGT1 (128,4), DGT2 (128,4), DGT3 (128,4), DGT4 (128,4),
3 F11V (128), F22V (128), F12V (128), F21V (128), DSD5 (128),
4 SIG1S (128), SIG2S (128), SIG3S (128), SIG15S (128),
5 DDP1 (128,4), DDP2 (128,4), DDP3 (128,4), DDP4 (128,4), DDP5 (128,4)
COMMON/VECT8/
1 DSAVE (4,4,128)
COMMON/VECT14/
1 AJ2 (128), AK2 (128), AK (128), SCLE (128), DEPI (128), DEPS (128),
2 F (128), DAVG (128), AJ1 (128), T1 (128), T2 (128), T3 (128), T4 (128),
3 DA1 (128), DA2 (128), DA3 (128), DA4 (128), QHS (128), DEFN (128),
4 SCALE1 (128), SCALE0 (128)
COMMON/VECT15/
1 D1 (128), D2 (128), D3 (128), D4 (128)

real*8 aki, ak1, ak2, ag1, ag2, cl, c2, c3, c4, c5, c6, c7, ac,xm
1 rcap, cap1, cap2, Xbeg, epsilon, ep (6), rz
real*8 sigz, sigy, sigz, depz, depz, depx, depx, depy, epbar, X
real*8 ac, xm
COMMON/moduli/ aki, ak1, ak2, ag1, ag2
COMMON/failc/ cl, c2, c3, c4, c5, c6, c7
COMMON/cap/ rcap, cap1, cap2, Xbeg
COMMON/convg/ miter, epsilon
COMMON/dtype/ mtype

DIMENSION PROP (1), SIG (IN,1), EPX (IN,1), ENER (IN,1)
DIMENSION jtype (128), sigi (4,128), ebari (128), Xi (128), xrr (128)
DIMENSION sigm (3,3), epp (3,3), cc (3,3,3,3)
DIMENSION aa (4,4), ac (4,4), dsig (4,4), xin (4)

aki = prop(1)
ak1 = prop(2)
ak2 = prop(3)
agi = prop(4)
ag1 = prop(5)
ag2 = prop(6)
c1 = prop(7)
c2 = prop(8)
c3 = prop(9)  
   rock
   c4 = prop(10)  
   rock
   c5 = prop(11)  
   rock
   c6 = prop(12)  
   rock
   c7 = prop(13)  
   rock
   recap = prop(14)  
   rock
   cap1 = prop(15)  
   rock
   cap2 = prop(16)  
   rock
   Xbeg = prop(17)  
   rock
   mxiter = int(prop(18))  
   rock
   epsilon = prop(19)  
   rock
C
   CALL ROTAT1 (SIG,ENER,LN)  
   rock
   DO 140 I=MFT,MLT  
         sigx = -sig(1,i)  
         sigy = -sig(2,i)  
         sigz = -sig(3,i)  
         sigxy = -sig(4,i)  
         epbar = expx(3,i)  
         X = expx(5,i)  
         depx = -d1(i)  
         depy = -d2(i)  
         depz = -d3(i)  
         depxy = -d4(i)/2.0  
         call premat(depx,depy,depz,depxy,0.0d0,0.0d0,epbar,
                      X,sigx,sigy,sigz,sigxy,0.0d0,0.0d0,ep,rr)  
   c
   jtype(i) = mtype  
   epx(6,i) = float(mtype)  
   epx(3,i) = epbar  
   epx(5,i) = X  
   epx(7,i) = epx(7,i) + ep(1)  
   epx(8,i) = epx(8,i) + ep(2)  
   epx(9,i) = epx(9,i) + ep(3)  
   epx(10,i) = epx(10,i) + ep(4)  
   rrr(i) = rr  
C
   SIG(1,I) = -sigx  
   SIG(2,I) = -sigy  
   SIG(3,I) = -sigz  
   SIG(4,I) = -sigxy  
C
   sigi(1,i) = -sigx  
   sigi(2,i) = -sigy  
   sigi(3,i) = -sigz  
   sigi(4,i) = -sigxy  
   epbar(1) = epx(3,i)  
   Xi(i) = expx(5,i)  
C
   c............Rotate stress back to deformed state  
C
   call rotat2s(sigi(1,i),sigi(2,i),sigi(4,i),i)  
C
   140 continue  
C
   IF (IPHASE.EQ.3) GO TO 200  
C
   c.....Calculate Tangent Constitutive Matrix  
C
   do 990 i=mft,mlt
c........If increment was elastic, directly define matrix
    if (jtype(i).eq.1) then
      press = (sisi(1,i)+sisi(2,i)+sisi(3,i))/3.0
      sx = sisi(1,i) - press
      sy = sisi(2,i) - press
      sx = sisi(4,i)
      sx1 = -press*3.0
      sx2 = sqrt(sx*sxx+sy*sy+y+y+sx*sxy)
      bmod = aki*(1.0-ak1*exp(-aktwo*x1))/((1.0-ak1)
      smod = agi*(1.0-ag1*exp(-ag2*agj))/((1.0-ag1)
      dsave(1,1,i) = bmod + 4.0/3.0*smod
      dsave(1,2,i) = bmod - 2.0/3.0*smod
      dsave(1,3,i) = bmod - 2.0/3.0*smod
      dsave(1,4,i) = 0.0
      dsave(2,1,i) = dsave(1,2,i)
      dsave(2,2,i) = bmod + 4.0/3.0*smod
      dsave(2,3,i) = bmod - 2.0/3.0*smod
      dsave(2,4,i) = 0.0
      dsave(3,1,i) = dsave(1,3,i)
      dsave(3,2,i) = dsave(2,3,i)
      dsave(3,3,i) = bmod + 4.0/3.0*smod
      dsave(3,4,i) = 0.0
      dsave(4,1,i) = dsave(1,4,i)
      dsave(4,2,i) = dsave(2,4,i)
      dsave(4,3,i) = dsave(3,4,i)
      dsave(4,4,i) = smod
    endif
    c........Calculate constitutive matrix for increments on
    c........Failure surface
    if (jtype(i).eq.2) then
      press = (sisi(1,i)+sisi(2,i)+sisi(3,i))/3.0
      sx = sisi(1,i) - press
      sy = sisi(2,i) - press
      sx = sisi(4,i)
      sx1 = -press*3.0
      sx2 = sqrt(sx*sxx+sy*sy+y+y+sx*sxy)
      bmod = aki*(1.0-ak1*exp(-aktwo*x1))/((1.0-ak1)
      smod = agi*(1.0-ag1*exp(-ag2*agj))/((1.0-ag1)
      do 6992 j=1,3
        do 6992 k=1,3
          sigm(j,k) = 0.0
          epp(j,k) = 0.0
        continue
      6992
      sigm(1,1) = -sisi(1,i)
      sigm(1,2) = -sisi(4,i)
      sigm(2,1) = sigm(1,2)
      sigm(2,2) = -sisi(2,i)
      sigm(3,3) = -sisi(3,i)
      epp(1,1) = epx(7,i)
epp(1,2) = epx(10,i)  
  epp(2,1) = epp(1,2)  
  epp(2,2) = epx(8,i)  
  epp(3,3) = epx(9,i)  

  c
call cmath(bmod,smod,rrr(i),sigm,epp(3,i),cc)  
  c
do 2500 j=1,3  
    dsave(j,1,i) = cc(j,j,1,2)  
    dsave(4,j,i) = dsave(j,4,i)  
    do 2500 k=1,3  
      dsave(j,k,i) = cc(j,j,k,k)  
    2500continue  
    dsave(4,4,i) = cc(1,2,1,2)  
  c  
  endif  
  c
c........Calculate constitutive matrix for increments on  
  c........Cap surface  
c........if (jtype(i).eq.3) then  
  c
    press = (sigt(j,1,i)+sigt(j,2,i)+sigt(3,i))/3.0  
    sxx = sigt(j,1,i) - press  
    syy = sigt(j,2,i) - press  
    sxy = sigt(j,4,i)  
    xil = -press*3.0  
    sj2 = sqrt(sxx*sxx+syy*syy+sxx*syy+sxy*sxy)  
  c
   bmod = aki*(1.0-ak1*exp(-aktwo*xil))/((1.0-ak1)*exp(-ak2*sj2))/(1.0-ak1)  
   smod = agi*(1.0-ag1*exp(-ag2*sj2))/(1.0-ag1)  
  c  
  do 6997 j=1,3  
    do 6997 k=1,3  
      sigm(j,k) = 0.0  
    6997continue  
  c
  sigm(1,1) = -sigt(j,1,i)  
  sigm(1,2) = -sigt(j,4,i)  
  sigm(2,1) = sigm(1,2)  
  sigm(2,2) = -sigt(j,2,i)  
  sigm(3,3) = -sigt(3,i)  
  c
  XX = epx(5,i)  
  c
call cmath(bmod,smod,rrr(i),sigm,XX,cc)  
  c
do 2505 j=1,3  
  dsave(j,4,i) = cc(j,j,1,2)  
  dsave(4,j,i) = dsave(j,4,i)  
  do 2505 k=1,3  
  dsave(j,k,i) = cc(j,j,k,k)  
  2505continue  
  dsave(4,4,i) = cc(1,2,1,2)  
  c  
  endif  
  c
c........Numerically approximate the constitutive matrix when  
  c........at the corner  
c........if (jtype(i).eq.4) then  
  c
dave = (abs(d1(i))+abs(d2(i))+abs(d3(i))+abs(d4(i)))/4.0  
factor = 0.0001  
6995 if (dave*factor.lt.0.0000005) then  
    factor = factor * 10.0  
goto 6995  
endif  
c  
do 7000 kk=1,4  
c    if (kk.eq.1) then  
        xm(1) = 1.0  
        xm(2) = 1.0  
        xm(3) = 1.0  
        xm(4) = 1.0  
    endif  
c    if (kk.eq.2) then  
        xm(1) = 0.92  
        xm(2) = 1.22  
        xm(3) = 1.12  
        xm(4) = 0.82  
    endif  
c    if (kk.eq.3) then  
        xm(1) = 0.84  
        xm(2) = 0.94  
        xm(3) = 1.24  
        xm(4) = 1.14  
    endif  
c    if (kk.eq.4) then  
        xm(1) = 1.17  
        xm(2) = 0.87  
        xm(3) = 0.97  
        xm(4) = 1.27  
    endif  
c    aa(kk,1) = d1(i)  
    aa(kk,2) = d2(i)  
    aa(kk,3) = d3(i)  
    aa(kk,4) = d4(i)  
    if (aa(kk,kk).eq.0) aa(kk,kk) = dave  
    aa(kk,kk) = aa(kk,kk)*10.0  
c  
do 6998 kki=1,4  
    aa(kk,kki) = aa(kk,kki)*factor*xm(kki)  
6998 continue  
c  
sigx = -sigi(1,i)  
sigy = -sigi(2,i)  
sigz = -sigi(3,i)  
sigxy = -sigi(4,i)  
epbar = epbari(i)  
X = Xi(i)  
c  
depx = -aa(kk,1)  
depy = -aa(kk,2)  
depz = -aa(kk,3)  
depxy = -aa(kk,4)/2.0  
c  
call premat(depx,depy,depz,depxy,0.0d0,0.0d0,epbar,  
1       X,sigx,sigy,sigz,sigxy,0.0d0,0.0d0,ep,rr)
```
c     write(8,*) ' Mtype = ',mtype

c     dsig(1,kk) = -sigt - sig(1,i)
     dsig(2,kk) = -sigt - sig(2,i)
     dsig(3,kk) = -sigt - sig(3,i)
     dsig(4,kk) = -sigt - sig(4,i)

c     7000 continue

   do 7100 kk=1,4

     do 7010 kki=1,4
     xm(kki) = dsig(kk,kki)
     do 7010 kkj=1,4
         ac(kki,kkj) = aa(kki,kkj)*aki
     7010 continue

   call solsys(ac,xm,4,4,ind)

   if (ind.eq.1) then
     write(6,*) ' WARNING: Singular matrix in solsys.'
     write(8,*) ' WARNING: Singular matrix in solsys.'
   endif

   do 7020 kki=1,4
     dsave(kk,kki,i) = xm(kki)*aki
   7020 continue

   dsave(1,2,i) = (dsave(1,2,i)+dsave(2,1,i))/2.0
   dsave(2,1,i) = dsave(1,2,i)
   dsave(1,3,i) = (dsave(1,3,i)+dsave(3,1,i))/2.0
   dsave(3,1,i) = dsave(1,3,i)
   dsave(1,4,i) = (dsave(1,4,i)+dsave(4,1,i))/2.0
   dsave(4,1,i) = dsave(1,4,i)
   dsave(2,3,i) = (dsave(2,3,i)+dsave(3,2,i))/2.0
   dsave(3,2,i) = dsave(2,3,i)
   dsave(2,4,i) = (dsave(2,4,i)+dsave(4,2,i))/2.0
   dsave(4,2,i) = dsave(2,4,i)
   dsave(3,4,i) = (dsave(3,4,i)+dsave(4,3,i))/2.0
   dsave(4,3,i) = dsave(3,4,i)
   endif

   990 continue

   200 CALL ROTAT2 (SIG,ENER,LN)
   DO 210 I=MFT,MLT
     SIG11(I) = SIG(1,I)
     SIG22(I) = SIG(2,I)
     SIG33(I) = SIG(3,I)
     SIG12(I) = SIG(4,I)
   210 continue
   return
end
```

```
subroutine s17out(sig, epx, thick)
    common/bk18/nummat, ityp2d, ako(20)
    common/bk48/stress(4), strain(4), c(4,4), ipt, nel, nstate
    dimension sig(1), epx(1)
    do 10 i=1,4
        stress(i) = sig(i)
    end
    do 10 continue
        nstate = int(epx(6))
        if (nstate.gt.1) nstate = nstate + 1
        strain(1) = epx(3)
        if (ityp2d.eq.2) stress(3) = thick
    end
    return
end

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

Subroutine Premat

This subroutine sequentially calls 'Matlaw', when the input
strain increment is too large.

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

subroutine premat(depsx, depsy, depsz, depsxy, depsyz, depsxz, ebar,
                   X, sigx, sigy, sigz, sigxy, sigyz, sigxz, ep, rr)
    implicit real*8 (a-h,o-z)
    dimension ep(6), epp(6)
    do 100 i=1,6
        ep(i) = 0.0
    end
    rr = 0.0
    eppmax = 0.0
    if (dabs(depsx).gt.eppmax) eppmax = dabs(depsx)
    if (dabs(depsy).gt.eppmax) eppmax = dabs(depsy)
    if (dabs(depsz).gt.eppmax) eppmax = dabs(depsz)
    if (dabs(depsxy).gt.eppmax) eppmax = dabs(depsxy)
    if (dabs(depsyz).gt.eppmax) eppmax = dabs(depsyz)
    if (dabs(depsxz).gt.eppmax) eppmax = dabs(depsxz)
    i = int(eppmax/0.0008) + 1
    xi = float(i)
    depsx = depsx/xi
    depsy = depsy/xi
    depsz = depsz/xi
    depsxy = depsxy/xi
    depsyz = depsyz/xi
    depsxz = depsxz/xi
    do 200 j=1,i
        call matlaw(depsx, depsy, depsz, depsxy, depsyz, depsxz, ebar,
                    X, sigx, sigy, sigz, sigxy, sigyz, sigxz, ep, rr)
    end
    do 110 k=1,6
        ...
ep(k) = ep(k) + epp(k)
continue

rr = rr + rrr/ksi
continue
return
end

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

.....Dirac Delta Function

function del(i,j)
if (i.eq.j) then
  del = 1.0
else
  del = 0.0
endif
return
end

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

.....Subroutine to calculate constitutive matrix for failure
.....surface increments

real*8 c1, c2, c3, c4, c5, c6, c7
common/failc/ c1, c2, c3, c4, c5, c6, c7
dimension sig(3,3), ep(3,3), c(3,3,3,3), d(3,3,3), dfdsig(3,3),
dkdep(3,3), s(3,3), q1(3,3), q2(3,3)

real j2, il

.....Calculate elastic constitutive matrix

do 100 i=1,3
  do 100 j=1,3
    do 100 k=1,3
      d(i,j,k,l) = (blk-2.0/3.0*g)*del(i,j)*del(k,l)
      + g*(del(i,k)*del(j,l)+del(i,l)*del(j,k))
    end do
  end do
100 continue

.....Calculate deviatoric stresses

sigkk = sig(1,1) + sig(2,2) + sig(3,3)
il = sigkk
do 200 i=1,3
  do 200 j=1,3
    s(i,j) = sig(i,j) - 1.0/3.0*sigkk*del(i,j)
  end do
200 continue

.....Calculate second invariant of deviatoric stress tensor

j2 = 0.0
do 210 i=1,3
  do 210 j=1,3
    j2 = j2 + 0.5*s(i,j)*s(i,j)
  end do
210 continue
210 continue
  c
  c.....Calculate partial of failure surface function with respect
  c..... to stress tensor components
  c
  do 220 i=1,3
    do 220 j=1,3
      dfdsig(i,j) = s(i,j)/2.0/sqrt(j2)
      1 = -(c2*c3*exp(-c3*i1)+c6*(1.0-exp(-c7*epbar)))*del(i,j)
  220 continue
  c
  c.....Calculate partial of kappa(epbar) with respect
  c..... to plastic strain tensor components
  c
  do 230 i=1,3
    do 230 j=1,3
      dkdep(i,j) = 2.0/3.0*ep(i,j)/epbar
  230 continue
  c
  c.....Calculate first term of the denominator
  c
  d1 = 0.0
  do 240 i=1,3
    do 240 j=1,3
      do 240 k=1,3
        do 240 l=1,3
          d1 = d1 + dfdsig(i,j)*d(i,j,k,l)*dfdsig(k,l)
  240 continue
  c
  c.....Calculate second term of the denominator
  c
  dfdk = -(c6*c7*exp(-c7*epbar)*i1-c4*c5*exp(-c5*epbar))
  d2 = 0.0
  do 250 i=1,3
    do 250 j=1,3
      d2 = d2 + dfdk*dkdep(i,j)*dfdsig(i,j)
  250 continue
  c
  denom = d1 - d2
  c
  c.....Calculate numerator
  c
  do 260 i=1,3
    do 260 j=1,3
      q1(i,j) = 0.0
    do 260 k=1,3
      do 260 l=1,3
        q1(i,j) = q1(i,j) + d(i,j,k,l)*dfdsig(k,l)
  260 continue
  c
  do 270 k=1,3
    do 270 l=1,3
      q2(k,l) = 0.0
    do 270 i=1,3
      do 270 j=1,3
        q2(k,l) = q2(k,l) + dfdsig(i,j)*d(i,j,k,l)
  270 continue
  c
  c.....Calculate the constitutive matrix
  c
  do 300 i=1,3
    do 300 j=1,3
      rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock

rock
do 300 k=1,3
  do 300 i=1,3
    c(i,j,k,l) = d(i,j,k,l) - (1.0-x)*q1(i,j)*q2(k,l)/denom
  300 continue

return
end

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

*****Subroutine to calculate constitutive matrix for cap
*****surface increments

subroutine cmats(blk,g,r,sig,X,c)
  real*8 rcap,cap1,cap2,Xbeg
  common/cap// rcap,cap1,cap2,Xbeg
  dimension sig(3,3),c(3,3,3,3),d(3,3,3,3),dfdsig(3,3),
          dkdep(3,3),s(3,3),q1(3,3),q2(3,3)
  real j2,il

*****Calculate elastic constitutive matrix

  do 100 i=1,3
    do 100 j=1,3
      do 100 l=1,3
        d(i,j,k,l) = (blk-2.0/3.0*g)*del(i,j)*del(k,l)
        + g*(del(i,k)*del(j,l)+del(i,l)*del(j,k))
      100 continue

*****Calculate deviatoric stresses

  sigkk = sig(1,1) + sig(2,2) + sig(3,3)
  il = sigkk
  if (il.gt.X) il = X
  do 200 i=1,3
    do 200 j=1,3
      s(i,j) = sig(i,j) - 1.0/3.0*sigkk*del(i,j)
    200 continue

*****Calculate second invariant of deviatoric stress tensor

  j2 = 0.0
  do 210 i=1,3
    do 210 j=1,3
      j2 = j2 + 0.5*s(i,j)*s(i,j)
    210 continue

*****Calculate partial of failure surface function with respect
*****to stress tensor components

  do 220 i=1,3
    do 220 j=1,3
      dfdsig(i,j) = s(i,j)/2.0/sqrt(j2)
        +(rcap*il/sqrt(X*il*il))*del(i,j)
    220 continue

*****Calculate partial of kappa (X) with respect
*****to plastic strain tensor components

epkk = cap2*(1.0-exp(-cap1*(X-Xbeg)))
do 230  i=1,3
do 230  j=1,3
dkdep(i,j) = del(i,j)/(cap1*(cap2-epkk))
230 continue
c.....Calculate first term of the denominator
c
d1 = 0.0
do 240  i=1,3
do 240  j=1,3
do 240  k=1,3
do 240  l=1,3
d1 = d1 + dfdsig(i,j)*d(i,j,k,l)*dfdsig(k,l)
240 continue
c.....Calculate second term of the denominator
c
dfdk = -rcap*X/sqrt(X*X-ii*ii)
d2 = 0.0
do 250  i=1,3
do 250  j=1,3
d2 = d2 + dfdk*dkdep(i,j)*dfdsig(i,j)
250 continue
c
denom = d1 - d2
c.....Calculate numerator
c
do 260  i=1,3
do 260  j=1,3
q1(i,j) = 0.0
do 260  k=1,3
do 260  l=1,3
q1(i,j) = q1(i,j) + d(i,j,k,l)*dfdsig(k,l)
260 continue

do 270  k=1,3
do 270  l=1,3
q2(k,l) = 0.0
do 270  i=1,3
do 270  j=1,3
q2(k,l) = q2(k,l) + dfdsig(i,j)*d(i,j,k,l)
270 continue

c.....Calculate the constitutive matrix
c
do 300  i=1,3
do 300  j=1,3
do 300  k=1,3
do 300  l=1,3
c(i,j,k,l) = d(i,j,k,l)
c1 = (1.0-r)*q1(i,j)*q2(k,l)/denom
300 continue
c
SUBROUTINE SOLSYS(A,C,NN,M,IND)
C PURPOSE: SOLUTION OF THE SYSTEM OF LINEAR EQUATIONS
C
C AX=C........ (1)
C
C TECHNIQUE: GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
C
C ARGUMENTS:
C
A IS AN NN X NN MATRIX
C IS A NN VECTOR
NN IS THE DIMENSION OF SYSTEM (1); 1<=NN<=M
M IS THE MAXIMUM VALUE OF NN; 1<=M<=20
IND IS THE ERROR INDICATOR
IF IND=0.....NO ERROR
IF IND=1.....SYSTEM (1) IS SINGULAR
NOTE: THE SOLUTION OF SYSTEM (1) IS GIVEN BACK IN VECTOR C.

C

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(M,1),C(1),B(20),IV(20)

C

INITIALIZATION OF VARIABLES

IND=0
IF (NN.EQ.1) GOTO 10
N=NN-1
DO 1 I=1,NN
  IV(I)=I
  B(I)=C(I)
1

C

LU DECOMPOSITION OF MATRIX A

DO 6 K=1,N
  KK=K+1

C

SEARCHING FOR THE PIVOT IN COLUMN K

RMAX=DABS(A(IV(K),K))
IRHO=K
DO 2 I=KK,NN
  D=DABS(A(IV(I),K))
  IF (RMAX.GE.D) GOTO 2
  RMAX=D
  IRHO=I
2 CONTINUE
  IF (RMAX.EQ.0.0D0) GOTO 9
  IF (IRHO.EQ.K) GOTO 3
  IMAX=IV(K)
  IV(K)=IV(IRHO)
  IV(IRHO)=IMAX
3 IK=IV(K)

C

ONE STEP OF LU DECOMPOSITION

DO 4 I=KK,NN
  II=IV(I)
  A(II,K)=A(II,K)/A(II,K)
  B(II)=B(II)-A(II,K)*B(K)
4 DO 5 J=KK,NN

C


DO 5 I=KK,NN
   II=IV(I)
   5 A(I,J)=A(I,J)-A(I,K)*A(IK,J)
   CONTINUE
   IN=IV(NN)
   IF (A(IN,NN) .EQ. 0.0D0) GOTO 9
   BACK SUBSTITUTION
   C(NN)=B(IN)/A(IN,NN)
   DO 8 I=1,N
      J=NN-I
      IJ=IV(J)
      SUM=0.0D0
      JJ=J+1
      DO 7 K=JJ,NN
         SUM=SUM+A(IJ,K)*C(K)
      7 CONTINUE
      C(J)=(B(IJ)-SUM)/A(IJ,J)
   8 RETURN
   CONTINUE
   IF (A(1,1) .EQ. 0.0D0) GOTO 9
   C(1)=C(1)/A(1,1)
   RETURN
   9 IND=1
   RETURN
END

C*************************************************************************
C SUBROUTINE MATIN (MATYPE, DEN, THICK, PROP, IDUMP)
C*************************************************************************
C READ IN MATERIAL PROPERTY CARDS
C COMMON/BKO2/100FC, IPHASE, IMASS, IPAR(9)
COMMON/BKO3/NUMDC, IMASSN, IDAMPN, IROLLER, IPARST
COMMON/BKO6/MAXINT
COMMON/BK8/NUMMAT, ITYP2D, AR0(20)
cfjc DIMENSION MATYPE(1), DEN(1), THICK(1), PROP(48,1), HEADING(12),
cfjc 1 NCON(20)
dimension matype(1), den(1), thiek(1), prop(48,1), ncon(20)  fjc
character*6 heading(12)  fjc
EQUIVALENCE (MODEL, IPAR(1))
DATA NCON/4,4,8,8,8,8,8,8,4,5,10,8,6,9,4,9,14,4,0/
   CALL HEADER
WRITE (6,110)
IPAR(9)=0
BULKMX=0.
DO 10 I=1,NUMMAT
   READ (5,20) N,MATYPE(N),DEN(N),THICK(N)
   READ (5,40) HEADING
   IF (ITYP2D.EQ.2.AND.THIK(N).EQ.0.) THICK(N)=1.0
   IF (MATYPE(N).EQ.1) READ (5,30) (PROP(J,N),J=1,6)
   IF (MATYPE(N).EQ.2) READ (5,60) (PROP(J,N),J=1,11)
   IF (MATYPE(N).EQ.3) READ (5,80) (PROP(J,N),J=1,21)
   IF (MATYPE(N).EQ.4) READ (5,70) (PROP(J,N),J=1,48)
   IF (MATYPE(N).EQ.5) READ (5,50) (PROP(J,N),J=1,5), (PROP(J,N),J=47,
   1 258), (PROP(J,N),J=6,20)
   IF (MATYPE(N).EQ.6) READ (5,90) (PROP(J,N),J=1,6)
   IF (MATYPE(N).EQ.7) READ (5,100) (PROP(J,N),J=1,6), (PROP(J,N),J=16
1, 18), (PROP (J, N), J=7, 11)

IF (MATYPE(N), EQ. 8) READ (5, 70) (PROP (J, N), J=1, 48)
IF (MATYPE(N), EQ. 9) READ (5, 30) (PROP (J, N), J=1, 6)
IF (MATYPE(N), EQ. 10) READ (5, 70) (PROP (J, N), J=1, 48)
IF (MATYPE(N), EQ. 11) READ (5, 70) (PROP (J, N), J=1, 48)
IF (MATYPE(N), EQ. 12) READ (5, 70) (PROP (J, N), J=1, 48)
IF (MATYPE(N), EQ. 13) READ (5, 70) (PROP (J, N), J=1, 48)
IF (MATYPE(N), EQ. 14) READ (5, 80) (PROP (J, N), J=1, 21)
IF (MATYPE(N), EQ. 16) READ (5, 70) (PROP (J, N), J=1, 48)
if (matype(n).eq.17) read (5,105) (prop(j,n),j=1,19)

C

LPAR(1)=MATYPE(N)
LPAR(9)=MAXO(NCON(MODEL)+IDUMP+MAXD(0,ITYP2D-1),LPAR(9))
CALL PRINTM(N,DEN(N),THICK(N),PROP(1,N),HEADING)
10 CALL BLKMAX(MATYPE,N,PROP,BULKMX)
PENSF=BULKMX

C

RETURN

C

20 FORMAT(2I5,2E10.0)
30 FORMAT(E10.0)
40 FORMAT(12X)
50 FORMAT(7E10.0/5E10.0/5E10.0/5E10.0/5E10.0/5E10.0)
60 FORMAT(3E10.0/3E10.0/3E10.0/3E10.0/3E10.0/3E10.0)
70 FORMAT(8E10.0)
80 FORMAT(E10.0/8E10.0/8E10.0/8E10.0/8E10.0/8E10.0)
90 FORMAT(8E10.0/8E10.0/8E10.0/8E10.0/8E10.0/8E10.0)
100 FORMAT(3E10.0/3E10.0/3E10.0/3E10.0/3E10.0/3E10.0)
105 FORMAT(3E10.0/3E10.0/3E10.0/3E10.0/3E10.0/3E10.0)
110 FORMAT(//"MATERIAL DEFINITIONS"/

C

1 ' MATERIAL MODELS /
2  ' EQ.1: ISOTROPIC /
3  ' EQ.2: ORTHOTROPIC /
4  ' EQ.3: ELASTOPLASTIC (VON MISES) /
5  ' EQ.4: THERMO-ELASTIC-PLASTIC /
6  ' EQ.5: SOIL AND CRUSHABLE FOAM /
7  ' EQ.6: VISCOELASTIC /
8  ' EQ.7: THERMO-ORTHOTROPIC /
9  ' EQ.8: THERMO-ELASTIC-CREEP /
10  ' EQ.9: RUBBER (BLATZ-KO) /
11  ' EQ.10: POWER LAW PLASTICITY /
12  ' EQ.11: UNIFIED CREEP PLASTICITY /
13  ' EQ.12: POWER LAW THERMO-PLASTIC /
14  ' EQ.13: STRAIN RATE SENSITIVE PLASTICITY /
15  ' EQ.14: AXISYMMETRIC SEGMENTED PLASTICITY /
16  ' EQ.15: THERMO-ELASTIC-PLASTIC-CREEP /
17  ' EQ.17: STRAIN-SOFTENING ROCK /
END

C

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

SUBROUTINE PRINTM (N, DEN, THICK, PROP, HEADING)

COMMON/BK02/IOOF, ICASE, IMASS, MODEL, LPAR(8)
DIMENSION DEN(1), PROP(1), HEADING(12)
character*5 heading

WRITE (6,100) HEADING, N, MODEL, DEN(1), THICK

GO TO (10,20,30,40,50,60,70,80,90,94,96,
     1 97,98,30,10,99,92), MODEL
10 WRITE (6,110) (PROP(I),I=1,2)
   CALL SETSE1 (PROP)
   RETURN

C
20 WRITE (6,120) (PROP(I),I=1,11)
   CALL SETSE2 (PROP,PROP(30))
   RETURN

C
30 WRITE (6,130) (PROP(I),I=1,21)
   PROP(30)=PROP(1)
   PROP(31)=PROP(2)
   CALL SETSE1 (PROP(30))
   RETURN

C
40 WRITE (6,140) (PROP(I),I=1,48)
   CALL SETSE4 (PROP)
   RETURN

C
50 CALL SETSE5 (PROP,MODEL)
   WRITE (6,150) (PROP(I),I=1,6),PROP(48), (PROP(10+I),I=1,9), (PROP(20
   1 +I),I=1,9)
   RETURN

C
60 WRITE (6,160) (PROP(I),I=1,5)
   RETURN

C
70 WRITE (6,170) (PROP(I),I=1,6), (PROP(I),I=16,18), (PROP(I),I=7,11)
   CALL SETSE2 (PROP,PROP(30))
   RETURN

C
80 WRITE (6,240) (PROP(I),I=1,48)
   RETURN

C
90 WRITE (6,190) PROP(I)
   RETURN

C
94 WRITE (6,200) (PROP(I),I=1,4)
   PROP(30)=PROP(1)
   PROP(31)=PROP(2)
   CALL SETSE1 (PROP(30))
   RETURN

C
C
C
MODEL = 11 BANNAN'S MODEL CALL 415 422-2585 FOR INFO.
C
ASK FOR DOUG
C
96 WRITE (6,210) (PROP(I),I=1,5), (PROP(I),I=9,12),
1 (PROP(I),I=17,20), (PROP(I),I=25,28)
   PROP(30)=PROP(1)
   PROP(31)=PROP(2)
   CALL SETSE1 (PROP(30))
   RETURN

C
97 WRITE (6,220) (PROP(I),I=1,48)
   CALL SETSE4 (PROP)
   RETURN

C
98 WRITE (6,230) (PROP(I),I=1,4)
   PROP(30)=PROP(1)
   PROP(31)=PROP(2)
   CALL SETSE1 (PROP(30))
   RETURN
C
99 WRITE (6,180) PROP(I), (PROP(I), I=1,16)
RETURN
C......Claborn's Rock Model
C......Reference: F. J. Claborn, "Development of a Strain Softening
C...... Constitutive Model for Rock", Rice University Thesis,
C
92 WRITE (6,250) (prop(i),i=1,17),int(prop(18)),prop(19)
RETURN
C
100 FORMAT(/1X,12A6,' MATERIAL CONSTANTS SET NUMBER .... ','I5,
1 4X,' MATERIAL MODEL .... ','I5//
2 5X,'DEN .................................. =', E12.4/
3 5X,'THICKNESS (PLANE STRESS) ........ =', E12.4)
110 FORMAT(
1 5X,'E ................................... =', E12.4/
2 5X,'VNU ................................ =', E12.4)
120 FORMAT(
1 5X,'E(A) .................................. =', E12.4/
2 5X,'E(B) .................................. =', E12.4/
3 5X,'E(C) .................................. =', E12.4/
4 5X,'VNU(AB) ................................ =', E12.4/
5 5X,'VNU(AC) ................................ =', E12.4/
6 5X,'VNU(BC) ................................ =', E12.4/
7 5X,'G(AB) .................................. =', E12.4/
8 5X,'MATERIAL AXES OPTION .......... =', E12.4/
9 5X,'YC (OPTION =1.0) .................. =', E12.4/
$ 5X,'ZC (OPTION =1.0) .................. =', E12.4/
$ 5X,'GAMMA (OPTION =2.0) ............. =', E12.4)
130 FORMAT(
1 5X,'E ................................... =', E12.4/
2 5X,'VNU ................................ =', E12.4/
3 5X,'YIELD ................................ =', E12.4/
4 5X,'E (HARDEN) ......................... =', E12.4/
5 5X,'HARDENING PARAMETER ............. =', E12.4/
6 5X,'EFFECTIVE PLASTIC STRAIN ........ =8(E9.2,1X)/
7 5X,'EFFECTIVE STRESS ................. =8(E9.2,1X))
140 FORMAT(
1 5X,'TEMP ................................ =8(E9.2,1X)/
2 5X,'E ................................... =8(E9.2,1X)/
3 5X,'VNU ................................ =8(E9.2,1X)/
4 5X,'ALPHA ................................ =8(E9.2,1X)/
5 5X,'YIELD ................................ =8(E9.2,1X)/
6 5X,'E (HARDEN) ......................... =8(E9.2,1X))
150 FORMAT(
1 5X,'SHEAR ................................ =', E12.4/
2 5X,'BULK ................................ =', E12.4/
3 5X,'A0 .................................... =', E12.4/
4 5X,'A1 .................................... =', E12.4/
5 5X,'A2 .................................... =', E12.4/
6 5X,'PRESSURE CUTOFF .................... =', E12.4/
7 5X,'UNLOADING PARAMETER .............. =', E12.4/
8 5X,' EQ.0, VOLUMETRIC CRUSHING ..... 
9 5X,' EQ.1, NO VOLUMETRIC CRUSHING . 
$ 5X,'IN (V/V0) .......................... =9(E8.1,1X)/
$ 5X,'PRESSURE .......................... =9(E8.1,1X))
160 FORMAT(
1 5X,'BULK ................................ =', E12.4/
2 5X,'VNU ................................ =', E12.4/
3 5X,'SHORT TIME SHEAR MODULUS           =', E12.4/
4 5X,'LONG TIME SHEAR MODULUS           =', E12.4/
5 5X,'DECAY CONSTANT                    =', E12.4/)

170 FORMAT(
  1 5X,'E(A)                                =', E12.4/
  2 5X,'E(B)                                =', E12.4/
  3 5X,'E(C)                                =', E12.4/
  4 5X,'VNU(AB)                             =', E12.4/
  5 5X,'VNU(AC)                             =', E12.4/
  6 5X,'VNU(BC)                             =', E12.4/
  7 5X,'ALPHA(A)                            =', E12.4/
  8 5X,'ALPHA(B)                            =', E12.4/
  9 5X,'ALPHA(C)                            =', E12.4/
  $ 5X,'G(AB)                               =', E12.4/
  $ 5X,'MATERIAL AXES OPTION               =', E12.4/
  $ 5X,'YC (OPTION =1.0)                    =', E12.4/
  $ 5X,'ZC (OPTION =1.0)                    =', E12.4/
  $ 5X,'GAMMA (OPTION =2.0)                 =', E12.4/

180 FORMAT(
  1 5X,'BULK MODULUS                      =', E12.4/
  2 5X,'LOAD CURVE FOR YOUNG'S MODULUS     =', E12.4/
  3 5X,'LOAD CURVE FOR POISSONS RATIO      =', E12.4/
  4 5X,'LOAD CURVE FOR YIELD STRENGTH      =', E12.4/
  5 5X,'LOAD CURVE FOR HARDENING MODULUS   =', E12.4/
  6 5X,'LOAD CURVE FOR A                   =', E12.4/
  7 5X,'LOAD CURVE FOR M                   =', E12.4/
  8 5X,'LOAD CURVE FOR N                   =', E12.4/)

190 FORMAT(
  1 5X,'SHEAR                                 =', E12.4/

200 FORMAT(
  1 5X,'E                                      =', E12.4/
  2 5X,'VNU                                    =', E12.4/
  3 5X,'K                                      =', E12.4/
  3 5X,'M                                      =', E12.4/)

210 FORMAT(
  & 5X,'E                                      =', E12.4/
  & 5X,'VNU                                    =', E12.4/
  & 5X,'INITIAL TEMPERATURE                    =', E12.4/
  & 5X,'HEAT GENERATION COEFFICIENT           =', E12.4/
  & 5X,'HARDENING PARAMETER                   =', E12.4/
  & 5X,'C1                                    =', E12.4/
  & 5X,'C2                                    =', E12.4/
  & 5X,'C3                                    =', E12.4/
  & 5X,'C4                                    =', E12.4/
  & 5X,'C5                                    =', E12.4/
  & 5X,'C6                                    =', E12.4/
  & 5X,'C7                                    =', E12.4/
  & 5X,'C8                                    =', E12.4/
  & 5X,'C9                                    =', E12.4/
  & 5X,'C10                                   =', E12.4/
  & 5X,'C11                                   =', E12.4/
  & 5X,'C12                                   =', E12.4/)

220 FORMAT(
  1 5X,'TEMP                                  =', 8(E9.2,1X)/
  2 5X,'E                                      =', 8(E9.2,1X)/
  3 5X,'VNU                                    =', 8(E9.2,1X)/
  4 5X,'ALPHA                                 =', 8(E9.2,1X)/
  5 5X,'K                                      =', 8(E9.2,1X)/
  6 5X,'M                                      =', 8(E9.2,1X)/

230 FORMAT(
  1 5X,'E                                      =', E12.4/
250 FORMAT(      
  1 5X,'BULK MODULUS PARAMETERS:','/',                   
  2 5X,' K (INITIAL)                        =', e12.4,/,                   
  3 5X,' K1                                   =', e12.4,/,                   
  4 5X,' K2                                   =', e12.4,/,                   
  5 5X,' SHEAR MODULUS PARAMETERS:','/',                   
  6 5X,' G1 (INITIAL)                        =', e12.4,/,                   
  7 5X,' G1                                   =', e12.4,/,                   
  8 5X,' G2                                   =', e12.4,/,                   
  9 5X,' FAILURE SURFACE PARAMETERS:','/',                   
  * 5X,' C1                                   =', e12.4,/,                   
 * 5X,' C2                                   =', e12.4,/,                   
 * 5X,' C3                                   =', e12.4,/,                   
 * 5X,' C4                                   =', e12.4,/,                   
 * 5X,' C5                                   =', e12.4,/,                   
 * 5X,' C6                                   =', e12.4,/,                   
 * 5X,' C7                                   =', e12.4,/,                   
 * 5X,' CAP PARAMETERS:','/',                   
 * 5X,' R-CAP                                =', e12.4,/,                   
 * 5X,' C8                                   =', e12.4,/,                   
 * 5X,' C9                                   =', e12.4,/,                   
 * 5X,' X-BEG                                =', e12.4,/,                   
 * 5X,' CONVERGENCE PARAMETERS:','/',                   
 * 5X,' MAXIMUM NO. ITERATIONS ...... =', i12,/,                   
 * 5X,' TOLERANCE.............................. =', e12.4)                   
END

SUBROUTINE BLKMAX(MATYPE,MT,PROP,BULKM)
DIMENSION MATYPE(1),PROP(1)

LC=48*(MT-1)+1
MTYPE=MATYPE(MT)
GO TO (10,20,30,40,50,60,70,80,90,94,90,30,100,96,97) MTYP
1

10 YM=PROP (LC+16)
PR=PROP (LC+17)
GO TO 100
20 YM=.33333* (PROP (LC)+PROP (LC+1)+PROP (LC+2))
PR=.33333* (PROP (LC+3)+PROP (LC+4)+PROP (LC+5))
GO TO 100

30 YM=PROP (LC)
PR=PROP (LC+1)
GO TO 100
40 YM=PROP (LC+8)
PR=PROP (LC+16)
GO TO 100
50 YM=3.*PROP (LC+1)
PR=0.0
GO TO 100
60 YM=3.*PROP (LC)
PR=0.0
GO TO 100
70 YM=3.*PROP(LC+16)
   PR=0.0
   GO TO 100
80 YM=2.926*PROP(LC)
   PR=.463
   GO TO 100
90 YM=PROP(LC)
   PR=PROP(LC+1)
   GO TO 100
94 YM=PROP(LC+8)
   PV=PROP(LC+16)
   GO TO 100
96 YM=3.*PROP(LC)
   PR=0.0
   GO TO 100
97 blkx = prop(lc)/(1.0-PROP(lc+1))
   GO TO 110
100 BLKX=YM/(3.*(1.-2.*PR))
110 BULKX=AMAX1(BLKX,BULKMX)
RETURN
END

C**************************************************************
C SUBROUTINE GETSTR(A,IA)
C
C COMMON/BK02/IOFOC,IPHASE,IMASS,MODEL,NUMEL,LPAR(7)
C COMMON/BK06/MPNRT,MPRINT,LOCSTR,NUMELT,JPRINT,IDUMP
C COMMON/BK16/MAXINT
C COMMON/BK46/STRESS(4),STRAIN(4),D(4,4),LST,NEL,NSTATE
C
C DIMENSION A(1),IA(1)
C
C IPT=MINO(MAXINT,LST)
C MTP=IA(IGTPNT(8)+NEL-1)
C NN=IGTPNT(12)+48*(MTP-1)
C II=IGTPNT(14)+MAXINT*LPAR(7)*(NEL-1)
C NN=II+LPAR(7)*(IPT-1)
C NT=NN+LPAR(7)-1-IDUMP
C
C GO TO (10,10,20,30,40,10,30,30,10,50,10,30,50,20,10,10,60), MODEL
C 1           rock
C
C 10 CALL S1OUT (A(NN),A(NT))
C RETURN
C 20 CALL S3OUT (A(NN),A(NN+4),A(NT))
C RETURN
C 30 CALL S4OUT (A(NN),A(NN+5),A(NT))
C RETURN
C 40 CALL S5OUT (A(NN),A(NN+4),A(NM))
C RETURN
C 50 CALL S10OUT(A(NN),A(NN+4),A(NT))
C RETURN
C 60 call s17out(A(NN),A(NN+4),A(NT))
C RETURN
C END
C
C**************************************************************
C SUBROUTINE PRTSTR (IPST,MAPT,MATYPE)
C INTEGER FILE
COMMON/BK02/IOFC,IPHASE,IMASS,MODEL,NUMEL,LPAR(7)
COMMON/BK05/IFIL,IADD,MKSIZ,HEAD(12)
COMMON/BK06/NPRNT,MPRINT,LOCSTR,NUMEL,PRINT,NDUMP
COMMON/BK08/KPRINT,NSTEP,ITE,ILIMIT,NEWSTP
COMMON/BK14/FILE(6)
COMMON/BK18/NUMMAT,ITYP2D,AKO(20)
COMMON/BK20/NTOTAL
COMMON/BK32/NSREP,NEXIT,TIME,TIMEP,LPRT,NPRINT
COMMON/BK48/STRESS(4),STRAIN(4),D(4,4),IPT,N,NSTATE
COMMON/ /A(1)
C DIMENSION IPST(1),MATP(1),MATYPE(1),AST(4,9)
COMMON/BK34/STF(1)
CHARACTER*8 ASTATE(5)          rock
CHARACTER*4 NITYP
DATA ASTATE/‘ELASTIC’,‘PLASTIC’,‘FAILURE’,‘CAP’,‘CORNER’/
C STRESS CALCULATIONS FOR ELEMENTS WITH CONSTANT MATERIAL VARIABLES
C NITYP=’RHOOP’
IF (LPAR(3).EQ.1) NITYP=’XX’
C DO 70 N=1,NUMEL
MYPE=MATEP(N)
IF (MYPE.EQ.0) GO TO 60
MODEL=MATYPE(MYPE)
IF (MPRINT.LE.0) GO TO 10
IF (IPST(N).EQ.1) GO TO 70
C CALCULATE AND PRINT ELEMENT STRESSES AT INTEGRATION POINTS
C 10 DO 50 LST=1,4
IPT=LST
C CALL GETSTR (A,A(1))
C IF (MPRINT.LE.0) GO TO 40
IF (NPRINT.GT.0) GO TO 20
C NPRNT=40
CALL HEADER
WRITE (6,80) NSTEP,TIMEP
IF (ITYP2D.LT.2) THEN
WRITE (6,90) NITYP
ELSE
WRITE (6,120)
ENDIF
20 NPRNT=NPRINT-1
C CALL PRINT
C AST(LST,2)=STRESS(1)
AST(LST,3)=STRESS(2)
AST(LST,4)=STRESS(3)
AST(LST,5)=STRESS(4)
AST(LST,6)=STRAIN(2)
AST(LST,7)=STRAIN(3)
AST(LST,8)=STRAIN(4)
AST(LST,9)=STRAIN(1)
IF (IPT.LT.4) GO TO 50

WRITE (6,100) N,MTYPE
DO 30 I=1,4
WRITE (6,110) I,ASTATE(NSTATE), (AST(I,J),J=2,9)
30 CONTINUE
GO TO 50

40 KLOC=5*(LST-1)

CALL BLKCPY (STRESS,STF(LOCSTR+KLOC+1),5)

50 CONTINUE
60 LOCSTR=LOCSTR+20
IF (KPRINT.LE.0) GO TO 70
IF (LOCSTR+20.LE.NTOTAL) GO TO 70
CALL WRABSF (FILE(3),STF,LOCSTR,IADD)
CALL RIOSTAT (FILE(3))
IADD=IADD+LOCSTR
LOCSTR=0
70 CONTINUE
IF (LOCSTR.EQ.0.OR.KPRINT.LE.0) RETURN
CALL WRABSF (FILE(3),STF,LOCSTR,IADD)
CALL RIOSTAT (FILE(3))
IADD=IADD+LOCSTR
LOCSTR=0

RETURN

80 FORMAT(//' ELEMENT STRESS CALCULATIONS')
90 FORMAT(//' FOR TIME STEP ',15,7X,' (AT TIME ',1PE10.4,')')
100 FORMAT(,' ELEMENT STRESS ',1A4, ' SIG-YY SIG-ZZ SIG-',
110 FORMAT(,' MAX SIG MIN SIG',26X, 'SHYIELD /',
120 FORMAT(,1' NUM/IPT STATE',82X,'ANGLE',9X,'FUNCTION')
END
A5. INGRID INPUT DATASET FOR INDENTATION PROBLEM

SIMULATION OF OYA TUFF INDENTATION

N3D  
ANAL STAT
BWMO ON
NSTEP 20
DELT 0.001
IPRT 4
IPLT 1

MAT 1 3  
C MATERIAL #1 - PLASTIC MATERIAL

HEAD
MATERIAL #1 - PLASTIC MATERIAL
E 320000.0
PR 0.306
SIGY 1400.0
ETAN 1000.0
BETA 1  
C ISOTROPIC HARDENING
ENDMAT

MAT 2 1  
C MATERIAL #2 - ELASTIC STEEL

HEAD
MATERIAL #2 - ELASTIC STEEL
E 3000000.0
PR 0.300
ENDMAT

C LOAD CURVE #1: APPLY PRELOAD IN TWO STEPS

LCD 1 3 0.0 0.0 0.002 400.0 0.020 400.0

C LOAD CURVE #2: APPLY FORCING LOAD

LCD 2 3 0.0 0.0 0.002 10.0 0.020 800.0

SI 1 SV PNLT 0.0  
C SLIDE LINE PARAMETERS

C ***************************************************************

START  
C PART #1: ROCK SPECIMEN

1 9 24 30 36 ;
1 9 24 30 36 ;
-1 ;
0.0 0.206 0.206 0.91 1.25
0.0 0.206 0.206 0.91 1.25
0.0

MATE 1  
C MADE OF MATERIAL #1

RES 2 1 1 3 21 I 1.10  
C CONTINUOUS MESH GRADING
RES 1 2 1 2 31 J 1.10
C DELETE REGION
C FORM ARCS
C MOVE INTERIOR CORNER OF ME
C MESH SQUARE
C MOVE A FEW NODES:
C AS A CONSEQUENCE OF THE
C REGION DELETION
C RADIAFLY FIX CENTERLINE
C FOR GOOD MEASURE
C RADIAFLY FIX ROCK O.D.
C FIX TOP OF ROCK AXIALLY
C PRESSURE PRELOAD ON BOTTOM
C DEFINE SLIDE-LINE SLAVES

C ********************************************

START C PART #2: STEEL INDETER
1 6 8 ;
1 6 11 13 ;
-1 ;
0.0 0.142 0.219
-0.435 -0.219 -0.077 0.0
0.0

MATE 2 C MADE OF MATERIAL #2
D 230 340
PA 220 1 0.1334
MA 220 2 0.0486
PA 320 1 0.2058
MA 320 2 0.0749
PA 330 1 0.1256
MAC 330 2 0.0374
PA 240 1 0.1256
MA 240 2 -0.0396
PA 230 1 0.0946
MA 230 2 -0.0068
PA 310 1 0.3063
PA 210 1 0.2293
AC 141 2 4 13 0 -0.219 0 0.219 0 0 1
AC 3 2 1 3 3 1 3 0 -0.219 0 0.219 0 0 1
B 1 1 0 1 4 0 100000
PR 1 1 0 3 1 0 2 1.0 0 1 0
SI+ 1 4 0 2 4 0 1 M 0.03 0.03 0.0
SI+ 3 2 0 3 3 0 1 M 0.20 0.0 0.0
END
END

C RADially FIX CENTERLINE NODES
C PRESSURE LOAD ON TOOTH
C DEFINE SLIDE-LINE MASTERS