SCRUBS.BYU a Two Dimensional Finite Element Package for Continuum Analysis Using Quadratic Isoparametric Elements

Michael Glenn Long
Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Civil and Environmental Engineering Commons

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/etd/3458

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
SCRUBS.BYU A TWO DIMENSIONAL FINITE ELEMENT PACKAGE FOR CONTINUUM ANALYSIS USING QUADRATIC ISOPARAMETRIC ELEMENTS

MICHAEL GLENN LONG
SCRUBS.BYU A TWO DIMENSIONAL FINITE ELEMENT PACKAGE FOR CONTINUUM ANALYSIS USING QUADRATIC ISOPARAMETRIC ELEMENTS

A Thesis
Presented to the
Department Of Civil Engineering
Brigham Young University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Michael Glenn Long
April 1983
This thesis, by Michael Glenn Long, is accepted in its present form by the department of Civil Engineering of Brigham Young University as satisfying the thesis requirement for the degree of Master of Science.

Steven E. Benzley
Committee Chairman

Cliff S. Barton
Committee Member

Date

Henry N. Christiansen
Department Chairman
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ........................................ iv
LIST OF FIGURES ........................................ v

CHAPTER

1. INTRODUCTION ........................................ 1
2. SCRUBS.BYU ........................................ 3
   2.1 MATERIAL MODELING ................................ 7
   2.2 ISOPARAMETRIC ELEMENT FEATURES ............... 13
   2.3 FRACTURE MECHANICS ................................ 26
   2.4 SUBSIDENCE MODELING ................................ 41
3. PRESCRUBS.BYU USER'S GUIDE ......................... 52
4. PRESCRUBS.BYU PROGRAMMER'S GUIDE ............... 75
5. CONCLUSIONS AND RECOMMENDATIONS ............... 115

BIBLIOGRAPHY ........................................ 117

APPENDIXES

A. PRESCRUBS.BYU PROGRAM LISTING .................. 121
B. SCRUBS CARD IMAGES ................................ 184
C. SAMPLE SCRUBS FILES ................................ 202
D. QUATER POINT ALGORITHM LISTING ............... 211
ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Dr. Steven E. Benzley and Cliff S. Barton for the many hours of guidance and support provided during the course of this research.

A special thanks is given to family and friends who have helped in a number of ways.

Finally, the author wishes to express his gratitude to his loving and devoted wife Catherine, for her untiring patience and for her financial support.
LIST OF FIGURES

Figure:

1. Program Capabilities
2. Von Mises Yield Surface
3. Yield Surface--Plotted on the plane
4. Projection of Yield Surfaces on * Planes
5. Mohr-Coulomb Hypothesis
6. Drucker-Prager Yield Surface
7. Types of Elements
8. Gaussian Quadrature
9. Finite Element Meshes
10. Accuracy vs. Computational Costs
11. Zero-Energy Deformation Modes in Plane Elements
12. Von Mises Stress Contours showing hourglass effect
13. Deformed Grid showing hourglass effect
14. Basic Modes of Crack Deformation
15. Formation of Plastic Zone at a Crack Tip
16. Original Isoparametric Element Mesh
17. Singularity Elements Around Crack Tip (node 179)
18. Welded T Joint
19. T Joint Finite Element Mesh
20. Stress Fringes Without a Singular Point
21. Stress Fringes With a Singular Point
22. Mesh of Plate
23. Stress Fringes Using a Singular Point
24. Butt Weld
25. Butt Weld Mesh
26. Stress Fringes at a Crack Tip
27. Stratigraphy and Finite Element Grid for Transverse Section
28. Rubbleized Section and Unmagnified Deformed Equilibrium State of Failed Transverse Cross Section
29. Comparison of Computer and Field Data of Transverse Section of Old Ben 24
30. Height of Roof Break Above Coal Seam as Reported in Reference 19
31. Types of Elements
32. Gaussian Quadrature
33. Plastic Stress/Strain Curve
34. True Stress/Strain Curves
35. Boundary Angle
36. Point Load in -Z Direction
37. Pressure Load and Shear Traction
CHAPTER 1
Introduction

During the recent years advances in computer technology have made the use of finite element analysis more and more popular for complex continuum mechanics problems. A finite element analysis is a numerical procedure approximation which offers solutions to complex geometric bodies or complex loadings which are generally too lengthy or impossible to solve with the classic equations of equilibrium. The use of this type of analysis is well established in the aerospace industry; but, with the advent of smaller and faster computers, finite elements are beginning to find a place in smaller industries. Finite elements are indeed proving to be invaluable tools in the hands of wise engineers.

During the past several years, work has continued in the development of versatile finite element package SCRUBS. SCRUBS is a powerful two-dimensional, elastic-plastic finite element code.
which is capable of determining stresses or deformations in plane or axisymmetric bodies.

This thesis develops some new and powerful capabilities for the SCRUBS.BYU package. A quarter point algorithm was created making possible crack tip fracture mechanics solutions. An interactive user-friendly preprocessor PRESCRUBS.BYU was developed to create a SCRUBS input file and to automate the boundary condition definition. Example problems using quadratic isoparametric elements, rubbleization subsidence, and fracture mechanics situations are presented here demonstrating the capabilities of this package.

The basic equations of the finite element procedure, can be found in references 1 and 2. The work presented here focuses on special features which incorporate isoparametric elements, fracture mechanics, and solutions to subsidence problems.
CHAPTER 2

SCRUBS.BYU

The program SCRUBS is a two-dimensional, elastic-plastic finite element program which determines deformations and stress states for plane or axisymmetric bodies. These bodies may be composed of materials whose properties are represented as elastic, elastic perfectly-plastic; or elastic, work hardening. Four possible yield conditions which can be used to describe materials are: 1) Von Mises, 2) Tresca, 3) Drucker-Prager, or 4) Beltrami. SCRUBS also has material rubbleization and element mining capabilities. A crack tip algorithm is available for a fracture mechanics analysis. SCRUBS offers several modes of loading which include surface pressure loads, point loads, gravity loads, centrifugal loads or specified displacements. There are currently two types of isoparametric elements available (i.e. linear or quadratic), which can be integrated using either one, two, or three Gaussian quadratic points.
THE pre- and post-processors coupling to SCRUBS are shown in Figure 1. These capabilities include mesh generation by QMESH, data definition by PRESCRUBS, and line drawing and continuous tone color graphics by MOVIE.BYU.

QMESH is a mesh generator, which was developed at Sandia National Laboratories to create two dimensional and axisymmetric finite element meshes.

PRESCRUBS is an interactive user-friendly preprocessor developed by the author to assist in rapid data input. This capability proved to be invaluable while processing the numerous solutions performed for this thesis.

The INTERFACE program takes a SCRUBS output file and creates the necessary geometry, displacement, and special function files for MOVIE.BYU. The special function option of INTERFACE selects a specified stress or strain component, which can then be displayed using MOVIE. For example: axial stress, radial stress, shearing stress, or displacements may be displayed.

The SCRUBS 2-D finite element package has been under development for a number of years, originating at Sandia National Laboratories, and continuing on at Brigham Young University. As part of this development
this thesis adds some new and powerful capabilities to the SCRUBS package. This chapter presents these capabilities, along with some of the package's existing functions and capabilities.
FIGURE 1. Program Capabilities

1. Mesh generation
2. Interactive solution controls and definition
3. Linear plane stress, strain, or axisymmetric analysis
4. Linear fracture mechanics
5. Nonlinear plane stress, strain, or axisymmetric analysis
6. Nonlinear fracture mechanics
7. Rubbleization and subsidence
8. Line drawing and continuous tone color graphics post-processing
Section 2.1
Material Modeling

SCRUBS is an elastic-plastic code with the capability of using a wide range of materials. Four different yield conditions (i.e., Von Mises, Tresca, Drucker-Prager, and Beltrami) are available which accurately model many materials. This section presents a brief description of these yield theories.

The Von Mises yield theory is based on the assumption that the point of yielding occurs as the shear strain energy per unit volume reaches a critical value. [3] This stress, when expressed in terms of principal stresses is:

$$\sigma = \left(\left((\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right)/2\right)^{1/2}$$

As long as the stress remains within the smooth cylindrical yield surface shown in Figure 2, yielding will not occur [4]. The vector OH represents the hydrostatic pressure or the hydrostat. Obviously, as long as the stress vector remains on the hydrostat yielding will not occur. However, as the stress state deviates from the hydrostat, (i.e., vector NP of Figure 2, the deviatoric stress) yielding and plastic flow become imminent.
Another way to represent the yield surface, when unaffected by the hydrostat, is to project this surface on what is known as the \( \tau \) plane, represented by plane 0 in Figure 2. On this plane, the hydrostatic component is perpendicular and therefore induces no plastic work because there is no increase in the deviatoric component.

FIGURE 2. Von Mises Yield Surface [5]
The Tresca yield theory is similar to Von Mises but predicts that yielding will occur when the maximum value of shear stress reaches a critical value. [7] Expressed in terms of the principal stresses this yield stress is:

$$\sigma = \left| \sigma_1 - \sigma_3 \right|$$

The Tresca yield cylinder as shown in Figure 2 is similar to the Von Mises surface except as it is hexagonal rather than circular in cross section.

Although both Von Mises and Tresca model materials of a metallic nature, the Von Mises yield condition is generally used in numerical calculations because it is well defined by a smooth curve.
The Drucker-Prager yield theory is another condition that has been primarily used to model geomechanical problems involving such things as slope stability, bearing capacity, and soil pressure loads. This theory assumes an idealized material which behaves elastically up to some state of stress at which slip or yielding occurs. The shear stress required for simple slippage is considered to depend upon the cohesion and linearly upon the normal pressure on the slip surface. In complete plane investigations an extended Coulomb's rule is used.

\[ R = C \cos \phi - (\sigma_x + \sigma_y)/2 \sin \phi \]

As shown in Figure 5, R is the radius of Mohr's circle at slip; the maximum shearing stress is \( [(\sigma_x - \sigma_y)^2/4 + T_{xy}^2]^{1/2} \); c is the cohesion; \( c \cos \phi \) is the radius of Mohr's circle at slip when the mean normal stress, \((\sigma_x + \sigma_y)/2\) in the plane is zero, and \( \phi \) is the angle between the tangents of the Mohr's circles at slip and negative \( \sigma \) axis.

**FIGURE 5.** Mohr-Coulomb hypothesis
The yield function, which is a proper generalization of the Mohr-Coulomb hypotheses is:

\[ \sigma = \alpha J_1 + J_2^{1/2} \]

where \( \alpha \) is a positive constant at each point of the material, \( J_1 \) is the sum of the principal stresses:

\[ J_1 = \sigma_1 + \sigma_2 + \sigma_3 \]

\( J_2 \) is the second invariant of the stress-deviation:

\[ J_2 = \frac{1}{6}(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2 + \tau_{12}^2 + \tau_{23}^2 + \tau_{31}^2 \]

The yield surface in principal stress space is a right circular cone with its axis equally inclined to the coordinate axis about the hydrostat and with the apex in the tension octant of the stress space". [8]
The Beltrami stress theory assumes that yielding occurs when the total strain energy per unit volume equals the total strain energy per unit volume at yielding in uniaxial tension or compression. [7]

The yield condition then expressed in terms of principal stress is:

\[ \sigma = \left( \sigma_1^2 + \sigma_2^2 + \sigma_3^2 - 2 \mu(\sigma_1 \sigma_2 + \sigma_2 \sigma_3 + \sigma_1 \sigma_3) \right)^{1/2} \]

A problem arises as Beltrami asserts that yielding occurs under high hydrostatic pressures, which is contrary to experimental data. For this reason Beltrami is not often used.

In summary, SCRUBS currently offers four yield stress theories for material modeling. These four options provide flexibility and accuracy while modeling a variety of materials. The Von Mises theory is generally preferred over Tresca, and the Drucker-Prager theory is generally used for soils. Beltrami is seldom used.
Section 2.2
Isoparametric Element Features

The use of finite element can be a very useful tool for engineers solving complex continuum mechanics problems. Caution must always be exercised when analyzing finite element solutions because the finite element procedure is a numerical approximation and is subject to error. The degree of error is a function of many things, some of which are: 1) the number of elements used in the finite element mesh, 2) the type of element used, and 3) the number of Gaussian integration points used. To acquaint the user with these factors and thus minimize problems due to these factors; a thorough analysis was performed, comparing the isoparametric finite elements available in SCRUBS. This analysis is presented in this section.

The purpose of this analysis was to determine the relationship between the accuracy of a given solution verses computation costs for isoparametric finite element models of 1) varying number of elements, 2) linear or quadratic elements, and 3) varying number of Gaussian points of integration.
When modeling a particular structure an important consideration is the number of elements to be placed in the finite element mesh. Refined meshes generally produce more accurate results than coarse meshes, this is due to the increased amount of nodal points which provide additional accurate basis for the numerical integration of the strain energy in the structure. As a general rule, the more elements that are in the mesh, the better the solution. However, the more elements used adds to the computational costs.

Another consideration is the order of the element used. Considering the types of elements it may be difficult to predict which element will be most effective without some experimentation.

\[ U \]

![Linear and Quadratic Elements](image)

**FIGURE 7. Types of Elements**

Figure 7. shows the two elements that are available in SCRUBS. The linear element has straight sides and the sides of the quadratic element may contain quadratic curves. Lower order elements are economical and often yield accurate solutions, but care must be taken to ensure that the elements behave in a
manner compatible to the structure being modeled. For example, linear elements yield economical and accurate solutions under axial stresses but yield poor solutions when subjected to bending.

While higher order elements tend to yield accurate solutions they also have higher computational costs due to the added processing of the increased number of nodal points. Therefore a balance must be achieved between accuracy and the cost.

As well as the number and type of element used in the mesh, another important step in the finite element process is to determine the number of Gaussian integration points to be used. SCRUBS has the capability of one, two, and three Gauss points.

1st Order 2nd Order 3rd Order

Figure 8. Gaussian Quadrature

The number of Gauss points, often referred to as Gaussian quadrature, is the order of sampling points over which the strain energy is integrated (see Figure 8). Higher orders of Gaussian quadrature provide an increase in accuracy, but also an increase the computational expense. As a general rule, it is best
to use the lowest order quadrature which gives suitable results, remembering that lower quadratures impose limitations on various elements under various loading conditions.

Lower order quadratures are desireable for two reasons. First, fewer points result in lower computational costs, the cost of integration being proportional to the number of Gauss points multiplied by the square of the number of degrees of freedom. Second, lower order quadratures tend to soften the elements countering the stiff behavior which may be associated with the assumed displacement field. [9]

The following analysis was performed in order to familiarize the SCRUBS user with the advantages and disadvantages caused by variations of the factors previously mentioned: i.e. the density of the mesh, the type of element used, and number of Gauss points.

This analysis consists of eighteen test solutions of specific cantilever beam. Each solution has a different number of elements in the mesh and uses a different type of element or contains a different number of Gauss points. The first variable considered was the number of elements in the mesh. Meshes of 10, 100, and 500 elements were used (see Figure 9). The second variable was the type of element used in the mesh, i.e. linear, or quadratic. The third variable
Figure 9. Finite Element Meshes
was the number of Gauss points in the element. One, two, and three Gauss points were used.

The cantilever beam was ten inches long, one inch thick, and one inch deep. It was fixed at the left end and loaded with a shear traction at the right end. All of the test solutions had the above defined dimensions and loadings but differed by the variables mentioned.

The combination of these three variables defined the eighteen test problems which were then input into SCRUBS and the accuracy of the solution versus the computation cost was plotted. The results were then analyzed to determine the advantages and disadvantages of different combinations of the three variables discussed.
TABLE 1. Results of Isoparametric Analysis

For each solution the deflection, computation time, and a subscripted solution identifier (used in Figure 10) are given.

Linear Elements

<table>
<thead>
<tr>
<th></th>
<th>1 GAUSS</th>
<th>2 GAUSS</th>
<th>3 GAUSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 ELEMENTS</td>
<td>NO SOLUTION</td>
<td>-0.0072 in</td>
<td>NO SOLUTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.15 sec</td>
<td></td>
</tr>
<tr>
<td>100 ELEMENTS</td>
<td>-0.263 in</td>
<td>-0.0115 in</td>
<td>-0.0115 in</td>
</tr>
<tr>
<td></td>
<td>9.39 sec</td>
<td>17.54 sec</td>
<td>28.58 sec</td>
</tr>
<tr>
<td>500 ELEMENTS</td>
<td>0.0212 in</td>
<td>0.0314 in</td>
<td>NO SOLUTION</td>
</tr>
<tr>
<td></td>
<td>52.78 sec</td>
<td>165.8 sec</td>
<td></td>
</tr>
</tbody>
</table>

Quadratic Elements

<table>
<thead>
<tr>
<th></th>
<th>1 GAUSS</th>
<th>2 GAUSS</th>
<th>3 GAUSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 ELEMENTS</td>
<td>NO SOLUTION</td>
<td>-0.0122 in</td>
<td>-0.0122 in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6.81 sec</td>
<td>7.24 sec</td>
</tr>
<tr>
<td>100 ELEMENTS</td>
<td>NO SOLUTION</td>
<td>-0.0123 in</td>
<td>-0.0124 in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>46.0 sec</td>
<td>66.8 sec</td>
</tr>
<tr>
<td>500 ELEMENTS</td>
<td>DIMENSION EXCEEDED</td>
<td>DIMENSION EXCEEDED</td>
<td>DIMENSION EXCEEDED</td>
</tr>
</tbody>
</table>
Figure 10. Accuracy vs. Computational Costs
The results of this analysis are given in Table 1 and in Figure 10. It is obvious that test solutions seven and eight, (the ten element mesh, with quadratic elements, and containing either two or three Gauss points) produces the most accurate results with minimal cost. This is most significant because the accuracy of the solution was maintained while computational time was minimized.

As was mentioned earlier, the finite element method is a numerical approximation of physical relationships and the accuracy of the solution is dependent on the assumptions made in the finite element procedure. Disregard of these assumptions can produce erroneous results as was demonstrated by this analysis. The linear elements as a whole performed poorly, scarcely within fifty percent accuracy. The reason for this is that linear elements deform with straight edges and when subjected to bending stresses such as the cantilever test solutions, these elements must try to model a deformation pattern beyond their capability.

The test problems using finer meshes of linear elements illustrated an interesting phenomenon which is sometimes encountered in finite
element solutions, i.e. "hourglassing" or "keystoning." Hourglassing is caused by elements with zero-energy deformation modes in that the element deforms without a change in the strain energy. [2]

Figure 10. Zero-energy deformation modes in plane elements. (a) The hourglass modes in a linear element integrated by one Gauss point. (b) A quadratic element integrated by a four point rule. Differential elements of the Gauss points rotated and do not strain. [2]

If a finite element mesh contains one or more elements in a zero-energy deformation mode the stiffness matrix becomes singular. When the stiffness matrix becomes singular the matrix cannot be inverted and thus no solution can be obtained. For the cases demonstrated here, the use of one and two Gauss points had enough numerical roundoff to produce "a solution." However, the deflection was in the upward direction rather than downward in the direction of the loading as shown in Figures 11, 12. In the case using three Gauss
Figure 11. Von Mises Stress Contours showing hourglass effect.

Figure 12. Deformed Grid showing hourglass effect.
points the element integration is more precise and shows that the stiffness matrix is in fact singular.

For the other meshes using linear elements the solution indicated displacements in the direction of the loading but were far from the correct solution of -0.0135. This again is due to the misuse of linear elements.

The test problems using quadratic elements, on the other hand, produced solutions which were accurate. Quadratic elements may have edges which take on a quadratic surface which models bending much more accurately than linear elements.

No solution was obtained for quadratic elements using only one Gauss point. This was caused by the situation defined above.

The meshes containing 500 elements exceeded the current dimensional capacity of SCRUBS. When using meshes containing a large number of elements and higher order elements the number of nodal points is dramatically increased and the program dimensions can easily be exceeded. Dimension statements in SCRUBS can easily be modified to accommodate this situation.

The quadratic element meshes using two and three Gauss points produce very good results. The ten element mesh took approximately nine times as long to run with only a minor increase in accuracy. The extra
computational costs are due to the increase in the amount elements which must be processed.

In conclusion it was found that linear elements perform poorly when subjected to bending stresses but quadratic elements perform quite accurately. It was shown that the use of only one Gauss point is inadequate, as the integration is not accurate enough to produce suitable results. Also it was shown that the proper selection of element order and integration procedure has significant bearing on accuracy and cost.
Fracture mechanics is the study of cracks and flaws in structural materials. "Fracture demands the existence of a crack or flaw somewhere in a solid, and stresses which will induce the crack to propagate. It is, apparently, impossible to produce a solid body, regardless of the particular material involved, so perfect in a structure that it contains no flaws or microcracks of any kind." In real structures, flaws and cracks can originate for a variety of reasons. Defects due to welding, the effects of stress corrosion or even the presence of microcracks indicate potential fractures. The existence of a crack by itself is not of true concern; it is when the service conditions cause the crack to propagate that cracking becomes a hazard. Beginning with the premise that structures will contain flaws or cracks, it becomes essential to consider their size and shape in the initial design considerations, this introduces the importance of fracture mechanics." [4]

To upgrade the already powerful capabilities of SCRUBS, a crack tip algorithm was written which
simulates the stresses induced by cracks and flaws adding a fracture analysis option to the SCRUBS package.

"Traditionally, three possible loading modes have been considered in regard to crack opening. They are called Modes I, II, and III as shown in Figure 14, and the stress intensity factors for these modes are denoted as KI, KII, and KIII. Because the Mode I opening is encountered far more often than the other two and the majority of research studies have been devoted to this type of loading, no further detailed discussion will be devoted to Modes II or III. In fact, the bulk of the remainder of this section will be concerned with a Mode I loading under plane strain conditions, since at the present time it is the only situation for which standardized test procedures have been developed and accepted." [4] The critical parameter of concern is written as K.
FIGURE 14. Basic Modes of Crack Deformation [3]

"The equations that describe the stress field for Mode I loading come from a method attributed to Westergaard [10] and are as follows:

\[
\begin{align*}
\sigma_y &= \frac{K_1}{(2\pi r)^{\frac{1}{2}}} \cos \frac{\theta}{2} \left( 1 + \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) \\
\sigma_x &= \frac{K_1}{(2\pi r)^{\frac{1}{2}}} \cos \frac{\theta}{2} \left( 1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) \\
\sigma_z &= v(\sigma_x + \sigma_y), \quad \tau_{xz} = \tau_{yz} = 0 \\
\tau_{xy} &= \frac{K_1}{(2\pi r)^{\frac{1}{2}}} \sin \frac{\theta}{2} \cos \frac{\theta}{2} \cos \frac{3\theta}{2}
\end{align*}
\]

where Fig. 15 defines the notations of the coordinate system used." [4]
As was mentioned earlier, the case of plane strain will receive major attention and Westergaard's equations express that physical situation where higher order terms in $r$ have been omitted. As $r$ approaches the edge of the crack, these equations provide a good prediction of the stresses of that region. Of course it is obvious that these predictions indicate the possibility of extremely high stresses in the vicinity of the crack tip, as $r \to 0$, and could not be expected in a real material.

FIGURE 15. Stress State in the Vicinity of a Crack Tip [5]
The limit to any actual stress magnitude would obviously be the yield strength of the solid; regardless of whatever the values predicted by Westergaard's equations the maximum value would cut off when yielding occurred. Figure 15 illustrates this physical situation for $\sigma_y$, at $\theta=0$. As may be seen by these equations, $K_I$ is related to stress and the square root of some characteristic length so in terms of dimensional correctness, it must have units of stress times $(\text{length})^{1/2}$. 

**FIGURE 16. Formation of Plastic Zone at Crack Tip [4]**
It is now established that $K_I$ is related to the applied stress, $\sigma_\alpha$, and the square root of some function of crack length, $a$; thus in a general form $K_I$ is:

$$K_I = C\sigma(a)^{1/2}$$

The functional coefficient $C$ depends upon the geometry of the body and is subject to stress and to the crack itself. Many excellent analysis have provided the solutions for $K_I$ and may be found in reference 4.

In order to obtain a correct representation of the stress state in the vicinity of a crack tip, singularity elements or crack tip elements are used. "Singularity or crack-tip elements incorporate the theoretical singular solution in the element itself. Hybrid elements employ an assumed stress field within the element and therefore use the following equation.

$$\sigma = \frac{1}{(r)^{1/2}} (K_I \sigma_{I} + K_{II} \sigma_{II} + K_{III} \sigma_{III})$$

Displacement-based singularity elements add the following equation to the assumed displacement field.

$$f = (r)^{1/2} (K_I f_{I} + K_{II} f_{II} + K_{III} f_{III})$$

Both hybrid and displacement-based elements can have a stress intensity factor as one of the primary unknowns, so that the factor becomes part of the solution vector $D$. Other singularity elements augment their fields
without explicitly including a stress intensity factor. The elements are then able to approximate the equations above, but they yield a stress intensity factor only by additional calculations such as the energy-release or equation-matching methods. Singularity elements may be attached to standard elements that model the remainder of the structure away from the crack.

"In whatever form, singularity elements can promote accuracy and efficiency. It is better to surround the crack tip by a few good singularity elements than by an extravagantly refined mesh of standard elements." [2]

A singularity element can be produced simply and efficiently by moving the side nodes of a quadratic element toward the crack tip until they are at the quarter points (Fig. 17). This patch of elements that surround the crack tip is embedded in a mesh of standard elements that model the rest of the structure.

To create the quarter point singularity, a crack tip in SCRUBS, an algorithm was written which geometrically takes the midside nodes and translates them one half the distance between the midside node and the singular point, i.e. to the quarter points. Figure 16 shows the an original mesh with the tip of the crack located at node 179. Figure 17 shows the
FIGURE 16. Original Isoparametric Element Mesh

FIGURE 17. Singularity Elements Around Crack Tip (node 179)
mesh after the midside nodes have been translated to the quarter points. This relocation of midside nodes can create a $1/\sqrt{r}$ stress field or a $\sqrt{r}$ displacement field which exists at the tip of the crack when free surfaces are also incorporated along a crack face. To incorporate this condition of singularity in a SCRUBS solution the node number at the tip of the crack is defined and the crack tip algorithm is then accessed. When this singular point is entered it sets a flag which invokes the quarter point option. Using this method the stress intensity factor is not calculated directly, but may be obtained with some additional calculations.

To illustrate the capabilities of the crack tip algorithm three example problems involving cracks were solved and displayed using MOVIE.BYU.

The first sample modeled was a welded T joint. Since this particular weld is common in actual structures, and because the butted plates were joined only by the welds, leaving an unjoined interface between the two plates, this problem made an excellent crack tip example.

To model the joint realistically, the weld was comprised of a material slightly stronger than the base plate and geometrically penetrating into the parent
FIGURE 18. Welded T Joint

FIGURE 19. T Joint Finite Element Mesh
FIGURE 20. Stress Fringes Without a Singular Point

FIGURE 21. Stress Fringes With a Singular Point
metal of both the base and flange plates, as shown in Figure 18. Because the flange and the butt plate were connected only by the welds, this connection allowed the plates to separate when a tensile loading was applied to the upper plate. The forces were completely transferred through the welds to the bottom of the base plate. This loading condition caused a plate separation which was a good example of a Mode I crack deformation.

The T joint was loaded by fixing the bottom of the base plate against displacements in the Y direction and then displacing the top surface of the flange plate by 0.001 inches. This displacement creates a reasonable loading and when processed with the SCRUBS package illustrates stress concentrations expected in the typical structure, as shown in Figures 19, and 20. Figure 19 shows the SCRUBS solution without defining a singular point at the crack tip and Figure 20 is the SCRUBS solution with a singular point at the crack tip. The obvious increase in stresses at the crack tip can easily be seen by the color fringes. The midside nodes of the Quadratic element as shown in Figure 17.

The next example was a flat plate with a hole in the center in which cracks have formed perpendicular to the load, as shown in Figure 21. A singular point was
FIGURE 22. Mesh of Plate

FIGURE 23. Stress Fringes using a singular point
FIGURE 24. Butt Weld

FIGURE 25. Butt Weld Mesh

FIGURE 26. Stress Fringes at Crack Tip
defined at the end of the crack. The left end of the plate was fixed against displacements in the X and Y directions. The right end was loaded in tension by a pressure load of 10,000 psi. As the load increased the cracks begin to open and the stress concentrations at the crack tip became prominent.

The last example was a simple welded butt joint. Again the weld was defined to be stronger than the parent material and cracking was assumed to occur at the interface of the two materials. The crack is defined in Figure 23. The left end of the plate was fixed against displacement either the X or Y directions and the right end was loaded with a shear traction. A singular point was defined at the end of the crack and the structure was then loaded. As the loading increased the crack separation became visible and the stresses at the crack tip increased proportionally.
Section 2.4
Rubbleization and Subsidence

"Underground mining is a geomechanical process that provides many challenges for mathematical modeling. Such modeling, if done correctly, can be an effective tool in the hands of mining engineers to assist in answering many difficult problems associated with underground mining. Some of these problems include 1) provision for the safety of the mine personnel and equipment, 2) evaluation of both the surface and subsurface environmental effects of the mine, and 3) assessment of the economic feasibility of the mine plan. Massive ground movement resulting from the collapse of underground mining, i.e., subsidence, is a complex geomechanical process that can affect all three of the items listed above. This section describes the nonlinear continuum formulation that is used to model the subsidence process.

In the past, problems associated with subsidence have often been circumvented by using traditional room and pillar mining techniques. This method assumes that ground motion will be minimized, if not entirely eliminated, if adequate pillars are left to continually support the overburden. However, current pressures for higher extraction ratios have made longwall [11] mining methods attractive. This
technique allows essentially all of a mineral in a seam to be recovered. The obvious consequence of longwall mining is the total collapse of the area above the mined region and the resulting subsidence of all overlying strata. The ability to accurately predict such ground motion requires effective models of several different geomechanical processes such as 1) the failure of a rock mass above a mined region, 2) the free-fall of this mass into the mined-out volume, 3) the associated bulking of the rock rubble, and 4) the recompaction of this rubble.

Applications of the finite element method to geomechanics problems have become increasingly popular during the past several years [12]. Several numerical approaches to subsidence have been used [13, 14, 15, 16]. Cundall [17] has effectively modeled the kinematics of rubble motion using a discrete block technique. This approach has been incorporated into a finite difference computer program [18] to treat small deformation, linear elastic situations.

The formulation described here is a finite element technique that treats the failure and subsequent rubble formation of a rock mass. The initial work on this model has been reported in References 19-22. The goal is to capture only the major features of rubble formation. Careful study of
specific materials, initial jointing, and loading conditions could, no doubt, produce refinements for particular mining sites. The method differs from that of Cundall in that the rubble and associated void space is treated as a continuum rather than as discrete blocks. The main advantage to this formulation is that it allows an originally intact finite element to lose its strength, "fall" into a void space, and continue to respond in the continuum model as a post failed material.

The use of the rubbleization capabilities are illustrated using the example of the Old Ben Mine. The mesh was generated with QMESH; the solution was given by SCRUBS; and the graphics were processed with MOVIE.BYU.
Old Ben Mine Prediction

The Old Ben mine number 24 has been used as a longwall demonstration project. The results of this project have been measured and documented by Dames and Moore [23]. The results of the Old Ben demonstration are used here as a comparison with a numerical prediction using the continuum rubbleization finite element model.

The incorporation of joints, faults and bedding planes will only be approximated in this formulation by using a "reduced modulus" characterization as described in Reference 24. Also, recompaction of the rubble used is ignored because it is of little interest in this application.

Figure 27 shows a strip chart of the stratigraphy over Old Ben 24. Note the number and distribution of the limestone beds. Geotechnically, limestone beds are more competent and thus more resistant to collapse than shale.

The finite element mesh used to model the subsidence event is shown in Figure 27 and a list of the material properties for the stratigraphy is given in Table 2. The two dimensional model simulated here depicts one-half of the transverse profile of the mine on overburden. As shown in Figure 27 one-half of the mine width is equivalent to the first five elements of
<table>
<thead>
<tr>
<th>MATERIAL</th>
<th>YOUNG'S MODULUS (lb/in)</th>
<th>POISSON'S RATIO</th>
<th>SHEAR MODULUS (lb/in)</th>
<th>BULK MODULUS (lb/in)</th>
<th>DENSITY (lb/ft)</th>
<th>COHESION (lb/in)</th>
<th>SHEAR HARDENING ANGLE **</th>
<th>MODULUS (lb/in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coal</td>
<td>0.26(10)</td>
<td>0.22</td>
<td>0.11(10)</td>
<td>0.15(10)</td>
<td>56.</td>
<td>10</td>
<td>45</td>
<td>10</td>
</tr>
<tr>
<td>Composite*</td>
<td>2.0(10)</td>
<td>0.22</td>
<td>0.82(10)</td>
<td>1.14(10)</td>
<td>156.</td>
<td>10</td>
<td>45</td>
<td>10</td>
</tr>
<tr>
<td>Limestone</td>
<td>4.0(10)</td>
<td>0.22</td>
<td>1.64(10)</td>
<td>2.38(10)</td>
<td>168.</td>
<td>10</td>
<td>45</td>
<td>10</td>
</tr>
<tr>
<td>Strong Shale</td>
<td>0.6(10)</td>
<td>0.22</td>
<td>0.25(10)</td>
<td>0.36(10)</td>
<td>144.</td>
<td>10</td>
<td>45</td>
<td>10</td>
</tr>
<tr>
<td>Weak Shale</td>
<td>0.2(10)</td>
<td>0.22</td>
<td>0.08(10)</td>
<td>0.12(10)</td>
<td>144.</td>
<td>10</td>
<td>45</td>
<td>10</td>
</tr>
<tr>
<td>Soil</td>
<td>0</td>
<td>0.5</td>
<td>50.0</td>
<td>1000.</td>
<td>95.</td>
<td>10</td>
<td>45</td>
<td>10</td>
</tr>
</tbody>
</table>

* This composite material is used to define several thin layers of limestone and shale immediately over the coal seam.

** These values needed for a Drucker-Prager criterion were chosen to reflect a loss of strength order hydrostatic tension. They may not be realistic in hydro compression.

TABLE 4.1 Material Properties for Calculations
FIGURE 27. Stratigraphy and Finite Element Grid for Transverse Section
the lowermost row. A listing of the SCRUBS.DAT data file is given in Table 2.

The numerical simulation proceeds as follows:

1) In the first load step gravity is applied to the finite element model and an "in situ" state is developed.

2) The second load step involved "mining" or removing the first five elements of the lowermost row. This simulates the actual mining of this material by the longwall mining equipment.

3. The removal of the material as described in 2) above so alters the boundary conditions that a new equilibrium state is sought by the numerical calculations. This new state can involve both plastic flow and/or roof collapse as previously described. The roof collapse again alters the boundary conditions such that further failures insue. This process continues until a stable equilibrium state is reached.

Figure 28 shows the rubbleized cross section and the surface subsidence resulting from mining the first five elements of the coal seam. The subsidence profile from the calculation is compared with the experimental data in Figure 29.
Figure 28 shows the equilibrium state predicted after all roof failures have taken place. The criteria used to specify roof failure of this analysis was a rapid drop in vertical $\sigma_{yy}$ stress. A bulking factor, $\alpha$, was used to predict the bulking that is associated with roof fall. This factor is used to compute the new volume of an element after roof fall as

$$V_{\text{new}} = V_{\text{old}} / \alpha$$

Figure 30 as repeated from Reference 19 was used to define . As seen in Figure 30, roof failure and bulking proceeded from mine depth to a height of 410 ft above mine depth. Material above 410 ft can still fail, but if it should, it would not bulk, but fall as a continuous mass with no bulking. Consequently, was defined as .996 for materials from mine depth to 410 ft above mine depth, then as 1.0 from this level to the surface.

A significant comparison shows a computed elastic subsidence profile with the field data and rubble calculations. Note the gentle trough as predicted by an elastic curve compared to the true steep trough and the computed rubbleization formulation discussed here." [25]
FIGURE 28. Rubbleized Section and Unmagnified Deformed Equilibrium State of Failed Transverse Cross Section
FIGURE 29. Comparison of Computer and Field Data of Transverse Section of Old Ben 24
FIGURE 30. Height of Roof Break Above Coal Seam as Reported in Reference 19
PRESCRUBS.BYU USER'S GUIDE

PRESCRUBS.BYU is an interactive user friendly preprocessor developed by the author to assist in rapid data input and modification. PRESCRUBS creates a data input file used by SCRUBS. This input file consists of: the solution controls, material properties, rubbleization properties, singular point definitions, boundary conditions, loading definitions, and printout controls.

PRESCRUBS has four features which makes it user-friendly: 1) fail-safe data input to prevent program abortion due to minor errors such as format errors, 2) data echo checks, which displays the set of data just entered to verify that it was input correctly, 3) modification of incorrect data, and 4) extensive help files to assist the user with program functions and options throughout the program.
This chapter explains the following basic functions of PRESCRUBS which are listed below. After explaining these options, the remainder of the chapter describes the input data which is used to run SCRUBS.

1. Creating a SCRUBS input file
2. Modifying or viewing a SCRUBS file
3. Reading a SCRUBS file
4. Writing a SCRUBS file
5. Printing a hard copy of the SCRUBS input data

Program Options

Option 1 (Enter SCRUBS data file)

Option 1 interactively guides the user through the data input necessary to run the program SCRUBS.FOR. The program is sub-divided into common data sets shown below. Data entry is structured so that the specification of certain flags and solution controls guides the user through only the necessary input for the particular solution selected. If the user desires help at any point, he may type "H" for help.

Solution controls
Material properties
Tabulated stress/strain data
Rubbleization data
Elemental mining data
Boundary conditions
Load description, point loads
Gravity and centrifugal, densities
Pressure loads
Increment, load factors
Impact nodes
Print-out controls

As the data is input it is checked to insure it is that valid. If the entry is valid, the user is prompted for following information. If the data is not valid, the user is prompted for the required data until it is correctly input.
The dimension of the arrays in the program require some practical limits. As values are entered they are checked against the program's current maximum dimensions. If the value entered is less than the maximum, execution continues. If the value exceeds this maximum an error message is printed, stating that this maximum value has been exceeded, and then the current maximum is printed. If a greater value than that of the current maximum is desired the program must be altered, to alter consult the the section on changing dimensions.

After data is entered, the information is displayed on the screen of the terminal. The user is prompted if changes are desired. If modifications are desired, a menu is displayed and the user can then enter the option to be corrected. If no further changes are desired the user may enter the last menu option (END MODIFICATIONS) or may simply type a carriage return, and program execution transfers to the next data set.

For example, after entry of the material properties the data is viewed and the user is prompted if changes are desired. If changes are desired, the user types Y and the menu shown below is displayed.

<SELECT OPTION TO BE MODIFIED>
1. YOUNG'S MODULUS
2. POISSON'S RATIO
3. YIELD STRESS
4. YIELD ANGLE
5. THICKNESS OF MATERIAL
6. RE-ENTER DATA
7. END MODIFICATIONS

If Poisson's ratio of material number 2 needed to be modified the user simply types 2 when prompted for the option to be selected. The user then enters the correct information corresponding to the prompts given. Upon re-entry of the modified data the complete data set is again viewed. If no further changes are desired the user may type 8 or simply type a carriage return <CR>, and program execution continues. If further modifications are desired, the above process is repeated.

If the data set is too large to be shown on the screen, the user may select the (RE-SHOW) option and the data will be again displayed. Using <CR> S the user may stop data display; using <CR> Q data display is continued.
Option 2 (View or modify SCRUBS data)

This option allows the user to quickly and conveniently modify a data set that has just been entered or a set which was read from a disc file.

With the selection of Option 2, the main modification menu listed below is displayed and the user may choose the following options: 1. View or modify all data sets, or 2. View or modify a single data set. With this the corresponding data is then displayed on the terminal screen and the user is prompted whether changes are to be made. If no changes are desired the user types the return key, which transfers execution to the next related data block if the general modification option was selected or back to the main modification menu, if the single data option was selected.

If changes are desired another menu is displayed from which the user may modify the desired data. The data set is then re-displayed on the screen. If further modifications are desired the user may simply repeat this process. When all changes have been made the user types return and execution continues to the next data set or back to view or modify menu.

1. GENERAL MODIFICATION
2. SOLUTION CONTROLS
3. MATERIAL PROPERTIES
4. TABULATED STRESS/STRAIN DATA
5. RUBELEIZATION DATA
6. ELEMENTAL MINING DATA
7. BOUNDARY CONDITIONS
8. LOAD DESCRIPTION, POINT LOADS
9. GRAVITY AN CENTRIFUGAL, DENSITIES
10. PRESSURE LOADS
11. INCREMENT, LOAD FACTORS
12. IMPACT NODES
13. PRINT-OUT CONTROLS
14. END MODIFICATIONS

Option 3 (Read from a disc file)

This option allows the user to read an existing set of data which is stored on a disc file. The file specified is read in the identical format used in SCRUBS.FOR.
The user may specify the data file to be read. The file name determines where the file is located and which file is to be read. If the correct file name is not entered or the file does not exist the user is again prompted for the file name until correctly entered.

Data files created by this pre-processor are named SCRUBS.DAT, hence the file name entered is simply SCRUBS.DAT, unless the file has been renamed.

If the file has been renamed or is in another area, the corresponding file name must be entered. Listed below is the maximum possible file name specifications.

DISC:[USER .SUBDIR]FILENAME.EXT;VERSION

DISC- disc on which users account is located
USER - user's number account
SUBDIRECTORY- user's subdirectory
FILENAME- name of file to be read or created
EXTENSION- type of file
DAT= default file
FOR= fortran file
OBJ= machine code file
EXE= executable file
DAT= data file
VERSION NUMBER- current version of file

For example, DRC1:[204026.FINITE]SCRUBS.DAT would read the file named SCRUBS.DAT located in subdirectory FINITE of user 204026 on disc 1.

Option 4 (Writes to a disc file)

This option allows the user to write the current data set to a disc file. The data is written on the file in a format compatible with that necessary to run SCRUBS.

Note: The user may wish to insure the existence of the file created by checking his directory, after program termination.
Option 5 (Prints a hard copy of data set)

Selecting this option sends a hard copy of the current data appropriately labeled to the user's default line printer.

Option 6 (Ends program execution)

This option terminates program execution.

NOTE: All menus are designed to end current operations by selecting the last menu option, but for convenience the carriage return may also be used.

All "yes" or "no" questions will default to "no" with a carriage return.

If data set is too large to fit onto the screen, <CR> S may be used to stop the display and <CR> Q may be used to continue display.

If at any point the user desires help, type "H" and a help message will be printed.
SOLUTION CONTROLS

This section of SCRUBS will input controls which specify how the problem will be analyzed. The solution controls in this section are used as flags throughout the data input process and later during the execution of SCRUBS. Care should be taken in specifying the correct controls and flags corresponding to the problem being considered.

<ENTER TITLE OF PROBLEM>
This is the title or description of current problem (80 characters or less).

<ENTER NUMBER OF PROBLEMS TO BE SOLVED>
This is the number of problems to be considered per run (usually one).

<ENTER NUMBER OF LOAD CASES PER PROBLEM>
This is the number of different loading combinations to be considered per problem.

<ENTER NUMBER OF DOF PER NODE>
DOF (Degrees of Freedom) is the number of coordinate directions in which nodes may translate.

<ENTER NUMBER OF DIFFERENT MATERIALS>
Input the number of materials of which the body is composed. Current maximum is 10 materials.

<ENTER ELEMENT TYPE NUMBER: 1=LINEAR, 2=QUADRATIC, 3=CUBIC>
SCRUBS is capable of handling the three types of elements shown in Figure 31. Quadratic and cubic elements have mid-side nodes which allow more complex modes of deformation, but computation costs are slightly higher. At the present time QMESH.BYU, the mesh generator, will not process the cubic element.
FIGURE 31. Types of Elements

The number of Gauss points refers to the order of Gaussian quadrature defined for an element. A higher order of quadrature increases the number of integrations points, which can give a higher degree of accuracy. Increasing the number of integrations points increases the computational cost.

1st Order 2nd Order 3rd Order

Figure 32. Gaussian Quadrature

ENTER SOLUTION ALGORITHM NUMBER>
0=ELASTICITY ONLY
1=CONSTANT STIFFNESS
2=TWO STEP PROCESS
3=TANGENT STIFFNESS

The solution algorithm determines which type of stress/strain relationship is to be considered, linear or non-linear. The first option analyzes elastic solutions only and care should be taken to ensure that the stresses remain elastic. The last three options treat materials in the plastic region. The tangent stiffness is the best of the three.
<ENTER STRESS/STRAIN TYPE NUMBER>
0=PLANE STRAIN
1=PLANE STRESS
2=AXISYMMETRIC PROBLEM

The stress/strain type allows the user to model different geometric conditions. Plane strain is used for thick shapes. Plane stress is used for thin shapes, such as plates. Axisymmetrical problems model geometrical shapes which are symmetrical about the Z axis.

<ENTER YIELD CONDITION PARAMETER NUMBER>
1=VON MISES
2=TRESCA
3=DRUCKER-PRAGER
4=BELTRAMI

This parameter specifies the yield theory used. Von Mises and Tresca are generally used with metallic materials and Drucker-Prager and Beltrami are used for soils. See section 2.1 on material modeling.

<ENTER THE STIFFNESS CONTROL NUMBER>
0=NO. ELASTICALLY COUPLED NODES
1=INPUT STIFFNESS COEFFICIENTS

The stiffness control allows the input of the stiffness matrix, this option is rarely used (use 0).

<ENTER RUBBLE FLAG NUMBER, 0=NO RUBBLE, 1=RUBBLE>

At this point the user may specify if a material or materials are to be allowed to rubbleize.

<ENTER BANDWIDTH>

This refers to the size of the stiffness matrix and is not used unless a stiffness control of 1 is specified.

<ENTER NUMBER OF COORDINATES PER NODE,(DEFAULT=2)>

The number of coordinates is the dimension of the space considered.

<ENTER NUMBER OF GAUSS POINTS>

This is usually the same as the Gaussian quadrature entered earlier. This prompt refers to the number of gauss points for nodal force residual calculation and stress storage.
MATERIAL PROPERTIES

This section defines properties which model a given material. Material properties are input for the total number of materials in the body. To reduce needless interactive prompting the user is not be prompted for the yield angle or the material thickness unless the corresponding solution controls were previously entered.

<ENTER YOUNG'S MODULUS OF ELASTICITY FOR MATERIAL I>
Young's modulus is the slope of the stress/strain curve in the elastic region, for material I. A negative value in this position activates the low shear material description. In this case the bulk modulus K, is entered as a negative value here and the shear modulus G, is entered in the place of Poisson's ratio.

<ENTER POISSON'S RATIO OF MATERIAL I>
Poisson's ratio is the ratio of lateral to longitudinal strains, of material I.

<ENTER YIELD STRESS OF MATERIAL I>
This is the magnitude of stress at which the material begins to yield and deform plastically. If the Drucker-Prager option is selected the cohesion, c, is entered here.

<ENTER HARDENING MODULUS OF MATERIAL I>
At this point the user may specify how the plastic stress/strain curve is to be defined. This stress/strain curve defines the curve used by SCRUBS to interpolate the magnitudes of stress and strain for the plastic solutions. The hardness modulus is the slope of the stress/strain curve in the plastic region. Note: this is true stress, true strain rather than engineering stress strain. If the modulus is entered in as a positive value, the plastic stress/strain is defined by the hardening modulus. If it is entered in as negative, the user will be prompted for tabulated stress/strain points defining the plastic stress/strain curve. The two options for defining the plastic region are illustrated in Figure 33. If the Drucker-Prager option is selected, the cohesion, c, is input as a function of plastic strain, rather than yield stress as a function of plastic strain.
Defined by the hardening modulus
Defined by tabulated stress/strain points

FIGURE 33. Plastic Stress/Strain Curve

<ENTER CONICAL YIELD SURFACE ANGLE OF MATERIAL I>
This is the angle of friction and is entered if the Drucker-Prager yield condition is chosen.

<ENTER THICKNESS OF MATERIAL I>
Enter the material thickness if plane stress conditions are considered.
TABULATED STRESS/STRAIN DATA

Tabulated yield stress as a function of plastic strain is to be entered for a given material if the corresponding hardening modulus for that material was set negative. These stress/strain points define the plastic stress/strain curve point by point rather than using a single slope hardening modulus, (see hardening modulus in the Material Properties Section). Sample true stress/strain curves are given in Figure 34.

<ENTER NUMBER OF TABULATED STRAIN POINTS FOR MATERIAL I>

For the specified material enter the number of tabulated stress/strain points to be input.

<ENTER STRESS VALUE FOR POINT I>
This is the value of plastic true stress for a given point for the previously specified material.

<ENTER VALUE OF STRAIN FOR POINT I>
This strain is the plastic true strain corresponding to the stress just input.

FIGURE 34. True Stress Strain Curves
MATERIAL RUBBLEIZATION/ELEMENTAL DEATH DATA

This section defines the rubble parameters and element death (or mining) sequence. The rubble bulking parameter and the failure stress level for each material are entered for the rubble description. The specific elements that are mined are also defined.

<ENTER BULKING PARAMETER OF MATERIAL I>

The bulking parameter is the ratio of material expansion, the initial volume over the new volume, always less than one.

<ENTER FAILURE STRESS OF MATERIAL I>

The failure stress is the magnitude of stress at which the material fails and rubbleization occurs.

<ENTER MINIMUM Z COORDINATE, OF WHICH ELEMENTS MAY NOT PASS (E FORMAT)>

The minimum Z coordinate specifies the coordinate through which the rubbleized elements may not pass.

<ENTER TOTAL NUMBER OF INCREMENTS>

The total number of increments specifies the number of successive increments for which a given number of elements are to be removed.

<ENTER NUMBER OF ELEMENTS TO BE MINED ON INCREMENT I>

This defines the number of elements which are to be removed, or mined, on increment I.

<ENTER N ELEMENT NUMBERS TO BE MINED ON INCREMENT I>

The N specific elements which are to be removed on increment I are now input.
BOUNDARY CONDITIONS

This section defines the boundary conditions and provides the option of specifying a singular quarter point crack tip element. The user inputs the boundary flag assigned in QMESH along with the desired boundary condition. PRESCRUBS assigns the desired boundary condition to each nodal point that has that boundary flag.

<DO YOU WANT TO DEFINE A SINGULAR POINT (Y OR N)>

If a singular point is to be defined for use in a crack-tip analysis enter 'Y' for yes. Note: a crack-tip analysis is valid only for the quadratic element.

<ENTER NODE NUMBER OF SINGULAR POINT>

Enter the node number of the point of the crack-tip. Defining a singular point will automatically translate the midside nodes of the surrounding elements to their 'quarter point' positions.

<ENTER FILE NAME>, TYPE H FOR HELP

Enter the name of the QMESH file containing the finite element mesh and boundary flags of the body.

DO YOU TO CONSIDER BOUNDARY ANGLES? (Y or N)

This option is used if a boundary of a given body is inclined with respect to the coordinate axis. By typing 'Y' the angle between the axis and boundary can be defined.

DO YOU WANT TO READ A BOUNDARY FLAG? (Y or N)

This prompt allows the user to interactively process boundary conditions. Answering "yes" to the prompt starts the input loop; answering "no" ends the boundary processing.

<ENTER BOUNDARY FLAG NUMBER AS SET IN QMESH>

This is the boundary flag set in QMESH identifying a given boundary.
This boundary code defines a given set of displacement constraints which act on the nodal points corresponding to the flag set in QMESH.

**<ENTER VALUE OF R DISPLACEMENT>**
This is the specified displacement in the R (i.e. X) direction.

**<ENTER VALUE OF Z DISPLACEMENT>**
This is the specified displacement in the Z (i.e. Y) direction.

**<ENTER INCLINED BOUNDARY ANGLE>**
Enter the boundary angle described previously. If no boundary angles were specified this prompt will not be given.

![Figure 35. Boundary Angle](image)

**<ENTER NUMBER OF SUMMED REACTION NODES>**
This is the total number of reaction nodes which are to be summed.

**<ENTER SUMMED REACTION NODE NUMBER>**
Enter the reaction node numbers.
LOADING CONTROLS

This section specifies the manner in which the body is to be loaded. The user may specify independent loadings, loadings proportional to a load factor array, or loads equal on all increments. The user may also specify the possible load combinations which are: concentrated loads, pressure or shear tractions, gravitational loads, or centrifugal forces.

<ENTER TITLE FOR LOADING CASE I>
This is the title or description of the current loading case (80 characters or less).

<ENTER INCREMENT CONTROL FOR PRESCRIBED LOADS AND DISPLACEMENTS>
0=EQUAL INCREMENTS
1=INCREMENTS PROPORTIONAL TO FACTOR ARRAY
2=INDEPENDENT SET OF LOADS FOR EACH INCREMENT

The increment control allows the user to specify how loads or displacements are to be handled over a given time increment. Displacements and loads can be treated as constant, proportional to the elements of the factor array, or as different loadings for each increment.

<DO YOU WANT TO CONSIDER POINT LOADINGS? (Y or N)>
This allows the application of concentrated loads onto nodal points.

<DO YOU WANT TO CONSIDER GRAVITY LOADS? (Y or N)>
This allows the processing of forces due to gravitational or magnetic fields.

<DO YOU WANT TO CONSIDER PRESSURE LOADS? (Y or N)>
This allows surface pressures and shear tractions.

<DO YOU WANT TO CONSIDER CENTRIFUGAL FORCES? (Y or N)>
This considers forces caused by centripetal acceleration.
POINT LOADING DATA

This section enters concentrated load data. The total number of loaded nodes is entered, then each individual node with the corresponding load in the X and Y direction are entered.

<ENTER NUMBER OF NODES WHERE POINT LOADS ARE APPLIED>

Enter the total number of nodal points in the body which are to be loaded.

<ENTER NODE NUMBER AND FORCES IN X,Y DIRECTION>

Enter three numbers: the nodal point to be loaded, the magnitude of the concentrated load in the X direction, and the magnitude of the concentrated load in the Y direction.

FIGURE 36. Point Load in -Z Direction
GRAVITATIONAL, CENTRIFUGAL LOADINGS

This section enters the gravitational constant and the angular velocity and material density; if either gravitational or centrifugal loading was previously specified.

<IF GRAVITY AXIS IS TILTED ENTER GRAVITY ANGLE>
If the gravitational axis is not along the negative Z axis enter the value of the angle between the negative Z axis and the gravity axis. The angle entered is positive clockwise.

<ENTER CONSTANT OF ACCELERATION DUE TO GRAVITY>
Enter the magnitude of the gravitational constant or the constant of the magnetic field.

<ENTER ANGULAR VELOCITY IN RADIANS/UNIT TIME>
Enter the angular velocity of the body as it revolves about the Y axis.

<ENTER DENSITY FOR MATERIAL 1>
For the material number prompted, enter the material density. Enter densities for all materials.
PRESSURE LOADINGS

This section allows the input of pressure loadings or tractions on a given element surface. The total number of elements on which pressure or shear tractions are applied is entered, then each element along the loaded surface is entered with its corresponding node number and the pressure and shear tractions at the nodes.

<ENTER NUMBER OF LINE ELEMENTS ON WHICH PRESSURES ARE APPLIED>

Enter the total number of elements on which pressure loads are applied.

<ENTER I SPECIFIED NODE NUMBERS ON ELEMENT PRESSURE SURFACE>

Enter 1 nodal points which lie on the edge of the element being loaded. The number of nodes varies depending on the type of element being considered. Linear elements have two loaded nodes, quadratic elements have three, and cubic elements have four.

<ARE SAME LOADS APPLIED ON ALL NODES OF THE LINE ELEMENT? (Y or N)>

If the normal or tangential pressures are constant across the loaded surface only those pressures need to be entered rather than the individual pressures at each nodal point. If a linear element is being used the individual pressure loads must be applied for each node.

<ENTER CONSTANT NORMAL PRESSURE>

This is the magnitude of a uniformly distributed load applied across the loaded surface.

<ENTER CONSTANT TANGENTIAL PRESSURE>

This is the magnitude of a uniformly distributed shear traction applied across the loaded surface.
<ENTER NORMAL PRESSURE OF NODE I>
Enter the magnitude normal pressure applied on the nodes previously entered. Note these pressures must be entered in the same order as were the corresponding nodal points.

<ENTER TANGENTIAL PRESSURE OF NODE I>
Enter the magnitude of the shear traction in the same manner as the normal pressures for individual nodal points.
LOAD FACTORS AND INCREMENT DATA

This section specifies increment and load factor data. The load factors are used to increase or decrease the magnitude of the current loadings over a given time increment. At this point the user may specify the type of output data which is to be printed. Extensive output is available or a reduced amount of output may be specified according to the user's desire.

<ENTER NUMBER OF LOAD INCREMENTS>
This is the number of incremental steps in which the loading will be applied.

<ENTER LOAD FACTOR FOR INCREMENT I>
This is a factor applied to the specified loadings or displacements to define the magnitude of the load to be added for increment I. The total load at any step is, of course, the sum of all the increments to that point.
The SCRUBS.LIS file will contain a listing of the solution for each loading increment.

<ENTER INITIAL AND FINAL INCREMENT WRITE-OUT INDICATOR>
0=NO OUTPUT
1=DISPLACEMENTS AT NODAL POINTS
2=REACTIONS AT CONSTRAINED NODES AS WELL
3=STRESSES AT GAUSS POINTS AS WELL
4=RESIDUAL FORCES AT NODES AS WELL

The write-out indicator is a three digit number specifying the type of output data for the initial and final increment. The first of the three digits indicates the type of output for the initial increment, the second digit is zero, and the third digit indicates the type of output for the final increment. The type of output is specified by choosing one of the five conditions listed above. For example, the number 100 will cause the displacements at the nodal points to be printed for the initial increment and no output will be printed for the final increment. For most cases simply use 003.
<ENTER MAXIMUM NUMBER OF ITERATIONS>

The maximum number of iterations is the number of iterations allowed for convergence to occur. If convergence has not occurred calculations will be terminated.

<ENTER CONVERGENCE FACTOR IN PERCENT>

The convergence factor is the value used to determine whether the solution is within a certain degree of accuracy (i.e. convergence has been obtained). SCRUBS checks convergence on the ratio of the sum of the norm of force residuals divided by the sum of the norm of applied forces.
IMPACT BOUNDARY NODES

This section enters impact boundary nodal points.

<ENTER NUMBER OF IMPACT BOUNDARY NODES>
This is the total number of boundary nodes subjected to impact.

<ENTER IMPACT NODE NUMBER>
This is the boundary node number subjected to impact.

PRINT-OUT CONTROLS

This section allows the user to control the printed output data. Output for all elements or for a few selected elements may be defined.

<ENTER NUMBER OF ELEMENTS TO BE PRINTED>
This is the total number of elements for which output data is to be printed.

<ENTER NODAL OUTPUT CONTROL>
0=ALL NODES PRINTED
1=NO NODES PRINTED
This control specifies whether data for all or no nodes are to output.

<ENTER ELEMENT TO BE PRINTED>
Enter the number of the element which is to be printed.
CHAPTER 4

PRESCRUBS.BYU PROGRAMMER'S GUIDE

This chapter presents a programmer's guide for PRESCRUBS.BYU and contains information pertaining to the operation and maintenance of this program. This chapter is designed to help those who will be using and modifying this code. Seven basic topics are treated here:

1) an introduction to the main options in the program
2) subroutine descriptions
3) variable definitions index
4) current program dimensions
5) changing program dimensions
6) system dependent features
7) subroutine flow charts
Program Options

Option 1 (Enter new set of data)

This option interactively prompts the user for the necessary data required for the execution of SCRUBS.FOR. The data is then read using the following fail-safe subroutines READIN, READRL, or CHOOSE. If an invalid entry is specified, program execution returns and prompts for a new value until a valid number is entered. If the user desires help, by typing "H" subroutine HELP is called and the appropriate help message is displayed.

Upon completion of the data input, the data is then printed on the terminal screen and the user is prompted if modifications are desired. If "yes", a modification menu is then displayed and MFLAG is set to be 1. The user then selects the the option to be modified and the program branches to where the data was originally input. The user is again prompted for the necessary data after which execution is returned to viewing of the data.

If no further changes are desired for the current set of data, the user will select the last menu option or type a carriage return and program execution will continue to the next group of data to be input.

Option 2 (View or modify a set of data)

Option 2 allows the user to view or modify an existing set of data. This option sets the MODIFY flag to 1 and displays the modification menu. The user selects the option to be viewed or modified, the program then branches to the subroutine associated with that data set. With the MODIFY flag set to 1, program execution by-passes the subroutine's interactive data input section and goes directly to the viewing of the data. If the data is correct, the user may select the last option of the modify menu and execution will return to another associated data set or to the main view or modification menu. If the user desires to modify data, the program follows the same program execution as data modified in option 1.
Option 3 (Reads data from a disc file)

This option reads the file called (NAME) written on the logical unit number (NUNIT) defined in the main program. The file is then closed, using the logical unit number (NUNIT).

Option 4 (Writes data on a disc file)

This option takes the file name (NAME) and the logical unit number (NUNIT) defined in the main, opens the file corresponding to NAME and writes the present set of data. The file is then closed, using the logical unit number (NUNIT).

Option 5 (Prints a hard copy of present data)

In selecting this option, the IPRINT flag is set to 1. Each subroutine is called as in Option 1, but upon entering the subroutine the IPRINT flag is tested and found to be one. The logical unit number is changed from that of the terminal (OUT), to the unit number of the line printer (LPT). Program execution by-passes the interactive input and branches only to the data viewing section. With the unit number changed to that of the line printer a file is created and the data viewing is sent to that file. The modification option is also by-passed as execution is transferred from the current subroutine to following subroutines. The process continues until all subroutines have been called; at this point, the file created is sent to the line printer by means of the dispose command.

Note: The copy printed is sent to the user's default line printer.

Option 6 (Ends program execution)
Subroutine SOLCON

This subroutine enters title and solution controls which define the problem. The solution controls specified in this subroutine will be used throughout the program, controlling the data input. For example, if rubble is specified (NL=1) subsequent sections dealing with rubble will be accessed. If no rubble were selected these sections are by-passed.

Arguments Input:
MAXLD = maximum number of load cases per problem
MATMAX = maximum number of materials per problem
MODIFY = main modification flag
IMPRINT = hard copy print-out flag

Arguments Output:
TITLE = title of problem
NPROB = total number of problems to be solved
NLD = total number of load cases per problem
NDF = number of degrees of freedom
NMAT = number of materials considered
NSFR = element type: 1=linear, 2=parabolic, 3=cubic
NGAUS = number of gauss points for stiffness
NALGO = solution algorithm
NPP = type of stress/strain
NYIELD = yield condition parameter
NT = stiffness control
NL = rubble flag 0=no rubble, 1=rubble
NJZ = bandwidth
NCOORD = number of coordinates per node
MGAUS = number of gauss points for force residual

Common Input: IN- device number of terminal input
OUT- device number of terminal output
LPT- device number of line printer

Dimension: TITLE(20)

Maximums Tested: MATMAX
MATLD
Flags:
NOTRD= not read flag, returns for data re-entry if
NOTRD=0
MFLAG= modification flag, returns program execution from
data modification to view, and then to modify menu
if MFLAG=1
MODIFY= modification flag, by-passes interactive data input
if MODIFY=1
IPRINT= hard copy flag, This changes NDEV from OUT to LPT,
creating a file which is sent to the line printer.
Program execution bypasses interactive input and modification.

Help Files: S1

Format:

<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>20A4</td>
<td>20A4</td>
</tr>
<tr>
<td>ALL OTHERS</td>
<td>*</td>
<td>15</td>
</tr>
</tbody>
</table>
Subroutine MATPRP

Subroutine MATPRP enters the material properties of the body.

Arguments Input:
NMAT = number of material
NYIELD = yield condition parameter 3=Drucker-Prager
NPP = stress/strain type 1=plane stress
NL = rubble flag 0=no rubble, 1=rubble
MATMAX = maximum number of materials
MODIFY = main modification flag
IPRINT = hard copy print-out flag

Arguments Output:
ORT(N,1) = Young's modulus of elasticity
ORT(N,2) = Poisson's ratio
ORT(N,3) = yield stress
ORT(N,4) = hardening modulus
ORT(N,5) = conical yield surface angle
ORT(N,6) = thickness of material

Common Input: See subroutine SOLCON

Dimension: ORT(MATMAX,8)

Flags: NL, NYIELD, NPP, also see subroutine SOLCON

Help Files: M1,HF

Format:
<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORT(N,1)</td>
<td>G10.4</td>
</tr>
<tr>
<td>ORT(N,2)</td>
<td>G10.5</td>
</tr>
<tr>
<td>ORT(N,3)</td>
<td>G10.5</td>
</tr>
<tr>
<td>ORT(N,4)</td>
<td>G10.4</td>
</tr>
<tr>
<td>ORT(N,5)</td>
<td>G10.5</td>
</tr>
<tr>
<td>ORT(N,6)</td>
<td>G10.5</td>
</tr>
</tbody>
</table>

Note: if changes are to be made in formatting, the formats for both the unrubbleized and rubbleized must be modified.
Subroutine TABBY

Subroutine TABBY inputs tabulated stress/strain data if the plastic stress/strain curve is to be defined by tabulated points. If the hardening modulus is negative, ORT(N,4), is negative for a given material tabulated points are entered. If positive, no points are entered.

Arguments Input:
- NMAT = number of materials
- NL = rubble flag 0=no rubble, 1=rubble
- ORT(N,4) = hardening modulus
- MAXTAB = maximum number of tabulated str/stn points
- MAXMAT = maximum number of materials
- MODIFY = main modification flag
- IPRINT = hard copy print-out flag

Arguments Output:
- NTAB(N) = number of tabulated str/stn points
- TABSTR(N,I) = tabulated stress at point I
- TABSTN(N,I) = tabulated strain at point I

Common Input: See subroutine SOLCON

Maximum Tested: MAXTAB

Dimension:
ORT(MATMAX,8), NTAB(MATMAX), TABSTR(MATMAX,MAXTAB)
TABSTN(MATMAX,MAXTAB)

Flags: NL, ORT(N,4), also see subroutine SOLCON

Help Files: T1

Comments: The passing of ORT in the subroutine argument could be reduced to the passing of the single array ORT(N,4).

Format:

<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTAB</td>
<td>*</td>
<td>-</td>
</tr>
<tr>
<td>TABSTR</td>
<td>*</td>
<td>G10.5</td>
</tr>
<tr>
<td>TABSTN</td>
<td>*</td>
<td>G10.5</td>
</tr>
</tbody>
</table>
Subroutine RUBBLE

Subroutine RUBBLE inputs material rubbleization data
if NL=1. Also elemental mining data is entered if material
is removed from a body.

Arguments Input:

NMAT = number of materials
NL  = rubble flag 0=no rubble, 1=rubble
MATMAX = maximum number of materials
MODIFY = main modification flag
IPRINT = hard copy print-out flag

Arguments Output:

ORT(N,7) = failure stress
ORT(N,8) = bulking parameter
AK= minimum z coordinate of which nodes not pass

Common Input: See subroutine SOLCON

Dimension: ORT(MATMAX,8)

Flags: NL= rubble flag 0=no rubble, 1=rubble
Also see subroutine SOLCON

Help Files: RL, HF

Format:

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORT(N,7)</td>
<td>G10.5</td>
</tr>
<tr>
<td>ORT(N,8)</td>
<td>G10.5</td>
</tr>
<tr>
<td>AK</td>
<td>G10.4</td>
</tr>
</tbody>
</table>
Subroutine MINE

Subroutine MINE enters elemental mining data. This is a capability which models material removed from a body.

Arguments Input:
NMAT = number of materials
MATMAX = maximum number of materials
MAXINC = maximum number of rubble increments
MAXELY = maximum number of mined elements
MODIFY = main modification flag
IPRINT = hard copy print-out flag

Arguments Output:
NRFF = total number of increments
NME(I) = number of increments mined on increment I
MINES(I,J) = element mined on increment I

Common Input: See subroutine SOLCON

Dimension: NME(MAXINC), MINES(MAXINC,MAXELY)

Maximums Tested: MAXINC
MAXELY

Flags: See subroutine SOLCON

Help Files: RL

Possible Problems:
The current output viewing format is dependent on the number of elements to be 16 or less. The present format prints up to 8 values on one line, if the number of elements is more than 8, the remainder is printed on the next line. If the number of elements is increased, simply increase the test value and the do-loop limit to allow continuing lines of elements.

Format:

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRFF</td>
<td>* I3</td>
</tr>
<tr>
<td>NME(I)</td>
<td>* I4</td>
</tr>
<tr>
<td>MINES(I,J)</td>
<td>* I5</td>
</tr>
</tbody>
</table>
Subroutine BOUND

Subroutine BOUND reads a geometric mesh file QMESH9.DAT and processes the boundary conditions. BOUND also enters the summed reaction nodes.

Arguments Input:
- MAXBND= maximum number of boundary conditions
- MAXSUM= maximum number of summed reaction nodes
- MODIFY= main modification flag
- IPRINT= hard copy print-out flag

Arguments Output:
- NNBB= number of boundary nodes
- MPR= number of summed reaction nodes
- NBC= boundary node number
- NFIX= boundary code
- US= specified displacement
- ANG= inclined boundary angle
- NOPR= summed reaction nodes
  Not used: DANG, IBFLAG, IBC, BCODE, PNOR, PTOZ

Common Input: See subroutine SOLCON

Dimension: IBFLAG(MAXBND), IBC(MAXBND), BCODE(MAXBND)
           ICODE(MAXBND), PNOR(MAXBND), PTOZ(MAXBND)
           NBC(MAXBND), NFIX(MAXBND), US(MAXBND,2)
           ANG(MAXBND), NOPR(MAXSUM), DANG(MAXBND)

Flags: See subroutine SOLCON

Help Files: L4

Format:

<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNBB</td>
<td></td>
<td>I5</td>
</tr>
<tr>
<td>MPR</td>
<td>*</td>
<td>I5</td>
</tr>
<tr>
<td>NBC</td>
<td></td>
<td>I5</td>
</tr>
<tr>
<td>NFIX</td>
<td></td>
<td>I5</td>
</tr>
<tr>
<td>US</td>
<td></td>
<td>G10.4</td>
</tr>
<tr>
<td>ANG</td>
<td></td>
<td>G10.4</td>
</tr>
</tbody>
</table>

Note: DANG, IBFLAG, IBC, BCODE, PNOR, PTOZ, are not used outside of this subroutine but are passed out with the argument list to facilitate variable dimensioning. These values are switched to NBC, NFIX, US which are passed out and ultimately used by SCRUBS.
Subroutine LOAD1

Subroutine LOAD1 enters loading description, point loading data, gravity and centrifugal force data.

Arguments Input:
- LNUM = number of current load case
- MAXPNT = maximum number of point loadings
- MAXLD = maximum number of load cases per problem
- MODIFY = main modification flag
- IPRINT = hard copy print-out flag

Arguments Output:
- LDTYPE(N) = increment control for load case N
- TITLE1(N) = title for load case N
- NRE(N) = number of loads with point loads
- NRG(N) = gravity load flag 0=no, 1=gravity loads
- NRPRS(N) = number of pressure loads
- NRC(N) = centrifugal forces 0=no, 1=considered
- ND(N,I) = node number with point loading
- FX(N,I) = force at node in x direction
- FY(N,I) = force at node in y direction

Common Input: See subroutine SOLCON

Dimension: TITLE1(MAXLD,20),LDTYPE(MAXLD),NRE(MAXLD)
NRG(MAXLD),NRPRS(MAXLD),NRC(MAXLD)
ND(MAXLD,MAXPNT),FX(MAXLD,MAXPNT)
FY(MAXLD,MAXPNT)

Maximums Tested: MAXPNT

Flags: See subroutine SOLCON

Help Files: L1

Format:

<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE1</td>
<td>20A4</td>
<td>20A4</td>
</tr>
<tr>
<td>LDTYPE(N)</td>
<td>*</td>
<td>A</td>
</tr>
<tr>
<td>NRE(N)</td>
<td>A1</td>
<td>A</td>
</tr>
<tr>
<td>NRG(N)</td>
<td>A1</td>
<td>A</td>
</tr>
<tr>
<td>NRPRS(N)</td>
<td>A1</td>
<td>A</td>
</tr>
<tr>
<td>NRC(N)</td>
<td>A1</td>
<td>A</td>
</tr>
<tr>
<td>ND(N,I)</td>
<td>*</td>
<td>I2</td>
</tr>
<tr>
<td>FX(N,I)</td>
<td>*</td>
<td>G10.4</td>
</tr>
<tr>
<td>FY(N,I)</td>
<td>*</td>
<td>G10.4</td>
</tr>
</tbody>
</table>
Subroutine LOAD2

Subroutine LOAD2 enters gravity and centrifugal constants, and material densities if NRC=1 or NRG=1.

Arguments Input:
- LNUM = number of current load case
- NMAT = number of materials
- NL = rubble flag 0=no, 1=rubble considered
- NRC = centrifugal flag
- NRG = gravity flag
- MAXLD = maximum number of load cases per problem
- MATMAX = maximum number of materials
- MODIFY = main modification flag
- IPRINT = hard copy print-out flag

Arguments Output:
- THETA(LNUM) = gravity angle of load case LNUM
- GRAV(LNUM) = gravity constant of load case LNUM
- ANGVEL(LNUM) = angular velocity of load LNUM
- DENS(LNUM,I) = density load case LNUM, material I

Common Input: See subroutine SOLCON

Dimension: NRC(LNUM), NRG(LNUM), THETA(MAXLD), GRAV(MAXLD), ANGVEL(MAXLD), DENS(MAXLD, MATMAX)

Flags: NL, NRC, NRG, also see subroutine SOLCON

Help Files: L2,HF

Format:

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>THETA(LNUM)</td>
<td>*</td>
</tr>
<tr>
<td>GRAV(LNUM)</td>
<td>*</td>
</tr>
<tr>
<td>ANGVEL(LNUM)</td>
<td>*</td>
</tr>
<tr>
<td>DENS(LNUM,I)</td>
<td>*</td>
</tr>
</tbody>
</table>
Subroutine LOAD3

Subroutine LOAD3 inputs elemental pressure loads and surface tractions.

Arguments Input:
NRPRS(LNUM) = pressure load flag 0=no, 1=yes
NSFR = type of element
MODIFY = main modification flag
IPRINT = hard copy print-out flag

Arguments Output:
LNE(LNUM) = number of elements with pressure loads
NOPL(LNUM, I, K) = node numbers of element
PN(LNUM, I, K) = normal pressure at: node K, element I
PT(LNUM, I, K) = tangential pressure at: K, I

Common Input: See subroutine SOLCON
MFLAG not used

Maximums Tested: MAXPRS

Dimension: NRPRS(MAXLD), LNE(MAXLD), NOPL(MAXLD, MAXPRS, 4)
PN(MAXLD, MAXPRS, 4), PT(MAXLD, MAXPRS, 4), MODARR(30)

Help Files: L3, H4

Format:

<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNE(LNUM)</td>
<td>*</td>
<td>I5</td>
</tr>
<tr>
<td>NOPL(LNUM, I)</td>
<td>*</td>
<td>I5</td>
</tr>
<tr>
<td>PN(LNUM, I, K)</td>
<td>*</td>
<td>G10.4</td>
</tr>
<tr>
<td>PT(LNUM, I, K)</td>
<td>*</td>
<td>G10.4</td>
</tr>
</tbody>
</table>
Subroutine LOAD4

Subroutine LOAD4 inputs load factors, print-out indicators, maximum number of iterations, convergence factor.

Arguments Input:
MAXLD = maximum number of load cases per problem
MAXINC = maximum number of increments
MODIFY = main modification flag
IPRINT = hard copy print-out flag

Arguments Output:
NINC(LNUM) = number of increments
FAC(LNUM,I) = load factor on increment I
NOUT(LNUM,I) = print-out indicator for increment I
NIT(LNUM) = maximum number of iterations
CONFAC(LNUM) = convergence factor

Common Input: See subroutine SOLCON

Maximums Tested: MAXINC

Dimension:
NINC(MAXLD),FAC(MAXLD,INCMAX),NOUT(MAXLD,INCMAX)
NIT(MAXLD),CONFAC(MAXLD)

Flags: See subroutine SOLCON

Help Files: L4

Format: Input  Output
NINC(LNUM)  *   I5
FAC(LNUM,I)  *   G10.4
NOUT(LNUM,I)  *   I5
NIT(LNUM)  *   I5
CONFAC  *   G10.4
Subroutine LOAD5

Subroutine LOAD5 enters impact boundary nodes.

Arguments Input:
MAXLD = maximum number of load cases per problem
MAXIMP= maximum number of impact nodes
MODIFY= main modification flag
IPRINT= hard copy print-out flag

Arguments Output:
NIMP(LNUM) = number of impact nodes
IMPNOD(LNUM,I) = boundary impact node

Common Input: See subroutine SOLCON

Maximums Tested: MAXIMP

Dimension: NIMP(MAXLD), IMPNOD(MAXLD,MAXIMP)

Flags: See subroutine SOLCON

Help Files: L4

Format: Input Output
NIMP(LNUM) * I5
IMPNOD(LNUM,I) * I5
Subroutine PRTCON

This subroutine enters the reduced print-out controls which determine the quantity of output desired.

Arguments Input:
MAXPRT = maximum number of printed elements
MODIFY = main modification flag
IPRINT = hard copy print-out flag

Arguments Output:
NELP = number of elements to printed
NONODE = nodal output control
NPEL(I) = number of elements to be printed

Common Input: See subroutine SOLCON

Dimension: NPEL(MAXPRT)

Maximum Tested: MAXPRT

Flags: See subroutine SOLCON

Help Files: Pl

Format:

<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>NELP</td>
<td>*</td>
<td>I5</td>
</tr>
<tr>
<td>NONODE</td>
<td>*</td>
<td>I2</td>
</tr>
<tr>
<td>NPEL</td>
<td>*</td>
<td>I5</td>
</tr>
</tbody>
</table>
Subroutine WRTDSK

Subroutine WRTDSK writes current data onto a disc file in a format compatible for SCRUBS.FOR.

Arguments Input:
\[ \text{NAME} = \text{name of file created} \]
\[ \text{NUNIT} = \text{device number where file is written} \]
\[ \text{MAXIMUM PARAMETERS} = \]
\[ \text{MAXLD,MATMAX,MAXTAB,MAXINC,MAXELY,} \]
\[ \text{MAXPNT,MAXPRS,MAXIMP,INCMAX,MAXPRT} \]

Common Input: all variables considered in program

Dimensions: same as main program

Comments: write formatting uses G format to facilitate F or E format

Format: see individual subroutine associated with variable

Subroutine READSK

Subroutine READSK reads data from a disc file in same format as SCRUBS.FOR.

Arguments Input:
\[ \text{NAME} = \text{name of file created} \]
\[ \text{NUNIT} = \text{device number from which file is read} \]
\[ \text{MAXIMUM PARAMETERS} = \]
\[ \text{MAXLD,MATMAX,MAXTAB,MAXINC,MAXELY,} \]
\[ \text{MAXPNT,MAXPRS,MAXIMP,INCMAX,MAXPRT} \]

Common Input: See subroutine SOLCON

Common Output: all variables considered in program

Dimensions: same as main program

Comments: read format is structured the same as SCRUBS.FOR

Format: see individual subroutine associated with variable
Subroutine READIN

This is a fail-safe data input routine which reads integer values in free format. If an error is detected in input the flag NOTRD is set to zero. NOTRD is passed back to where this routine was called and then tested if equal to zero. If NOTRD is equal to zero execution is transferred back to the write statement re-prompting for the incompatible value. If no errors are detected, program execution continues.

Arguments Input: HLPFIL = help file identifier

Arguments Output: NUM = integer value to be read
NOTRD = not read flag 0= no read

Common Input: See subroutine SOLCON

Format: free format

Subroutine READRL

This is a fail-safe data input routine which reads real formatted values in free format. If an error is detected in input the flag NOTRD is set to zero. NOTRD is passed back to where this routine was called and tested if equal to zero. If NOTRD is equal to zero execution is transferred back to the write statement re-prompting for un-entered value. If no errors are detected, program execution continues.

Arguments Input: HLPFIL = help file identifier

Arguments Output: XNUM = integer value to be read
NOTRD = not read flag 0= no read

Common Input: See subroutine SOLCON

Format: free format
Subroutine CHOOSE

CHOOSE is a fail safe routine which is used to check input menu options for validity. If an invalid entry is input the user is prompted for a new value until something valid is entered. If a carriage return is typed, CHOOSE is set equal to NUM; this sets the option selected equal the last menu option which termination program execution at the current menu location.

Arguments Input: NUM = number of choices in menu
MENU = help file identifier

Arguments Output: ICHOOSE = option selected

Common Input: See subroutine SOLCON

Dimension: ICHAR(19)
Format: A format

Subroutine HELP

Subroutine HELP consists of a number of interactive help messages available to user. Helpfiles are accessed by typing 'H', or by doing something stupid like entering alpha characters when numerical input is required.

Arguments Input:
HLPFIL= HELP FILE INDICATOR
S1= Solution controls
M1= Material properties
T1= Tabulated stress/strain
R1= Rubbleization
L1= Load description
L2= Gravity and centrifugal forces
L3= Pressure loads
L4= Increment controls, impact nodes
P1= Print-out controls
NM= File name identifier
BF= Boundary conditions

Common Input: See subroutine SOLCON
SUBROUTINE RUBBLE

VARIABLE

AK = MINIMUM Z COORDINATE
ANG(I) = INCLINED BOUNDARY ANGLE
ANGVEL = ANGULAR VELOCITY

LOAD2 LOAD4 LOAD2 LOAD4 LOAD2 LOAD2

LOAD4 CONFAC(LNUM) = CONVERGENCE FACTOR IN PERCENT
LOAD2 DENS(LNUM,N) = DENSITY FOR MATERIAL N
LOAD4 FAC(LNUM,I) = LOAD FACTOR OF INCREMENT I
LOAD1 FX(I) = POINT LOAD AT NODE IN X DIRECTION
LOAD1 FY(I) = POINT LOAD AT NODE IN Y DIRECTION

LOAD2 GRAV(LNUM) = CONSTANT OF ACCELERATION
LOAD5 IMPNOD(LNUM,I) = IMPACT NODE NUMBER

SOLCON MGAUS = GAUSS POINTS FOR NODAL FORCE RESIDUAL
MINE MINES(INC,I) = ELEMENTS MINED ON INCREMENT I

LOAD1 LDTYPE(LNUM) = INCREMENT CONTROL
LOAD1 LNUM = NUMBER OF LOAD CASE
LOAD3 LNE(LNUM) = NUMBER OF LOAD CASE

MATPRP ORT(I,1) = YOUNGS MODULUS
MATPRP ORT(N,2) = POISSONS RATIO
MATPRP ORT(N,3) = YIELD STRESS
MATPRP ORT(N,4) = HARDENING MODULUS
MATPRP ORT(N,5) = CONICAL YIELD SURFACE ANGLE
MATPRP ORT(N,6) = MATERIAL THICKNESS
RUBBLE ORT(N,7) = BULKING PARAMETER
RUBBLE ORT(N,8) = FAILURE STRESS

SOLCON NALGO = SOLUTION ALGORITHM
SOLCON NB = NUMBER OF RESTRAINED BOUNDARY POINTS
SOLCON NCORD = NUMBER OF COORDINATES PER NODE (DEFAULT 2)
SOLCON NDF = NUMBER OF DEGREES OF FREEDOM
SOLCON NE = NUMBER OF ELEMENTS
PRCON NHELP = NUMBER OF ELEMENTS TO BE PRINTED
BOUND NBC(I) = BOUNDARY NODE NUMBER
BOUND NFIX = BOUNDARY CODE
LOAD5 NIMP(LNUM) = NUMBER OF IMPACT NODES
LOAD4 NINC(LNUM) = NUMBER OF LOAD INCREMENTS
LOAD4 NIT(LNUM) = MAXIMUM OF ITERATIONS
SOLCON NL= RUBBLE CONTROL
SOLCON NLD= TOTAL OF LOAD CASES/PROBLEMS
SOLCON NGAUS= GAUSS POINTS FOR STIFFNESS CALC.
SOLCON NMAT= OF DIFFERENT MATERIALS (MAX 10)
BOUND NNBB= NUMBER OF BOUNDARY NODES
PRTCON NONODE=NODAL OUTPUT CONTROL
BOUND NOPR(I)= SUMMED REACTION NODE NUMBER
LOAD3 NOPL(LNUM,I)=NODE WITH APPLIED PRESSURE
PRTCON NOUT(LNUM,I)= OUTPUT CONTROL
SOLCON NP= TOTAL OF NODES (1000)
SOLCON NPP= STRESS STRAIN TYPE
SOLCON NPROB= TOTAL NUMBER OF PROBLEMS TO BE SOLVED
LOAD1 NRC(LNUM)= CENTRIFUGAL FORCE
LOAD1 NRC(LNUM)= GRAVITY LOADS
LOAD1 NRE(LNUM)= OF NODES WHERE POINTS ARE APPLIED
MINE NRFF= INCREMENTS WITH DEFINED ELEMENTS REMOVED
SOLCON NSFR= ELEMENT TYPE 1=LINEAR,2=PARABALIC,3=CUBIC
LOAD1 NRPS(LNUM)= NO PRESSURE LOADS APPLIED
SOLCON NSIZ= BANDWIDTH
LOAD1 NSTRS= (NOT USED)
SOLCON NT= STIFFNESS CONTROL
TABBY NTAB= OF TABULATED STRAIN POINTS
TABBY NTABSTN(N)= STAIN INCREMENT
TABBY NTABSTR(N)= STRESS INCREMENT
SOLCON NYIELD= YIELD CONDITION PARAMETER
MINE MINES(N,I)= MINED ELEMENT NUMBER
BOUND MPR= NUMBER OF SUMMED REACTION NODES
LOAD3 PN(LNUM,I,K)=NORMAL PRESSURE
LOAD3 PT(LNUM,I,K)=TANGENTIAL PRESSURE
LOAD2 THETA(LNUM)= ANGLE OF Y OR Z AXES
SOLCON TITLE(I)= TITLE OF PROBLEM
LOAD1 TITLE1(LNUM,I)= TITLE OF LOAD CASE
BOUND US(I,J)= SPECIFIED DISPLACEMENT
CURRENT PROGRAM MAXIMUMS

Title 80 characters
Number of Load Cases 3 cases
Number of Materials 10 including rubble material
Number of Tabulated Points 10 points
Number of Rubble Increments 10 increments
Number of Mined Elements 16 elements
Number of Summed Reaction Nodes 20 nodes
Nodes With Point Loads 50 nodes
Elements With Pressure Loads 50 elements
Number of Impact Nodes 100 nodes
Number of Loading Increments 100 increments
Number of Printed Elements 100 elements
Changing Dimensions

This program is structured using variable dimensioning to facilitate the ease in the changing of dimensions. To change dimensions two things must be done: 1. The maximum corresponding to the array being altered must be modified and 2. All arrays corresponding to that maximum must be modified. The maximum is modified by simply inserting the new dimensional value in the data statement shown below, and is located after the dimension statements in the main program. The dimension of the individual arrays corresponding to the new maximum are then altered. These dimension statements are located at the beginning of the main program. Because of variable formatting the dimension statements of the subroutines need not be altered.

E.G. To increase the maximum number of point loads to 100, the second argument in ND,FX,FY must be increased from 50 to 100. Then the statement MAXPNT=50 must be modified to MAXPNT=100

```
DATA MAXLD/3/,MATMAX/10/,MAXTAB/10/,MAXINC/10/,MAXELY/16/
MAXPNT/50/,MAXPRS/50/,MAXIMP/100/,INCMAX/100/,MAXPRT/100/
MAXBND/100/,MAXSUM/20/
```

```
DIMENSION TITLE(20)
DIMENSION ORT(10,8)
DIMENSION NTAB(10),TABSTR(10,10),TABSTN(10,10)
DIMENSION NME(10),MINES(10,16)
DIMENSION NBC(100),NFX(100),US(100,2),ANG(100)
DIMENSION NOPR(20)
DIMENSION IBFLAG(100),IBC(100),BCODE(100)
DIMENSION ICODE(100),PNOR(100),PTOZ(100)
DIMENSION TITLE1(3,20),LDTYPE(3),NRE(3),NRG(3)
DIMENSION NRPRS(3),NRC(3),ND(3,50),FX(3,50),FY(3,50)
DIMENSION THETA(3),GRAV(3),ANGVEL(3),DENS(3,10)
DIMENSION LNE(3),NOPL(3,50,4),PN(3,50,4),PT(3,50,4)
DIMENSION NINC(3),FAC(3,100),NOUT(3,100),NIT(3)
DIMENSION CONFCAC(3),NIMP(3),IMPNO(3,100)
DIMENSION NPEL(100)
```
The maximum value definitions are listed below.

- **MATMAX** - MATERIALS
- **MXTAB** - TABULATED STRESS/STRAIN POINTS
- **MIXINC** - RUBBLIZED INCREMENTS
- **MIXELY** - REMOVED MINING ELEMENTS
- **HAXLD** - LOAD CASES PER PROBLEM
- **HAXPNT** - NODES WITH POINT LOADS
- **MXPRS** - ELEMENTS WITH PRESSURE LOADS
- **HAXIMP** - IMPACT NODES
- **INCMAX** - INCREMENTS
- **HAXPRT** - PRINTED ELEMENTS

The arrays associated with the maximum values are shown below.

```plaintext
DIMENSION TITLE(20)  
DIMENSION ORT(MATMAX,8)  
DIMENSION NTAB(MATMAX),TABSTR(MATMAX,MXTAB)  
DIMENSION TABSTN(MATMAX,MXTAB)  
DIMENSION NME(MIXINC),MINES(MIXINC,MIXELY)  
DIMENSION NBC(MAXBND),NFIX(MAXBND),US(MAXBND,2)  
DIMENSION NRE(MAXLD),NOPR(MAXSUM)  
DIMENSION TITLE1(MAXLD,20),LDTYPE(MAXLD)  
DIMENSION NRE(MAXLD),NRG(MAXLD)  
DIMENSION NRPRS(MAXLD),NRC(MAXLD),ND(MAXLD,MAXPNT)  
DIMENSION FX(MAXLD,MAXPNT),FY(MAXLD,MAXPNT)  
DIMENSION THETA(MAXLD),GRAV(MAXLD),ANGVEL(MAXLD)  
DIMENSION DENS(MAXLD,MATMAX)  
DIMENSION LNE(MAXLD),NOPL(MAXLD,MAXPRS,4)  
DIMENSION PN(MAXLD,MAXPRS,4),PT(MAXLD,MAXPRS,4)  
DIMENSION NINC(MAXLD),FAC(MAXLD,INCMAX)  
DIMENSION NOUT(MAXLD,INCMAX),NIT(MAXLD)  
DIMENSION CONPAC(MAXLD),NIMP(MAXLD),IMPNOI(MAXLD,MIXIMP)  
DIMENSION NPEL(MAXPRT)  
```
System Dependant Features

A. Fortran 5
   1. Do while statement
      Subroutine BOUND
   2. Error Statement
      a. All input
      b. Open statements
         Subroutines BOUND, WRTDSK, READSK
      c. Accept Statement
         All subroutines

B. Byte size
   1. A4 Format
      a. Subroutine SOLCON (TITLE)
      b. Subroutine LOAD1 (TITLE1)

C. Logical Device Number
   1. IN-Logical device number of terminal input
   2. OUT-Logical device number of terminal output
   3. LPT-Logical device number of line printer
   4. NUNIT-Device number of QMESH, SCRUBS files
After re-entry of modified value MFLAG (which is now equal to 1) is tested and execution transferred to line 185.
As part of a work to develop a versatile two-dimensional, elastic-plastic finite element package at Brigham Young University this thesis added several new and powerful capabilities to the existing SCRUBS.BYU package. A quarter point algorithm was created making possible crack tip fracture mechanics solutions. An interactive user-friendly preprocessor PRESCRUBS.BYU was developed to create a SCRUBS input file and to automate the boundary condition definition. The subject matter presented in here focuses on the special features which incorporate isoparametric elements, fracture mechanics, and solutions to subsidence problems. Example problems using quadratic isoparametric elements, rubbleization subsidence, and fracture mechanics situations were presented demonstrating the capabilities of this package.

Although this thesis completes the interactive pre- and post-processing capabilities of the SCRUBS.BYU package there are many other capabilities which might
be added to enhance this package. Some of these possibilities are: the use of quadratic isoparametric elements for rubbleization subsidence problems, three-dimensional capabilities, a better understanding of low shear materials, a material model for regularly jointed media, and the capability to generate the cubic isoparametric element.
BIBLIOGRAPHY


DIMENSION TITLE(20)
DIMENSION ORT(10, 8)
DIMENSION NTAB(10), TABSTR(10, 10), TABSTN(10, 10)
DIMENSION NME(10), MINES(10, 16)
DIMENSION NBC(100), NFIX(100), US(100, 2), ANG(100)
DIMENSION NOPR(20)
DIMENSION IBFLAG(100), IBC(100), BCODE(100), DANG(100)
DIMENSIONICODE(100), PNDR(100), PTOZ(100)
DIMENSION TITLE1(3, 20), LTYPE(3), NRE(3), NRG(3)
DIMENSION NRPRS(3), NRCS(3), ND(3, 50), FX(3, 50), FY(3, 50)
DIMENSION THETA(3), GRAV(3), ANGVEL(3), DEN(3, 10)
DIMENSION LNE(3), NOPL(3, 50, 4), PNF(3, 50, 4), PT(3, 50, 4)
DIMENSION NINC(3), FAC(3, 100), NOUT(3, 100), NIT(3)
DIMENSION CONFC(3), NIMP(3), IMPNOD(3, 100)
DIMENSION NPEL(100)
COMMON IN, OUT, LPT
INTEGER OUT
DATA MAXLD/3/, MATMAX/10/, MAXINC/10/, MAXELY/16/,
+ MAXPNT/50/, MAXPRS/50/, MAXIMP/100/, INCMAX/100/, MAXPRT/100/
+ MAXBND/100/, MAXSUM/20/
CHARACTER*2 HLPFIL

5 CONTINUE
MFLAG=0
MODIFY=0
IPRINT=0
WRITE(OUT, 10)
10 FORMAT(1X, 'MAIN PROGRAM MENU:']/
+ '1. ENTER SCRUBS DATA FILE'/
+ '2. VIEW OR MODIFY SCRUBS DATA '/
+ '3. READ SCRUBS DATA FILE'/
+ '4. WRITE SCRUBS DATA FILE'/
+ '5. PRINT HARD COPY OF SCRUBS DATA'/
+ '6. TERMINATE PROGRAM' )

NUM=6
CALL CHOOSE(IOPT, NUM)

GOTO(20, 30, 110, 120, 130, 150) IOPT

20 CONTINUE
CALL SOLCON(TITLE, NPRS, NLD, NDF, NMAT, NSFR, NGAUS, NALQD, NPP,
+ NYIELD, NT, NL, NS1Z, NCORD, MGAUS, MATMAX, MAXLD, MODIFY, IPRINT)
CALL MATPRP (NMAT, NYIELD, NPP, NL, ORT, MATMAX, MODIFY, IPRINT)
CALL TABBY (NMAT, NL, NPP, NL, ORT, TABSTR, TABSTN, MATMAX, MAXTAB,
        + MODIFY, IPRINT)
CALL RUBBLE (NMAT, NL, ORT, AK, MATMAX, NRFF, NME, MINES,
        + MAXINC, MAXELY, MODIFY, IPRINT)
CALL BOUND (NSING, NNBB, MPR, NBC, NFIX, US, ANG, NOFR, DANG,
        + IBFLAG, IBC, BCODE, ICODE, PNR, PTO2, MAXBND, MAXSUM, IPRINT, MODIFY)
DO 25 LNUM=1, NLD
   CALL LOAD1 (LNUM, LDTYPE, TITLE1, NRE, NRG, NRPRS, NRC,
             + ND, FX, FY, MAXPNT, MAXLD, MODIFY, IPRINT)
   CALL LOAD2 (LNUM, NMAT, NL, NRG, THETA, GRAY, ANGVEL, DENS,
             + MODIFY, IPRINT, MAXLD, MATMAX)
   CALL LOAD3 (LNUM, NRPRS, NSFR, LNE, NOPL, PN, PT,
             + MAXPRS, MAXLD, MODIFY, IPRINT)
   CALL LOAD4 (LNUM, NINC, FAC, NOUT, NIT, CONFACT,
             + MAXLD, INCMAX, MODIFY, IPRINT)
   CALL LOAD5 (LNUM, NIMP, IMPNOD, MAXLD, MAXIMP, MODIFY, IPRINT)
25 CONTINUE
CALL PRTCON (NELP, NONODE, NPEL, MAXPR, MODIFY, IPRINT)
GOTO 140

C****** modify set of data ******

C
30 CONTINUE
MODIFY=1
WRITE (OUT, 40) FORMAT(' (SELECT OPTION TO BE MODIFIED')
+  1. GENERAL MODIFICATION'/,
+  2. SOLUTION CONTROLS'/,
+  3. MATERIAL PROPERTIES'/,
+  4. TABULATED STRESS/STRAIN DATA'/,
+  5. RUBBLEIZATION DATA'/,
+  6. BOUNDARY CONDITIONS'/,
+  7. LOAD DESCRIPTION, POINT LOADS'/,
+  8. GRAVITY AN CENTRIFUGAL, DENSITIES'/,
+  9. PRESSURE LOADS'/,
+  10. LOAD FAC. PRINT IND, ITERATIONS, CONVERGENCE'/,
+  11. IMPACT NODES'/,
+  12. PRINT-OUT CONTROLS'/,
+  13. END MODIFICATIONS'/)
NUM=13
CALL CHOOSE (IOPT, NUM)
GOTO (50, 55, 60, 65, 70, 73, 75, 75, 75, 75, 100, 107) IOPT
50 CONTINUE
CALL SOLCON (TITLE, NPROB, NLD, NDF, NMAT, NSFR, NCAUS, NALGO, NPP,
        + NYIELD, NT, NL, NSIZ, NORD, NGAUS, MAXMAT, MAXLD, MODIFY, IPRINT)
CALL MATPRP (NMAT, NYIELD, NPP, NL, ORT, MATMAX, MODIFY, IPRINT)
CALL TABBY (NMAT, NL, ORT, TABSTR, TABSTN, MATMAX, MAXTAB,
        + MODIFY, IPRINT)
CALL RUBBLE (NMAT, NL, ORT, AK, MATMAX, NRFF, NME, MINES,
        + MAXINC, MAXELY, MODIFY, IPRINT)
CALL BOUND (NSING, NNBB, MPR, NBC, NFIX, US, ANG, NOFR, DANG,
        + IBFLAG, IBC, BCODE, ICODE, PNR, PTO2, MAXBND, MAXSUM, IPRINT, MODIFY)
DO 52 LNUM=1, NLD
   CALL LOAD1 (LNUM, LDTYPE, TITLE1, NRE, NRG, NRPRS, NRC,
             + ND, FX, FY, MAXPNT, MAXLD, MODIFY, IPRINT)
   CALL LOAD2 (LNUM, NMAT, NL, NRG, THETA, GRAY, ANGVEL, DENS,
MODIFY, IPRINT, MAXLD, MATMAX
CALL LOAD3(LNUM, NRPRS, NSFR, LNE, NOPL, PN, P1,
+ MAXPRS, MAXLD, MODIFY, IPRINT)
CALL LOAD4(LNUM, NINC, FAC, NOUT, NIT, CONFAC,
+ MAXLD, INCMAX, MODIFY, IPRINT)
CALL LOAD5(LNUM, NIMP, IMPNOD, MAXLD, MAXIMP, MODIFY, IPRINT)
CONTINUE
CALL PRTCON(NELP, NODE, NPEL, MAXPRT, MODIFY, IPRINT)
GOTO 105
CONTINUE
CALL SOLCON(TITLE, NPROB, NLD, NDF, NMAT, NSFR, NGAUS, NALGO, NPP,
+ NYIELD, NT, NL, NSIZ, NCORD, MGAUS, MATMAX, MAXLD, MODIFY, IPRINT)
GOTO 105
CONTINUE
CALL MATPRP(NMAT, NYIELD, NPP, NL, ORT, MATMAX, MODIFY, IPRINT)
GOTO 105
CONTINUE
CALL RUBBLE(NMAT, NL, ORT, AK, MATMAX, NRFF, NME, MINES,
+ MAXINC, MAXELY, MODIFY, IPRINT)
GOTO 105
CONTINUE
CALL BOUND(NSING, NNBB, MPR, NBC, NFIX, US, ANG, NCP, DANG,
+ IBFLAG, IBC, ECODE, ICODE, PTOF, PTOZ, MAXBG, MAXSUM, MAXSUM, IPRINT, MODIFY)
GOTO 105
CONTINUE
IF(NLD .EQ. 1) GOTO 78
WRITE(OUT, 76)
FORMAT(/, 'ENTER LOAD CASE TO BE MODIFIED: $')
CALL READIN(LNUM, NOTRD, NH)
IF(NOTRD .EQ. 0) GOTO 75
CONTINUE
IF(IOPT .EQ. 7) GOTO 80
IF(IOPT .EQ. 9) GOTO 85
IF(IOPT .EQ. 10) GOTO 90
IF(IOPT .EQ. 11) GOTO 97
GOTO 105
CONTINUE
CALL LOAD1(LNUM, LDTYPE, TITLE1, NRE, NRQ, NRPRS, NRC,
+ ND, FX, FY, MAXPNT, MAXLD, MODIFY, IPRINT)
GOTO 105
CONTINUE
CALL LOAD2(LNUM, NMAT, NL, NRC, NRQ, THETA, GRAV, ANGVEL, DENS,
+ MODIFY, IPRINT, MAXLD, MATMAX)
GOTO 105
CONTINUE
CALL LOAD3(LNUM, NRPRS, NSFR, LNE, NOPL, PN, PT,
+ MAXPRS, MAXLD, MODIFY, IPRINT)
GOTO 105
95 CONTINUE
CALL LOAD4(LNUM, NINC, FAC, NOUT, NIT, CONFAC,
+ MAXLD, INCMAX, MODIFY, IPRINT)
GOTO 105
97 CONTINUE
CALL LOADS(LNUM, NIMP, IMPNOD, MAXLD, MAXIMP, MODIFY, IPRINT)
GOTO 105
100 CONTINUE
CALL PRTCON(NELP, NONODE, NPML, MAXPRT, MODIFY, IPRINT)
105 CONTINUE
GOTO 30
107 CONTINUE
GOTO 140

****** read data set from a disk file ******
110 CONTINUE
CALL READSKC(UNIT, MAXLD, MATMAX, MAXTAB, MAXINC, MAXELY,
+ MAXPNT, MAXPRS, MAXIMP, INCMAX, MAXPRT, MAXBD, MAXSUM,
+ TITLE, NP0B, NLD, NDF, NMAT, NSFR, NGAUS, NALGO,
+ NPP, NYIELD, NT, NL, NSIZ, NQRD, MGAUS, ORT, NTAB, TABSTR, TABSTN,
+ AK, NRFF, NME, MINES, NSING, NNB, MPR, NBC,
+ NFIX, US, ANG, NOPR, LDTYPE, TITLE1, NRE, NRG, NRPRS, NRC,
+ ND, FX, FY, THEITA, GRAV, ANGVEL, DENS, LNE, NOPL, PN, PT, NINC, FAC,
+ NOUT, NIT, CONFAC, NIMP, IMPNOD, NELP, NONODE, NPML)
GOTO 140

****** write data set to a disk file ******
120 CONTINUE
CALL WRDSK(UNIT, MAXLD, MATMAX, MAXTAB, MAXINC, MAXELY,
+ MAXPNT, MAXPRS, MAXIMP, INCMAX, MAXPRT, MAXBD, MAXSUM, IWROTE,
+ TITLE, NP0B, NLD, NDF, NMAT, NSFR, NGAUS, NALGO,
+ NPP, NYIELD, NT, NL, NSIZ, NQRD, MGAUS, ORT, NTAB, TABSTR, TABSTN,
+ AK, NRFF, NME, MINES, NSING, NNB, MPR, NBC,
+ NFIX, US, ANG, NOPR, LDTYPE, TITLE1, NRE, NRG, NRPRS, NRC,
+ ND, FX, FY, THEITA, GRAV, ANGVEL, DENS, LNE, NOPL, PN, PT, NINC, FAC,
+ NOUT, NIT, CONFAC, NIMP, IMPNOD, NELP, NONODE, NPML)
GOTO 140

****** print hard copy of input data on line printer ******
130 CONTINUE
IPRINT=1
OPEN(UNIT=LP1, ACCESS='SEQUENTIAL', STATUS='NEW',
+ FILE='PRESCRBS.LIS')
CALL S0LCON(TITLE, NP0B, NLD, NDF, NMAT, NSFR, NGAUS, NALGO, NPP,
+ NYIELD, NT, NL, NSIZ, NQRD, MGAUS, MATMAX, MAXLD, MODIFY, IPRINT)
CALL MATTAB(NMAT, NYIELD, NPP, NL, ORT, MATMAX, MODIFY, IPRINT)
CALL TABBY(NMAT, NL, ORT, NTAB, TABSTR, TABSTN, MATMAX, MATTAB,
+ MODIFY, IPRINT)
CALL RUBBLE(NMAT, NL, ORT, AK, MATMAX, NRFF, NME, MINES,
c
+ MAXINC, MAXELY, MODIFY, IPRINT)
+ CALL BOUNN<NSING. NNBB.HPR, NBC,FIX, US. ANG, NO<PR, DANG,
+ IFLAG. IBC, BCODE. ICODE. PNOR. PTOZ. MAXEND. MAXSUM, IPRINT. MODIFY)
+ DO 135 LNUM=1, NLD
+ CALL LOAD1(LNUM. LDTYP. TITLE, NRE. NRG. NFRS. NRC,
+ ND, FX, FY, MAXPNT. MAX<. MODIFY, IPRINT)
+ CALL LOAD2(LNUM. NMAT. NL. NRC. NRG. THETA. GRAV. ANGVEL. DENS,
+ MODIFY. IPRINT. MAXLD. MATMAX)
+ CALL LOAD3(LNUM. NFRS. NSFR. LNE. NOPL. PN, PT,
+ MAXFRS. MAXLD. MODIFY. IPRINT)
+ CALL LOAD4(LNUM. NINC. FAC. NGUT. NIT. CONFAC,
+ MAXLD. INCMAX. MODIFY. IPRINT)
+ CALL LOAD5(LNUM. NIMP. IMPNOD. MAXLD. MAXIM. MODIFY. IPRINT)
+ CONTINUE
+ CALL PRTC<NI(NELP. NONODE. NP<. MAXPRT. MODIFY. IPRINT)
+ CLOSE(UNIT=LPT. DISPOSE='PRINT/DL¥E')
GOTO 140

c
+**** return to main menu *******

C
+ 140 CONTINUE
+ GOTO 5

C
+ end program execution ******

C
+ 150 CONTINUE
+ IF (IWROTE. NE. 1) GOTO 165
+ WRITE(OUT. 175)
+ FORMAT(/,' ********** DATA FILE SCRUBS. DAT COMPLETED ********'//
+ + 10X. '********* ADIOS HERMANO ********* ')
+ GOTO 180

C
+ 165 WRITE(OUT. 160)
+ 160 FORMAT(/,' S. DO YOU WANT TO SAVE THE DATA? (Y or N) * *')
+ ACCEPT 170. ANS

C
+ 170 FORMAT(A1)
+ WRITE(OUT. 173)
+ 173 FORMAT(/,' ********* ADIOS HERMANO ********* ')
+ IF (ANS. NE. 'Y') GOTO 180
+ CALL WRO<TS(NUNIT. MAXLD. MATMAX. MAXTAB. MAXINC. MAXELY,
+ MAXPNT. MAXFRS. MAXIMP. INCMAX. MAXPRT. MAXEND. MAXSUM. IWROTE,
+ TITLE. NPROB. NLD. NDF. NMAT. NSFR. NSAUS. NALGO,
+ NPP. NYIELD. NT. NL. NSIZ. NORD. MGAUS. ORT. NTAB. TABSTR. TABSTN,
+ AK. NRAF. MME. MINES. NSING. NNBB. MPR. NBC,
+ NFIX. US. ANG. NOPR. LDTYP. TITLE1. NRE. NRG. NFRS. NRC,
+ ND. FX. FY. THETA. GRAV. ANGVEL. DENS. LNE. NOPL. PN. PT. NINC. FAC,
+ NGUT. NIT. CONFAC. NIMP. IMPNOD. MAXLD. MAXIM. MODIFY. IPRINT)
+ CONTINUE
+ GOTO 180
+ END

C
+ SUBROUTINE SOLCON: ENTERS TITLE AND SOLUTION CONTROLS

C
+ SUBROUTINE SOLCON(TITLE. NPROB. NLD. NDF. NMAT. NSFR. NSAUS. NALGO,
+ NPP. NYIELD. NT. NL. NSIZ. NORD. MGAUS. MATMAX. MAXLD. MODIFY. IPRINT)
+ DIMENSION TITLE(20)
+ COMMON IN. OUT. LPT
+ INTEGER OUT
CHARACTER*2 HLPFIL
CHARACTER*2 HLPMOD
CHARACTER*2 NOHELP
HLPFIL='SI'
NOHELP='NH'
HLPMOD='MD'

NDEV=OUT
IF(IPRINT .NE. 1) GOTO 2
NDEV=LPT
GOTO 385
2 CONTINUE
IF(MODIFY.EQ.1) GOTO 385
3 CONTINUE
MFLAG=0

C****** enter solution controls ********

10 WRITE(OUT,20)
20 FORMAT(/,' ENTER TITLE OF PROBLEM')
ACCEPT 25.(TITLE(I),I=1,20)
25 FORMAT(20A4)
   IF(TITLE(I).EQ.'H') CALL HELP(HLPFIL)
   IF(TITLE(I).EQ.'H') GOTO 10
   IF(MFLAG.EQ.1) GOTO 385
30 WRITE(OUT,40)
40 FORMAT(/,' ENTER NUMBER OF PROBLEMS TO BE SOLVED')
   CALL READIN(NPROB,NOTRD,NOHELP)
   IF(NOTRD.EQ.0) GOTO 30
   IF(MFLAG.EQ.1) GOTO 385

70 WRITE(OUT,80)
80 FORMAT(/,' ENTER NUMBER OF LOAD CASES PER PROBLEM')
   CALL READIN(NLD,NOTRD,NOHELP)
   IF(NOTRD.EQ.0) GOTO 70
   IF(NLD.LE.MAXLD) GOTO 87
   WRITE(OUT,85) MAXLD
85 FORMAT(/,' MAXIMUM NUMBER OF LOAD CASES PER PROBLEM EXCEEDED',/,'CURRENT MAXIMUM=',I15,///)
   GOTO 70
87 CONTINUE
   IF(MFLAG.EQ.1) GOTO 385
90 WRITE(OUT,100)
100 FORMAT(/,' ENTER NUMBER OF DOF PER NODE')
   CALL READIN(NDF,NOTRD,NOHELP)
   IF(NOTRD.EQ.0) GOTO 90
   IF(MFLAG.EQ.1) GOTO 385
110 WRITE(OUT,120)
120 FORMAT(/,' ENTER NUMBER OF DIFFERENT MATERIALS')
   CALL READIN(NMAT,NOTRD,NOHELP)
   IF(NOTRD.EQ.0) GOTO 110

C****** test max number of materials, (NMAT + NL) le. MATMAX ********

   IF(NMAT.LE.MATMAX) GOTO 127
   WRITE(OUT,125) MATMAX
125 FORMAT(/,' EXCESIVE NUMBER OF MATERIALS RE-ENTER ')
ALL DATA *******/.

CONTINUE

GOTO 110

127 WRITE(OUT,150)
150 FORMAT(/,'<ENTER ELEMENT TYPE NUMBER: 1=LINEAR, 2=QUADRATIC, ' + '3=CUBIC> $')
 CALL READIN(NSFR,NOTRD,NOHELP)
 IF(NOTRD.EQ.0) GOTO 130
 IF(MFLAG.EQ.1) GOTO 385

160 WRITE(OUT,170)
170 FORMAT(/,'<ENTER GAUSSIAN QUADRATURE (NO. OF GAUSS POINTS)> $')
 CALL READIN(NGAUS,NOTRD,NOHELP)
 IF(NOTRD.EQ.0) GOTO 160
 IF(MFLAG.EQ.1) GOTO 385

180 WRITE(OUT,190)
190 FORMAT(/,'<ENTER SOLUTION ALGORITHM NUMBER> $')
+ '3X,'0=ELASTICITY ONLY'
+ '3X,'1=CONSTANT STIFFNESS'
+ '3X,'2=TWO STEP PROCESS'
+ '3X,'3=TANGENT STIFFNESS $')
 CALL READIN(NALGO,NOTRD,NOHELP)
 IF(NOTRD.EQ.0) GOTO 180
 IF(MFLAG.EQ.1) GOTO 385

200 WRITE(OUT,210)
210 FORMAT(/,'<ENTER STRESS/STRAIN TYPE NUMBER> $')
+ '3X,'0=PLANE STRAIN'
+ '3X,'1=PLANE STRESS'
+ '3X,'2=AXISYMMETRIC PROBLEM $')
 CALL READIN(NPP,NOTRD,NOHELP)
 IF(NOTRD.EQ.0) GOTO 200
 IF(MFLAG.EQ.1) GOTO 385

220 WRITE(OUT,230)
230 FORMAT(/,'<ENTER YIELD CONDITION PARAMETER NUMBER> $')
+ '3X,'1=MISES'
+ '3X,'2=TRESCA'
+ '3X,'3=DRUCKER-PRAGER $')
 CALL READIN(NYIELD,NOTRD,NOHELP)
 IF(NOTRD.EQ.0) GOTO 220
 IF(MFLAG.EQ.1) GOTO 385

250 WRITE(OUT,260)
260 FORMAT(/,'<ENTER THE STIFFNESS CONTROL NUMBER > $')
+ '3X,'0=NO ELASTICALLY COUPLED NODES'
+ '3X,'1=INPUT STIFFNESS COEFFICIENTS $')
 CALL READIN(NT,NOTRD,NOHELP)
 IF(NOTRD.EQ.0) GOTO 250
 IF(MFLAG.EQ.1) GOTO 385

270 WRITE(OUT,290)
290 FORMAT(/,'<ENTER RUBBLE FLAG NUMBER, 0=NO RUBBLE, 1=RUBBLE> $')
 CALL READIN(NL,NOTRD,NOHELP)
 IF(NL.EQ.0) GOTO 290
IF(NMAT.EQ.1) GOTO 110
NTST=NMAT+1
IF(NTST.GT.MATHAX) GOTO 110
CONTINUE
IF(NOTRD.EQ.0) GOTO 270
IF(MFLAG.EQ.1) GOTO 385


**** omit bandwidth if stiffness coefficients are input, NT=1 ****

NSIZ=0
IF (NT.NE.1) GOTO 340
WRITE(OUT,330)
330 FORMAT(/, '<ENTER BANDWIDTH>' ','$)
CALL READIN(NSIZ,NOTRD,NOHELP)
IF(NOTRD.EQ.0) GOTO 320
IF(MFLAG.EQ.1) GOTO 385
CONTINUE
NCORD=2
WRITE(OUT,360)
360 FORMAT(/, '<ENTER NUMBER OF COORDINATES PER NODE>'  ', $')
CALL READIN(NCORD,NOTRD,NOHELP)
IF(NOTRD.EQ.0) GOTO 350
IF(MFLAG.EQ.1) GOTO 385
WRITE(OUT,380)
380 FORMAT(/, '<ENTER NUMBER OF GAUSS POINTS>'  '$')
CALL READIN(MGAUS,NOTRD,NOHELP)
IF(NOTRD.EQ.0) GOTO 370
IF(MFLAG.EQ.1) GOTO 385


**** view solution controls ****

CONTINUE
WRITE(NDEV,387)
387 FORMAT(/, 20(''), 'SOLUTION CONTROLS ', 20(''), '/)
WRITE(NDEV,395) (TITLE(I),I=1,20)
395 FORMAT(/, 1X, 20A4) /
WRITE(NDEV,400) NPGB, NLD, NDF, NMAT, NSFR, NGAUS, NALGO, NPP, + NYIELD, NT, NL, NSIZ, NCORD, MGAUS
400 FORMAT(15X, 'CONTROL PARAMETERS'/
+ 1. TITLE/
+ 2. NUMBER OF PROBLEMS-----------------', I5/
+ 3. NUMBER OF LOAD CASES-------------', I5/
+ 4. NUMBER OF DEGREES OF FREEDOM---', I5/
+ 5. NUMBER OF MATERIALS-------------', I5/
+ 6. ELEMENT TYPE---------------', I5/
+ 7. NUMBER OF GAUSS POINTS--------', I5/
+ 8. SOLUTION ALGORITHM CONTROL-----', I5/
+ 9. STR/STN CONTROL----------------', I5/
+ 10. YIELD CONDITION CONTROL-------', I5/
+ 11. ARBITRARY STIFF. CONTROL------', I5/
+ 12. RUBBLE FLAG CONTROL---------', I5/
+ 13. BAND WIDTH (NOT NEEDED)-------', I5/
+ 14. NUMBER OF COORDINATES--------', I5/
+ 15. GAUSS PTS FOR. RES. CALC.------', I5/
+ 16. RE-ENTER DATA SET'/
+ 17. END MODIFICATIONS')
IF(IPRINT.EQ.1) GOTO 440
IF(MFLAG.EQ.1) GOTO 425

405 WRITE(OUT,410)
410 FORMAT(/,10X,'DO YOU WISH TO MAKE ANY CHANGES? (Y OR N) ',*)
ACCEPT 420, ANS

420 FORMAT(A1)
IF(ANS.EQ.'H') CALL HELP(HLPMOD)
IF(ANS.EQ.'Y') GOTO 405
IF(ANS.NE.'Y') GOTO 440
MFLAG=1

c***** modify controls *****
c
425 CONTINUE
NUM=17
CALL CHOOSE(IDOPT,NUM)
GOTO(10,30,70,90,110,130,160,180,200,220,250,270,320,350,370,3440),IDOPT
440 CONTINUE
RETURN
END

C**********************************************************************
C* SUBROUTINE MATPRP: ENTERS MATERIAL PROPERTIES
C**********************************************************************

SUBROUTINE MATPRP(NMAT,NYIELD,NPP,NL,ORT,MATMAX,MODIFY,IPRINT)
DIMENSION ORTCMATMAX,B>
COMMON IN.QUT.LPT
INTEGER OUT
CHARACTER*2 HLPFIL
CHARACTER*2 NOHELP
CHARACTER*2 HLPMOD
HLPMOD='MD'
NOHELP='NH'
HLPFIL='M1'
NDEV=OUT
IF(IPRINT.NE.1) GOTO 2
NDEV=LPT
GOTO 300
2 CONTINUE
IF(MODIFY.EQ.1) GOTO 300
3 CONTINUE
MFLAG=0
WRITE(OUT,4)
4 FORMAT(/,20('<<'),' MATERIAL PROPERTIES ',20(''))
c***** modify do-loop parameter if rubble considered *****
c
IF(NL.NE.1) GOTO 5
NMAT=NMAT-1
GOTO 7
5 CONTINUE
NMAT=NMAT
7 CONTINUE
cin***** enter material properties *****

DO 150 N=1, NMAT
   WRITE(OUT,20) N
   10 FORMAT(//,'<ENTER YOUNGS MODULUS OF ELASTICITY FOR'
      + ' MATERIAL', I2, '>', '*')
      CALL READRL(ORT(N,1), NOTRD, HLPFIL)
      IF(NOTRD.EQ.0) GOTO 10
   20 WRITE(OUT,40) N
   30 FORMAT(//,'<ENTER POISSONS RATIO OF MATERIAL', I2, '>', '*')
      CALL READRL(ORT(N,2), NOTRD, HLPFIL)
      IF(NOTRD.EQ.0) GOTO 30
   40 WRITE(OUT,60) N
   50 FORMAT(//,'<ENTER YIELD STRESS OF MATERIAL', I2, '>', '*')
      CALL READRL(ORT(N,3), NOTRD, HLPFIL)
      IF(NOTRD.EQ.0) GOTO 50
   60 WRITE(OUT,80) N
   70 FORMAT(//,'<ENTER HARDENING MODULUS OF MATERIAL', I2, '>', '*')
      CALL READRL(ORT(N,4), NOTRD, HLPFIL)
      IF(NOTRD.EQ.0) GOTO 70
   80 CONTINUE
   90 IF(NYIELD.EQ.3) GOTO 120
   100 FORMAT(//,'<ENTER CONICAL YIELD SURFACE ANGLE OF MATERIAL',
      + ' I2, '>', '*')
      CALL READRL(ORT(N,5), NOTRD, HLPFIL)
      IF(NOTRD.EQ.0) GOTO 95
   110 CONTINUE
   120 IF(NP=1) GOTO 140
   130 IF(NP=1) GOTO 125
   140 CONTINUE
   150 CONTINUE

cin***** if NL=1 (rubbleized condition) enter rubbleized data *****

IF(NL.NE.1) GOTO 300
N=NMAT
   WRITE(OUT,170)
   170 FORMAT(//,'<ENTER YOUNGS MODULUS OF ELASTICITY OF RUBBLEIZED'
      + ' MATERIAL>', '*')
      CALL READRL(ORT(N,1), NOTRD, HLPFIL)
      IF(NOTRD.EQ.0) GOTO 170
   180 WRITE(OUT,190)
   190 FORMAT(//,'<ENTER POISSONS RATIO OF RUBBLEIZED MATERIAL>', '*')
      CALL READRL(ORT(N,2), NOTRD, HLPFIL)
      IF(NOTRD.EQ.0) GOTO 180
   200 WRITE(OUT,210)
   210 FORMAT(//,'<ENTER YIELD STRESS OF RUBBLEIZED MATERIAL>', '*')
CALL READRL(ORT(N, 3), NOTRD, HLFPIL)
IF(NOTRD. EQ. 0) GOTO 200
220 WRITE(OUT, 230)
230 FORMAT(/, ' <ENTER HARDENING MODULUS OF RUBBLEIZED MATERIAL> 
+ *)
CALL READRL(ORT(N, 4), NOTRD, HLFPIL)
IF(NOTRD. EQ. 0) GOTO 220
ORT(N, 5) = 0.0
IF(NYIELD NE. 3) GOTO 260
240 WRITE(OUT, 250)
250 FORMAT(/, ' <ENTER CONICAL YIELD SURFACE ANGLE OF RUBBLEIZED 
+ MATERIAL> *)
CALL READRL(ORT(N, 5), NOTRD, HLFPIL)
IF(NOTRD. EQ. 0) GOTO 240
260 CONTINUE
ORT(N, 6) = 0.0
IF(NPP. NE. 1) GOTO 300
270 WRITE(OUT, 280)
280 FORMAT(/, ' <ENTER THICKNESS OF RUBBLEIZED MATERIAL> 
+ *)
CALL READRL(ORT(N, 6), NOTRD, HLFPIL)
IF(NOTRD. EQ. 0) GOTO 270

C***** view material properties *****
C
300 CONTINUE
WRITE(NDEV, 310)
310 FORMAT(/, /, 20('**'), ' MATERIAL PROPERTIES ', 20('**'))
WRITE(NDEV, 320)
320 FORMAT(/, /, 1X, 'MAT', 4X, 'YOUNGS', 4X, 'POISSONS', 6X, 'YIELD', 6X, 
+ 'HARDENING', 5X, 'YIELD', 5X, 'MATERIAL', '/2X, #', 5X, 'MODULUS', 5X, 
+ 'RATIO', 7X, 'STRESS', 6X, 'MODULUS', 6X, 'ANGLE', 5X, 'THICKNESS', /)
DO 340 N = 1, NMAT
WRITE(NDEV, 330), N, ORT(N, 1), ORT(N, 2), ORT(N, 3), ORT(N, 4), ORT(N, 5), 
+ ORT(N, 6)
330 FORMAT(/, 1X, I2, 4X, G10.4, 2X, G10.4, 2X, G10.4, 2X, G10.5, 2X))
340 CONTINUE
IF(IPRINT EQ. 1) GOTO 340
IF(MFLAG. EQ. 1) GOTO 390
345 WRITE(OUT, 350)
350 FORMAT(/, 10X, ' DO YOU WISH TO MAKE ANY CHANGES? (Y or N) 
+ *)
ACCEPT 360, ANS
360 FORMAT(A1)
IF(ANS. EQ. 'H') CALL HELP(HLPMOD)
IF(ANS. EQ. 'H') GOTO 345
IF(ANS. NE. 'Y') GOTO 540
MFLAG = 1

C***** modify material properties *****
C
390 CONTINUE
WRITE(OUT, 400)
400 FORMAT(/, '/ SEFECT OPTION TO BE MODIFIED'> '/ 
+ 1. YOUNGS MODULUS '/ 
+ 2. POISSONS RATIO '/
NUM=8
CALL CHOOSE(IOPT, NUM)

IF(IOPT.EQ.7) GOTO 3
IF(IOPT.EQ.8) GOTO 540

405 WRITE (OUT, 407)
407 FORMAT('ENTER MATERIAL NUMBER TO BE MODIFIED: ',$)
CALL READIN(N, NOTRD.HLPFIL)
IF(NOTRD.EQ.0) GOTO 405

c****** branch to option ******
c
GOTO(410, 430, 450, 470, 490, 510) IOPT

410 WRITE(OUT, 420) N
420 FORMAT('RE-ENTER YOUNG'S MODULUS OF ELASTICITY FOR','MATERIAL', I2, ' $')
CALL READRLCORT(N, 1), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 410
GOTO 530

430 WRITE(OUT, 440) N
440 FORMAT('RE-ENTER POISSON'S RATIO OF MATERIAL', I2, ' $')
CALL READRLCORT(N, 2), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 430
GOTO 530

450 WRITE(OUT, 460) N
460 FORMAT('RE-ENTER YIELD STRESS OF MATERIAL', I2, ' $')
CALL READRLCORT(N, 3), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 450
GOTO 530

470 WRITE(OUT, 480) N
480 FORMAT('RE-ENTER HARDENING MODULUS OF MATERIAL', I2, ' $')
CALL READRLCORT(N, 4), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 470
GOTO 530

490 WRITE(OUT, 500) N
500 FORMAT('RE-ENTER CONICAL YIELD SURFACE ANGLE OF MATERIAL', I2, ' $')
CALL READRLCORT(N, 5), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 490
GOTO 530

510 WRITE(OUT, 520) N
520 FORMAT('RE-ENTER THICKNESS OF MATERIAL', I2, ' $')
CALL READRLCORT(N, 6), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 510
GOTO 300

CONTINUE
SUBROUTINE TABBY(NMAT, NL, ORT, NTAB, TABSTR, TABSTN, MATMAX, MAXTAB, MODIFY, IPRINT)
DIMENSION ORT(MATMAX, 8)
DIMENSION NTAB(MATMAX), TABSTR(MATMAX, MAXTAB), TABSTN(MATMAX, MAXTAB)
COMMON IN, OUT, LPT
INTEGER OUT
CHARACTER*2 HLPFIL
CHARACTER*2 HLPMOD
HLPFIL='TI'
HLPMOD='MB'
NDEV=OUT
IF (IPRINT .NE. 1) GOTO 1
NDEV=LPT
GOTO 180
1 CONTINUE
IF (MODIFY .EQ. 1) GOTO 180
C****** test hardening modulus if negative *******
DO 2 N=1, NMAT
IF (ORT(N, 4) .LT. 0) NOPTS=1
2 CONTINUE
IF (NOPTS .EQ. 0) GOTO 370
3 CONTINUE
MFLAG=0
WRITE(OUT, 4)
4 FORMAT (•/•/••/, 'TABULATED STRESS/STRAIN DATA', •/•/••/)
C****** modify do-loops parameters if NL=1 ******
IF (NL .NE. 1) GOTO 5
NMAT1=NMAT-1
GOTO 7
5 CONTINUE
NMAT1=NMAT
7 CONTINUE
C****** enter tab str/stn points if hardening mod is zero ******
DO 80 N=1, NMAT1
IF (ORT(N, 4). GE. 0) GOTO 70
80 CONTINUE
WRITE(OUT, 20) N
20 FORMAT (•/•/••/, 'ENTER NUMBER OF TABULATED STRAIN POINTS FOR', •/•/••/)
+ ' MATERIAL', •/•/••/)
CALL READIN(NTAB(N), NOTRD, HLPFIL)
IF (NOTRD .EQ. 0) GOTO 10
IF(NTAB(N) .LE. MAXTAB) GOTO 25
WRITE(OUT,22)
22 FORMAT(//,'****** MAXIMUM NUMBER OF POINTS EXCEEDED *****'//)
GOTO 10
25 CONTINUE
DO 70 J=1,NTAB(N)
30 WRITE(OUT,40) J
40 FORMAT(/,'<ENTER STRESS VALUE FOR POINT',I3,'>'($) )
CALL READLR(TABSTR(N,J),NOTRD,HLPFLI)
IF(NOTRD.EQ.0) GOTO 30
55 WRITE(OUT,60) J
60 FORMAT(/,'<ENTER VALUE OF STRAIN FOR POINT',I3,'>'($) )
CALL READLR(TABSTN(N,J),NOTRO,HLPFIL)
IF(NOTRD.EQ.0) GOTO 55
70 CONTINUE
80 CONTINUE
C******** if hardening mod. of rubbl. mat. is neg. enter str/stn points ****
C
85 CONTINUE
IF(NL.NE.1) GOTO 180
N=NMAT
IF(ORT(N,4).GE.0) GOTO 180
90 WRITE(OUT,100)
100 FORMAT(/, '<ENTER NUMBER OF TABULATED STRAIN POINTS FOR'
+ ' RUBBLEIZED MATERIAL>' '('$) )
CALL READIN(NTAB(N),NOTRD,HLPFIL)
IF(NOTRD.EQ.0) GOTO 90
IF(NTAB(N) .LE. MAXTAB) GOTO 105
WRITE(OUT,102)
102 FORMAT(//,'****** MAXIMUM NUMBER OF POINTS EXCEEDED *****'//)
GOTO 90
105 CONTINUE
DO 170 JR=1,NTAB(N)
110 WRITE(OUT,120) JR
120 FORMAT(/,'<ENTER STRESS VALUE FOR POINT',I3,'>'($) )
CALL READLR(TABSTR(N,JR),NOTRD,HLPFIL)
IF(NOTRD.EQ.0) GOTO 110
140 WRITE(OUT,150) JR
150 FORMAT(/,'<ENTER VALUE OF STRAIN FOR POINT',I3,'>'($) )
CALL READLR(TABSTN(N,JR),NOTRD,HLPFIL)
IF(NOTRD.EQ.0) GOTO 140
170 CONTINUE
C******** view tab str/stn data *****
C
190 CONTINUE
WRITE(NDEV,200)
200 FORMAT(///.2O('**'), 'TABULATED STRESS/STRAIN POINTS',
+ ' **'))
DO 240 N=1,NMAT
240 IF(ORT(N,4).GE.0) GOTO 240
WRITE(NDEV,210) N
```plaintext
210 FORMAT(/,7X,'MATERIAL=',I2,/,2X,'POINT',4X,'STRESS',5X,'STRAIN')
     DO 230 I=1,NTAB(N)
        WRITE(NDEV,220) I,TABSTR(N,I),TABSTN(N,I)
220    FORMAT(2X,I2,5X,G10.5,3X,G10.5)
     CONTINUE
230 CONTINUE
     IF(IPRINT.EQ.1) GOTO 370
     IF(MFLAG.EQ.1) GOTO 265
240 CONTINUE
245 WRITE(OUT,250)
250 FORMAT(/,10X,'DO YOU WISH TO MAKE ANY CHANGES? (Y or N)?')
     ACCEPT 260,ANS
     IF(ANS.EQ.'H') CALL HELP(HLPMOD)
     IF(ANS.EQ.'H') GOTO 245
     IF(ANS.NE.'Y') GOTO 370
     MFLAG=1
     c***** modify data *****
     265 CONTINUE
     WRITE(OUT,270)
270 FORMAT(/,'SELECT OPTION TO BE MODIFIED'/
     + 1. TABULATED STRESS-STRAIN POINTS'/
     + 2. PESHOW DATA'/
     + 3. RE-ENTER DATA SET'/
     + 4. END MODIFICATIONS'/)
     NUM=4
     CALL CHOOSE(IOPT,NUM)
     GOTO(280,180,3370,370)IOPT
280 WRITE(OUT,290)
290 FORMAT(/,'ENTER MATERIAL NUMBER TO BE MODIFIED$')
     CALL READIN(N,NOTRD,HLPFIL)
     IF(NOTRD.EQ.0) GOTO 280
     WRITE(OUT,320)N
310 WRITE(OUT,320)N
320 FORMAT(/,'RE-ENTER NUMBER OF TABULATED STRAIN POINTS FOR'
     + 'MATERIAL',I2,'>'$)
     CALL READIN(NTAB(N),NOTRD,HLPFIL)
     IF(NOTRD.EQ.0) GOTO 310
     IF(NTAB(N).LE.MAXTAB) GOTO 330
     WRITE(OUT,322)
322 FORMAT(/,**** MAXIMUM NUMBER OF POINTS EXCEEDED *****/)
     GOTO 310
330 CONTINUE
     DO 365 J=1,NTAB(N)
340     FORMAT(/,'RE-ENTER STRESS VALUE FOR POINT',I3,'$')
     CALL READRL(TABSTR(N,J),NOTRD,HLPFIL)
     IF(NOTRD.EQ.0) GOTO 330
350     WRITE(OUT,360)J
360     FORMAT(/,'RE-ENTER VALUE OF STRAIN FOR POINT',I3,'$')
     CALL READRL(TABSTN(N,J),NOTRD,HLPFIL)
```
SUBROUTINE RUBBLE: INPUTS RUBBLEIZATION DATA

SUBROUTINE RUBBLE(NMAT, NL, ORT, AK, IMAX, NRFF, NME, MINES,
    + MAXINC, MAXELY, MODIFY, IPRINT)
    DIMENSION ORT(MATMAX, 8)
    DIMENSION NME(MAXINC), MINES(MAXINC, MAXELY)
    COMMON IN, OUT, LPT
    INTEGER OUT
    CHARACTER*2 HLPFIL
    CHARACTER*2 HLPMOD
    HLPFIL = 'R1'
    HLPMOD = 'MD'

    IF(NL . NE. 1) GOTO 600
    NDEV = OUT
    IF(IPRINT . NE. 1) GOTO 2
    NDEV = LPT
    GOTO 165
    CONTINUE
    IF(MODIFY . EQ. 1) GOTO 165
    CONTINUE
    MFLAG = 0

    WRITE(OUT, 3)
    5 FORMAT(///, 20(' ', ' RUBBLEIZATION DATA ', 20(' '))
    DO 50 N = 1, NMAT
    10 WRITE(OUT, 20) N
    20 FORMAT(///, ' ENTER BULKING PARAMETER OF MATERIAL ', I2, ' ')
    CALL READRL(ORT(N, 8), NOTRD, HLPFIL)
    IF(NOTRD.EQ.0) GOTO 10
    30 WRITE(OUT, 40) N
    40 FORMAT(///, ' ENTER FAILURE STRESS OF MATERIAL ', I2, ' ')
    CALL READRL(ORT(N, 7), NOTRD, HLPFIL)
    IF(NOTRD.EQ.0) GOTO 30
    50 CONTINUE
    WRITE(OUT, 70)
    70 FORMAT(///, ' ENTER MINIMUM Z COORDINATE, OF WHICH ELEMENTS MAY ')
    + ' NOT PASS: ')
    READ(1, 80, ERR=60) AK
    80 FORMAT(E10.4)
CONTINUE  WRITE(NDEV, 170)
170 FORMAT(//, 'RUBBLEIZATION DATA', 'MATERIAL', 'FAILURE STRESS', 'BULKING PARAMETER'/)
    DO 190 N=1, NMAT
          WRITE(NDEV, 180) N, ORT(N, 7), ORT(N, 8)
180 FORMAT(5X, I2, 11X, G10.4, 11X, G10.5)
190 CONTINUE
WRITE(NDEV, 200) AK
200 FORMAT(1X, 'MINIMUM Z COORDINATE=', G10.4)

IF (PRINT.EQ.1) GOTO 590
IF (MFLAG.EQ.1) GOTO 300
WRITE(OUT, 280)
280 FORMAT(10X, 'DO YOU WISH TO MODIFY DATA? (Y or N) ', *)
      ACCEPT 290, ANS
290 FORMAT(A1)
      IF (ANS.EQ. 'H') CALL HELP(HLPMOD)
      IF (ANS.EQ. 'Y') GOTO 270
      IF (ANS.EQ. 'H') GOTO 590
      MFLAG=1
300 WRITE(OUT, 310)
310 FORMAT(//, '<SELECT OPTION TO BE MODIFIED>:'
               + 1. BULKING PARAMETER /
               + 2. FAILURE STRESS /
               - 3. MIN. Z COORDINATE /
               - 4. RESHOW DATA /
               - 5. REENTER DATA SET /
               - 6. END MODIFICATIONS ')
      NUM=6
      CALL CHOOSE(IOPT, NUM)

GOTO (320, 360, 400, 165, 3, 590) IOPT
320 WRITE(OUT, 330)
330 FORMAT(//, '<ENTER MATERIAL NUMBER TO BE MODIFIED> ', *)
      CALL READIN(N, NOTRD, HLPFIL)
      IF (NOTRD.EQ.0) GOTO 320
340 WRITE(OUT, 350) N
350 FORMAT(//, '<REENTER BULKING PARAMETER OF MATERIAL ', I2, ' > ', *)
      CALL READRL(ORT(N, 8), NOTRD, HLPFIL)
      IF (NOTRD.EQ.0) GOTO 340
      GOTO 580
360 WRITE(OUT, 370)
370 FORMAT(//, '<ENTER MATERIAL NUMBER TO BE MODIFIED> ', *)
      CALL READIN(N, NOTRD, HLPFIL)
      IF (NOTRD.EQ.0) GOTO 360
WRITE(OUT, 390) N
FORMAT(//, 'RE-ENTER FAILURE STRESS OF MATERIAL ', I2, ' $)
CALL READRL(GRT(N, 7), NOTRD, HLPFIL)
IF(NOTRD. EQ. 0) GOTO 380
GOTO 580

WRITE(OUT, 410)
FORMAT(//, 'RE-ENTER MINIMUM Z COORDINATE, OF WHICH ELEMENTS',
'MAY NOT PASS (E FORMAT) $'
READ(5, 420, ERR=400) AK

FORMAT(10.4)
CONTINUE
GOTO 165

CONTINUE
CALL MINE(NMAT, NRFF, NME, MINES, MAXINC, MAXELY, MODIFY, IPRINT)

CONTINUE
RETURN
END

SUBROUTINE MINE: INPUTS ELEMENTAL MINING DATA

SUBROUTINE MINE(NMAT, NRFF, NME, MINES, MAXINC, MAXELY, MODIFY, IPRINT)
DIMENSION NMECMAXINC, MINESCMAXINC, MAXELY)
COMMON/INOUT,LPT
INTEGER OUT
CHARACTER*2 HLPFIL
CHARACTER*2 HLPMOD
HLPFIL= 'HI
HLPMOD= ' MI'
NDEV= OUT
IF(IPRINT NE. 1) GOTO 2
NDEV= LPT
GOTO 165
2 CONTINUE
IF(MODIFY . EQ. 1) GOTO 165
3 CONTINUE
MFLAG= 0

**** enter elemental mining data ****
WRITE(OUT, 5)
FORMAT(//, 20('('), ' ELEMENTAL MINING DATA ', 20('') $)

WRITE(OUT, 90)
90 FORMAT(//, ' ENTER TOTAL NUMBER OF INCREMENTS TO BE MINED ', ' $)
CALL READIN(NRFF, NOTRD, HLPFIL)
IF(NOTRD. EQ. 0) GOTO 80

IF(NRFF. LE. MAXELY) GOTO 110

WRITE(OUT,100) MAXELY
FORMAT(//, ' MAXIMUM NUMBER OF INCREMENTS EXCEEDED TRY',
AGAIN **** ', /, 10X, ' CURRENT MAXIMUM= ', IS, //)
GOTO 80
110 CONTINUE

if no elements are removed, omit rest of data *****

IF(NRFF.LE.0) GOTO 590
DO 135 I=1,NRFF
115 WRITE(OUT,120) I
120 FORMAT(‘/,’‘ENTER NUMBER OF ELEMENTS TO BE MINED ON’
+’INCREMENT’,I3,’>’,$)
CALL READIN(NME(I),NOTRD,HLPFIL)
IF(NOTRD.EQ.0) GOTO 115
IF(NME(I).LE.MAXINC) GOTO 135
WRITE(OUT,130) MAXINC
130 FORMAT(‘/,’‘MAXIMUM NUMBER OF ELEMENTS REMOVED EXCEEDED’
+‘***’,’/10X,’CURRENT MAXIMUM=’,I5,’/)
GOTO 115
135 CONTINUE

DO 160 I=1,NRFF
140 WRITE(OUT,150) NME(I),I
150 FORMAT(‘/,’‘ENTER’,I4,’ELEMENT NUMBERS TO BE MINED ON’
+’INCREMENT’,I3,’>’,$)
READ(IN,*,'ERR=155),(MINES(I,J),J=1,NME(I))
GOTO 160
155 CONTINUE
WRITE(OUT,157)
157 FORMAT(‘/,’‘RE-ENTER ALL ELEMENTS AT ONE TIME’
+’(e.g. 1,15,23,...)’/$)
GOTO 140
160 CONTINUE

view data *****

CONTINUE
WRITE(NDEV,170)
170 FORMAT(‘//,’20(‘*’),’ELEMENTAL MINING DATA’,20(‘*’),’//)
WRITE(NDEV,210) NRFF
210 FORMAT(‘/,’TOTAL NUMBER OF MINED INCREMENTS=’,I3)
215 CONTINUE

IF(NRFF.LE.0) GOTO 270
WRITE(NDEV,220)
220 FORMAT(‘/,’1X,20(‘*’),’ELEMENTAL MINING DATA’,20(‘*’),’/8X,
+’NUMBER’,21X,’ELEMENTS TO BE MINED’’,/2X,’INC’,3X,’ELEMENTS’,
+’3X,’8(‘-----’,2X))

this section must be modified if size MAXINC is increased *****

DO 270 I=1,NRFF
IF(NME(I).GT.8) GOTO 240
WRITE(NDEV,230) I,NME(I),(MINES(I,J),J=1,NME(I))
230 FORMAT(‘/,’1X,I3,5X,I4,6X,8(I5,2X))
GOTO 270
240 CONTINUE
WRITE(NDEV,250) I,NME(I),(MINES(I,J),J=1,8)
250 FORMAT(‘/,’1X,I3,5X,I4,6X,8(I5,2X))
WRITE(NDEV, 260) (MINES(I, J), J=1, NME(I))
260 FORMAT(/, 19X, 6(I5, 2X))
270 CONTINUE
IF(IPRINT.EQ.1) GOTO 590
IF(MFLAG.EQ.1) GOTO 300
275 WRITE(OUT, 280)
280 FORMAT(/, 'DO YOU WISH TO MODIFY DATA? (Y or N) 'S)
ACCEPT 290, ANS
290 FORMAT(A1)
IF(ANS.EQ. 'H') CALL HELP(HLP.MOD)
IF(ANS.EQ. 'M') GOTO 275
IF(ANS.EQ. 'Y') GOTO 590
MFLAG=1

C******** modify data ********
300 WRITE(OUT, 310)
310 FORMAT(/, 'SELECT OPTION TO BE MODIFIED: 'S/
+ '1. NUMBER OF INCREMENTS'/
+ '2. NUMBER OF MINED ELEMENTS'/
+ '3. ELEMENTS TO BE MINED'/
+ '4. RESHOW DATA'/
+ '5. RE-ENTER DATA SET'/
+ '6. END MODIFICATIONS'
NUM=6
CALL CHOOSE(IOPT, NUM)

C***** branch to desired option *****
420 WRITE(OUT, 430)
430 FORMAT(/, 'RE-ENTER TOTAL NUMBER OF INCREMENTS'> 'S)
CALL READIN(NRF, NOTRD, HLP.FIL)
IF(NOTRD.EQ.0) GOTO 420
IF(NRF.LE.MAXELY) GOTO 580
WRITE(OUT, 440) MAXELY
440 FORMAT(/, '***** MAXIMUM NUMBER OF INCREMENTS EXCEEDED TRY /
+ 'AGAIN *****', /, 10X, 'CURRENT MAXIMUM=', I5, //)
GOTO 420
450 WRITE(OUT, 460)
460 FORMAT(/, 'ENTER INCREMENT NUMBER TO BE MODIFIED'> 'S)
CALL READIN(I, NOTRD, HLP.FIL)
IF(NOTRD.EQ.0) GOTO 450
470 WRITE(OUT, 480) I
480 FORMAT(/, 'RE-ENTER NUMBER OF ELEMENTS TO BE MINED ON '
+ 'INCREMENT', I3, ' > 'S)
CALL READIN(NME(I), NOTRD, HLP.FIL)
IF(NOTRD.EQ.0) GOTO 470
IF(NME(I).LE.MAXINC) GOTO 580
WRITE(OUT, 490) MAXINC
490 FORMAT(/, '*** MAXIMUM NUMBER OF ELEMENTS REMOVED EXCEEDED '
+ '***', /, 10X, 'CURRENT MAXIMUM=', I5, //)
GOTO 470
510 WRITE(OUT, 520)
520 FORMAT(/, '<ENTER INCREMENT NUMBER TO BE MODIFIED> ('*)
     CALL READIN(I, NOTRD, HLPFIL)
     IF(NOTRD.EQ.0) GOTO 510

530 WRITE(OUT, 540) NME(I), I
540 FORMAT(/, '<RE-ENTER', I4, '.' ELEMENT NUMBERS TO BE MINED ON'
     + ' 'INCREMENT ', I3, '> ('*)
     READ(IN, * , ERR=550) (MINES(I, J), J=1, NME(I))
     GOTO 560

550 WRITE(OUT, 560)
560 FORMAT(/, '***** RE-ENTER ALL ELEMENTS AT ONE TIME'
     + ' (e.g. 1,15,23,...) *****/)
     GOTO 530

580 CONTINUE
GOTO 165
590 CONTINUE
RETURN
END

C******************************************************************************
C* SUBROUTINE BOUND: ENTERS BOUNDARY CONDITIONS, SUMMED REACTIONS NODES *
C******************************************************************************
SUBROUTINE BOUND(NSING, NNBS, MFR, NB, NFIX, US, ANG, NOPR, DANG,
+ IBFLAG, IBC, ICODE, PNOR, PTOZ, MAXBN, MAXSM, IPRINT, MODIFY)
     DIMENSION NBC(MAXBN), NFIX(MAXBN), US(MAXBN, 2), ANG(MAXBN)
     DIMENSION NOPR(MAXSUM)
     DIMENSION IBFLAG(MAXBN), IBC(MAXBN), ICODE(MAXBN), DANG(MAXBN)
     DIMENSION ICODE(MAXBN), PNOR(MAXBN), PTOZ(MAXBN)
CHARACTER*64 NAME
CHARACTER*64 IFNAME
CHARACTER*2 HLPFIL
CHARACTER*2 HLPNME
CHARACTER*2 HLPBN
CHARACTER*2 NOHELP
COMMON IN, OUT, LPT
INTEGER IN,
     OUT=5
     HLPFIL='BF'
     HLPNME='NM'
     HLPBN='BD'
     NOHELP='NH'
     HLPMOD='MD'
     NDEV=OUT
     IF(IPRINT.NE.1) GOTO 2
     NDEV=LPT
     GOTO 351
     2 CONTINUE
     IF(MODIFY.EQ.1) GOTO 351

3 CONTINUE
MFLAG=0
NANGLE=0
NSING=0

DO 4 I=1, MAXBND
    NBC(I)=0
4 CONTINUE

WRITE(OUT, 5)
5 FORMAT(/, 10('(', ')'), ' BOUNDARY AND SINGULAR POINT CONDITIONS'
     + 10('(', ')'))

****** define a singular point ******

WRITE(OUT, 7)
7 FORMAT(/, '<DO YOU TO DEFINE A SINGULAR POINT? (Y or N)> ', $)
     ACCEPT S, ANS
8 FORMAT(A1)
9 WRITE(OUT, 10)
10 FORMAT(/, '<ENTER NODE NUMBER OF SINGULAR POINT> ', $)
     CALL READIN(NSING, NOTRD, NOHELP)
     IF(NOTRD.EQ.0) GOTO 9
11 CONTINUE

****** read in the name of the file ******

WRITE(OUT, 20)
20 FORMAT(/, '<ENTER NAME OF GMESH FILE), TYPE H FOR HELP ', $)
     GOTO 50

30 WRITE(OUT, 40)
40 FORMAT(/, '<RE-ENTER NAME OF GMESH FILE), TYPE H FOR HELP ', $)
50 READ(IN, 50) IFNAME
60 FORMAT(A)

****** determine if help is needed ******

IF(IFNAME.EQ.'H') CALL HELP(HLPNAME)
IF(IFNAME.EQ.'EXIT') GOTO 351

****** trims the trailing blanks, gives length of string ******

CALL STR$TRIM(NAME, IFNAME, LEN)

****** open and read file ******

OPEN(UNIT=9, FILE=NAME, ACCESS='SEQUENTIAL', STATUS='OLD'
     + FORM='UNFORMATTED', ERR=70)
90 CONTINUE
GOTO 30

70 CONTINUE
GOTO 30

90 CONTINUE
READ (9)
READ (9) NE, NP, NFF
READ (9)
READ (9)
READ (9) (IBFLAG(I), I=1, NFF)
CLOSE(UNIT=9)

95 WRITE(OUT,100)
100 FORMAT(///,' DO YOU WANT TO CONSIDER BOUNDARY ANGLES? (Y or N) ','$')
   ACCEPT 140, ANS
   IF(ANS.EQ.'H') CALL HELP(HLPBND)
   IF(ANS.EQ.'H') GOTO 95
   IF(ANS.EQ.'Y') NANGLE=1
   CONTINUE

I=0
120 CONTINUE
125 WRITE(OUT,130)
130 FORMAT(///,' DO YOU WANT TO READ A BOUNDARY FLAG? (Y or N) ','$')
   ACCEPT 140, ANS
140 FORMAT(A1)
   IF(ANS.EQ.'H') CALL HELP(HLPBND)
   IF(ANS.EQ.'H') GOTO 125
   IF(ANS.NE.'Y') GOTO 250
   I=I+1

150 WRITE(OUT,160)
160 FORMAT(///,' ENTER BOUNDARY FLAG NUMBER AS SET IN GMESH> ','$')
   CALL READIN(IBC(I),NOTRD,HLPFIL)
   IF(NOTRD.EQ.0) GOTO 150
170 WRITE(OUT,180)
180 FORMAT(///,' ENTER BOUNDARY CODE, TYPE "H" FOR HELP> ','$')
   CALL READRL(BCODE(I),NOTRD,HLPFIL)
   IF(NOTRD.EQ.0) GOTO 170
   IF(BCODE(I).EQ.1.0) GOTO 190
   IF(BCODE(I).EQ.2.0) GOTO 210
   IF(BCODE(I).NE.3.0) GOTO 170
190 WRITE(OUT,200)
200 FORMAT(///,' ENTER VALUE OF R DISPLACEMENT> ','$')
   CALL READRL(PNOI(I),NOTRD,HLPBND)
   IF(NOTRD.EQ.0) GOTO 190
   IF(BCODE(I).EQ.1.0) GOTO 225
210 WRITE(OUT,220)
220 FORMAT(///,' ENTER VALUE OF Z DISPLACEMENT> ','$')
   CALL READRL(PTOZ(I),NOTRD,HLPBND)
   IF(NOTRD.EQ.0) GOTO 210
   CONTINUE
225 WRITE(OUT,228)
228 FORMAT(///,' ENTER INCLINED BOUNDARY ANGLE> ','$')
   CALL READRL(DANG(I),NOTRD,HLPBND)
   IF(NOTRD.EQ.0) GOTO 227
229 CONTINUE
230 WRITE(OUT,240)
240 FORMAT(///,' DO YOU WISH TO MAKE CHANGES? (Y or N) ','$')
   ACCEPT 140, ANS1
   IF(ANS1.EQ.'H') CALL HELP(HLPMOD)
   IF(ANS1.EQ.'H') GOTO 230
IF(ANS1.EQ. 'Y') GOTO 150
GOTO 120

250 CONTINUE
NUMTB=1
ICOUNT=0
DO 300 K=1,NFF
   IF(IBFLAG(K) .LE. 0) GOTO 260
   ICOUNT=ICOUNT+1
   IF(ICOUNT .LE. MAXBND) GOTO 257
WRITE(OUT,255) MAXBND
   FORMAT(/,'***** MAXIMUM NUMBER OF BOUNDARY NODES EXCEEDED'
+ '*****','/,'.20X,'CURRENT MAXIMUM=',I4,/
+ 'RE-ENTER BOUNDARY CONDITIONS OR CONSULT PROGRAMMER'S MANUAL')
GOTO 3
257 CONTINUE

NNN=IBFLAG(K)
GOTO 270
260 NFLAG=ABS(IBFLAG(K))
GOTO 300
270 DO 290 I=1,NP
   IF(NNN .NE. I) GOTO 290
   DO 280 J=1,NUMTB
      IF(IBC(J).NE.NNFLAG) GOTO 280
      IF(BCODE(J).LT.0.0) GOTO 280
      NBC(ICOUNT)=I
      IF(BCODE(J).EQ.1.0) NFIX(ICOUNT)=10
      IF(BCODE(J).EQ.2.0) NFIX(ICOUNT)=1
      IF(BCODE(J).EQ.3.0) NFIX(ICOUNT)=11
      US(ICOUNT,1)=PNOR(J)
      US(ICOUNT,2)=PTOZ(J)
      ANG(ICOUNT)=DANG(J)
280 CONTINUE
290 CONTINUE
300 CONTINUE
NNBB=ICOUNT

C***** delete undefined or doubly defined boundary points *****
C

303 CONTINUE
NEXCH=1
ISAFE=0
DO 320 WHILE(NEXCH.GT.0)
   IF(ISAFE.GT.90000) GOTO 346
   NEXCH=0
   N1=NNBB-1
   DO 315 I=1,N1
      N=N1+1
      IF(NBC(I).NE.NBC(N)) GOTO 305
      NBC(I)=0
      IF(NFIX(I).EQ.NFIX(N)) GOTO 305
      NFIX(N)=11
305 CONTINUE
   IF(NBC(I).GE.NBC(N)) GOTO 310
      TEMP=NBC(I)
      NBC(I)=NBC(N)
      NBC(N)=TEMP
      TEMP2=NFIX(I)
NFIX(I) = NFIX(N)
NFIX(N) = TEMP2
TEMP3 = US(I, 1)
US(I, 1) = US(N, 1)
US(N, 1) = TEMP3
TEMP4 = US(I, 2)
US(I, 2) = US(N, 2)
US(N, 2) = TEMP4
TEMP5 = ANG(I)
ANG(I) = ANG(N)
ANG(N) = TEMP5
NEXCH = NEXCH + 1
ISAFE = ISAFE + 1

310 CONTINUE
315 CONTINUE
320 CONTINUE

****** determine number of defined boundary points ******

NOZERO = NNBB
DO 325 I = 1, NNBB
   IF (NBC(I) .EQ. 0) NOZERO = NOZERO - 1
325 CONTINUE
NNBB = NOZERO

****** rank score array using bubble sort algorithm ******

NEXCH = 1
ISAFE = 0
DO 345 WHILE (NEXCH .GT. 0)
   IF (ISAFE .GT. 90000) GOTO 346
   NEXCH = 0
   N1 = NNBB - 1
   DO 340 I = 1, N1
      IF (NBC(I) .LT. NBC(N)) GOTO 330
      TEMP = NBC(I)
      NBC(I) = NBC(N)
      NBC(N) = TEMP
      TEMP2 = NFIX(I)
      NFIX(I) = NFIX(N)
      NFIX(N) = TEMP2
      TEMP3 = US(I, 1)
      US(I, 1) = US(N, 1)
      US(N, 1) = TEMP3
      TEMP4 = US(I, 2)
      US(I, 2) = US(N, 2)
      US(N, 2) = TEMP4
      TEMP5 = ANG(I)
      ANG(I) = ANG(N)
      ANG(N) = TEMP5
      NEXCH = NEXCH + 1
      ISAFE = ISAFE + 1
330 CONTINUE
340 CONTINUE
345 CONTINUE
GOTO 351

346 WRITE (OUT, 347)
347 FORMAT (/*, ' **** NODAL SORTING NOT COMPLETE **** */)
c***** view boundary conditions *****
c 351 CONTINUE
DO 352 I=1,NNBB
  IF(NFIX(I).EQ.10) ICODE(I)=1
  IF(NFIX(I).EQ.1) ICODE(I)=2
  IF(NFIX(I).EQ.11) ICODE(I)=3
352 CONTINUE
WRITE(NDEV,353) 'BOUNDARY AND SINGULAR POINT CONDITIONS'
  + '10(*)')
  IF(NSING.GT.0) GOTO 355
WRITE(OUT,354) 'NO SINGULAR POINT CONSIDERED' NSING
GOTO 357
WRITE(OUT,356) NSING
FORMAT(I, 'SINGULAR POINT DEFINED AS NODE NUMBER=', I5, *)
CONTINUE
WRITE(NDEV,364) NNBB
FORMAT(I, 'NUMBER OF BOUNDARY POINTS=', I5)
WRITE(NDEV,366) INDEX=NBB
  INDEX=I
WRITE(NDEV,368) INDEX, NBC(I), ICODE(I), US(I, 1), US(I, 2), ANG(I)
FORMAT(3X, I5, 4X, I5, 5X, I5, 8X, G10. 4, 3X, G10. 4, 3X, G10. 4)
CONTINUE
DO 370 I=1,NNBB
INDEX=I
WRITE(NDEV,380) I
CONTINUE
WRITE(OUT,385) 'SELECT OPTION TO BE MODIFIED'
  + '1. SINGULAR POINT'
  + '2. BOUNDARY CODE'
  + '3. R DISPLACEMENT'
  + '4. Z DISPLACEMENT'
  + '5. BOUNDARY ANGLE'
  + '6. RE-SHOW DATA'
  + '7. RE-ENTER ALL BOUNDARY DATA'
  + '8. END MODIFICATIONS'
NUM=8
CALL CHOOSE(IOPT, NUM)
GOTO(386, 393, 393, 393, 393, 352, 3, 450) IOPT

386  WRITE(OUT, 390)
390  FORMAT(1, ' <RE-ENTER NODE NUMBER OF SINGULAR POINT> ')
     CALL READIN(NSING, NOTRD, NOHELP)
     IF(NOTRD.EQ.0) GOTO 389
391  CONTINUE
     GOTO 449

393  WRITE(OUT, 395)
395  FORMAT(1, ' <ENTER INDEX TO BE MODIFIED> ')
     CALL READIN(I, NOTRD, HLPFIL)
     IF(NOTRD.EQ.0) GOTO 393
     GOTO(400, 400, 425, 435, 445) IOPT

400  WRITE(OUT, 403)
403  FORMAT(1, ' <RE-ENTER BOUNDARY CODE, TYPE "H" FOR HELP> ')
     CALL READRL(BCODE(I), NOTRD, HLPFIL)
     IF(NOTRD.EQ.0) GOTO 400
     NFIX(I)=11
     IF(BCODE(I).EQ.1) NFIX(I)=10
     IF(BCODE(I).EQ.2) NFIX(I)=1

405  WRITE(OUT, 407)
407  FORMAT(1, ' <RE-ENTER VALUE OF R DISPLACEMENT> ')
     CALL READRL(US(I,1), NOTRD, HLPBND)
     IF(NOTRD.EQ.0) GOTO 405

410  WRITE(OUT, 415)
415  FORMAT(1, ' <RE-ENTER VALUE OF Z DISPLACEMENT> ')
     CALL READRL(US(I,2), NOTRD, HLPBND)
     IF(NOTRD.EQ.0) GOTO 410
420  CONTINUE
     GOTO 449

425  WRITE(OUT, 430)
430  FORMAT(1, ' <RE-ENTER VALUE OF R DISPLACEMENT> ')
     CALL READRL(US(I,1), NOTRD, HLPBND)
     IF(NOTRD.EQ.0) GOTO 425
     GOTO 449

435  WRITE(OUT, 440)
440  FORMAT(1, ' <RE-ENTER VALUE OF Z DISPLACEMENT> ')
     CALL READRL(US(I,2), NOTRD, HLPBND)
     IF(NOTRD.EQ.0) GOTO 435
     GOTO 449

445  WRITE(OUT, 447)
447  FORMAT(1, ' <RE-ENTER INCLINED BOUNDARY ANGLE> ')
     CALL READRL(ANG(I), NOTRD, HLPBND)
     IF(NOTRD.EQ.0) GOTO 445
449  CONTINUE
     GOTO 352

****** summed reaction nodes *****

450  CONTINUE
     IF(MODIFY.EQ.1) GOTO 520
CONTINUE
MFLAG=0
WRITE(OUT, 460)
FORMAT(///, 20(' ', ' SUMMVED REACTION NODES ', 20('')))
WRITE(OUT, 480)
WRITE(OUT, 480)
CALL READIN(MPR, NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 470
IF(MPR.LE.MAXSUM) GOTO 487
WRITE(OUT, 485) MAXSUM
WRITE(OUT, 500)
WRITE(NDEV, 530)
FORMAT(///, 20(' ', ' SUMMVED REACTION NODES ', 20('')))
WRITE(NDEV, 533) MPR
WRITE(NDEV, 540)
WRITE(NDEV, 540)
DO 510 I=1, MPR
WRITE(OUT, 500)
CALL READIN(NOPR, NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 490
CONTINUE
IF(NOTRD.LE.0) GOTO 660
DO 510 I=1, MPR
WRITE(OUT, 500)
CALL READIN(NOPR, NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 490
CONTINUE
WRITE(OUT, 500)
WRITE(NDEV, 530)
FORMAT(///, 20(' ', ' SUMMVED REACTION NODES ', 20('')))
WRITE(OUT, 500)
WRITE(NDEV, 530) MPR
WRITE(NDEV, 535)
WRITE(NDEV, 535)
WRITE(NDEV, 540)
WRITE(NDEV, 540)
DO 560 I=1, MPR
WRITE(OUT, 550)
FORMAT(///, 20(' ', ' SUMMVED REACTION NODES ', 20('')))
WRITE(OUT, 550) I, NOPR
CONTINUE
IF(ITEMPR.EQ.1) GOTO 660
IF(MFLAG.EQ.1) GOTO 580
WRITE(OUT, 570)
FORMAT(///, 10X, ' DO YOU WISH TO MAKE CHANGES (Y or N) ', ')
ACCEPT 140, ANS
IF(ANS.EQ. 'H') CALL HELP(HELPMOD)
IF(ANS.EQ. 'Y') GOTO 563
IF(ANS.EQ. 'Y') GOTO 563
MFLAG=1
CONTINUE
WRITE(OUT, 590)
FORMAT(///, ' SELECT OPTION TO BE MODIFIED', ')
+ 1. NUMBER OF REACTION NODES'
+ 2. REACTION NODE NUMBERS'
+ 3. RE-SHOW DATA'
+ 4. RE-ENTER ALL REACTION NODE DATA'
5. END MODIFICATIONS

MODIFICATIONS

NUM=5
CALL CHOOSE(IOPT, NUM)
GOTO(592, 600, 520, 455, 660) IOPT

592 WRITE(OUT, 594)
594 FORMAT(//, 'ENTER NUMBER OF SUMMED REACTION NODES', 
CALL READIN(MPR, NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 592

IF(MPR.LE.MAXSUM) WRITE(OUT, 596) MAXSUM
596 FORMAT(//, '***** MAXIMUM NUMBER OF SUMMED REACTION NODES', 
+ ' EXCEEDED *****', //, 20X, 'CURRENT MAXIMUM=', I4)
GOTO 592
598 CONTINUE
GOTO 650

600 WRITE(OUT, 610)
610 FORMAT(//, 'ENTER INDEX FOR NODE TO BE MODIFIED', 
CALL READIN(I, NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 600

630 WRITE(OUT, 640) I
640 FORMAT(//, 'RE-ENTER NODE FOR INDEX', I4)
CALL READIN(MPR(I), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 630
650 CONTINUE
GOTO 520
660 CONTINUE
RETURN
END

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

SUBROUTINE LOAD1: ENTERS LOADING DESCRIPTION, POINT LOADING DATA, 
GRAVITY AND CENTRIFUGAL FORCE DATA
*****************************************************************************

SUBROUTINE LOAD1(LNUM, LDTYPE, TITLE1, NRE, NR3, NRPR, NRC, 
NO, FX, FY, MAXPNT, MODIY, IPRINT)
DIMENSION TITLE1(MAXL, 20), LDTYPE(MAXL), NRE(MAXL), NR3(MAXL)
DIMENSION NRPR(MAXL), NRC(MAXL)
DIMENSION ND(MAXL, MAXPNT), FX(MAXL, MAXPNT), FY(MAXL, MAXPNT)
COMMON IN, OUT, LPT
INTEGER OUT
CHARACTER*2 HLPFIL
CHARACTER*2 HLPMOD
HLPFIL='L1'
HLPMOD='MD'
NDEV=OUT
IF(IPRINT NE. 1) GOTO 2
NDEV=LPT
GOTO 127
2 CONTINUE
IF(MODIFY.EQ.1) GOTO 127
3 CONTINUE
MFLAG=0

cccc

**** enter loading description ****

cccc

WRITE(OUT, 5)
  5 FORMAT(//, 20('C'), ' LOADING DESCRIPTION ' , 20('C'))

10 WRITE(OUT, 20) LNUM
20 FORMAT(//, ' <ENTER TITLE FOR LOADING CASE > , I3, '>' )
   ACCEPT 25, (TITLE1(LNUM, J), J=1, 20)
25 FORMAT(20A4)
   IF(TITLE1(LNUM, 1), EQ. 'H') CALL HELP(HLPFIL)
   IF(MFLAG.EQ.1) GOTO 129

30 WRITE(OUT, 40)
40 FORMAT(//, ' <ENTER INCREMENT CONTROL FOR PRESCRIBED LOADS >'
   + ' AND DISPLACEMENTS >'
   + ' 0=EQUAL INCREMENTS >'
   + ' 1=INCREASES PROPORTIONAL TO FACTOR ARRAY >'
   + ' 2=INDEPENDENT SET OF LOADS FOR EACH INCREMENT >'
   CALL READINC(LDTYPE(LNUM), NOTRD, HLPFIL)
   IF(NOTRD.EQ.0) GOTO 30
   IF(MFLAG.EQ.1) GOTO 129

50 CONTINUE
   IF(NRE(LNUM).GT.0) GOTO 55
   NRE(LNUM)=1
55 CONTINUE

WRITE(OUT, 60)
60 FORMAT(//, ' <DO YOU WISH TO CONSIDER POINT LOADS (Y or N) >'
   ' )
   ACCEPT 85, ANS
   IF(ANS.EQ. 'H') CALL HELP(HLPFIL)
   IF(ANS.EQ. 'H') GOTO 30
   IF(ANS.EQ. 'Y') NRE(LNUM)=0
   IF(MFLAG.EQ.1) GOTO 129

70 CONTINUE
   NRG(LNUM)=0
   WRITE(OUT, 80)
80 FORMAT(//, ' <DO YOU WANT TO CONSIDER GRAVITY LOADS? (Y or N) >'
   ' )
   ACCEPT 85, ANS1
85 FORMAT(4A1)
   IF(ANS1.EQ. 'H') CALL HELP(HLPFIL)
   IF(ANS1.EQ. 'H') GOTO 70
   IF(ANS1.EQ. 'Y') NRG(LNUM)=1
   IF(MFLAG.EQ.1) GOTO 129

90 CONTINUE
   NRPRS(LNUM)=0
   WRITE(OUT, 100)
100 FORMAT(//, ' <DO YOU WANT TO CONSIDER PRESSURE LOADS?'
   + ' (Y or N) >'
   + ' )
   ACCEPT 85, ANS2
   IF(ANS2.EQ. 'H') CALL HELP(HLPFIL)
   IF(ANS2.EQ. 'H') GOTO 90
   IF(ANS2.EQ. 'Y') NRPRS(LNUM)=1
   IF(MFLAG.EQ.1) GOTO 125

110 CONTINUE
   NRC(LNUM)=0
   WRITE(OUT, 120)
120 FORMAT(//, '<DO YOU WANT TO CONSIDER CENTRIFUGAL FORCES?'
+ ' (Y or N) ?> ')$
ACCEPT $5, ANS3
IF(ANS3.EQ. 'H') CALL HELP(HLPFILE)
IF(ANS3.EQ. 'H') GOTO 110
IF(ANS3.EQ. 'Y') NRC(LNUM)=1
IF(MFLAG.EQ. 1) GOTO 125

125 CONTINUE
MNRE='YES'
IF(ANS.EQ. 'Y') MNRE='NO'
MNRR='NO'
IF(ANS1.EQ. 'Y') MNRR='YES'
MPRS='YES'
IF(ANS2.EQ. 'Y') MPRS='NO'
MNRC='YES'
IF(ANS3.EQ. 'Y') MNRC='NO'
GOTO 129

127 CONTINUE
MNRE='NO'
IF(NRE(LNUM).GT.0) MNRE='YES'
MNRR='NO'
IF(NR(LNUM).GT.0) MNRR='YES'
MPRS='NO'
IF(NR(PR(LNUM)).GT.0) MPRS='YES'
MNRC='NO'
IF(NRC(LNUM).GT.0) MNRC='YES'
GOTO 129

c******* view loading description *****
c

129 CONTINUE
WRITE(NDEV,130)
130 FORMAT(//' , 'LOADING DESCRIPTION , 20( ' ')')
WRITE(NDEV,140) LNUM
140 FORMAT(//' , 'LOADING PROBLEM ,' , 4X)
WRITE(NDEV,150), (TITLE1(LNUM,J), J=1,20)
150 FORMAT(//' , 1X, 20A4)

LDTYP=LDTYPE(LNUM)+1
GOTO(160,180,200) LDTYP
160 WRITE(NDEV,170)
170 FORMAT(//' , 'INCREMENT CONTROL= EQUAL INCREMENTS')
GOTO 220
180 WRITE(NDEV,190)
190 FORMAT(//' , 'INCREMENT CONTROL= PROPORTIONAL TO LOAD FACTORS')
GOTO 220
200 WRITE(NDEV,210)
210 FORMAT(//' , 'INCREMENT CONTROL= DIFFERENT LOADS FOR EACH INCREMENT')
220 CONTINUE

WRITE(NDEV,230) MNRE
230 FORMAT(//' , 'CONSIDER POINT LOADS= 'A4)
WRITE(NDEV,240) MNRR
240 FORMAT(//' , 'CONSIDER GRAVITY LOADS= 'A4)
WRITE(NDEV,250) MPRS
250 FORMAT(//' , 'CONSIDER PRESSURE LOADS= 'A4)
WRITE(NDEV,260) MNRC
260 FORMAT(//' , 'CONSIDER CENTRIFUGAL FORCES= 'A4)
IF(IPRINT EQ. 1) GOTO 333

C********* modify loading description ******

C

IF(MFLAG EQ. 1) GOTO 277

265 WRITE (OUT,270)

270 FORMAT(///,10X,' DO YOU WISH TO MAKE CHANGES? (Y or N) ')

ACCEPT 275,ANS

IF(ANS.EQ.'H') CALL HELP(HLPMOD)

IF(ANS.EQ.'H') GOTO 265

IF(ANS.NE.'Y') GOTO 290

275 FORMAT(A1)

MFLAG=1

277 CONTINUE

WRITE(OUT,280)

280 FORMAT(///, <SELECT OPTION TO BE MODIFIED>''/

+ 1. TITLE '/

+ 2. INCREMENT CONTROL '/

+ 3. POINT LOADINGS '/

+ 4. GRAVITY LOADS '/

+ 5. PRESSURE LOADS '/

+ 6. CENTRIFUGAL FORCES '/

+ 7. RE-SHOW DATA'/

+ 8. RE-ENTER DATA SET'/

+ 9. END MODIFICATIONS '/)

NUM=9

CALL CHOOSE(IOPT,NUM)

GOTO(10,30,50,70,90,110,129,3,290) IOPT

CC

C******** point loading ******

CC

290 CONTINUE

IF(MODIFY EQ. 1) GOTO 333

IF(NRE(LNUM).EQ.0) GOTO 430

295 CONTINUE

MFLAG=0

WRITE(OUT,300)

300 FORMAT(///,20( '<.'),'POINT LOADING DATA ',20('<.'))//)

302 WRITE(OUT,304)

304 FORMAT(///, <ENTER NUMBER OF NODES WHERE POINT LOADS ARE APPLIED> ')

+ ')

CALL READIN(NRE(LNUM),NOTRD,HLPMOD)

IF(NOTRD.EQ.0) GOTO 302

IF(NRE(LNUM).LE.MAXPNT) GOTO 308

WRITE(OUT,306) MAXPN

306 FORMAT(///, '**** MAXIMUM NUMBER OF NODAL POINT LOADINGS EXCEEDED ',

+ '****',//,10X,' CURRENT MAXIMUM=',IS,///)

GOTO 302

308 CONTINUE

DO 330 I=1,NRE(LNUM)

310 WRITE(OUT,320)

320 FORMAT(///, <ENTER NODE NUMBER AND FORCES IN R,Z DIRECTION> ')

+ ')

READ(IN,*,ERR=325) ND(LNUM,I),FX(LNUM,I),FY(LNUM,I)
GOTO 330

325 CONTINUE
326 WRITE(OUT,327)
327 FORMAT(/,'**** ENTER NODE AND BOTH FORCES (e.g. 245,55,750)'/)
330 CONTINUE

C****** view point loadings *******
C
333 CONTINUE
334 IF(NRE(LNUM) EQ. 0) GOTO 430
335 WRITE(NDEV,335)
336 FORMAT(/,20('*'), ' POINT LOAD DATA ', 20('*')))
337 WRITE(NDEV,340)
339 DO 360 I=1,NRE(LNUM)
340 WRITE(NDEV,350) I, ND(LNUM, I), FX(LNUM, I), FY(LNUM, I)
341 CONTINUE
342 IF(IPRINT.EQ.1) GOTO 430
C****** modify loading *******
C
365 WRITE(OUT,370)
366 FORMAT(/,1X,' DO YOU WISH TO MAKE CHANGES? (Y or N) ')
367 ACCEPT 275,ANS
368 MFLAG=1
369 IF(ANS.EQ. 'H') CALL HELP(HLPMOD)
370 IF(ANS.EQ. 'H') GOTO 365
371 IF(ANS .NE. 'Y ') GOTO 430
372 CONTINUE
373 WRITE(OUT,376)
374 FORMAT(/, 'SELECT OPTION TO BE MODIFIED'/)
375 + 1. NUMBER OF POINT LOADS'/
376 + 2. POINT LOADINGS'/
377 - 3. RE-SHOW DATA'/
378 + 4. RE-ENTER DATA SET'/
379 + 5. END MODIFICATIONS'/)
380 NUM=5
381 CALL CHOOSE(IOPT,NUM)
382 GOTO(377, 400, 333, 295, 430) IOPT
383 WRITE(OUT,378)
384 FORMAT(/, 'RE-ENTER NUMBER OF POINT LOADINGS> ')
385 CALL READIN(NRE,LNUM),NOTRD,L1)
386 IF(NOTRD .EQ. 0) GOTO 377
387 IF(NRE(LNUM).LE.MAXPNT) GOTO 390
388 WRITE(OUT,379) MAXPNT
389 FORMAT(/,'**** MAXIMUM NUMBER OF NODAL POINT LOADINGS EXCEEDED '
390 + '****',/10X, 'CURRENT MAXIMUM=',I5,/)
391 GOTO 377
392 CONTINUE
393 GOTO 420
400  WRITE(OUT, 403)
403  FORMAT(/,' <ENTER INDEX FOR NODE TO BE MODIFIED> $')
      CALL READIN(I, NOTRD, HLPFIL)
      IF(NOTRD.EQ.0) GOTO 400

405  WRITE(OUT, 410)
410  FORMAT(/,' <RE-ENTER NODE NUMBER AND FORCES IN R, Z DIRECTION> $')
      READ(IN, *). ERR=415). ND(LNUM, I), FX(LNUM, I), FY(LNUM, I)
      GOTO 420

415  CONTINUE
      WRITE(OUT, 417)
417  FORMAT(/,' **** ENTER NODE AND BOTH FORCES (e.g. 245,55,750) $')
      GOTO 405

420  CONTINUE
      GOTO 333

430  CONTINUE
      RETURN
      END

C***************************************************************
C* SUBROUTINE LOAD2: ENTERS GRAVITY AND CENTRIFUGAL CONSTANTS, AND *
C* MATERIAL DENSITIES                                             *
C*****************************************************************

SUBROUTINE LOAD2(LNUM, NMAT, NL, NRC, NRG, THETA, GRAV, ANGVEL, DENS,
                  MODIFY, IPRINT, MAXLD, MATMAX)
   DIMENSION NRC(MAXLD), NRG(MAXLD), THETA(MAXLD), GRAV(MAXLD)
   DIMENSION ANGVEL(MAXLD), DENS(MAXLD, MATMAX)
   COMMON IN, OUT, LPT
   INTEGER OUT
   CHARACTER*2 HLPFIL
   CHARACTER*2 HLPMOD
   HLPFIL='L2'
   HLPMOD='MD'
   NRCG=NRC(LNUM) + NRG(LNUM)
   IF(NRCG.LE.0) GOTO 350
   NDEV=OUT
   IF(IPRINT.NE.1) GOTO 2
   NDEV=LPT
   GOTO 80
   2 CONTINUE
   IF(MODIFY.EQ.1) GOTO 80
   3 CONTINUE
   MFLAG=0

C***** enter loading constants ******

WRITE(OUT,10)
10  FORMAT(/,'(*', ' LOADING CONSTANTS ',20(')'))

WRITE(OUT,30)
30  FORMAT(/,' <IF GRAVITY AXIS TILTED- ENTER ANGLE> $')
      CALL READRL(THETA(LNUM), NOTRD, HLPFIL)
      IF(NOTRD.EQ.0) GOTO 20
IF (MFLAG .EQ. 1) GOTO 80

IF (NRC(LNUM).NE.1) GOTO 55

40 WRITE(OUT,50)
50 FORMAT(*', '<ENTER CONSTANT OF ACCELERATION DUE TO GRAVITY> ',S)
     CALL READRL(GRAV(LNUM), NOTRD, HLPFIL)
     IF (NOTRD .EQ. 0) GOTO 40
     IF (MFLAG .EQ. 1) GOTO 80

55 CONTINUE
     IF (NRC(LNUM).NE.1) GOTO 80
60 WRITE(OUT,70)
70 FORMAT(*', '<ENTER ANGULAR VELOCITY IN RADIANS/UNIT TIME> ',S)
     CALL READRL(ANGVEL(LNUM), NOTRD, HLPFIL)
     IF (NOTRD .EQ. 0) GOTO 60

80 CONTINUE
     WRITE(NDEV,85)
85 FORMAT(*', '/20(', '.'), ', 'LOADING CONSTANTS ',20(', '.'))

     WRITE(NDEV,90) THETA(LNUM)
90 FORMAT(*', '/20( 'GRAVITATIONAL ANGLE= ',G10.4)
     WRITE(NDEV,100) GRAV(LNUM)
100 FORMAT(*', '/20( 'GRAVITATIONAL CONSTANT= ',G10.5)
     IF (NRC(LNUM).NE.1) GOTO 113
     WRITE(NDEV,110) ANGVEL(LNUM)
110 FORMAT(*', '/20( 'ANGULAR VELOCITY= ',G10.5)
115 CONTINUE
     IF (IPRINT .EQ. 1) GOTO 240

120 FORMAT(*', '/10X,' DO YOU WISH TO MAKE CHANGES? (Y or N) ',S)
     ACCEPT 125,ANS
125 FORMAT(A1)
     IF (ANS .EQ. 'H') CALL HELP(HLPFIL)
     IF (ANS .EQ. 'H') GOTO 117
     IF (ANS .EQ. 'Y') GOTO 150
     MFLAG=1
130 CONTINUE
     WRITE(OUT,140)
140 FORMAT(*', '/SELECT OPTION TO BE MODIFIED>/ ',
      1. GRAVITY ANGLE '/
      2. GRAVITY CONSTANT '/
      3. ANGULAR VELOCITY '/
      4. RE-ENTER DATA SET'/
      5. END MODIFICATIONS '<)

     NUM=5
     CALL CHOOSE(IOPT, NUM)
     GOTO(20, 40, 60, 3, 150) IOPT

ccc

****** enter density of materials ******
ccc
150 CONTINUE
IF(MODIFY.EQ.1) GOTO 240
165 CONTINUE
MFLAG=0
WRITE(OUT,167)
167 FORMAT(//,20('*'), ' MATERIAL DENSITIES ' ,20('*')//)

****** modify do-loop parameter if rubble considered *****

IF(NL.NE.1) GOTO 170
NMAT1=NMAT-1
GOTO 180
170 CONTINUE
NMAT1=NMAT
180 CONTINUE

DO 210 N=1,NMAT1
WRITE(OUT,200) N
200 FORMAT(//' <ENTER DENSITY FOR MATERIAL',I2,' > '*')
CALL READRL(DENS(LNUM,N),NOTRD,HLPFIL)
IF(NOTRD.EQ.0) GOTO 190
210 CONTINUE
IF(NL.NE.1) GOTO 240
N=NMAT
220 WRITE(OUT,230)
230 FORMAT(//' <ENTER DENSITY FOR RUBBLEIZED MATERIAL> '*')
CALL READRL(DENS(LNUM,N),NOTRD,HLPFIL)
IF(NOTRD.EQ.0) GOTO 220

****** view densities *****

WRITE(NDEV,245)
245 FORMAT(//' ' ,20('*'), ' MATERIAL DENSITIES ','20('*'))
WRITE(NDEV,250)
250 FORMAT(//' *, MATERIAL',10X,' DENSITY')

DO 270 N=1,NMAT
WRITE(NDEV,260) N,DENS(LNUM,N)
260 FORMAT(//' *,12X,12X,G10.5)
270 CONTINUE
IF(IPRINT.EQ.1) GOTO 350

****** modify densities? *****

IF(MFLAG.EQ.1) GOTO 283
275 WRITE(OUT,280)
280 FORMAT(//' *X,' DO YOU WISH TO MAKE CHANGES? (Y or N) '*')
ACCEPT 125,ANS
IF(ANS.EQ.'H') CALL HELP(HLPMOD)
IF(ANS.EQ.'H') GOTO 275
IF(ANS.NE.'Y') GOTO 350
MFLAG=1

283 WRITE(OUT,285)
285 FORMAT(//' ' ,SELECT OPTION'/')
+      1. MATERIAL DENSITY'
+      2. RE-ENTER DATA SET'
+      3. END MODIFICATIONS')
NUM=3
CALL CHOOSE(IOPT, NUM)
GOTO(290, 165, 350) IOPT

290 WRITE(OUT, 300)
300 FORMAT(/, '<ENTER MATERIAL NUMBER FOR DENSITY TO BE ',
+$ 'MODIFIED> $')
CALL READIN(I, NOTRD, HLPFIL)
IF(NOTRD. EQ. 0) GOTO 290

320 WRITE(OUT, 330) I
330 FORMAT('<ENTER DENSITY FOR MATERIAL', I2, '>
+$ ')
CALL READRL(DENS(LNUM, I), NOTRD, HLPFIL)
IF(NOTRD. EQ. 0) GOTO 320
GOTO 240

350 CONTINUE
RETURN
END

C**********************************************************************
C* SUBROUTINE LOAD3: INPUTS ELEMENTAL PRESSURE LOADS
C**********************************************************************

SUBROUTINE LOAD3(LNUM, NRPRS, NSFR, LNE, NOPL, PN, PT,
+ MAXPRS, MAXLD, MODIFY, IPRINT)
DIMENSION NRPRS(MAXLD), LNE(MAXLD), NOPL(MAXLD, MAXPRS, 4)
DIMENSION PN(MAXLD, MAXPRS, 4), PT(MAXLD, MAXPRS, 4)
COMMON IN, OUT, LPT
INTEGER OUT
CHARACTER*2 HLPFIL
CHARACTER*2 HLPMOD
HLPFIL='L3'
HLPMOD='MD'
IF(NRPRS(LNUM), LE. 0) GOTO 480
NDEV=OUT
IF(IPRINT. NE. 1) GOTO 2
NDEV=LPT
GOTO 167
2 CONTINUE
IF(MODIFY. EQ. 1) GOTO 167
MFLAG=0

C**** if no pressure loads omit following section ****

WRITE(OUT, 5)
5 FORMAT(/, 'PRESSURE LOADING', 20(''), )
WRITE(OUT, 20)
20 FORMAT(/, '<ENTER NUMBER OF ELEMENTS ON WHICH PRESSURES'
+$ ' ARE APPLIED> $')
CALL READIN(LNE(LNUM), NOTRD, HLPFIL)
IF(NOTRD. EQ. 0) GOTO 10
IF(LNE(LNUM), LE. MAXPRS) GOTO 27
WRITE(OUT, 25) MAXPRS
25 FORMAT(/, '**** MAXIMUM NUMBER OF ELEMENTS EXCEEDED ****'/)
C
C CURRENT MAXIMUM='i5,' //
GOTO 10
27 CONTINUE
IC=NSFR+1
DO 165 I=1, LNE(LNUM)
30 WRITE(OUT, 40) IC
40 FORMAT(/, ' ENTER SPECIFIED NODE NUMBERS OF PRESSURE
+ SURFACE ELEMENT> ' )
READ(IN, *, ERR=45) (NOPL(LNUM, I, K), K=1, IC)
GOTO 48
45 CONTINUE
WRITE(OUT, 47)
47 FORMAT(/, ' ENTER ALL NODE NUMBERS AT ONE TIME'
+ ' (e.g. 23, 34, 45,...)/')
GOTO 30
48 CONTINUE
C
C constant loading? ******
C
50 WRITE(OUT, 60)
60 FORMAT(/, ' ARE THE SAME LOADS APPLIED ON ALL NODES OF THE'
+ ' ELEMENT? (Y or N)> ' )
ACCEPT 45, ANS
65 FORMAT(A1)
IF(ANS .EG. 'H') CALL HELP(HLPMOD)
IF(ANS .EG. 'H') GOTO 50
C ****** constant loads on all nodes not valid for linear elements ******
C
IF(NSFR .EG. 1) GOTO 110
IF(NSFR .NE. 'Y') GOTO 110
70 WRITE(OUT, 80)
80 FORMAT(/, ' ENTER NORMAL PRESSURE> ' )
CALL READRL(PN(LNUM, I, 1), NOTRD, HLPMIL)
IF(NOTRD .EQ. 0) GOTO 70
PN(LNUM, I, 2)=PN(LNUM, I, 1)
PN(LNUM, I, 3)=PN(LNUM, I, 1)
PN(LNUM, I, 4)=PN(LNUM, I, 1)
90 WRITE(OUT, 100)
100 FORMAT(/, ' ENTER TANGENTIAL PRESSURE> ' )
CALL READRL(PT(LNUM, I, 1), NOTRD, HLPMIL)
IF(NOTRD .EQ. 0) GOTO 90
PT(LNUM, I, 2)=PT(LNUM, I, 1)
PT(LNUM, I, 3)=PT(LNUM, I, 1)
PT(LNUM, I, 4)=PT(LNUM, I, 1)
GOTO 165
110 CONTINUE
DO 160 J=1, IC
120 WRITE(OUT, 130) NOPL(LNUM, I, J)
130 FORMAT(/, ' ENTER NORMAL PRESSURE OF NODE', I4, ' > ' )
CALL READRL(PN(LNUM, I, J), NOTRD, HLPMIL)
IF(NOTRD .EQ. 0) GOTO 120
140 WRITE(OUT, 150) NOPL(LNUM, I, J)
150 FORMAT(/, ' ENTER TANGENTIAL PRESSURE OF NODE', I4, ' > ' )
CALL READRL(PT(LNUM, I, J), NOTRD, HLPMIL)
IF(NOTRD .EQ. 0) GOTO 140
160 CONTINUE
***view data****

167 WRITE(NDEV, 170)
170 FORMAT('/1x, 'PRESSURE LOADING', 2x,'PRESSURE LOADING')
     IC=NSFR+1

WRITE(NDEV, 180) LNE(LNUM)
180 FORMAT(/, 2x, 'NUMBER OF PRESSURE ELEMENTS= ', IS)

WRITE(NDEV, 190)
190 FORMAT(/, 1x, 'ELEMENT', 18x, 'NORMAL', 17x, 'TANGENTIAL'
 + 1.2x, 'INDEX', 4x, 'NODE', 10x, 'PRESSURE', 17x, 'PRESSURE')

DO 270 I=1, LNE(LNUM)
220 WRITE(NDEV, 220)
250 WRITE(NDEV, 240) I, NCPL(LNUM, I, 1), PN(LNUM, I, 1), PT(LNUM, I, 1)
240 FORMAT(3x, I2, 4x, I5, 10x, G10.4, 17x, G10.4)
    DO 260 K=2, IC
     WRITE(NDEV, 250) NCPL(LNUM, I, K), PN(LNUM, I, K), PT(LNUM, I, K)
250 FORMAT(9x, I3, 10x, G10.4, 17x, G10.4)
260 CONTINUE
270 CONTINUE

IF(IPRINT .EQ. 1) GOTO 480
IF(MFLAG .EQ. 1) GOTO 295

WRITE(OUT, 280)
260 FORMAT(/1x,'DO YOU WISH TO VIEW OR MODIFY DATA? (Y or N) ', $'
     ACCEPT 290, ANS
290 IF(ANS .EQ. 'Y') CALL HELP(HLPMOD)
290 IF(ANS .EQ. 'H') GOTO 275
290 MFLAG=1

modify data****

295 WRITE(OUT, 300)
300 FORMAT(/, ' <SELECT OPTION TO BE MODIFIED> '/
 + 1. NUMBER OF PRESSURE ELEMENTS '/
 + 2. PRESSURE LOADS'/
 + 3. SHOW DATA ON SCREEN '/
 + 4. RE-ENTER DATA SET '/
 + 5. END VIEW OR MODIFY '/)

NUM=5
CALL CHOOSE(IOPT, NUM)
GOTO(302, 310, 167, 3, 480) IOPT

WRITE(OUT, 304)
304 FORMAT(/, '<ENTER NUMBER OF ELEMENTS ON WHICH PRESSURES'
 + ' ARE APPLIED> $'
     CALL READIN(LNE(LNUM), NORD, HLPFIL)
     IF(NORD .EQ. 0) GOTO 302
     IF(LNE(LNUM), LE, MAXPRS) GOTO 309
     WRITE(OUT, 308) MAXPRS
308 FORMAT(/, ' **** MAXIMUM NUMBER OF ELEMENTS EXCEEDED ****/')
+ 10X, 'CURRENT MAXIMUM=' , IS , //)
GOTO 360
308 CONTINUE
GOTO 460
310 WRITE (OUT, 320)
320 FORMAT (/ , '<ENTER INDEX FOR ELEMENT TO BE MODIFIED>' , '(')
CALL READIN(I, NOTRD, HLPFIL)
IF(NOTRD. EQ. 0) GOTO 310

**** reenter data *****

330 WRITE (OUT, 340) IC
340 FORMAT (/ , '<ENTER', I2, ' SPECIFIED NODE NUMBERS OF PRESSURE'
+ ' SURFACE ELEMENT>' , '(')
READ (IN , * , ERR=345) (NGPL(LNUM, I , K) , K=1 , IC)
GOTO 348
345 CONTINUE
WRITE (OUT, 347)
347 FORMAT (/ , 'RE-ENTER ALL NODE NUMBERS AT ONE TIME'
+ ' (e.g. 23, 34, 45 ...)')
GOTO 330
348 CONTINUE

**** constant loading? *****

350 WRITE (OUT, 360)
360 FORMAT (/ , '<ARE THE SAME LOADS APPLIED ON ALL NODES OF THE'
+ ' ELEMENT? (Y or N)> ')
ACCEPT 365.ANS
265 FORMAT (A1)
IF (ANS. EQ. 'H') CALL HELP (HLPFIL)
IF (ANS. EQ. 'H') GOTO 350

**** const loads all nodes not valid for linear elements *****

IF (NSFR. EQ. 1) GOTO 410
IF (ANS. NE. 'Y') GOTO 410
370 WRITE (OUT, 380)
380 FORMAT (/ , '<RE-ENTER NORMAL PRESSURE>' , '(')
CALL READRL(PN(LNUM, I, 1) , NOTRD, HLPFIL)
IF(NOTRD. EQ. 0) GOTO 370
PN(LNUM, I, 2) = PN(LNUM, I, 1)
PN(LNUM, I, 3) = PN(LNUM, I, 1)
PN(LNUM, I, 4) = PN(LNUM, I, 1)
390 WRITE (OUT, 400)
400 FORMAT (/ , '<RE-ENTER TANGENTIAL PRESSURE>' , '(')
CALL READRL(PT(LNUM, I, 1) , NOTRD, HLPFIL)
IF(NOTRD. EQ. 0) GOTO 390
PT(LNUM, I, 2) = PT(LNUM, I, 1)
PT(LNUM, I, 3) = PT(LNUM, I, 1)
PT(LNUM, I, 4) = PT(LNUM, I, 1)
GOTO 460
410 CONTINUE
DO 455 JJ=1, IC
420 WRITE (OUT, 430) NCPL(LNUM, I, JJ)
430 FORMAT (/ , '<RE-ENTER NORMAL PRESSURE OF NODE', I4, ' > ')
CALL READRL(PN(LNUM, I, JJ) , NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 420
440 WRITE(OUT,450) NOUT(LNUM,I,JJ)
450 FORMAT(/'RE-ENTER TANGENTIAL PRESSURE OF NODE'/'I4' ')' 
CALL READRL(P,T(LNUM,I,JJ),NOTRD,HLPFIL)
IF(NOTRD.EQ.0) GOTO 440
455 CONTINUE
460 CONTINUE
GOTO 167
480 CONTINUE
RETURN
END

C* SUBROUTINE LOAD4: INPUTS LOAD FACTORS, PRINT-OUT INDICATORS, MAXIMUM 
  NUMBER OF ITERATIONS, CONVERGENCE FACTOR
C*
C*---------------------------------------------------------------
SUBROUTINE LOAD4(NUIN, INC, FAC, NOT, NIT, MFAC, MAXLD, 
+ INCMA, MODIFY, IPRINT)
DIMENSION INC(MAXLD), FAC(MAXLD, INCMA), NOT(MAXLD, INCMA)
DIMENSION NIT(MAXLD), MFAC(MAXLD)
COMMON IN,QUT,LPT
INTEGER OUT
CHARACTER*2 HLPFIL
CHARACTER*2 HLPMD
HLPFIL='L4'
HLPMD='MD'
NDEV=OUT
IF(IPRINT NE. 1) GOTO 2
NDEV=LPT
GOTO 75
2 CONTINUE
IF(MODIFY.EQ.1) GOTO 75
3 CONTINUE
MFLAG=0
WRITE(OUT,5)
5 FORMAT(/'INCREMENT AND LOAD FACTOR DATA ',20('('))
10 WRITE(OUT,20)
20 FORMAT(/, 'ENTER NUMBER OF LOADING INCREMENTS' ) 
CALL READIN(NINC(LNUM), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 10
IF(MFLAG.EQ.1) GOTO 75
IF(NINC(LNUM).LE.INCMA) GOTO 40
WRITE(OUT,30) INCMA
30 FORMAT(/, '*++++* MAXIMUM NUMBER OF INCREMENTS EXCEEDED *++++* 
+ /,10X,' CURRENT MAXIMUM=',I5, //)
GOTO 10
40 CONTINUE
DO 70 I=1,NINC(LNUM)
50 WRITE(OUT,60) I
60 FORMAT(/, 'ENTER FACTOR APPLIED TO INCREMENT',I4,) 
CALL READRL(FAC(LNUM,I), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 50
CONTINUE
70 CONTINUE

C**** view load factors ****
C
75 CONTINUE
WRITE(NDEV,80)
80 FORMAT(1/A/20(' '), 'LOAD FACTOR DATA ', 20(' '))
WRITE(NDEV,85) NINC(LNUM)
85 FORMAT(1/A/ 'NUMBER OF LOAD INCREMENTS=', I5)
WRITE(NDEV,90)
90 FORMAT(1/A/2X, 'INCREMENT ', 8X, 'FACTOR')
DO 110 I=1, NINC(LNUM)
   WRITE(NDEV,100) I, FAC(LNUM, I)
100 FORMAT(2X, I5, 17X, F5.3)
110 CONTINUE
IF(IPRINT.EQ.1) GOTO 185
IF(MFLAG.EQ.1) GOTO 122
113 WRITE(OUT,115)
115 FORMAT(1/A/10X, 'DO YOU WISH TO MAKE CHANGES? (Y OR N) ', $)
   ACCEPT 120, ANS
120 FORMAT(A1)
   IF(ANS.EQ.'H') CALL HELP(HLPMOD)
   IF(ANS.EQ.'H') GOTO 113
   IF(ANS.NE.'Y') GOTO 194
   MFLAG=1
C
C**** modify load factors ****
C
122 WRITE(OUT,125)
125 FORMAT(1/A/ '<SELECT OPTION TO BE MODIFIED>' )
   + '(' NUMBER OF INCREMENTS '
   + '(' LOAD FACTOR '
   + '(' RE-SHOW DATA'
   + '(' RE-ENTER DATA SET'
   + '(' END MODIFICATIONS'
NUM=5
   CALL CHOOSE(IOPT, NUM)
   GOTO(132, 140, 75, 154) IOPT
132 WRITE(OUT,134)
134 FORMAT(1/A/ '<RE-ENTER NUMBER OF LOADING INCREMENTS> ', $)
   CALL READIN(NINC(LNUM), NOTRD, HLPFIL)
   IF(NOTRD.EQ.0) GOTO 132
   IF(NINC(LNUM).LE.INCMAK) GOTO 138
   WRITE(OUT,136) INCMAK
136 FORMAT(1/A/ '***** MAXIMUM NUMBER OF INCREMENTS EXCEEDED *****'
   + '/.10X, ' CURRENT MAXIMUM=', I5, /*)
   GOTO 132
138 CONTINUE
   GOTO 150
140 WRITE(OUT,142)
142 FORMAT(1/A/ '<ENTER INCREMENT NUMBER TO BE MODIFIED> ', $)
   CALL READIN(I, NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 140
144 WRITE(OUT,145) I
145 FORMAT(/,'<ENTER LOAD FACTOR APPLIED TO INCREMENT',I4,'$')
CALL READL(FAC(LNUM,I),NOTRD,HLPFIL)
IF(NOTRD.EQ.0) GOTO 144
150 CONTINUE
GOTO 75

C********* enter write-out indicators *********

154 CONTINUE
155 CONTINUE
MFLAG=0
156 FORMAT(/,,20('<'),'WRITE-OUT INDICATORS ',20('>'))
157 WRITE(OUT,159)
159 FORMAT(/,'<ENTER INITIAL AND FINAL INCREMENT WRITE-OUT',+ 'INDICATOR>',
+ '/.3X,'+'0=NO OUTPUT',
+ '/.3X,'+'1=DISPLACEMENTS AT NODAL POINTS',
+ '/.3X,'+'2=REACTIONS AT CONSTRAINED NODES AS WELL',
+ '/.3X,'+'3=STRESSES AT GAUSS POINTS AS WELL',
+ '/.3X,'+'4=RESIDUAL FORCES AT NODES AS WELL',
+ '/'' FOR MOST CASES USE AN INDICATOR= 3''/)
DO 190 I=1,NINC(LNUM)
160 WRITE(OUT,170) I
170 FORMAT(/,'<ENTER WRITE-OUT INDICATOR FOR INCREMENT',I4,+
+ '>')
CALL READIN(NOUT(LNUM,1),NOTRD,HLPFIL)
IF(NOTRD.EQ.0) GOTO 180
WRITE(OUT,159)
GOTO 160
180 CONTINUE

C********* view write-out indicators *********

185 CONTINUE
WRITE(NDEV,190)
190 FORMAT(/,,20('*'),'OUTPUT CONTROL DATA ',20('*'))
WRITE(NDEV,200)
200 FORMAT(/,2X,'INCREMENT',I4,CONTROL NUMBER'/)
DO 220 I=1,NINC(LNUM)
WRITE(NDEV,210) I,NOUT(LNUM,1)
210 FORMAT(2X,15,10X,15)
220 CONTINUE
IF(IPRINT.EQ.1) GOTO 230
IF(MFLAG.EQ.1) GOTO 231
WRITE(OUT,230)
230 FORMAT(/,10X,'DO YOU WISH TO MAKE CHANGES? (Y or N) ',$)
ACCEPT 120,ANS
IF(ANS.EQ. 'Y') CALL HELP(HLPMOD)
IF(ANS.EQ. 'N') GOTO 225
IF(ANS.NE. 'Y') GOTO 270
MFLAG=1
**modify write-out indicators ******

```c
231 WRITE(OUT,232)
232 FORMAT(//,' <SELECT OPTION TO BE MODIFIED> '/
+ ' 1. WRITE-OUT INDICATOR '/
+ ' 2. RE-SHOW DATA '/
+ ' 3. RE-ENTER DATA SET '/
+ ' 4. END MODIFICATIONS ')

NUM=4
CALL CHOOSE(IOPT, NUM)
GOTO(235, 185, 155, 270) IOPT

235 WRITE(OUT,240)
240 FORMAT(//,' <ENTER INCREMENT NUMBER TO BE MODIFIED> '*)
CALL READIN(I, NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 235

249 WRITE(OUT,250) I
250 FORMAT(//,' <RE-ENTER WRITE-OUT INDICATOR FOR INCREMENT ',I3
+ ' > '*)
CALL READIN(NOUT(LNUM, I), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 249
GOTO 165
```

**enter number iterations and convergence factor ******

```c
270 CONTINUE
IF(MODIFY.EQ.1) GOTO 320
273 CONTINUE
WRITE(OUT,275)
275 FORMAT(///,20('C'), ' MAX ITERATION AND CONVERGENCE FAC ',
+ ' 20('D')'))

280 WRITE(OUT,290)
290 FORMAT(///, ' <ENTER MAXIMUM NUMBER OF ITERATIONS> '*)
CALL READIN(NIT(LNUM), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 280

300 WRITE(OUT,310)
310 FORMAT(///, ' <ENTER CONVERGENCE FACTOR IN PERCENT> '*)
CALL READP(CONFAC(LNUM), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 300
```

**view or modify iter. or con. factor ******

```c
320 CONTINUE
WRITE(INDV,325)
325 FORMAT(///,20('C'), ' MAX ITERATION AND CONVERGENCE FAC ',
+ ' 20('D')'))
WRITE(INDV,330) NIT(LNUM)
330 FORMAT(///, ' MAXIMUM NUMBER OF ITERATIONS= ',I3)
WRITE(INDV,340) CONFAC(LNUM)
340 FORMAT(10X, ' CONVERGENCE FACTOR= ',G10.4)
IF(IFPRINT.EQ.1) GOTO 360

345 WRITE(OUT,350)
350 FORMAT(///,10X, ' DO YOU WISH TO MAKE CHANGES? (Y or N) '*)
ACCEPT 120, ANS
IF(ANS.EQ. 'H') CALL HELP(HLPMOD)
IF(ANS.EQ. 'H') GOTO 345
```
IF(ANS.EQ. 'Y') GOTO 273

CONTINUE
RETURN
END

C**********************************************************************
C*
C* SUBROUTINE LOADS: ENTERS BOUNDARY IMPACT NODES
C*
C**********************************************************************
SUBROUTINE LOADS(LNUM, NIMP, IMPNOD, MAXLD, MAXIMP, MODIFY, IPRINT)
DIMENSION NIMP(MAXLD), IMPNOD(MAXLD, MAXIMP)
COMMON IN, OUT, LPT
INTEGER
CHARACTER*2 HLPFIL
CHARACTER*2 HLPMOD
HLPFIL='LS'
HLPMOD='MD'
NDEV=OUT
IF(IPRINT.NE.1) GOTO 2
NDEV=OUT
GOTO 85
2 CONTINUE
IF(MODIFY.EQ.1) GOTO 85
3 CONTINUE
MFLAG=0

C******** enter impact nodal data ********
MFLAG=0
WRITE(CUT,55)
55 FORMAT('** ENTER NUMBER OF IMPACT BOUNDARY NODES >', ILE, A, A)
WRITE(CUT,65)
60 FORMAT('** CURRENT MAXIMUM= ', I5, A, A)
65 FORMAT('/** MAXIMUM NUMBER OF IMPACT NODES EXCEEDED ***/
+ 20X, ' CURRENT MAXIMUM= ', I5, A, A)
GOTO 60
69 CONTINUE
IF(NIMP(LNUM).LE.0) GOTO 180
DO 80 I=1,NIMP(LNUM)
70 WRITE(CUT,75)
75 FORMAT('** ENTER IMPACT NODE NUMBER >', A)
CALL READIN(IMPNOD(LNUM, I), NOTRD, HLPFIL)
IF(NOTRD.EQ.0) GOTO 70
80 CONTINUE

C******** view impact nodal data ********
85 CONTINUE
WRITE(NDEV,90)
90 FORMAT('** IMPACT NODAL DATA **')
WRITE(NDEV, 100) NIMP(LNUM)
100 FORMAT(//, 'NUMBER OF BOUNDARY IMPACT NODES=', I5)
WRITE(NDEV, 110)
110 FORMAT(//, 2X, 'INDEX', 8X, 'NODE'/)
DO 130 I = 1, NIMP(LNUM)
   WRITE(NDEV, 120) I, IMPNOD(LNUM, I)
120 CONTINUE
   IF(IPRINT.EQ.1) GOTO 180
   IF(MFLAG.EQ.1) GOTO 142
135 WRITE(OUT, 140)
140 FORMAT(//, 10X, 'DO YOU WISH TO MAKE CHANGES? (Y OR N) ',$)
   ACCEPT 141, ANS
141 FORMAT(A11)
   IF(ANS.EQ. 'H') CALL HELP(HLPMOD)
   IF(ANS.EQ. 'H') GOTO 135
   IF(ANS.EQ. 'Y') GOTO 180
   MFLAG = 1
C****** modify impact nodal data *****
C
142 WRITE(OUT, 145)
143 FORMAT(//, '<SELECT OPTION TO BE MODIFIED> '/
       + ' 1. NUMBER OF IMPACT NODES'/
       + ' 2. IMPACT NODES'/
       + ' 3. RE-SHOW DATA'/
       + ' 4. RE-ENTER DATA SET'/
       + ' 5. END MODIFICATIONS'/)
   NUM = 5
   CALL CHOOSE(IOPT, NUM)
   GOTO(146, 150, 155, 150, 180) IOPT
146 WRITE(OUT, 147)
147 FORMAT(//, '<ENTER NUMBER OF IMPACT BOUNDARY NODES > ',$)
   CALL READIN(NIMP(LNUM), NOTRD, HLPMOD)
   IF(NOTRD.EQ.0) GOTO 146
   IF(NIMP(LNUM).LE.MAXIMP) GOTO 149
   WRITE(OUT, 148) MAXIMP
148 FORMAT(//, '***** MAXIMUM NUMBER OF IMPACT NODES EXCEEDED *****'/
       + ' CURRENT MAXIMUM=', I5, '//')
   GOTO 146
149 CONTINUE
   GOTO 170
150 WRITE(OUT, 155)
155 FORMAT(//, '<ENTER INDEX NUMBER TO BE MODIFIED> ',$)
   CALL READIN(I, NOTRD, HLPMOD)
   IF(NOTRD.EQ.0) GOTO 150
160 WRITE(OUT, 165) I
165 FORMAT(//, '<RE-ENTER IMPACT NODE NUMBER FOR INDEX', I4, '> ',$)
   CALL READIN(IMPNOD(LNUM, I), NOTRD, HLPMOD)
   IF(NOTRD.EQ.0) GOTO 160
170 CONTINUE
GOTO 85
180 CONTINUE
RETURN
END

C*******************************************************************************
C* SUBROUTINE PRTCON: ENTERS REDUCED PRINT-OUT CONTROLS  *
C*******************************************************************************

SUBROUTINE PRTCON(NPEL, NONODE, NPSEL, MAXPRT, MODIFY, IPRINT)
DIMENSION NPSEL(MAXPRT)
COMMON IN, OUT, LPT
INTEGER OUT
CHARACTER*2 HLPFIL
CHARACTER*2 HLPMOD
HLPFIL='P1'
HLPMOD='MD'

NDEV=OUT
IF(IPRINT .NE. 1) GOTO 2
NDEV=LPT
GOTO 72
2 CONTINUE
IF(MODIFY .EQ. 1) GOTO 72
3 CONTINUE
MFLAG=0
WRITE(OUT,5)
5 FORMAT(/,','PRINTOUT CONTROLS ','),20('',''),).
10 WRITE(OUT,20)
20 FORMAT(/,'<ENTER NODAL OUTPUT CONTROL>','),/
       + 'O=ALL NODES PRINTED','),/
       + '1=NO NODES PRINTED>','(S)
CALL READIN(NONODE, NOTRD, HLPFIL)
IF(NOTRD .EQ. 0) GOTO 10
30 WRITE(OUT,35)
35 FORMAT(/,'<ENTER NUMBER OF ELEMENTS TO BE PRINTED>','(S)
CALL READIN(NPEL, NOTRD, HLPFIL)
IF(NOTRD .EQ. 0) GOTO 30
IF(NPEL .LE. MAXPRT) GOTO 45
WRITE(OUT,40) MAXPRT
40 FORMAT(/,'**** MAXIMUM NUMBER OF PRINTED ELEMENTS EXCEEDED *****/
        + '10X,'CURRENT MAXIMUM='),15,/')
45 CONTINUE

C******** skip if no elements printed ********
C
C******** enter element printout data ********
C
DO 70 I=1,NPEL
    WRITE(OUT,60)
70 FORMAT(/,'<ENTER ELEMENT TO BE PRINTED>','(S)
    CALL READIN(NPEL(I), NOTRD, HLPFIL)
    IF(NOTRD .EQ. 0) GOTO 50
CONTINUE

view data

CONTINUE

WRITE(ND8, 75)

FORMAT(///, 20(' '), 'PRINTOUT CONTROLS ', 20(' '), //)

WRITE(ND8, 80) NODE

FORMAT(///, ' NODAL OUTPUT (0=ALL NODES, 1=NO NODES)----', I2)

WRITE(ND8, 90) NELP

FORMAT(///, ' NUMBER OF ELEMENTS TO BE PRINTED------', I5)

IF(NELP LE 0) GOTO 110

WRITE(ND8, 95)

FORMAT(///, 'INDEX', 7X, 'PRINT NODES')

DO 110 I=1, NELP

WRITE(ND8, 100), I, NPEL(I)

110 CONTINUE

IF(IPRINT EQ. 1) GOTO 200

IF(MFLAG EQ. 1) GOTO 135

WRITE(OUT, 120)

FORMAT(///, 10X, 'DO YOU WISH TO MAKE ANY CHANGES? (Y or N) ' $)

ACCEPT 130, ANS

FORMAT(A1)

IF(ANS. EQ. 'H') CALL HELP(HLPMOD)

IF(ANS. EQ. 'M') GOTO 115

IF(ANS. NE. 'Y') GOTO 200

MFLAG=1

modify data

WRITE(OUT, 140)

FORMAT(///, 'SELECT OPTION TO BE MODIFIED: '/'
+
+ ' 1. NUMBER OF PRINTED ELEMENTS'/
+
+ ' 2. NODAL OUTPUT'/
+
+ ' 3. PRINTED ELEMENT NUMBERS'/
+
+ ' 4. RE-SHOW DATA'/
+
+ ' 5. RE-ENTER DATA SET'/
+
+ ' 6. END MODIFICATIONS'/)

NUM=6

CALL CHOOSE(IOPT, NUM)

GOTO(145, 142, 150, 72, 3, 200) IOPT

WRITE(OUT, 143)

FORMAT(///, 'ENTER NODAL OUTPUT CONTROL> ' , //
+
+ ' 0=ALL NODES PRINTED', //,
+
+ ' 1=NO NODES PRINTED>' , $)

CALL READIN(NONODE, NOTRD, HLPFIL)

IF(NOTRD. EQ. 0) GOTO 142

GOTO 190

WRITE(OUT, 146)

FORMAT(///, 'ENTER NUMBER OF ELEMENTS TO BE PRINTED> ' $)

CALL READIN(NELP, NOTRD, HLPFIL)

IF(NOTRD. EQ. 0) GOTO 143

IF(NELP LE MAXPRT) GOTO 149
WRITE(OUT, 147) MAXPRT
147 FORMAT( /, 'MAXIMUM NUMBER OF PRINTED ELEMENTS EXCEEDED ****'/, 10X, 'CURRENT MAXIMUM=', I5, //)
148 CONTINUE
GOTO 190
150 WRITE(OUT, 160)
160 FORMAT( 'ENTER INDEX FOR ELEMENT TO BE MODIFIED> $')
CALL READIN(I, NOTRD, HLPFIL)
IF(NOTRD .EQ. 0) GOTO 150
175 WRITE(OUT, 180)
180 FORMAT( 'ENTER ELEMENT TO BE PRINTED> $')
CALL READIN(NPEL(I), NOTRD, HLPFIL)
IF(NOTRD .EQ. 0) GOTO 175
190 CONTINUE
GOTO 72
200 CONTINUE
RETURN
END

C*************************************************************
C* SUBROUTINE READSK: READS DATA FROM A DISK FILE
C*************************************************************
C* SUBROUTINE READSK(NUNIT, MAXLD, MATMAX, MAXTAB, MAXINC, MAXLY,
 + MAXPNT, MAXPRS, MAXIMP, INCMAX, MAXPRT, MAXBND, MAXSUM,
 + TITLE, NPROB, NLDT, DDF, NMAT, NSFR, NGAUS, NALGO,
 + NNP, NFF, NT, NL, NSIZ, NCR, MGAUS, ORT, NTAB, TABSTN,
 + AK, NFF, NSNS, MINES, NSING, NNNS, NFF, NSNS,
 + NFF, U, ANG, NFF, LDTY, TITLE1, NRE, NR, NFRS, NRC,
 + ND, FX, FY, THETA, GRAV, ANGVEL, DENS, LNE, NOPL, PN, PT, NINC, FAC,
 + NOUT, NIT, CONFAC, NIMP, IMPNQO, NPEL, NONNQO, NPEL)
DIMENSION TITLE(20)
DIMENSION ORT(MATMAX, 8)
DIMENSION NTAB(MATMAX), TABSTR(MATMAX, MAXTAB), TABSTN(MATMAX, MAXTAB)
DIMENSION NME(MAXINC), MINES(MAXINC, MAXLY)
DIMENSION NBC(MAXBND), NFIX(MAXBND), US(MAXBND, 2), ANG(MAXBND)
DIMENSION NOPR(MAXSUM)
DIMENSION TITLE1(MAXLD, 20), LDTYTYPE(MAXLD), NRE(MAXLD), NRG(MAXLD)
DIMENSION NFRS(MAXLD), NRC(MAXLD), ND(MAXLD, MAXPNT), FX(MAXLD, MAXPNT)
DIMENSION FY(MAXLD, MAXPNT)
DIMENSION THETA(MAXLD), GRAV(MAXLD), ANGVEL(MAXLD), DENS(MAXLD, MATMAX)
DIMENSION LNE(MAXLD), NOPL(MAXLD, MAXPRT), PN(MAXLD, MAXPRT, 4)
DIMENSION PT(MAXLD, MAXPRT, 4)
DIMENSION NINC(MAXLD), FAC(MAXLD, INCMAX), NOUT(MAXLD, INCMAX), NIT(MAXLD)
DIMENSION CONFAC(MAXLD), NIMP(MAXLD), IMPNQO(MAXLD, MAXIMP)
DIMENSION NPEL(MAXPRT)

COMMON IN, OUT, LPT
INTEGER OUT
CHARACTER*64 IFNAME
CHARACTER*64 NAME
CHARACTER*2 HLPNME
HLPNME= 'NM'
NOTCPN=0
**c***** read in the name of the file ******** 

```plaintext
1  CONTINUE
   IF(NOTOPN .EQ. 1) GOTO 3
2  WRITE(OUT,2)
   FORMAT(/,'<ENTER FILE NAME TO BE READ>, TYPE H FOR HELP '*)
   GOTO 3
3  WRITE(OUT,4)
   WRITE(OUT,4)'<RE-ENTER FILE NAME TO BE READ>, TYPE H FOR HELP '*)
5  READ(IN,6) IFNAME
6  FORMAT(A)
```

**c***** determine if help is needed ******** 

```plaintext
IF(IFNAME.NE.'H') GOTO 7
7  CALL HELP(HLPNAME)
   GOTO 1
```

**c***** trim the trailing blanks, gives length of string ******** 

```plaintext
CALL STRTRIM(NAME,IFNAME,LEN)
```

**c***** open and read file ******** 

```plaintext
OPEN(UNIT=NUNIT,ACCESS='SEQUENTIAL',STATUS='OLD',
     FILE=NAME,ERR=8)
```

**c***** read solution controls ******** 

```plaintext
READ(NUNIT,10) NDUM
READ(NUNIT,10) NPROB
READ(NUNIT,20) (TITLE(I),I=1,20)
READ(NUNIT,10) NP,NE,NS,NL,D,ND,NMAT,NSFR,NGAUS,NALGO,NPP,
   + NYIELD,NT,NL,NSIZ,NCORD,MGAUS
10  FORMAT(16I5)
20  FORMAT(20A4)
```

**c***** read material properties ******** 

```plaintext
DO 30 N=1,NMAT
   K=N
   READ(NUNIT,40) K,(ORT(N,I),I=1,6)
30  CONTINUE
40  FORMAT(10,6F10.3)
```

**c***** read tabulated stress/strain data ******** 

```plaintext
DO 50 N=1,NMAT
   IF(ORT(N,4).GT.0) GOTO 50
   READ(NUNIT,60) NTAB(N)
50  CONTINUE
```

**Read rubbleization data**

```plaintext
C**** read rubbleization data *****

IF(NL . NE. 1) GOTO 111
DO 75 N=1, NMAT
  K=N
  READ(NUNIT, 100) K, (ORT(N, I), I=7, 8)
75 CONTINUE
READ(NUNIT, 90) NK
IF(NRFF . LE. 0) GOTO 111
READ(NUNIT, 90) (NME(I), I=1, NRFF)
DO 80 I=1, NRFF
  READ(NUNIT, 90) (MINES(I, J), J=1, NME(I))
80 CONTINUE
READ(NUNIT, 110)
```

**Read boundary condition data**

```plaintext
C***** read boundary condition data *****

111 CONTINUE
READ(NUNIT, 114) NDUM, NSINC, NNBB, MPR
IF(NNBB . LE. 0) GOTO 113
DO 112 I=1, NNBB
  READ(NUNIT, 116) NBC(I), NFIX(I), UL(I), US(I, 2), ANG(I)
112 CONTINUE
113 CONTINUE
IF(MPR . LE. 0) GOTO 120
READ(NUNIT, 114) (NOPR(I), I=1, MPR)
114 FORMAT(1615)
116 FORMAT(215, 4F10.3)
```

**Read loading description**

```plaintext
C***** read loading description *****

120 CONTINUE
DO 250 LNUM=1, NLD
  READ(NUNIT, 140) NDUM, LDTYPE(LNUM)
  READ(NUNIT, 150) (TITLE1(LNUM, I), I=1, 20)
  READ(NUNIT, 140) NDUM, NRE(LNUM), NRG(LNUM), NRPRS(LNUM),
                   NDUM, NRC(LNUM)
  IF(NRE(LNUM) . LE. 0) GOTO 165
  DO 130 I=1, NRE(LNUM)
    READ(NUNIT, 160) NO(LNUM, I), FX(LNUM, I), FY(LNUM, I)
 130 CONTINUE
140 FORMAT(15S)
150 FORMAT(20A4)
160 FORMAT(I10, 2F10.3)
```

**Read loading constants**

```plaintext
C***** read loading constants *****

165 CONTINUE
NRCG=NRC(LNUM)+NRG(LNUM)
IF(NRCG . LE. 0) GOTO 175
READ(NUNIT, 170) THETA(LNUM), GRAV(LNUM), ANGVEL(LNUM)
READ(NUNIT, 170) (DENS(LNUM, I), I=1, NMAT)
```
170 FORMAT(8F10.3)
c
***** read pressure loads *****
c
175 CONTINUE
IF(NPRTSL(LNUM) LE 0) GOTO 210
READ(NUNIT, 190) LNE(LNUM)
IC=NSFR+1
DO 180 I=1, LNE(LNUM)
    READ(NUNIT,190)(NOPL(LNUM, I, J), J=1, IC)
    READ(NUNIT, 200)((PLN(LNUM, I, J), PT(LNUM, I, J)), J=1, IC)
180 CONTINUE
190 FORMAT(B15)
200 FORMAT(BF10.3)
c
***** read increment impact data *****
c
210 CONTINUE
READ(NUNIT, 220) NINC(LNUM)
READ(NUNIT, 230)(FAC(LNUM, I), I=1, NINC(LNUM))
READ(NUNIT, 220)(NOUT(LNUM, I), I=1, NINC(LNUM))
READ(NUNIT, 240)(NIT(LNUM), CONFA(LNUM))
READ(NUNIT, 220)(NIMP(LNUM))
IF(NIMP(LNUM) LE 0) GOTO 250
READ(NUNIT, 220)(IMPNO(LNUM, I), I=1, NIMP(LNUM))
220 FORMAT(16I5)
230 FORMAT(16FS.3)
240 FORMAT(110.F10.3)
250 CONTINUE
c
***** read printout controls *****
c
READ(NUNIT, 260) NELP, NONODE
IF(NELP LE 0) GOTO 270
READ(NUNIT, 260)(NPFL(I), I=1, NELP)
260 FORMAT(16IS)
270 CONTINUE
CLOSE(UNIT=NUNIT)
RETURN
END

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

SUBROUTINE WRDATA: WRITES DATA ONTO DISK FILE (SCRUBS.DAT)
**********************************************************************

SUBROUTINE WRDATA(NUNIT, MAXLD, MATMAX, MXTAB, MAXINC, MAXELY, + MAXPNT, MAXPRS, MAXIMP, INCMAX, MAXPRT, MAXBND, MAXSUM, IWRITE, + TITLE, NRCCB, NLD, NDF, NMAK, NSFR, NGAS, NLAG0, + NPFF, NYIELD, NT, NL, NSII, NORD, MGAUS, ORT, NTA0, TABSH, TABSN, + AK, NRFF, NFE, MINES, NSING, NNNB, MPN, NBC, + NFDY, US, ANG, NPRS, LDTYP, TITL0, NRE, NRG, NPRS, NRC, + ND, FX, FV, THETA, GRAV, ANGVEL, DENS, LNE, NOPL, PN, PT, NINC, FAC, + NOUT, NIT, CONFA, NIMP, IMPNO, NELP, NONODE, NPEL) +

DIMENSION TITLE(20)
DIMENSION ORT(MATMAX, B)
DIMENSION NTA0(MATMAX), TABSH(MATMAX, MXTAB), TABSN(MATMAX, MXTAB)
DIMENSION NFE(MAXINC), MINES(MAXINC, MAXELY)
DIMENSION NBC(MAXND), NFIX(MAXND), US(MAXND,2), ANG(MAXND)
DIMENSION NDFR(MAXSUM)
DIMENSION TITL(MAXLD,20), LDTYP(MAXLD), NRE(MAXLD), NRG(MAXLD)
DIMENSION NFRS(MAXLD), NRC(MAXLD), ND(MAXLD, MAXPN), FX(MAXLD, MAXPN)
DIMENSION FY(MAXLD, MAXPN)
DIMENSION THETA(MAXLD), GRAV(MAXLD), ANGVEL(MAXLD), DENS(MAXLD, MATMAX)
DIMENSION LNE(MAXLD), NDFL(MAXLD, MAXPRT, 4), FN(MAXLD, MAXPRT, 4)
DIMENSION PT(MAXLD, MAXPRT, 4)
DIMENSION NINC(MAXLD), FAC(MAXLD, INCMAX), NOUT(MAXLD, INCMAX), NIT(MAXLD)
DIMENSION CONFA(MAXLD), NIMP(MAXLD), IMPND(MAXLD, MAXIMP)
DIMENSION NPEL(MAXPRT)
COMMON [N,OUT,LPT
INTEGER OUT
CHARACTER*64 IFNAME
CHARACTER*64 NAME
NQTOPN=0
IWRITEN=0

C******** open file to be written ********
C
OPEN_UNITSUNIT, ACCESS='SEQUENTIAL', STATUS='NEW',
+ FILE='SCRUBS.DAT', CARRIAGECONTROL='LIST'
NDUM=O
WRITE(NUNIT, 10) NDUM
WRITE(NUNIT, 10) NPROB
WRITE(NUNIT, 20) (TITL(I), I=1,20)
WRITE(NUNIT, 30) NDUM, NDUW, NNB, NLDS, NDF, NMAT, NSFR, NGAUS, NALGO, NPP,
+ NIELD, NT, NL, NSIZ, NORD, NGAUS
10 FORMAT(15)
20 FORMAT(20A4)
30 FORMAT(1615)

C******** write material properties ********
C
DO 40 N=1,NMAT
K=N
WRITE(NUNIT, 50) K, (CRT(N, I), I=1,6)
40 CONTINUE
50 FORMAT(I10,G10.4,4,G10.4,4,G10.5)

C******** write tabulated stress/strain data ********
C
DO 60 N=1,NMAT
IF(CRT(N, 4).GT.0) GOTO 60
WRITE(NUNIT, 70) NTAB(N)
WRITE(NUNIT, 80) (TABST(N, I), I=1,NTAB(N))
WRITE(NUNIT, 80) (TABSN(N, I), I=1,NTAB(N))
60 CONTINUE
70 FORMAT(I10)
80 FORMAT(I168.0)

C******** write rupplication data ********
C
IF(NL, NE, .) GOTO 141
DO 90 N=1,NMAT
K=N
WRITE(NUNIT, 110) K, (CRT(N, I), I=7,8)
90 CONTINUE
WRITE(NUNIT, 120) AK
WRITE(NUNIT, 130) NRFF
IF(NRFF.LE.0) GOTO 141
WRITE(NUNIT, 140) (NME(I), I=1, NRFF)
DO 100 I=1, NRFF
   WRITE(NUNIT, 140) (MINES(I, J), J=1, NME(I))
100 CONTINUE
110 FORMAT(I10, G10.4, G10.5)
120 FORMAT(G10.4)
130 FORMAT(I5)
140 FORMAT(I6, I5)

c******** write boundary data ******
c
141 CONTINUE
WRITE(NUNIT, 144) NDUM, NSING, NNBB, MPR
IF(NNBB.LE.0) GOTO 143
DO 142 I=1, NNBB
   WRITE(NUNIT, 146) NBC(I), NFIX(I), US(I, 1), US(I, 2), ANG(I)
142 CONTINUE
143 CONTINUE
IF(MPR.LE.0) GOTO 150
WRITE(NUNIT, 144) (NQPR(I), I=1, MPR)
144 FORMAT(I6, I5)
146 FORMAT(2I5, 3G10.4)

c******** write loading data ******
c
150 CONTINUE
DO 360 LNUM=1, NLD
   NDUM=0
   WRITE(NUNIT, 190) NDUM, LDTYPE(LNUM)
   WRITE(NUNIT, 210) (TITLE1(LNUM, I), I=1, 20)
   WRITE(NUNIT, 200) NDUM, NRE(LNUM), NRG(LNUM), NRPRS(LNUM), NDUM, NRC(LNUM)
190 FORMAT(2I5)
200 FORMAT(6I5)
210 FORMAT(2A4)

160 CONTINUE
IF(NRE(LNUM).LE.0) GOTO 240
DO 220 I=1, NRE(LNUM)
   WRITE(NUNIT, 230) ND(LNUM, I), FX(LNUM, I), FY(LNUM, I)
220 CONTINUE
230 FORMAT(I10, 2G10.4)

240 CONTINUE
NRG=NRG(LNUM) + NRG(LNUM)
IF(NRG.LE.0) GOTO 280
WRITE(NUNIT, 260) THEA(LNUM), GRAV(LNUM), ANGVEL(LNUM)
WRITE(NUNIT, 270) (DENS(LNUM, I), I=1, NMAT)
260 FORMAT(G10.4, 2G10.5)
270 FORMAT(G10.5)

270 CONTINUE

280 CONTINUE
280 CONTINUE
    IF(MRPRSLNUM).LE.O) GOTO 320
    WRITE(NUNIT,300) LNE(LNUM)
    IC=NSFR+1
    DO 290 I=1,LNE(LNUM)
       WRITE(NUNIT,300) (NOPL(LNUM,I), J=1, IC)
       WRITE(NUNIT,310) ((PN(LNUM,I), PT(LNUM,I,J)), J=1, IC)
    290 CONTINUE
    WRITE(NUNIT,310) (NOPL(LNUM,I), J=1, IC)
    WRITE(NUNIT,320) NINC(LNUM)
    WRITE(NUNIT,330) (FAC(LNUM,I), I=1, NINC(LNUM))
    WRITE(NUNIT,330) (OUT(LNUM,I), I=1, NINC(LNUM))
    WRITE(NUNIT,330) NIT(LNUM), CONFAC(LNUM)
    WRITE(NUNIT,330) NIMP(LNUM)
    IF(NIMP(LNUM).EQ.0) GOTO 360
    WRITE(NUNIT,330) (IMPNOD(LNUM,I), I=1, NIMP(LNUM))
    330 FORMAT(IS)
    340 FORMAT(16FS.3)
    350 FORMAT(I10.C10.5)
    360 CONTINUE

C***** write increment, impact nodes *****
C
320 CONTINUE
    WRITE(NUNIT,330) NINC(LNUM)
    WRITE(NUNIT,340) (FAC(LNUM,I), I=1, NINC(LNUM))
    WRITE(NUNIT,330) (OUT(LNUM,I), I=1, NINC(LNUM))
    WRITE(NUNIT,330) NIT(LNUM), CONFAC(LNUM)
    WRITE(NUNIT,330) NIMP(LNUM)
    IF(NIMP(LNUM).EQ.0) GOTO 360
    WRITE(NUNIT,330) (IMPNOD(LNUM,I), I=1, NIMP(LNUM))
    330 FORMAT(IS)
    340 FORMAT(16FS.3)
    350 FORMAT(I10.C10.5)
    360 CONTINUE

C***** write print-out data *****
C
    WRITE(NUNIT,370) NELP, NODE
    IF(NELP.LE.0) GOTO 380
    WRITE(NUNIT,370) (NPEL(I), I=1, NELP)
    370 FORMAT(16IS)
    380 FORMAT(16FS.3)
732 CONTINUE
C
    CLOSE(UNIT=NUNIT)
    IWRITE=1
    RETURN
END

C***********************************************************************
C* SUBROUTINE READIN: READS INTEGER VALUES IN FREE FORMAT
C***********************************************************************
SUBROUTINE READIN(NUM,NOTRD,HLPFIL)
COMMON IN,OUT,LPT
INTEGER IN,OUT,LPT
CHARACTER*2 HLPFIL
NOTRD=1

C***** read integer number, NUM *****
C
    10 CONTINUE
    READ(IN,*,ERR=30) NUM
    GOTO 60
C
C***** error- optional call help file, then return and re-read *****
C
    30 CALL HELP(HLPFIL)
    NOTRD=0
C
C***** return NOTRD=0 no value read *****

60 CONTINUE
RETURN
END

C*****************************************************************************
C* SUBROUTINE READRL: READS REAL VALUES IN FREE FORMAT
C*****************************************************************************

C********************************************************************************
SUBROUTINE READRL(XNUM, NOTRD, HLPFIL)
COMMON IN, OUT, LPT
INTEGER OUT
CHARACTER*2 HLPFIL
NOTRD=1

C****** read real number, XNUM *****

20 CONTINUE
READ(IN, *, ERR=30) XNUM
GOTO 60

C****** error- optional call help file, then return and re-read *****

30 CALL HELP(HLPFIL)
NOTRD=0

C****** return NOTRD=0 no value read *****

60 CONTINUE
RETURN
END

C*****************************************************************************
C* SUBROUTINE CHOOSE: FAIL SAFE MENU INPUT
C*****************************************************************************

C********************************************************************************
SUBROUTINE CHOOSE(ICHOOSE, NUM)
DIMENSION ICHAR(19)
DATA ICHAR/2H1, 2H2, 2H3, 2H4, 2H5, 2H6, 2H7, 2H8, 2H9, 2H10, + 2H11, 2H12, 2H13, 2H14, 2H15, 2H16, 2H17, 2H18, 2H19,
+ COMMON IN, OUT, LPT
CHARACTER*2 HLP OPT
INTEGER OUT
HLP OPT='HP'

5 WRITE(CUT, 10)
10 FORMAT(/, 'ENTER YOUR OPTION, FOR HELP TYPE "H"')
READ(IN, 15) ICPT
15 FORMAT(A2)

DO 20 ICHOOSE=1, 19
   IF(ICPT.EQ.ICHOOSE) GOTO 50
20 CONTINUE

C****** error message: invalid character *****
CONTINUE
WRITE(OUT,35) IOPT, (ICHAR(I), I=1, NUM)
35 FORMAT(' *** ',A2.' IS AN INVALID RESPONSE ****'/
     +' THE ACCEPTABLE CHARACTER ARE: H ',17(A2,1X))
  GOTO 5

c***** check ICHOOSE to make sure that it is a legal number *****
c
50 CONTINUE
  IF(ICHOOSE .GT. 17) GOTO 70
  IF(ICHOOSE .GT. NUM) GOTO 30
  GOTO 900

70 CONTINUE
  IF(ICHOOSE .NE. 19) GOTO 80
  ICHOOSE=NUM
  GOTO 900.

80 CONTINUE
  IF(ICHOOSE .EQ. 18) CALL HELP(HLPOPT)
  GOTO 5

c***** return with ICHOOSE *****
c
900 CONTINUE
RETURN
END

C**************************
C* SUBROUTINE HELP: PRINT HELP MESSAGES *
C*
C******************************************************************************
SUBROUTINE HELP(HLPFIL)
COMMON IN, OUT, LPT
INTEGER OUT
CHARACTER*2 HLPFIL

  IF(HLPFIL .EQ. 'Sl') GOTO 5
  IF(HLPFIL .EQ. 'M1') GOTO 15
  IF(HLPFIL .EQ. 'T1') GOTO 25
  IF(HLPFIL .EQ. 'RI') GOTO 35
  IF(HLPFIL .EQ. 'MI') GOTO 45
  IF(HLPFIL .EQ. 'L1') GOTO 45
  IF(HLPFIL .EQ. 'L2') GOTO 55
  IF(HLPFIL .EQ. 'L3') GOTO 65
  IF(HLPFIL .EQ. 'L4') GOTO 75
  IF(HLPFIL .EQ. 'LS') GOTO 82
  IF(HLPFIL .EQ. 'P1') GOTO 85
  IF(HLPFIL .EQ. 'NM') GOTO 95
  IF(HLPFIL .EQ. 'BF') GOTO 105
  IF(HLPFIL .EQ. 'BD') GOTO 112
  IF(HLPFIL .EQ. 'HP') GOTO 115
  IF(HLPFIL .EQ. 'NH') GOTO 125
  IF(HLPFIL .EQ. 'MD') GOTO 135
  GOTO 500

  5 WRITE(OUT, 10)
10 FORMAT(/,' **** ENTERING SOLUTION CONTROLS ****'//)
1. TITLE MUST BE 80 CHARACTERS OR LESS
2. DOF IS DEGREES OF FREEDOM
FOR FURTHER ASSISTANCE CONSULT USERS GUIDE:
SOLUTION CONTROLS

GOTO 500

15 WRITE(OUT,20)
20 FORMAT(//,'****ENTERING MATERIAL PROPERTIES****'/

+ 'ENTER FOR EACH MATERIAL:'/
  - '1. YOUNG'S MODULUS OF ELASTICITY'/
  - '2. POISSON'S RATIO'/
  - '3. YIELD STRESS'/
  - '4. HARDENING MODULUS'/
  - '5. CONICAL SURFACE ANGLE (IF YIELD CONDITION IS 'DRUCKER PRAGER)'/
  - '6. THICKNESS OF MATERIAL (UNLESS PLANE STRESS IS 'CONSIDERED)'/

+ 'ENTER VALUES AS REAL EXPRESSION. USE EITHER FOR F' /
  OR E FORMAT'/
- 'FOR FURTHER ASSISTANCE CONSULT USERS GUIDE '
- 'MATERIAL PROPERTIES')//'
GOTO 500

25 WRITE(OUT,30)
30 FORMAT(//,'****ENTERING TABULATED STRESS STRAIN DATA****'/

+ 'IF THE HARDENING MODULUS OF A GIVEN MATERIAL WAS' /
  'SET TO BE NEGATIVE (AS A FLAG): TABULATED STRESS'/
- 'STRAIN POINT DATA MUST BE ENTERED FOR THAT MATERIAL'/
- 'RATHER THAN USE THE HARDENING MODULUS TO DEFINE' /
- 'THE PLASTIC STRESS/STRAIN CURVE THESE TABULATED' /
  'POINTS DEFINE THE CURVE.'/

+ 'ENTER FOR EACH MATERIAL WITH NEG. HARDENING MODULUS:'/
  - '1. NUMBER OF POINTS'/
  - '2. STRESSES AT POINTS'/
  - '3. STRAINS AT POINTS'/
- 'FOR FURTHER ASSISTANCE CONSULT USERS GUIDE.
- 'TABULATED STRESS/STRAIN')//'
GOTO 500

35 WRITE(OUT,40)
40 FORMAT(//,'****ENTERING RUBBLEIZATION DATA****'/

+ 'ENTER FOR EACH MATERIAL:'/
  + '1. BULKING PARAMETER'/
  + '2. FAILURE STRESS OF MATERIAL'/

+ 'ENTER:'/
  + '1. VALUE OF THE MINIMUM Z COORDINATE FOR WHICH' /
  'ELEMENT NODAL POINTS CANNOT PASS'/
  + '2. TOTAL NUMBER OF MINING INCREMENTS'//
1. NUMBER OF ELEMENTS TO BE MINED IN AN INCREMENT
2. ELEMENT NUMBERS MINED IN THIS INCREMENT

FOR FURTHER ASSISTANCE CONSULT USERS GUIDE: RUBBLEIZATION DATA

GOTO 500

WRITE (OUT, 43)
FORMAT ('****ENTERING RUBBLEIZATION DATA****')

+ ENTER: TOTAL NUMBER OF MINING INCREMENTS
+ ENTER IF THE NUMBER OF MINING INCREMENTS NOT ZERO
  1. NUMBER OF ELEMENTS TO BE MINED IN AN INCREMENT
  2. ELEMENT NUMBERS MINED IN THIS INCREMENT
+ FOR FURTHER ASSISTANCE CONSULT USERS GUIDE: RUBBLEIZATION DATA

GOTO 500

WRITE (OUT, 50)
FORMAT ('****ENTERING LOAD DESCRIPTION, POINT LOADS, GRAVITY OR CENTRIFUGAL FORCES ****')

+ ENTER LOAD DESCRIPTION:
  1. TITLE OF CURRENT LOADING CASE (80 CHARACTERS)
  2. INCREMENT CONTROL
+ ENTER:
  1. IF POINT LOADS WILL BE CONSIDERED
  2. IF GRAVITY LOADS WILL BE CONSIDERED
  3. IF PRESSURE LOADS WILL BE CONSIDERED
  4. IF CENTRIFUGAL FORCES WILL BE CONSIDERED
+ ENTER IF NUMBER OF POINT LOADS IS NOT ZERO:
  1. NODE NUMBER
  2. FORCE IN X DIRECTION
  3. FORCE IN Y DIRECTION
+ FOR FURTHER ASSISTANCE CONSULT USERS GUIDE: TABLE OF CONTENTS

GOTO 500

WRITE (OUT, 60)
FORMAT ('****ENTERING GRAVITY AND CENTRIFUGAL CONSTANTS, DENSITIES ****')

+ ENTER:
  1. GRAVITY AXIS IF TILTED
  2. CONSTANT OF ACCELERATION DUE TO GRAVITY
  3. ANGULAR VELOCITY
+ ENTER FOR EACH MATERIAL:
  1. MATERIAL DENSITY


**FOR FURTHER ASSISTANCE CONSULT USERS GUIDE:**

**TABLE OF CONTENTS**

GOTO 500

65 WRITE(OUT, 70)
70 FORMAT(/, ' ****ENTERING ELEMENTAL PRESSURE LOADS ****'//)
    + ' ENTER:'//
    + ' NUMBER OF ELEMENTS ON WHICH PRESSURE'//
    + ' LOADS ARE APPLIED:'//
    + ' ENTER FOR EACH ELEMENT:'//
    + ' 1. NODAL POINTS OF ELEMENT:'//
    + ' 2. NORMAL AND TANGENTIAL PRESSURE:'//
    + ' A. ON ELEMENT (IF CONSTANT STRESS):/'
    + ' B. OR CORRESPONDING NODAL LOADS:'//
    + ' NOTE: WHEN ENTERING THE NODAL PRESSURES YOU SHOULD://'
    + ' NOTE THAT THE NUMBER OF NODAL POINTS IS A:'//
    + ' FUNCTION OF THE TYPE OF ELEMENT CHOOSEN:'//
    + ' THEREFORE THE USER IS PROMPTED FOR THE:'//
    + ' CORRESPONDING NUMBER OF NODAL POINTS:'//
    + ' NOTE: CONSTANT LOADING ON LINEAR ELEMENTS IS NOT:'//
    + ' VALID. EXECUTION WILL DEFAULT TO A NON-:'//
    + ' UNIFORM LOADING CONDITION.'//
    + ' FOR FURTHER ASSISTANCE CONSULT USERS GUIDE:'//
    + ' PRESSURE LOADS.'//'!

GOTO 500

75 WRITE(OUT, 80)
80 FORMAT(/, ' ENTERING LOAD FACTORS, PRINT OUT INDICATORS,'//
    + ' MAXIMUM NUMBER OF ITERATIONS, CONVERGENCE FACTOR.'//'!
    + ' NOTE:'//
    + ' THE WRITE-OUT INDICATORS ARE COMPOSED OF A:'//
    + ' 3 DIGIT NUMBER WHICH DETERMINES THE AMOUNT OF:'//
    + ' DATA WHICH IS TO BE OUTPUT. THE INDICATOR IS:'//
    + ' CONSTRUCTED BY CHOOSING THE DESIRED INITIAL:'//
    + ' AND FINAL INDICATORS, AND THEN PLACING THE:'//
    + ' SECOND DIGIT, AND THE FINAL INDICATOR AS THE:'//
    + ' LAST DIGIT OF THE 3 DIGIT WRITE-OUT INDICATOR.'//'!
    + ' FOR FURTHER ASSISTANCE CONSULT USERS GUIDE:'//
    + ' TABLE OF CONTENTS'//'!

GOTO 500

82 WRITE(OUT, 83)
83 FORMAT(/, ' **** ENTERING BOUNDARY IMPACT NODES ****'//)
    + ' 1. TOTAL NUMBER OF BOUNDARY IMPACT NODES:'//
    + ' 2. NODE NUMBER OF IMPACT NODE (IF TOTAL NUMBER'//
    + ' IS NOT ZERO)'//
    + ' FOR FURTHER ASSISTANCE CONSULT USERS GUIDE:'//
    + ' TABLE OF CONTENTS'//'!

GOTO 500
WRITE(OUT,90)
90 FORMAT(/, '***** ENTERING PRINT-OUT CONTROLS *****'/
+ ' ENTER: '/
+ 1. NUMBER OF ELEMENTS TO BE PRINTED'/
+ 2. NODAL OUTPUT CONTROL'/
+ 3. ELEMENT NUMBERS (IF # OF ELEMENTS IS NOT 'ZERO')'/
+ ' FOR FURTHER ASSISTANCE CONSULT USERS GUIDE: '/
+ 'PRINT-OUT CONTROLS'//)
GOTO 500

WRITE(OUT,100)
100 FORMAT(/, '***** FILE NAME SPECIFICATION *****'/
+ ' MAXIMUM POSSIBLE FILENAME: '/
+ ' DISK: [USER #. SUBDIR]. FILENAME. EXT; VERSION #'/
+ ' DISK- DSK ON WHICH USERS ACCT. IS LOCATED'/
+ ' USER #: USER NUMBER ACCOUNT'/
+ ' SUBDIRECTORY- USERS SUBDIRECTORY'/
+ ' FILENAME- NAME OF FILE TO BE READ OR CREATED'/
+ ' EXTENSION- TYPE OF FILE; DEFAULT= DAT'/
+ ' FORTRAN= FOR'/
+ ' MACHINE CODE= OBJ'/
+ ' EXECUTABLE= EXE'/
+ ' DATA= DAT'/
+ ' VERSION NUMBER- CURRENT VERSION OF FILE'/
+ ' INCREMENTED EACH TIME FILE IS ACCESSED'/
+ ' MAX. VERSION #: 99'/
+ ' DEFAULT FILENAME: '/
+ ' FILENAME- DEFAULTS TO: FILENAME. DAT'/
+ ' EXAMPLE MINIMUM FILENAME: GMESH'/
+ ' FILE OPENED: GMESH.DAT'/
+ ' IF FILE CANNOT BE OPENED, TYPE EXIT WHEN'/
+ ' PROMPTED FOR FILE NAME. USER MAY ENTER REMAINING'/
+ ' PROGRAM DATA AND LATER RETURN TO MODIFY BOUNDARY'/
+ ' CONDITIONS'//)
GOTO 500

WRITE(OUT,110)
110 FORMAT(/, '***** BOUNDARY CODE INDEX *****'/
+19X ' SELECT APPROPRIATE BOUNDARY CODE'//
+10X' *******************************************************'/
+10X' * BOUNDARY DISPLACEMENT DISPLACEMENT *'/
+10X' * CODE R DIR. Z DIR. *'/
+10X' *******************************************************'/
+10X' * 1.0 SPECIFIED ------- *'/
+10X' * 2.0 SPECIFIED SPECIFIED *'/
+10X' * 3.0 SPECIFIED SPECIFIED *'/
+10X' *******************************************************'//
+ ' IF DISPLACEMENT IS SPECIFIED FOR R DIRECTION ENTER THE''/
+ ' CORRESPONDING FORCE OR DISPLACEMENT''/
+ ' IF DISPLACEMENT IS SPECIFIED FOR Z DIRECTION ENTER THE''/
+ ' CORRESPONDING FORCE OR DISPLACEMENT''/
GOTO 500

112 WRITE(OUT, 113)
113 FORMAT(// ' BOUNDARY CODE INDEX ****'//)
+ ' ENTER: '/
- ' 1. BOUNDARY FLAG AS SET IN GMESH'//
- ' 2. SELECT APPROPRIATE BOUNDARY CODE'//
+ ' 3. ASSOCIATED DISPLACEMENTS'//
+ ' 4. INCLINED BOUNDARY ANGLE IF SPECIFIED'//
+ ' IF DISPLACEMENT IS SPECIFIED FOR R DIRECTION ENTER THE''/
+ ' CORRESPONDING FORCE OR DISPLACEMENT''/
+ ' IF DISPLACEMENT IS SPECIFIED FOR Z DIRECTION ENTER THE''/
+ ' CORRESPONDING FORCE OR DISPLACEMENT''/
GOTO 500

115 WRITE(OUT, 120)
120 FORMAT(// ' OPTION SELECTION *****'//)
+ ' TYPE IN THE INDEX CORRESPONDING TO THE DESIRED OPTION'//
+ ' FOR FURTHER ASSISTANCE, CONSULT THE PROGRAMMERS GUIDE'//
GOTO 500

125 WRITE(OUT, 130)
130 FORMAT(/// ' NO HELP AVAILABLE ************' '///)
GOTO 500

135 WRITE(OUT, 140)
140 FORMAT(/// ' MODIFY OR VIEW DATA *****'///)
+ ' IF MODIFICATIONS ARE DESIRED TYPE Y FOR THE MODIFICATION'//
+ ' MENU TO BE DISPLAYED. SELECT OPTION TO BE MODIFIED BY TYPING'//
+ ' THE CORRESPONDING INDEX'///)
GOTO 500

500 CONTINUE
RETURN
END
APPENDIX B

SCRUBS CARD IMAGES
SCRUBS is a FORTRAN computer program designed to compute the failure, collapse, and resulting subsidence of geologic materials. The uniqueness of this program is its ability to model rubble formation and collapse in a continuum, as opposed to a discrete, sense. Both pre and post failure aspects of particular problems are treated. SCRUBS is a nonlinear finite element program which is capable of determining the deformation and state of stress in plane and axi-symmetric bodies. Pre and post failed material properties may be elastic-perfectly plastic, or elastic work hardening. Linear, quadratic or cubic isoparametric elements are used to represent the region and its boundary. Four different yield conditions can be imposed on the materials: von Mises, Tresca, Drucker-Prager, or Beltrami. Three different algorithms for solving the nonlinear discretized equations are available. They are: 1) a constant stiffness initial stress method, 2) a two step process where the stiffness matrix is updated on the second iteration of an otherwise initial stress process and 3) a regular tangent stiffness method. A check on force residuals is used to evaluate convergence for any load increment.

Specific files are created by SCRUBS. The three files that are read have the following names and functions.

QMESH9.DAT = Mesh information as created by QMESH.BYU
SCRUBS.DAT = SCRUBS input data
SCRUBS.RST = Restart file

The three files that are created have the following names and functions.

SCRUBS.MOV = Displacement, stress and strain data at each load step
RUBBLE.MOV = Element rubbleization plot file
SCRUBS.RST = Restart file
This is the standard QMESH.BYU renumbered file that contains the two dimensional finite element meshing information. It consists of five records written in blocked binary form. The content of these records is as follows.

Record 1. (8 words)
A comment, packed 4 characters per word. This comment is from the COMMENT card in the QMESH input.

Record 2. (4 words)
KKK, the number of elements in the mesh
NNN, the number of nodes in the mesh
NFF, the number of words in the boundary flag table
MAXDIF, the maximum difference of node numbers for any element in the renumbered mesh

Record 3. (2 x NNN words)
the lists of nodes, in renumbered order; that is
(X(N), Y(N), N=1, NNN)
or
(R(N), Z(N), N=1, NNN)

Record 4. (5 x KKK words)
The list of elements:
(N1(K), N2(K), N3(K), N4(K), MAT(K), K=1, KKK)
where N1(K) to N4(K) are the node numbers for the K-th element in counterclockwise order, and MAT(K) is the material number of the K-th element.

Record 5. (NFF words)
the list of boundary flags and nodes,
(IFLAG(I), I=1, NFF). Flags will be negated to distinguish them from nodes, and the corresponding node or list of nodes will follow each flag. If NFF=0, this record will not be written.
This file contains the data necessary to define
the problem to be analyzed. The groups of lines of
this file and the specified format of each line
follows.

GROUP 1  INTERNAL MESH GENERATION OR RESTART
SPECIFICATION

Line 1. (4I5) Geometry parameters

  1 - 5 Number of elements (NEL) **
  6 - 10 Number of nodes (NODES)
 11 - 15 Number of load cards (NUMPC)

** NOTE if NEL = 0, omit remaining lines of the group
and read total mesh from the input file QMESH9.DAT. If
NEL = -1, the problem is to be restarted and the
remaining lines of this group are omitted.

Line 2. (6I5) N,IX array

  1 - 5 Element number (N)
  6 - 10 Ith nodal point (IX array)
 10 - 15 Jth nodal point (IX array)
 16 - 20 Kth nodal point (IX array)
 21 - 25 Lth nodal point (IX array)
 26 - 30 Material number (MAT, IX array)

In general, every element must be defined; but
with the semi-automatic mesh generation feature, a
minimum of one element per row need be input. For
example, if element 10 is read with values I=12, J=13,
k=24, L=23, and MAT=1, and the next element is read is
element 15 with values I=23, J=24, K=35, L=34, and
MAT=1, then element 11 would be assigned values 13, 14,
25,24, and 1.

Line 3. (I5,F5.0,4F10.0)

  1 - 5 Nodal point number (N)
  6 - 10 Boundary condition code (CODE)
 11 - 20 Radial coordinate (R)
 21 - 30 Axial coordinate (Z)

In general, every nodal point must be defined, but
since the program has a semi-automatic mesh generation
feature, a minimum of two nodal points per row need be
input and the intervening points will be assigned
coordinates based on a linear interpolation procedure. For example, if nodal point 1 is the first point in a row with coordinates (2.5, 5.4), and nodal point 11 is the next point defined with coordinates (12.5, 10.4), then nodal point 2 will be located at (3.5, 5.9), etc.
GROUP 2  SOLUTION CONTROLS

Line 1.  Problem identification

1 - 5 Total number of problems to be solved in one run (NPROB)

Line 2.  Title (20A4)

1 - 72 Title of problem.

Line 3.  Control data (16I5)

1 - 5 Total number of nodes, NP
6 - 10 Total number of elements, NE
11 - 15 Total number of restrained boundary points, NB
16 - 20 Number of load cases/problem, NLD
21 - 25 Number of d.o.f. per node, NDF
26 - 30 Number of different materials, NMAT
31 - 35 Element type 1=linear, 2=parabolic, 3=cubic, NSTR
36 - 40 Number of gauss points for stiffness calculations NGAUS
41 - 45 Solution algorithm, NALGO
   0 = elasticity only
   1 = constant stiffness
   2 = two step process
   3 = tangent stiffness
46 - 50 Stress/strain type NPP
   0 = Plane strain
   1 = Plane stress
   2 = Axisymmetric problem
51 - 55 Yield condition parameter, NYIELD
   1 = Mises
   2 = Tresca
   3 = Drucker-Prager
   4 = Beltrami
56 - 60 Input stiffness control, NT
   0 = Number of elastically coupled nodes
   1 = input stiffness coefficients
61 - 65 Flag for Rubble calculation, NL
   0 = No rubble
   1 = rubble
66 - 70 Bandwidth (leave blank unless NT = 1)
71 - 75 Number of coordinates per node NCORD (default 2)
76 - 80 Number gauss points for nodal force residual calculation an stress storage (MGAUS) result even for elastic solution NGAUS defaults for NGAUS.
Line 10. Load increment data (I5)
   1 - 5 Number of load increments
      NINC

Line 11. Load factor data (16F5.3)
   1 - 5 Multiplier for increment 1     FAC(1)
   6 - 10 Multiplier for increment 2     FAC(2)
      etc.

Line 12. Output control data (16I5) NOUT(I), I=1,NIC
   1 - 5 s 0 t   s = 1st iteration write out
                  indicator
                  t = final iteration write
                  out indicator

   all printout is given up to and
   including s or t

   s or t = 0   no output
       = 1   displacements at nodal points
       = 2   reactions at constrained nodes as well
       = 3   stresses at gauss points as well
       = 4   residual forces at nodes as well

   6 - 10 NOUT(2)
   11 - 15 NOUT(3)
      etc.

Line 13. Iteration and convergence data (I10,Fl0.3)
   1 - 10 Maximum number of iterations
      NIT
   11 - 20 Convergence factor in percent
      CONFAC checks on 100 x(Sum of force residuals) divided by (Sum of
      applied forces)
GROUP 3 - MATERIAL PROPERTIES

Line 1. Material data (I10, 7F10.2) NMAT Lines,
Note, if NL = 1, card NMAT of this section must give unconsolidated rubble constants)

1 - 10 Material property number N
11 - 20 Young's modulus ORT(N,1)*
21 - 30 Poisson's ratio ORT(N,2)
31 - 40 Yield stress ORT(N,4)
41 - 50 Hardening modulus ORT(N,5)
51 - 60 Conical yield surface angle ORT(N,6)
61 - 70 Thickness (leave as zero if NPP = 1)

* A negative value in this position activates the low shear material description. In this case, the bulk modulus, K, is the absolute value of ORT(N,1), and the shear modulus, g, is the value of ORT(N,2)

Line 2. Tabulated plastic stress-strain data
(I10, F10.3)
(Only for materials with ORT(N,4) negative)

1 - 10 Number of tabulated strain points, NTAB
11 - 20 Strain increment in percent strain, TABSTN(N)
(If ORT(N,4) > 0 these values default to 2 and 100 respectively.)

Line 3. String of stress values, NTAB in number, one line required for each material (15F8.0)

1 - 8 Stress value for point 1
9 - 16 Stress value for point 2
17 - 24 Stress value for point 3
25 - 32 Stress value for point 4

.
.
.
.

etc.

Line 4. String of strain values, NTAB in number, one line required for each material (15F8.0)

1 - 8 Strain value for point 1
9 - 16 Strain value for point 2
17 - 24 Strain value for point 3
.
.
.
.

etc.
GROUP 4 RUBBLEIZATION DATA (Omit this group if NL=0)

Line 1. Rubble material data, one line required for each material (I10,2F10.3)

   1  - 10 Material number, N
   11 - 20 Bulking parameter, ORT(N,7)
   21 - 30 Failure stress ORT(N,8)

Line 2. Minimum boundary coordinate (E10.3)

   1 - 10 Value of minimum Z coordinate for which element nodal points cannot pass through

Line 3. Material failure flag (I5)

   1 - 10 Failure flag, NRFF
   If this flag is non zero, a subroutine, READF is called. This subroutine defines a list of elements that can be removed (i.e. mined) on a particular increment. NRFF is the total number of increments allowed (10 max).

Line 4. Element mining data, omit if NRFF=0. (16I5)

   1 - 5 Number of elements to be mined on increment 1, NME(1)
   6 - 10 Number of elements to be mined on increment 2, NME(2)
   11 - 15 Number of elements to be mined on increment 3, NME(3)
   16 - 20 Number of elements to be mined on increment 4, NME(4)

   etc.

Note: Currently 16 is the maximum number of elements that can be mined on any given increment.

Line 5. Specified elements to be mined, one line required for increment, NRFF total, omit if NRFF=0. (16I5)

   1 - 5 Element to be mined on this increment
   6 - 10 Element to be mined on this increment
   11 - 15 Element to be mined on this increment
   16 - 20 Element to be mined on this increment

   etc.
GROUP 5 BOUNDARY CONDITIONS

Line 1. Ancillary control data (16I5)
   1 - 5 Not used - leave blank
   6 - 10 Not used - leave blank
   11 - 15 Number of boundary points NNBB
   16 - 20 Number of reactions to be summed MPR
   21 - 80 Indexing vector, automatic generation of NOPE array
       IB(I), I=1,12

Line 2. Boundary conditions (2I5,4F10.3)
       NNBE cards in ascending order of boundary nodes.
       1 - 5 Boundary node number NBC(I)
           9 u condition, 0=free, 1=fixed
           10 v condition, 0=free, 1=fixed
       11 - 20 Prescribed u displacement US(I,1)
       21 - 30 Prescribed v displacement US(I,2)
       31 - 40 Boundary angle (if inclined)
           ANG(I) in degrees of X' axes from X. Positive
counterclockwise.

Line 3. Summed reaction nodes (16I5)
       (If MPR>0, otherwise omit)
       1 - 5 Node number
       6 - 10 Node number
       NOPR(I), I = 1,MPR
       11 - 15 Node number
           ...
           ...
           ...
etc.
GROUP 6  LOADING DEFINITION

Line 1.  Ancillary load control data (2I5)
    1 -  5 Initial stress counter ISTS
    6 - 10 Increment control (Note
        LDTYPE applies to both prescribed
        loads and displacements.)
        0 ... equal increments
        1 ... increments in proportion
            to total according to the
            factor array
            \{x\} = FAC \{x\}
            inc inc total
        2 ... independent set of loads
            for each increment

Line 2.  Loading title (20A4)

Line 3.  Loading control (8I5)
    1 -  5 NSTRS (Not used)
    6 - 10 Number of nodes where point loads
        are applied, NRE
    11 - 15 Gravity load flag, if nonzero read
        lines 5 and 6.
    16 - 20 Pressure load flag, if nonzero read
        lines 7, 8 and 9.
    21 - 25 Temperature load flag, currently not
        available.
    26 - 30 Centrifugal force flag, if nonzero
        read lines 5 and 6.
Line 4. External Point load data (I10,3F10.3)
NRE lines
  1 - 10  Node number
  11 - 20  Force in the X direction
  21 - 30  Force in the Y direction

Line 5. Gravity and centrifugal force data (3F10.3)
two or three lines if NRC + NRG > 0. Otherwise go to line 7.
  1 - 10  Angle of Y or Z axes from gravity axes, clockwise positive. THEDA
  11 - 20  Number of g's applied for system (when NRC = 1) GRAV
  21 - 30  Angular velocity in radians/unit time (only when NRC = 1) ANGVEL

Line 6. Density (8F10.3)
  1 - 10  Density for material 1  DENS(1)
  11 - 20  Density for material 2  DENS(2)
  
  etc. (Density units are weight/volume)

Line 7. Pressure element control, if no pressure loads
i.e. NRPRS=0 leave out and go to line 10.
  1 - 5  Number of line elements on which pressures are applied LNE

Note each line 8 and 9 should be in sequence
i.e., 8,1
      9,1
      8,2
      9,2
      etc.
Line 8. Element nodes with applied pressure, one line/element (4I5)

1 - 5 node 1
6 - 10 Node 2
11 - 15 Node 3
16 - 20 Node 4

NOPL(I), I=1, NSFR+1

Node numbers on element pressure surfaces numbered in counterclockwise direction.

Line 9. Pressure data (8F10.3)

(a) same loads on each node of line
1 - 10 Normal pressure
11 - 20 Tangential pressure
21 - 30

(b) Different loads at each node of line element Pn
1 - 10 Pn Node 1
11 - 20 Pt
21 - 30 Node 2
31 - 40

e tc.

Return to 8 for next line element.

Note (a) does not apply for linear elements
GROUP 7 REDUCED PRINTOUT CONTROLS

Line 1. Print control (2I5)
   1 - 5 Number of elements to be printed, 
      If zero, all elements printed.
   6 - 10 Nodal point output control, if zero, 
      no nodes printed, if 1, all nodes 
      printed.

Line 2. List of elements to be printed (16I5) 
   (only if NELP > 0) 
   1 - 5 element number to be output 
   6 - 10 element number to be output 
   11 - 15 element number to be output 
     . 
     . 
     . etc.
This file is written in blocked binary form as follows:

Record 1. (22 words)

the first twenty words are the title of the problem and
the last two words are NE, the total number of
elements, NP, the total number of nodal points, that
is,

\[ \text{TITLE,NE,NP} \]

Record 2. (2xNP+5xNE words)

the first two lists are the nodal point coordinates,
the next is the element connectivity, and the last is
the element type.

\[
\begin{align*}
(&\text{CORD}(I,1), I=1, NP), (\text{CORD}(I,2), I=1, NP), \\
(&\text{NOP}(4*(K-1)+M), K=1, NE), M=1, 4), \\
(&\text{IMAT}(I), I=1, NE)
\end{align*}
\]

The following records are repeated for each load
increment.

Record 1'. (1 word)

Increment number, TINC.

Record 2'. (3x2xNP words)

Nodal point displacements, repeated 3 times.

\[
\begin{align*}
(&\text{TDIS}(1,I), \text{TDIS}(2,I), I=1, NP), \\
(&\text{TDIS}(1,I), \text{TDIS}(2,I), I=1, NP), \\
(&\text{TDIS}(1,I), \text{TDIS}(2,I), I=1, NP)
\end{align*}
\]

Record 3'. (4xNE words)

Average element stresses

\[
\begin{align*}
(&\text{SPLT}(I,J), J=1, NE), I=1, 4)
\end{align*}
\]

Record 4'. (4xNE words)

Average element strains

\[
\begin{align*}
(&\text{SPLT}(I,J), J=1, NE), I=1, 4)
\end{align*}
\]
RUBBLE.MOV

The records of this file are written to draw rubbleized groups of subelements to effectively display a failed region.

Record 1. (2 words, E15.5, I5)
Increment number, TT, and total number of parts, ICNT.

Record 2. (4 words, 4I5)
Number of "parts", NONE; number of nodal points, NUMNP; number of unfailed elements, NP2; number of entries in connectivity array, NCON.

Record 3. (2 words, 2I5)
Beginning element number, NP1; Ending element number, NP2.

Record 4. (3xNUMNP words, 6e12.5)
Nodal point coordinates (CORD(I,1), CORD(I,2), 0.0), I=1, NUMNP

Record 5. (4xNUMEL WORDS, 16I5)
connectivity array (IXAR(I), I=1, NCON)
The above records provide geometry data for the original mesh. The following records are written, for each and every failed or rubble element, to describe the rubbleization as a "rubble" region.

Record 1'. (4 words, 4I5)
The variables NONE, NJ, NPT and NCON. These standard descriptors for the group of rubble subelements as used to describe a rubble region (see below). NONE is the number of "parts" for this group (i.e. 1), NJ is the number nodes (i.e. 12), NPT is the number of elements, (i.e. 5), and NCON is the number of entries in the connectivity (i.e. 20).
Record 2'. (2 words, 2I5)

The variables NPl and NP2 which are the smallest and largest element numbers associated with the subelement group.

Record 3'. (36 words, 6E12.5)

the coordinates of the subelement group.
(RN(J),ZN(J),0.0), J=1,12)

Record 4'. (20 words, 16I5)

the connectivity of the subelement group
(IXN(J), J=1,20)
This file is the restart file that is used to both restart from and write subsequent restart files. A file is automatically written after each converged increment. To restart a run, the following lines of SCRUBS.DAT, along with the proper SCRUBS.RST file, are all that are required.

1. Group 1, line 1.
2. Group 3, all lines.
APPENDIX C

SAMPLE SCRUBS FILES
### Old Ben SIGIZ Failure, Low Shear Soil

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 2e00e+06 22000</td>
<td>50 000</td>
<td>10 000</td>
<td>0 7900</td>
<td>.00000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20 2e00e+07 22000</td>
<td>50 000</td>
<td>10 000</td>
<td>0 7900</td>
<td>.00000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30 4e00e+07 22000</td>
<td>50 000</td>
<td>10 000</td>
<td>0 7900</td>
<td>.00000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>40 5e00e+06 22000</td>
<td>50 000</td>
<td>10 000</td>
<td>0 7900</td>
<td>.00000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50 2e00e+06 22000</td>
<td>50 000</td>
<td>10 000</td>
<td>0 7900</td>
<td>.00000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-1000</td>
<td>10 000</td>
<td>50 000</td>
<td>10 000</td>
<td>0 7900</td>
<td>.00000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70 6e00e+05 22000</td>
<td>50 000</td>
<td>10 000</td>
<td>0 7900</td>
<td>.00000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80 4e00e+06 22000</td>
<td>50 000</td>
<td>10 000</td>
<td>0 7900</td>
<td>.00000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90 2e00e+05 22000</td>
<td>50 000</td>
<td>10 000</td>
<td>0 7900</td>
<td>.00000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 1000</td>
<td>.45000</td>
<td>50 000</td>
<td>10 000</td>
<td>0 7900</td>
<td>.00000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-200</td>
<td>.50 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-400</td>
<td>.99200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-500</td>
<td>.99200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-225</td>
<td>.99200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-190</td>
<td>.99200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-5 000</td>
<td>1 0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-50 00</td>
<td>1 0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-50 00</td>
<td>1 0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-100 0</td>
<td>1 0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10-100 0</td>
<td>1 0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-7524</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>74</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>11 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>10 0 000e+00 0.000e+00 0 000e+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>146</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>147</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>148</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>149</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>152</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>157</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>164</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>165</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>167</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>168</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>169</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>171</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>173</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>174</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>175</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>176</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>177</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>178</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>179</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>181</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>182</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>183</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>184</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>185</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>186</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>187</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>188</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>189</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>191</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>193</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>194</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>195</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>196</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>197</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>198</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>199</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>202</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>204</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>205</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>207</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>208</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>209</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>211</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>212</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>213</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>214</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>215</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>216</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>217</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>218</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>219</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>220</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>221</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>222</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>223</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>224</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>225</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>226</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>227</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>228</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>229</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>230</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>231</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>232</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>233</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>234</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>235</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>236</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>237</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>238</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>239</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>241</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>242</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>243</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>244</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>245</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>246</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>247</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>248</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>249</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>251</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>252</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>253</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>254</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>255</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OLD BEN MINING SEQUENCE

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0.0000E+01 0000 000000E+00
33000E-01 63300E-01 83300E-01 83300E-01 83300E-01 83300E-01
97200E-01 97200E-01

3

3

2512 000

0

12 1

439 439 430 403 404 405 418 419 420 433 434 435

204
ULD GEN SIG-ZZ FAILURE, LOW SHEAR SOIL

CONTROL PARAMETERS

1  TITLE
2  NUMBER OF PROBLEMS------ ----- ----  1
3  NUMBER OF LOAD CASES------------------  1
4  NUMBER OF DEGREES OF FREEDOM----  2
5  NUMBER OF MATERIALS------------------  10
6  ELEMENT TYPE---------------------  1
7  NUMBER OF GAUSS POINTS--------  2
8  SOLUTION ALGORITHM CONTROL------  3
9  STR/STN CONTROL-------------------  0
10 YIELD CONDITION CONTROL----------  3
11 ARBITRARY STIFF. CONTROL-------  0
12 RUBBLE FLAG CONTROL-----------  1
13 BAND WIDTH (NOT NEEDED)--------  0
14 NUMBER OF COORDINATES---------  0
15 GAUSS PTS FOR RES. CALC.-------  0
16 RE-ENTER DATA SET
17 END MODIFICATIONS

MATERIAL PROPERTIES

<table>
<thead>
<tr>
<th>MAT #</th>
<th>YOUNG'S MODULUS</th>
<th>POISSON'S RATIO</th>
<th>YIELD STRESS</th>
<th>HARDENING MODULUS</th>
<th>YIELD ANGLE</th>
<th>MATERIAL THICKNESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2600E-06</td>
<td>.22000</td>
<td>50.000</td>
<td>10.00</td>
<td>.79000</td>
<td>.00000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.2000E-07</td>
<td>.22000</td>
<td>50.000</td>
<td>10.00</td>
<td>.79000</td>
<td>.00000E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.4000E-07</td>
<td>.22000</td>
<td>50.000</td>
<td>10.00</td>
<td>.79000</td>
<td>.00000E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.6000E-06</td>
<td>.22000</td>
<td>50.000</td>
<td>10.00</td>
<td>.79000</td>
<td>.00000E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.2000E+06</td>
<td>.22000</td>
<td>50.000</td>
<td>10.00</td>
<td>.79000</td>
<td>.00000E+00</td>
</tr>
<tr>
<td>6</td>
<td>-1000.</td>
<td>10.000</td>
<td>50.000</td>
<td>10.00</td>
<td>.79000</td>
<td>.00000E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.4000E+05</td>
<td>.22000</td>
<td>50.000</td>
<td>10.00</td>
<td>.79000</td>
<td>.00000E+00</td>
</tr>
<tr>
<td>8</td>
<td>0.4000E+06</td>
<td>.22000</td>
<td>50.000</td>
<td>10.00</td>
<td>.79000</td>
<td>.00000E+00</td>
</tr>
<tr>
<td>9</td>
<td>0.2000E+06</td>
<td>.22000</td>
<td>50.000</td>
<td>10.00</td>
<td>.79000</td>
<td>.00000E+00</td>
</tr>
<tr>
<td>10</td>
<td>1000.</td>
<td>.45000</td>
<td>50.000</td>
<td>10.00</td>
<td>.79000</td>
<td>.00000E+00</td>
</tr>
</tbody>
</table>
************ RUBBLIZATION DATA ************

<table>
<thead>
<tr>
<th>MATERIAL</th>
<th>FAILURE STRESS</th>
<th>BULKING PARAMETER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-200.0</td>
<td>50.000</td>
</tr>
<tr>
<td>2</td>
<td>-400.0</td>
<td>.99200</td>
</tr>
<tr>
<td>3</td>
<td>-300.0</td>
<td>.99200</td>
</tr>
<tr>
<td>4</td>
<td>-225.0</td>
<td>.99200</td>
</tr>
<tr>
<td>5</td>
<td>-180.0</td>
<td>1.0000</td>
</tr>
<tr>
<td>6</td>
<td>-5.000</td>
<td>1.0000</td>
</tr>
<tr>
<td>7</td>
<td>-50.000</td>
<td>1.0000</td>
</tr>
<tr>
<td>8</td>
<td>-50.000</td>
<td>1.0000</td>
</tr>
<tr>
<td>9</td>
<td>-100.0</td>
<td>1.0000</td>
</tr>
<tr>
<td>10</td>
<td>-100.0</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

MINIMUM Z COORDINATE: -7524

************ ELEMENTAL MINING DATA ************

TOTAL NUMBER OF INCREMENTS: 2

************ ELEMENTAL MINING DATA ************

<table>
<thead>
<tr>
<th>INC</th>
<th>ELEMENTS</th>
<th>ELEMENTS TO BE MINED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>11 12 13 14 15</td>
</tr>
</tbody>
</table>

************ BOUNDARY CONDITIONS ************

NUMBER OF BOUNDARY POINTS: 74

<table>
<thead>
<tr>
<th>NODE INDEX</th>
<th>NODE NUMBER</th>
<th>BOUNDARY CODE</th>
<th>DISPLACEMENT R DIR.</th>
<th>DISPLACEMENT Z DIR.</th>
<th>BOUNDARY ANGLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>11</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
<td>0.0000DE+00</td>
</tr>
</tbody>
</table>
NUMBER OF SUMMED REACTION NODES = 0

INDEX NODES

LOADING DESCRIPTION

LOADING PROBLEM 1
OLD BEN MINING SEQUENCE
INCREMENT CONTROL = PROPORTIONAL TO LOAD FACTORS
CONSIDER POINT LOADS = NO
CONSIDER GRAVITY LOADS = YES
CONSIDER PRESSURE LOADS = NO
CONSIDER CENTRIFUGAL FORCES = NO

LOADING CONSTANTS

GRAVITATIONAL ANGLE = 0.0000E+00
GRAVITATIONAL CONSTANT = 1.0000

MATERIAL DENSITIES

<table>
<thead>
<tr>
<th>MATERIAL</th>
<th>DENSITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32400E-01</td>
</tr>
<tr>
<td>2</td>
<td>90300E-01</td>
</tr>
<tr>
<td>3</td>
<td>97200E-01</td>
</tr>
<tr>
<td>4</td>
<td>83300E-01</td>
</tr>
<tr>
<td>5</td>
<td>83300E-01</td>
</tr>
<tr>
<td>6</td>
<td>55000E-01</td>
</tr>
<tr>
<td>7</td>
<td>83300E-01</td>
</tr>
</tbody>
</table>
9 0.97200E-01
9 0.83300E-01
10 0.83300E-01

++++++++++++++++++++ LOAD FACTOR DATA +++++++++++++++++++++

NUMBER OF LOAD INCREMENTS = 2

INCREMENT     FACTOR
1              1.000
2              0.000

++++++++++++++++++++ OUTPUT CONTROL DATA +++++++++++++++++++++

INCREMENT     CONTROL NUMBER
1              3
2              3

++++++++++++++++++++ MAX ITERATION AND CONVERGENCE FACTOR +++++

MAXIMUM NUMBER OF ITERATIONS = 25
CONVERGENCE FACTOR = 12.00

++++++++++++++++++++ IMPACT NODAL DATA +++++++++++++++++++++

NUMBER OF BOUNDARY IMPACT NODES = 0

INDEX NODE

++++++++++++++++++++ PRINTOUT CONTROLS +++++++++++++++++++++

NODAL OUTPUT (0=ALL NODES, 1=NO NODES) --- 1
NUMBER OF ELEMENTS TO BE PRINTED --- 12

INDEX PRINT NODES
1 398
2 353
APPENDIX D

QUARTER POINT ALGORITHM LISTING
**GEOMETRICAL DATA, BOUNDARY CONDITIONS AND GAUSSIAN INTEGRATION DATA**

```fortran
READ(5, 1) NNP, NSING, NNBB, MPR, (IB(I), I=1, 12)
IF(NB.GT.0 .AND. NNBB.EQ.0 ) NNBB=NB
WRITE(6, 101) NNP, NNBB, MPR, (IB(I), I=1, 12)
IF(IB(1).NE.0 ) GO TO 9
DO 6 I=1, 12
6 IB(I)=NSFR
IF(NSFR-2).GT.0 .7
C
7 IB(5)=IB(11)=IB(12)=1
7 IB(5)=1
IB(6)=1
IB(11)=1
IB(12)=1
GO TO 9
C
8 IB(4)=IB(8)=1
8 IB(4)=1
IB(8)=1
9 CONTINUE
DO 10 II=1, NNP
DO 10 JJ=1, NCORD
10 CORD(II, JJ) = 0.

READ NODAL POINT DATA
READ(9) (CORD(N, 1), CORD(N, 2), N=1, NNP)

READ ELEMENT DATA
IF (NCN .EQ. 8) GO TO 445
READ(9)((NOP(NCN*(K-1)+M), M=1, NCN), IMAT(K)), K=1, NE)
GO TO 446

445 READ(9):(NOP(B*(K-1)+1), NOP(B*(K-1)+3), NOP(B*(K-1)+5),
* NOP(B*(K-1)+7), NOP(B*(K-1)+2), NOP(B*(K-1)+4),
* NOP(B*(K-1)+6), NOP(B*(K-1)+8), IMAT(K)), K=1, NE)
IF(NSING.EQ.0 ) GO TO 446
DO 100 K=1, NE
DO 11 I=1, B
ICK=NOP((K-1)+I-1)
11 CONTINUE
QOTO 100
20 IFP=I-2
IF(IFP.EQ.-1) IFP=7
IF(IFP.EQ.0) IFP=8
NFAR=NOP((K-1)*B+IFP)
XQP=0.25*CORD(NFAR, 1)+0.75*CORD(NSING, 1)
ZQP=0.25*CORD(NFAR, 2)-0.75*CORD(NSING, 2)
IGP=I-1
IF(IGP.EQ.0) IGP=8
NGP=NOP((K-1)*B+IGP)
CORD(NGP, 1)=XQP
CORD(NGP, 2)=ZQP
IFP=I+2
IF(IFP.EQ.9) IFP=1
IF(IFP.EQ.10) IFP=2
NFAR=NOP((K-1)*B+IFP)
XQP=0.25*CORD(NFAR, 1)+0.75*CORD(NSING, 1)
ZQP=0.25*CORD(NFAR, 2)-0.75*CORD(NSING, 2)
IGP=I-1
IF(IGP.EQ.9) IGP=8
NGP=NOP((K-1)*B+IGP)
CORD(NGP, 1)=XQP
CORD(NGP, 2)=ZQP
100 CONTINUE
446 CONTINUE
```
SCRUBS.BYU A TWO DIMENSIONAL FINITE ELEMENT PACKAGE FOR CONTINUUM ANALYSIS USING QUADRATIC ISOPARAMETRIC ELEMENTS

Michael Glenn Long
Department of Civil Engineering
M.S. Degree, April 1983

ABSTRACT

This thesis develops a two-dimensional, axisymmetric finite element package for solving continuum problems including rubbleization subsidence and nonlinear fracture mechanics. This package includes both a user-friendly preprocessor, PRESCRUBS.BYU, and a versatile analysis code SCRUBS.BYU. PRESCRUBS.BYU systematically creates the data file necessary to run SCRUBS.BYU. SCRUBS.BYU provides many options of nonlinear static analysis using either linear or quadratic isoparametric finite elements. Sample problems are presented that demonstrate the capabilities of this package.

COMMITTEE APPROVAL:

Steven E. Benzley
Committee Chairman

Cliff S. Barton
Committee Member

Henry N. Christiansen
Department Chairman