1984-12-01

Application of Combined Jointed Media and Discrete Slip Plane Characteristics to Subsidence Predictions

David W. Basinger
Brigham Young University - Provo

BYU ScholarsArchive Citation
Basinger, David W., "Application of Combined Jointed Media and Discrete Slip Plane Characteristics to Subsidence Predictions" (1984). *Theses and Dissertations*. 3450.
https://scholarsarchive.byu.edu/etd/3450

APPLICATION OF COMBINED JOINTED MEDIA AND DISCRETE SLIP PLANE CHARACTERISTICS TO SUBSIDENCE PREDICTIONS

DAVID W. BASINGER

APPLICATION OF COMBINED JOINTED MEDIA AND DISCRETE SLIP PLANE CHARACTERISTICS TO SUBSIDENCE PREDICTIONS

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
David W. Basinger
December 1984

This Thesis, by David W. Basinger, 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

Steven E. Benzley
Committee Chairman

Reese J. Goodwin

Reese J. Goodwin
Committee Member

4 SEPTEMBER 84

Date

Henry N. Christiansen

Henry N. Christiansen
Department Chairman

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I wish to express my appreciation to Dr. Steven E. Benzley for his patient help in the preparation of this thesis, for providing this research opportunity, and for his expert advice. Appreciation goes also to Royd R. Nelson and Randy Nish, who helped me get started, and to the Brigham Young University Department of Civil Engineering for providing software, hardware, graphical aids, and moral support. Above all, love, appreciation and thanks go to the my wife, Janette, for her patience, understanding and overwhelming confidence through many long hours.

CHAPTER ONE

INTRODUCTION

Ground settlement, also called subsidence, has increased in significance as the use of longwall mining and underground coal gasification have become more prominent. Many methods have been used to predict the amount of subsidence that may occur. These include empirical methods, physical models, and numerical models. This thesis investigates the use of finite element material modeling to incorporate slippage planes in the analysis of subsidence problems.

Chapter two of this thesis reviews current research in finite-element numerical simulations of subsidence predictions. The area emphasized is the ability to incorporate fractures and joints in rock mechanics problems.

In chapter three, the theory behind two methods of modeling fractures in finite element technique is presented. Low shear theory is presented as the first method. Among the items discussed in low shear theory are: (1) the incorporation of low shear slip planes into the model, (2) the formulation of the low shear, and (3) the installation of a low shear element row into the model. The second method used to modeling fractures is to model the material as a jointed media continuum. In this section, the topics are: (1) the approximation of the media as a continuum, (2) the formulation of a constitutive model, (3) the assumptions of model behavior, and (4) the solution of the constitutive equations. Three basic assumptions are included in the jointed media formulation. These involve: (1) the behavior of intact material, (2) the neglection of joint dilatation, and (3) the behavior of joint shear.

In chapter four, various models used to predict subsidence in a coal mine problem are shown. First, a physical centrifuge model carried out at Sandia National Laboratories is described in detail. The results of this model are also given. Second, several numerical models set up and analyzed using the SCRUBS.BYU finite element program are described. Included in the descriptions are: (1) the equipment used, (2) the construction of the model, (3) the dimensions of the model, (4) the materials used, (5) the creation of the subsurface cavity, and (6) the results of selected runs. Finally, comparisons are made between the models.

In chapter five, conclusions are made and recommendations listed. The successes, the problems, and the proposed solution to these problems are presented.

In the attached appendices, a current listing of the program SCRUBS.BYU is provided. Also included are a subroutine map, a short description of each subroutine, additional results of simulation runs, and sample files used in the simulations.

CHAPTER TWO

SUBSIDENCE PREDICTION

Background Information

The collapse of the earth at the surface, called surface subsidence, can seriously effect ground water, surface hydrology, and surface structures. The mining of natural resources, especially of coal, can result in significant amounts of subsidence that in turn can cause significant damage.

Consequently, accurate subsidence prediction is of great interest to mining engineers. Recently, increased use of longwall mining techniques and increased research in underground coal gasification have heightened the concern in predicting the behavior of rocks and soils subject to mining stresses [17]. Longwall mining and in-situ (underground) coal gasification are often preferred as methods that maximize resource removal, as both of these methods have the result that all the coal in a seam is removed. The result is, of course, the total collapse of the area above the mined region and the resulting subsidence of all overlying strata. This is in contrast to the older mining method of room and pillar mining, where pillars of coal were left behind to support the mine roof.

Methods of Prediction

Historical methods. Empirical methods have historically been used to predict surface subsidence [3,14]. These methods are based mostly on engineering judgement and require a large data base from which to derive the design formulas. In Europe, where empirical methods have long been used, many records exist that provide a long history from which to gather a sufficient data base. In the United States such a data base does not usually exit and alternative methods are sought. Methods that have been investigated include both physical and numerical models. It has been shown that mutual enhancement can occur when both physical and numerical models are applied to subsidence problems [19]. In this thesis, physical and numerical methods are applied to a subsidence problem in an effort to obtain the additional understanding that this mutual enhancement can afford.

Physical methods. A centrifuge can be used to physically simulate mining subsidence. Subsidence above a mine is caused primarily by the force of gravity. Gravity interacting with the mine geometry, the material properties of the rock and soil, and the natural ground movement of the area, creates a stress field in the overburden. For an accurate model, the stress field in the full-size structure must be duplicated in the reduced-scale model. To do that, the force of gravity can be increased in the same proportion that the model has been reduced in scale. A centrifuge can be used to apply these greater-than-gravity loads to the scaled model. For example, a coal mine at a depth of 100 m below ground level can be simulated by a scale model only 1 m high and constructed from representative site materials by loading the scaled model in a centrifuge to 100 times the acceleration of gravity (g's).

Numerical methods. Numerical models have increased in use with the onset of the computer, and the corresponding increase in computational power, speed, and accuracy. Many numerical approaches to subsidence prediction have been used [1,7,13,22,23]. One important numerical technique is called the finite element technique.

The finite element technique is a numerical method that has been adapted to modeling geologic materials. The nonhomogeneity of soil and rocks make their modeling intrinsically complicated and uncertain. Material properties are often vague. The presence of cracks and fissures increase the complexity and difficulty of predicting the behavior of geologic materials. As a result, numerical simulations of geologic materials have almost necessarily involved major idealizations and simplifications involving the material properties and the effect of fractures and joints on subsidence mechanics.

While accurately determining soil properties will always be a problem, determining the location of cracks and joints and incorporating their effects into a numerical model leads to better and more accurate predictions of soil and rock mechanics problems. Many finite element codes have incorporated joints and fractures into their analysis by adding sliding interfaces or slip planes, with the usual result that computation time has increased dramatically [16]. At Brigham Young University, several unique geologic modeling techniques have been included in a finite element program called SCRUBS.BYU.

Current Research

The computer code SCRUBS.BYU. SCRUBS.BYU is a two-dimensional, elastic-plastic finite element computer program developed to analyze resource extraction problems resulting in significant surface subsidence [9]. SCRUBS.BYU can model the subsidence process associated with either longwall coal mining or in-situ coal gasification. Of special interest in SCRUBS.BYU'S subsidence formulations are the abilities to model failure and rubbleization, to mine elements, and to incorporate joints and cracks.

Fracture modeling. In 1983 Royd Nelson, a master's candidate at Brigham Young University, incorporated joints, cracks, and fractures into SCRUBS.BYU [16]. Two separate crack definitions were used. In the first, a discrete crack model was used, and in the other a continuum approach defining a new material with regularly spaced joints was used. These capabilities were demonstrated by several simplified examples.

Recommendations for future study. Further study in this area could be made. Nelson specifically recommended that areas of future work include [16, p.56]:

1) Studying the effects of faults on a variety of geomechanical problems using the slip plane addition to model the faults.
2) Evaluating the effect of joints and fractures on problems where the rubble model available in SCRUBS.BYU is utilized.
3) Investigating plastic yielding of the low shear slip plane material, particularly with utilization of the Drucker- Prager failure criteria.

4) Performing experimental studies to determine single-joint slip responses in order to more accurately estimate the material constants that define the slip response.

5) Performing of large-scale tests to verify analytical calculations.

To some extent this thesis presents further study in each of the recommended areas except number 3. The investigation of plastic yielding and the Drucker-Prager failure criteria was deemed to be beyond the scope of this study.

The Scope of this thesis. This thesis is a continuation of Nelson's work to incorporate fracture modeling into the finite element code SCRUBS.BYU. Verification of the discrete slip plane and jointed media formulations is sought by their application to a physical model. In addition, the effort to combine discrete slip planes and jointed media continua in models will increase the understanding of numerical geologic materials modeling.

CHAPTER THREE

LOW SHEAR AND JOINTED MEDIA THEORY

Low Shear Theory

Modeling with low shear element slip planes. The modeling of joints and fractures has been incorporated into many finite element codes by using sliding interfaces or slip planes. The implementation of these interfaces into the codes involves complex formulations and often increases the computing time of these programs considerably. A slip plane formulation is part of the finite element computer code SCRUBS.BYU. In this program a "low shear element" slip plane was formulated. The low shear element slip plane has the advantages of being easy to implement and has a limited effect on the run time of the program. In this method, a thin row of elements is added into the model wherever a slip plane is desired, as shown in Figure 1. This thin row of elements is then defined as a low shear material. This formulation simply imposes regions of minimal shear resistance along the slip plane element row.

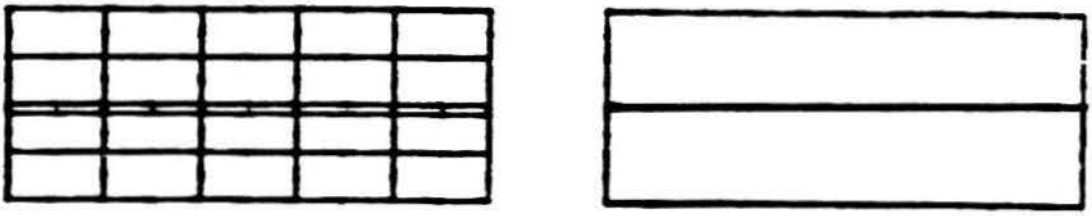


Figure 1. Low shear slip plane addition.

The low shear formulation.

> "In standard plane strain computer programs the relationship between stresses and strains is expressed as a generalized form of Hooke's law,
>
> $$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & 0 \\ \nu & (1-\nu) & 0 \\ 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix}$$
>
> Problems occur in this formulation. When Poisson's ratio becomes .5, division by zero occurs. Another disadvantage for a material that has little shear strength is that as the shear modulus G approaches zero, the modulus of Elasticity E also approaches zero and the equations for stress and strain break down.
>
> In a manner similar to Clough and Woodward [4] we can circumvent the above problems by writing the stress strain relations in terms of the bulk modulus (K) and the shear modulus (G). Doing this,
>
> $$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = \begin{bmatrix} K+\frac{4}{3}G & K-\frac{2}{3}G & 0 \\ K-\frac{2}{3}G & K+\frac{4}{3}G & 0 \\ 0 & 0 & G \end{bmatrix} \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix}$$
>
> where
>
> $$K=\frac{E}{3(1-2\nu)}$$
>
> $$G=\frac{E}{2(1+\nu)}$$
>
> The above equation allows G to approach zero and still maintain a valid stress strain relationship." [2]

This formulation is incorporated in the SCRUBS.BYU program by entering the bulk modulus K as a negative number in place of Young's modulus, and by entering the shear modulus G in place of Poisson's ratio. The negative sign on the bulk modulus is a flag that causes the elasticity matrix to be created using equation 3.2 rather than equation 3.1. To model a low shear material, the shear modulus is entered as a number low relative to the bulk modulus. As stated before, this formulation is not limited to use in low shear materials, but can be utilized any time the material is described by K and G rather than by E and ν.

PRESCRUBS.BYU provides the capability to input low shear slip planes. A solution control parameter is required that indicates whether any slip planes exist. If so, the user is automatically prompted for the bulk and shear moduli for the slip plane material. Note that this formulation requires a separate boundary flag for each discrete slip plane.

Addition of the slip plane element row. The geometric modeling is done on an interactive mesh generator called QMESH.BYU. This program accepts the perimeter data for specified regions of the geometry, and automatically creates and numbers the finite element mesh. Boundary conditions are imposed by assigning flags to desired lines or line segments that can later (in PRESCRUBS.BYU) be coupled to specified displacement or force criteria. To define low shear element rows as described earlier, the user would have to enter perimeter data for each row. The SCRUBS.BYU program automatically creates this thin row of elements. Any line can be defined as a slip plane by simply assigning it a boundary flag in QMESH.BYU, thus saving considerable preprocessing time. (Readers not familiar with the use of QMESH.BYU should consult Reference 8.)

Jointed Media Theory

Modeling as jointed media. A second method of modeling the effects of cracks and fractures in geologic materials is by using a material model for the jointed material based on continuum theory. In this method, joints in the materials are implemented by allowing for the effects of jointing on the material response rather than discrete models for each joint. This formulation is taken from work done by Thomas [21], but is modified neglecting joint dilatation response, and limited to only two dimensional or axisymmetric continua.

The model is composed of two parts: (1) a continuum approximation in which the joint displacements are averaged through the material, and (2) a continuum description based on linear behavior of the base material and nonlinear shear at the joints.

The continuum approximation. The continuum model used for this material is based on the published work of Moreland [10,11,12]. Consider a "representative elementary volume" containing regularly spaced parallel fractures as sketched in Figure 2. The orientation of this joint set is characterized by a unit normal vector $\underset{\sim}{n}$ with respect to fixed x_1, x_2, x_3 coordinate axes. The spacing between fractures is denoted by δ. Additional unit vectors $\underset{\sim}{s}$ and $\underset{\sim}{t}$ in the plane of the joints are introduced such that $\underset{\sim}{s}$, $\underset{\sim}{t}$, and $\underset{\sim}{n}$ form a local Cartesian coordinate system.

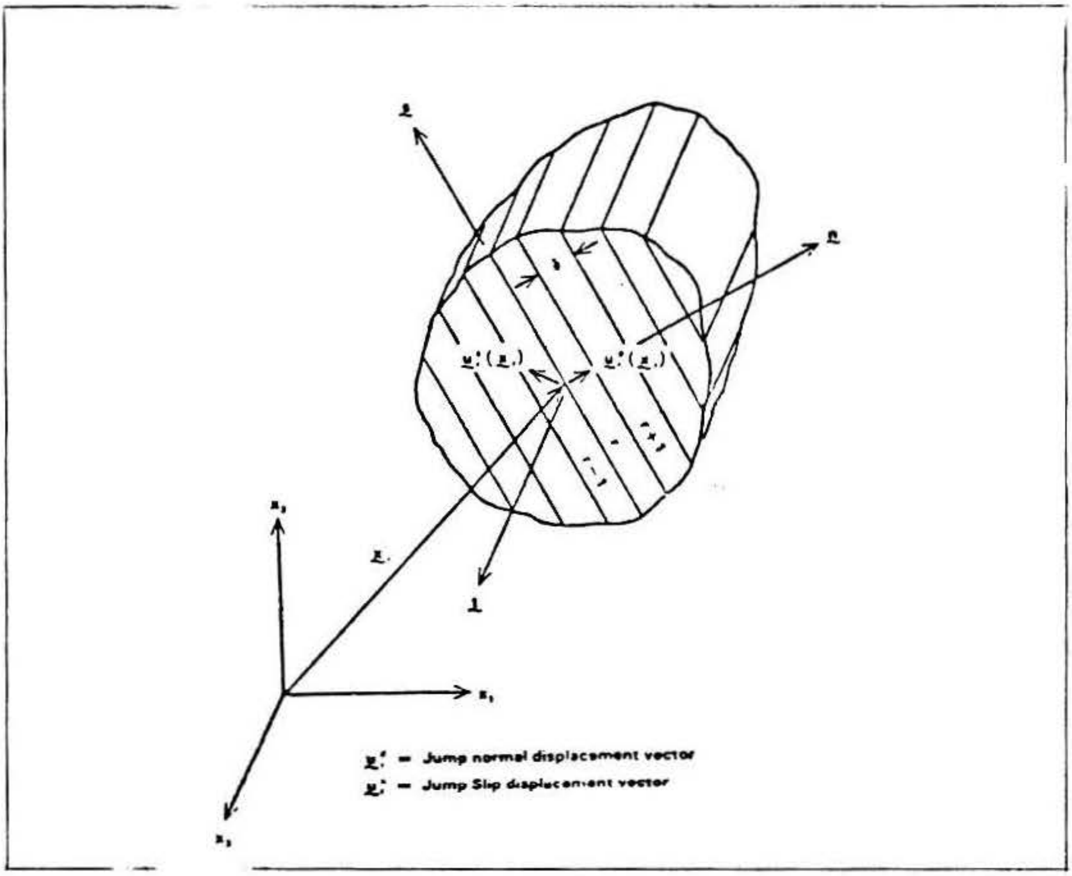


Figure 2. Representative elementary volume containing regularly spaced fractures.

It is assumed that the relative motion at the interface of the rth fracture at position x_r can be measured by a jump "dilatation" vector $u_r^d\,(x_r)$ normal to the fracture plane, and a jump slip displacement vector $u_r^s\,(x_r)$ parallel to the fracture plane. The net jump displacements for R fractures in the representative volume will then be

$$R\underset{\sim}{\bar{u}}^{d}(\underset{\sim}{x}) = \sum_{r=1}^{R} \underset{\sim}{u}_{r}^{d}(\underset{\sim}{x}_{r})$$

$$R\underset{\sim}{\bar{u}}^{s}(\underset{\sim}{x}) = \sum_{r=1}^{R} \underset{\sim}{u}_{r}^{s}(\underset{\sim}{x}_{r}) \tag{3.3}$$

where $\underset{\sim}{\bar{u}}^{d}$ and $\underset{\sim}{\bar{u}}^{s}$ are average displacements and $\underset{\sim}{x}$ is any position in the element. The continuous displacement fields $\underset{\sim}{\bar{u}}^{d}$ and $\underset{\sim}{\bar{u}}^{s}$ with respect to the x_1, x_2, x_3 axes are introduced,

$$\underset{\sim}{u}^{d}(\underset{\sim}{x} + d\underset{\sim}{n}) = \underset{\sim}{u}^{d}(\underset{\sim}{x}) + \left[\frac{\bar{u}^{d}}{\delta}\right] d\,\underset{\sim}{n}$$

$$\underset{\sim}{u}^{s}(\underset{\sim}{x} + d\underset{\sim}{n}) = \underset{\sim}{u}^{s}(\underset{\sim}{x}) + \left[\frac{\bar{u}^{s}}{\delta}\right] d\,\underset{\sim}{n}\,, \tag{3.4}$$

where

$$\underset{\sim}{\bar{u}}^{d} = |\underset{\sim}{\bar{u}}^{d}|\,\underset{\sim}{n} = \bar{u}^{d}\underset{\sim}{n}$$

$$\underset{\sim}{\bar{u}}^{s} = |\underset{\sim}{\bar{u}}^{s}|\,\underset{\sim}{v} = \bar{u}^{s}\underset{\sim}{v} \tag{3.5}$$

In equation (3.5) the direction of slip displacement is in the direction of the unit vector $\underset{\sim}{s}$. The total displacements can be written as

$$\underset{\sim}{u}(\underset{\sim}{x}) = \underset{\sim}{u}^{b}(\underset{\sim}{x}) + \underset{\sim}{u}^{d}(\underset{\sim}{x}) + \underset{\sim}{u}^{s}(\underset{\sim}{x}) \tag{3.6}$$

where $\underset{\sim}{u}^{b}$ is the displacement field of the intact material between fractures. From equation (3.6), the strain decomposition is taken to be

$$\underset{\approx}{e} = \underset{\approx}{e}^{b} + \underset{\approx}{e}^{d} + \underset{\approx}{e}^{s} \tag{3.7}$$

The dilatation and slip strains are defined in terms of the continuous displacement,

$$2\,\underset{\approx}{e}^{d} = \underset{\sim}{u}^{d}\nabla + (\underset{\sim}{u}^{d}\nabla)^{T}$$

$$2\,\underset{\approx}{e}^{s} = \underset{\sim}{u}^{s}\nabla + (\underset{\sim}{u}^{s}\nabla)^{T} \tag{3.8}$$

where ∇ is the gradient operator with respect to the x_1, x_2, x_3 axes. Equation (3.8) can be reduced by decomposing $\underset{\sim}{u}^{d}$ and $\underset{\sim}{u}^{s}$ into components in the local coordinate system.

$$\underset{\sim}{u}^{d} = |\underset{\sim}{u}^{d}|\;\underset{\sim}{n} = u^{d}\;\underset{\sim}{n}$$

$$\underset{\sim}{u}^{s} = |\underset{\sim}{u}^{s}|\;\underset{\sim}{v} = u_{s}^{s}\;\underset{\sim}{s} \tag{3.9}$$

Since both $\underset{\sim}{u}^{d}$ and $\underset{\sim}{u}^{s}$ have nonzero gradients only in the n direction, equation (3.8) becomes

$$\underset{\approx}{e}^{d} = \frac{\partial u^{d}}{\partial n}\,(\underset{\sim}{n} \times \underset{\sim}{n})$$

$$\underset{\approx}{e}^{s} = \frac{1}{2}\,\frac{\partial u_{s}^{s}}{\partial n}\,(\underset{\sim}{s} \times \underset{\sim}{n} + \underset{\sim}{n} \times \underset{\sim}{s}) \tag{3.10}$$

From equation (3.4)

$$\frac{\partial u^d}{\partial n} = \frac{\bar{u}^{\,d}}{\delta}$$

$$\frac{\partial u^s}{\partial n} = \frac{\bar{u}^{\,s}}{\delta} \tag{3.11}$$

so the final form for the strains is

$$\underset{\approx}{e}^{d} = \frac{\bar{u}^{\,d}}{\delta}\,(\underset{\sim}{n} \times \underset{\sim}{n})$$

$$\underset{\approx}{e}^{s} = \frac{\bar{u}^{\,s}_{\,s}}{2\delta}\,(\underset{\sim}{s} \times \underset{\sim}{n} + \underset{\sim}{n} \times \underset{\sim}{s}) \tag{3.12}$$

The constitutive model. We will first introduce the components of a stress tensor $\underset{\approx}{T}$ and a total strain tensor $\underset{\approx}{E}$ which refer to the local $\underset{\sim}{s}$, $\underset{\sim}{t}$, $\underset{\sim}{n}$ coordinate system. If σ and e are stress and strain tensors in the x_1, x_2, x_3 coordinate system, the transformation equations are

$$T_{nn} = \underset{\approx}{\sigma}\,\underset{\sim}{n} \cdot \underset{\sim}{n}$$

$$T_{ss} = \underset{\approx}{\sigma}\,\underset{\sim}{s} \cdot \underset{\sim}{s}$$

$$T_{tt} = \underset{\approx}{\sigma}\,\underset{\sim}{t} \cdot \underset{\sim}{t}$$

$$T_{ns} = \underset{\approx}{\sigma}\,\underset{\sim}{n} \cdot \underset{\sim}{s} \tag{3.13}$$

and

$$E_{nn} = \underset{\approx}{e}\,\underset{\sim}{n} \cdot \underset{\sim}{n}$$

$$E_{ss} = \underset{\approx}{e}\,\underset{\sim}{s} \cdot \underset{\sim}{s}$$

$$E_{tt} = \underset{\approx}{e}\,\underset{\sim}{t} \cdot \underset{\sim}{t}$$

$$E_{ns} = \underset{\approx}{e}\,\underset{\sim}{n} \cdot \underset{\sim}{s} \tag{3.14}$$

The behavior of intact material. In the present formulation, the intact material is assumed to behave as a linear elastic solid, having a strain rate

$$\underset{\approx}{\dot{e}}^{b} = \frac{\underset{\approx}{\dot{\sigma}}}{2G} - \left(\frac{K-\frac{2}{3}G}{6KG} \right) (\mathrm{tr}\,\underset{\approx}{\dot{\sigma}})\,\underset{\approx}{I} \tag{3.15}$$

Neglection of joint dilatation. The main purpose of using this model is to predict the weakening effects of slip along fractures and joints in materials. For many problems the fracture opening is small relative to the fracture spacing. For these problems, the joint dilatation normal to the crack is also small for cracks in compression; therefore the dilatation response will be neglected; thus

$$\underset{\approx}{\dot{e}}^{d} = 0 \tag{3.16}$$

The only condition imposed on the joint dilatation is that a joint cannot support a tensile load; therefore

$$T_{nn} = 0\,, \ \bar{u}^{\,d} \geq 0 \tag{3.17}$$

The behavior assumption for joint shear. The joint shear stress-displacement behavior is assumed to be elastic perfectly plastic. In the elastic range

$$\dot{\bar{u}}^{s}_{s} = \frac{\dot{T}_{ns}}{G_s} \tag{3.18}$$

where G_s is an elastic modulus of slip determined for a single joint. From equation (3.12) the slip strain rate is

$$\dot{\underset{\sim}{e}}^{s} = \frac{\dot{T}_{ns}}{2\delta G_s}\left(\underset{\sim}{s} \times \underset{\sim}{n} + \underset{\sim}{n} \times \underset{\sim}{s}\right) \tag{3.19}$$

The onset of plastic behavior is assumed to be governed by a linear Mohr-Coulomb criterion, based on a scalar "slip function", F, defined as

$$F = | T_{ns} | + \mu T_{nn} - C_o \tag{3.20}$$

where μ is the coefficient of friction and C_o is the cohesion. The joint behavior is elastic for F less than zero and plastic for F greater than zero. The stress-displacement behavior in shear is shown in Figure 3.

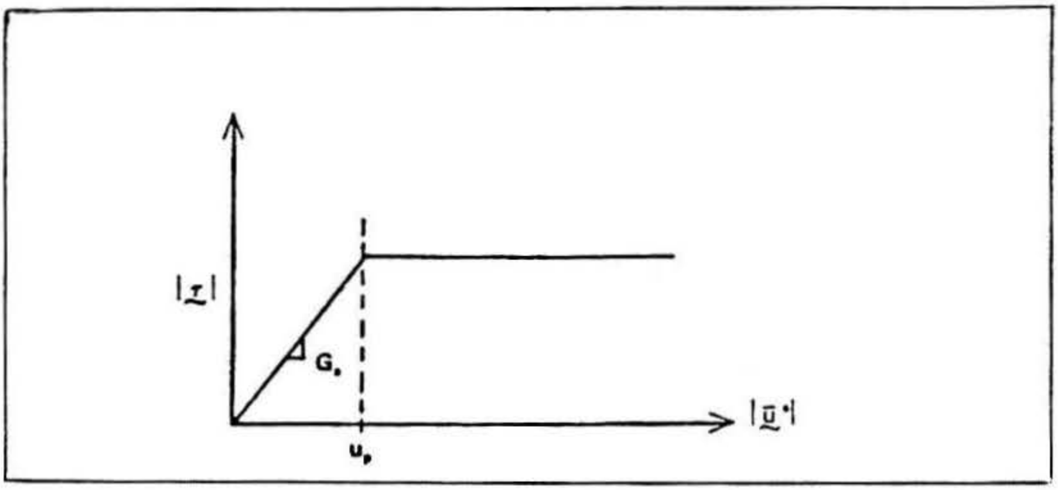


Figure 3. Assumed stress-displacement behavior in shear.

The solution of the constitutive equations. The SCRUBS.BYU finite element program uses an incremental method where the stresses and strains are known at the last stress increment, the current incremental strains are known, and only the current incremental stresses are needed.

From equation (3.7), the elastic strain rate can be defined as

$$\underset{\approx}{\dot{e}} = \underset{\approx}{\dot{e}^{b}} + \underset{\approx}{\dot{e}^{d}} + \underset{\approx}{\dot{e}^{s}}$$

Using the constitutive relations in equations (3.15), (3.16), and (3.19), the elastic strain rate [Eq 3.21] becomes

$$\underset{\approx}{\dot{e}} = \frac{\underset{\approx}{\dot{\sigma}}}{2G} - \left(\frac{K - \frac{2}{3}G}{6KG} \right) (\mathrm{tr}\,\underset{\approx}{\dot{\sigma}})\, \underset{\approx}{I} + \frac{\dot{T}_{ns}}{2\delta G_s} (\underset{-}{s} \times \underset{-}{n} + \underset{-}{n} \times \underset{-}{s})$$

Transferring equation (3.22) to scalar equations in terms of the local stress components, and omitting the out-of-plane shears, we have

$$\dot{E}_{nn} = \frac{\dot{T}_{nn}}{2G} - \left(\frac{K-\frac{2}{3}G}{6KG} \right) (\mathrm{tr}\underset{\approx}{\dot{T}})$$

$$\dot{E}_{ns} = \left(\frac{1}{2G} + \frac{1}{2\delta G_s} \right) \dot{T}_{ns}$$

$$\dot{E}_{ss} = \frac{\dot{T}_{ss}}{2G} - \left(\frac{K-\frac{2}{3}G}{6KG} \right) (\mathrm{tr}\underset{\approx}{\dot{T}})$$

$$\dot{E}_{tt} = \frac{\dot{T}_{tt}}{2G} - \left(\frac{K-\frac{2}{3}G}{6KG} \right) (\mathrm{tr}\underset{\approx}{\dot{T}}) \tag{3.23}$$

Solving for incremental stresses in terms of incremental strains produces

$$\dot{T}_{ns} = \left(\frac{2G}{1+\frac{G}{\delta G_s}} \right) \dot{E}_{ns}$$

$$\dot{T}_{nn} = 2GE_{nn} + \left(K-\frac{2}{3}G \right) \mathrm{tr}\,(\underset{\approx}{E})$$

$$\dot{T}_{ss} = 2GE_{ss} + \left(K-\frac{2}{3}G \right) \mathrm{tr}\,(\underset{\approx}{E})$$

$$\dot{T}_{tt} = 2GE_{tt} + \left(K-\frac{2}{3}G \right) \mathrm{tr}\,(\underset{\approx}{E}) \tag{3.24}$$

Therefore for the elastic case, all components of incremental stress in the local coordinate system can be obtained directly from incremental strains. Next, it must be determined if plastic slip occurs. After the stress components have been obtained the slip function [Eq 3.20] must be tested.

If the slip function F was less than zero, the strain rate was entirely elastic and the stress rates determined in equation (3.24) are correct. However, if F was greater than zero, plastic slip has occurred.

For plastic slip, determination of the incremental stresses can easily be maded by assuming elastic and perfectly plastic behavior such that

$$\dot{E}_{ns} = \dot{E}_{ns}{}^{e} + \dot{E}_{ns}{}^{p} \tag{3.25}$$

where $\dot{E}_{ns}$ is the total strain increment, and $\dot{E}_{ns}{}^{e}$ and $\dot{E}_{ns}{}^{p}$ are the elastic and plastic components of incremental strain, and the shear stress increment is then defined as

$$\dot{T}_{ns} = \left(\frac{2G}{1+\dfrac{G}{\delta G_s}} \right) E_{ns}{}^{p} \tag{3.26}$$

and the normal shear stresses are still defined as in equation (3.24) since they are not dependent on the incremental shear stress.

Using equations (3.25) and (3.26) and substituting into equation (3.20), the plastic strain, $E_{ns}{}^{p}$ can be solved for directly by solving the slip function, F, equal to zero. After the plastic strain is determined the incremental shear stress can be determined by repeating equations (3.25) and (3.26).

In chapter four a physical centrifuge simulation performed previously at Sandia National Laboratories along with several simulations performed for this thesis are presented. The results of the numerical simulations are then compared with each other and with the results of the physical simulation.

CHAPTER FOUR
MODEL VERIFICATION

Description of Problem

This chapter summarizes the procedure and results of various mine subsidence simulations. First, a physical centrifuge experiment of a coal mine model done at Sandia National Laboratories and the results of that experiment are presented. Second, numerical simulations using SCRUBS.BYU are introduced, the variables of the numerical models are listed and the results shown. Finally, comparisons are made.

Centrifuge Model

The following description of the test was taken basically from the laboratory report of the test [20].

The prototype. This centrifuge experiment modeled the response of shale overburden to the mining of a long wall panel. The prototype mine was located 86.3 m below the surface. The primary material above this mine was Devonian shale with horizontal bedding planes. The shale was fractured along and across these bedding planes. Above the shale was a relatively thin layer of sandy soil. A cavity of increasing width beneath the shale was created and the resulting failures and rubble formation were recorded.

The Sandia 25-Foot Centrifuge was used to conduct this test. The scaled-down model described in this section was mounted on a "swing platform" connected to the centrifuge.

Construction of the model. To construct the model, several shale boulders were cut into 202 mm slabs using a large rock saw. Three of these slabs were chosen. Their lengths were trimmed to 1.167 m (or less) and their top and bottom surfaces were machined flat and parallel to their bedding planes. They were then placed in a holding fixture that duplicated the internal dimensions of the test fixture. After this initial processing, dental plaster was used to fill voids between the three slabs of shale and to fill voids around the ends of each slab.

The model was fractured to simulate a jointed shale overburden. The fracturing process involved machining grooves into the exposed surface of the model (along the bedding planes). The grooves were 6 mm wide by 13 mm deep and on 25 mm centers. A 1.167 m long knife blade was then pressed into each groove to crack the shale along the bedding planes (the plaster was also broken). Some vertical cracks were also generated in the model during handling, machining, and fracturing processes. A complete description of the crack system, as observed from the non-grooved face of the model, is given in Figure 4.

Dimensions of the model. After the model was fractured, a thin layer of silica sand was placed above the shale to represent the soil layer. The thickness of this layer was 63.5 mm. With the shale layers being 0.511 m high, the completed model was rectangular, 1.167 m long, 0.202 m deep and 0.575 m high (again, refer to Figure 4).

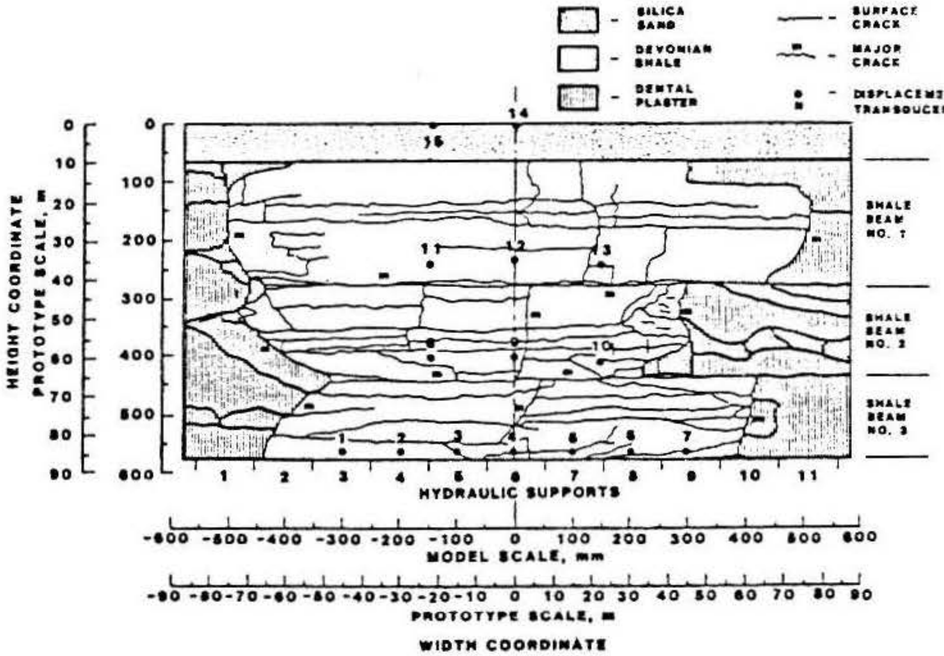


Figure 4. Schematic diagram of the centrifuge test [19].

The materials. As mentioned, three materials were used to construct the model mine structure. They were Devonian shale, silica sand, and dental plaster. The specific gravity of the Devonian shale was 2.64 Mg/m3. Its material properties varied between samples and with load range. For samples tested below 34.5 MPa, the average Modulus of Elasticity was 69.0 Gpa and Poisson's Ratio was 0.29. The unconfined compressive strength was 216 Mpa. The density of the dental plaster was

1.65 Mg/m3. The silica sand had a maximum partical size of 400 micrometers. When compacted at 150 g's, the density of this material was 1.25 Mg/m3.

Modeling the cavity. To create the subsurface cavity, the model was placed into the fixture with its bottom surface (length dimension) resting on a series of hydraulic supports. Each unit supported an area 102 mm wide by 203 mm deep. A total of 11 units were used to support the entire width (1.167 m) of the model. Using an electrically controlled hydraulic system, each support could be made to fall while under gravity loads. The cavity created by the fall of each support was 102 mm wide, 51 mm high, and 203 mm deep. Figure 4 includes a schematic diagram of this experimental configuration.

Between each segment of the simulation experiment (i.e., increase in width of the mine cavity), the centrifuge was stopped and additional records were made.

The experiment was conducted at an acceleration of 150 g's. Since linear dimensions scale in direct proportion with the acceleration, this model simulates a mine that is 86.3 m below the surface. The release of any single hydraulic support creates a prototype scale void that is 15.24 m wide and 7.62 high. All subsequent presentation of dimensions in this section will be specified in the prototype scale.

The results. Before the cavity was created in the experiment, the model was taken to 150 g's for approximately 5 minutes. This period was used to stabilize it. This was considered to be the first "time step."

As discussed earlier, a total of 11 hydraulic units supported the model. The mine cavity was created by lowering chosen supports in four additional time steps. For this simulation, the chosen sequence for lowering the supports was symmetric about the centerline of the model. First, the centerline unit was dropped. Subsequently, symmetric pairs about the centerline were dropped. Using the numbering sequence shown in Figure 4, support number 6 was dropped first (time step two), next supports 5 and 7 were dropped simultaneously (time step three), then supports 4 and 8 (time step four), and, finally, supports 3 and 9 (time step five).

Dropping support 6 produced a cavity of 15.24 m wide by 7.62 m high below the total strata depth of 84.3 m.

When the width of the drift was increased to 45.7 m by dropping jacks 5 and 7 (a total of three supports removed), a failure arch was formed. This arch extended to a depth of 49.1 m below the surface (a height of 35.2 m above the original roof of the cavity).

When the width was increased in the next step to 76.2 m (a total of 5 supports removed), the failure extended to the surface. Some new cracks were formed. The surface subsidence trough that was introduced into the model by these motions had a lateral extent of 94 m with a maximum subsidence of 5.33 m (see Figure 5).

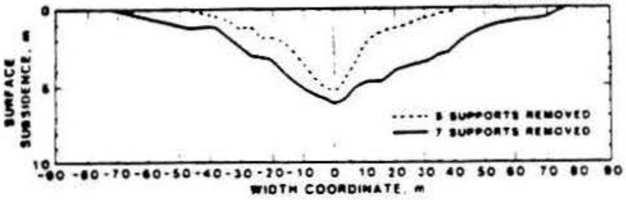


Figure 5. Centrifuge subsidence profiles [4].

With the final increase of the cavity width to 106.7 m (a total of 7 supports removed), many new cracks were formed and/or opened (see Figure 6). The subsidence trough extended to a width of 133 m with the maximum subsidence increasing slightly to 6.10 m (see Figure 5).

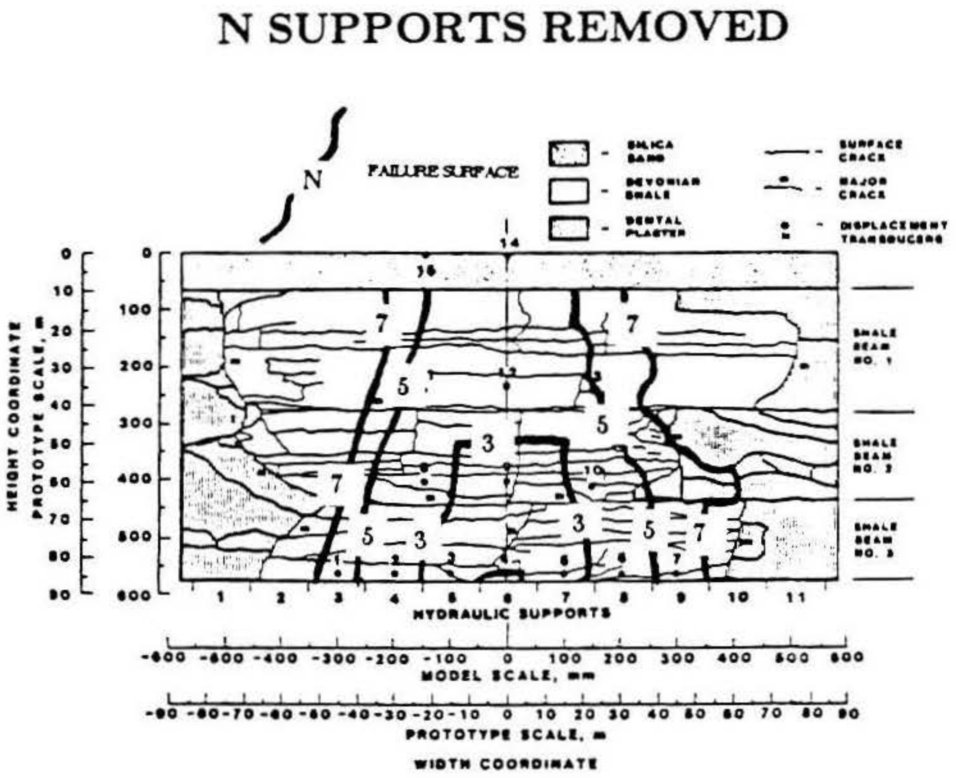
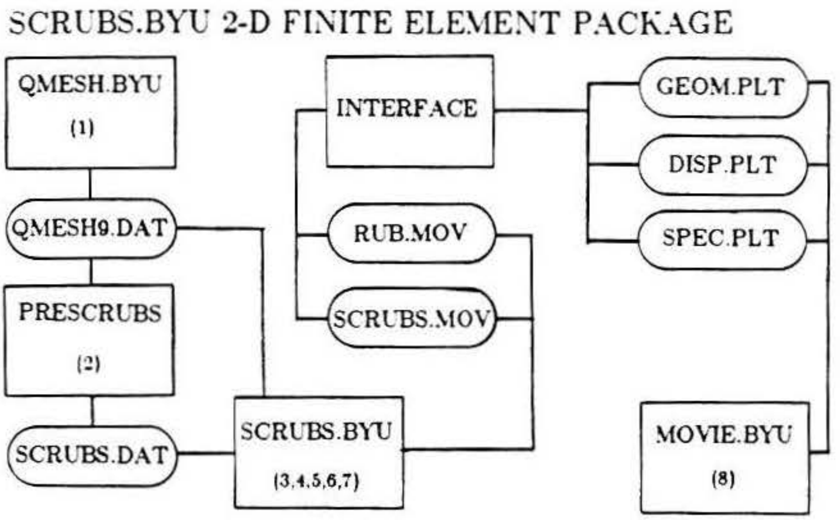


Figure 6. Failure progression of centrifuge model [4].

Finite Element Models

Equipment used. The numerical simulation of the mine was performed using SCRUBS.BYU on a VAX 11750 computer. Computer programs used in addition to the finite element code included a mesh generator (QMESH.BYU), a pre-processor (PRESCRUBS.BYU), an interface for the output (INTERFACE), and the graphics package (MOVIE.BYU). A short description of each of these programs and their use is contained in this section (See also Figure 7).



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.

Figure 7. Program flow and capabilities [9].

Description of the various models. Six numerical models were used to simulate the response of the shale overburden; two different meshes were generated, and each mesh was used with three different combinations of jointed media continuum/discrete slip planes characteristics. The first mesh used the major vertical and horizontal fracture lines as determined in the Sandia test as region boundaries. This mesh approximated these major fracture lines as closely as possible (see figure 8). The horizontal joints were straight, but the vertical joints were not, which introduced some difficulties in constructing a uniform elements. As a result, this mesh was labeled the "skewed mesh." The second mesh approximated the major fracture lines with strictly vertical and horizontal boundaries. This resulted in a much more uniform mesh (see figure 9), and was labeled the "straight mesh."

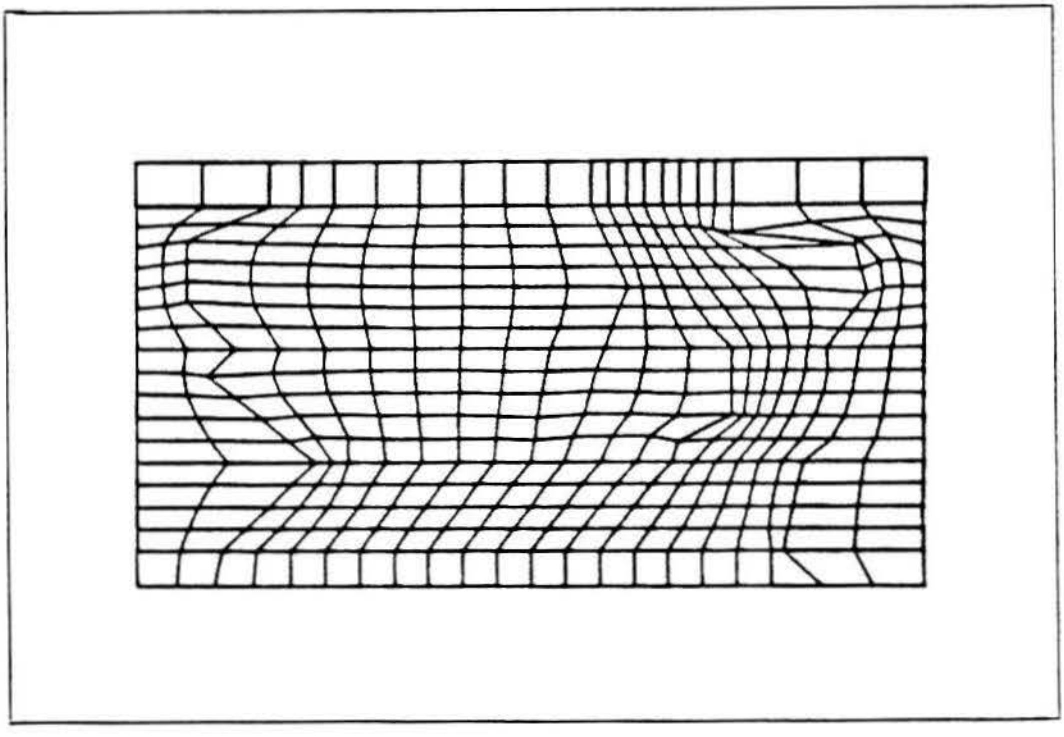


Figure 8. The skewed mesh.

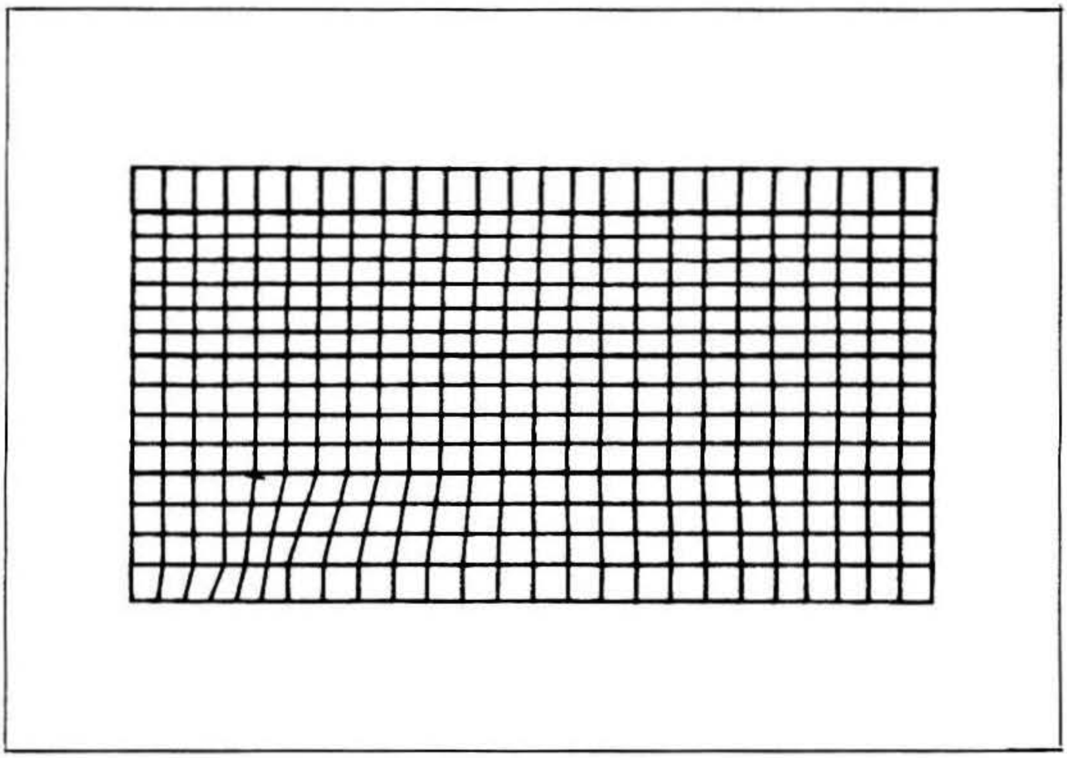


Figure 9. The straight mesh.

As stated, three different combinations of discrete slip plane/ jointed media continuum were used with each mesh. The fracture combinations were:

1) Horizontal slip planes only (modeling of major slip planes).
2) Jointed media continuum only (modeling of minor fractures).
3) Combined jointed media and horizontal slip planes.

Construction of the meshes. Included in the mesh generation were the assignment of regions, material types, boundary conditions, application of smoothing algorithms to obtain even meshes, and renumbering of the mesh for more efficient use in the finite element code. The final meshes each contained 14 regions of four different materials (refer to Figures 8 and 9). There were a total of 462 elements and 506 nodes in the skewed mesh, and 375 elements and 416 nodes in the straight mesh. The number of elements in the mesh was kept as low as it was judged possible and still maintain reasonable accuracy and reasonable computation time. Each potential slip plane required a unique flag to be set in QMESH.BYU. The mesh parameters, once processed by QMESH.BYU, were stored in a binary file named QMESH9.DAT to be used later by SCRUBS.BYU. Anyone not familiar with the operation of QMESH.BYU should consult Reference 8.

The problem parameters were all input using PRESCRUBS.BYU, the preprocessor for SCRUBS.BYU. These problem parameters included the material properties, the stress/strain field used, the yield condition, the element type, the number of gauss points to use in integration, the mining data, the convergence criteria, and the loading conditions. A state of plane strain was used throughout. Elasticity was assumed, the Von Mises failure criteria was chosen, gravity loads were applied, and linear isoparametric elements were integrated using four gaussian points per element. The rubbleization, slip plane, and jointed media characteristics were activated or deactivated by changing the corresponding flag in PRESCRUBS.BYU. The output from PRESCRUBS.BYU was stored in the file SCRUBS.DAT.

Model dimensions. The outside dimensions of both meshes were the same as those of the centrifuge model. The length was 112.2 cm, the height was 624.5 cm (from the bottom of the supports to the top of the sand), and the thickness was 20.2 cm. The steel elements had the same depth, 51 cm, as the supports in the Sandia model. The 14 interior steel elements were defined to have exactly one half the width of the Sandia supports, 51 cm, so that two elements were equivalent to one Sandia support.

Materials modeled. Four materials were modeled in the numerical simulation. The four modeled materials were 1) silica sand, 2) Devonian shale, 3) dental plaster, and 4) steel. The silica sand, dental plaster, and Devonian shale were set up the same as in the centrifuge model. Steel was used for the material of the supports at the bottom of the model.

Material properties which were needed for the numerical simulation but not available from the Sandia centrifuge test information were determined from soil mechanics texts and judgement. A consistent system of units was used of centimeters, seconds, grams, and dynes for all measurements and quantities. For silica sand, a modulus of elasticity of 17 MPa ($170(10^6)$dynes/cm^2), a Poisson's Ratio of 0.3, and a density of 1.25 g/cm^3 were used. For both the shale and the plaster properties of 69 GPa ($690(10^9)$dynes/cm^2) for Young's modulus, 0.29 for Poisson's Ratio, and 2.64 g/cm^3 for the densities were used. For the steel, the modulus of elasticity was 200000 N/m^2 ($20(10^{11})$dynes/cm^2), Poisson's Ratio 0.3, and the density 7.8 g/cm^2.

To use the rubbleization, slip plane, and jointed media capabilities of SCRUBS.BYU, additional material properties were required. Properties of the rubbleized material were set the same as for the shale. The slip material was given a bulk modulus of $3.45(10^{11})$dynes/cm^2 and a shear modulus of $3.45(10^{10})$dynes/cm^2. The jointed media was given a fracture spacing of 10 cm and a slip modulus of 10^7dynes/cm^2.

All materials were given a very high cohesion of 10^{20}dynes/cm^2 and a hardening modulus of one. The cohesion was set arbitrarily high so that the materials failed while still in the elastic range. The hardening modulus was arbitrarily set at one as it is not used in this formulation.

Failure condition. The failure condition was chosen to be when either the horizontal or vertical stress exceeded the failure stress. This condition is a user-supplied option in the subroutine FTEST (see appendix E). The failure stress for each model is noted in each figure caption (see Figures 10-15). The bulking parameter for rubbleization was set at 0.98 for all materials. The failure stress and the bulking parameter were set in PRESCRUBS.BYU. These choices gave generally good results, and were chosen for that reason.

The finite element program SCRUBS.BYU was used to analyze the model. The model was initially subjected to a gravity load of 150 gravities (1471.5 cm/s^2). The subsidence of the material under a gravity load was then determined using basic finite element techniques. This stabilizing step was called the first time step. Supports were removed in four additional time steps, with the center one (two elements) removed in the second time step and two neighboring supports (four elements) removed in each successive step, until seven supports (fourteen elements) had been removed.

The results. Results of the simulation run were stored in three files: (1) SCRUBS.LIS contained the parameters and the progression of failure, (2) SCRUBS.MOV contained the geometry, displacements, and stresses of the final mesh, and (3) RUB.MOV contained the rubbleized incremental geometry. SCRUBS.LIS contained the results in a readable form, while SCRUBS.MOV and RUB.MOV were binary files. Using an interface to the graphics package MOVIE.BYU, final subsidence profiles and rubbleized incremental profiles could be obtained from. correspondingly, SCRUBS.MOV and RUB.MOV.

Comparisons

The failure progression of the physical centrifuge simulation can be seen in Figure 6. This progression was used as a standard from which the numerical simulations were adjusted and refined in an attempt to match, and to which the results were compared.

The different meshes seemed to have some effect on the results. There was a tendency for the failure in the skewed mesh to tilt in the direction of the skewed elements, and in the straight mesh the failure would often pipe to the surface long before the failure in the skewed mesh would reach the surface. This piping is evident in both the jointed media and the discrete slip plane straight mesh models, although in slightly different forms. The centrifuge model had more of a pyramid failure form. The tilting of the failure in the skewed meshes did not seem to alter the overall pattern of failure, which was also pyramid in form.

In the jointed media straight mesh model (see Figures 10 a-d), piping to the surface occurred at the second time step (first support removed). After this initial piping, the failure seemed to follow the same general pyramid pattern of the centrifugal failure (and the skewed meshes).

In the jointed media skewed mesh model (see Figures 11 a-d), failure occurred in pyramid form, with the time steps favorably comparable to the progression which had occurred in the centrifuge model, including the formation of a failure arch, failure to the surface occurring at the fourth time step (5 supports removed), and a wide subsidence trough at the end.

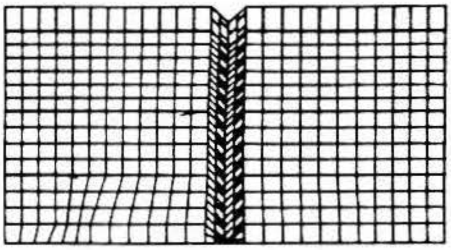
In the discrete slip plane straight mesh model (see Figures 12 a-d), no piping occurred until the third time step, but failure to the surface was then direct for all the remaining time steps. The resulting trough was comparable to the physical model, but there was no formation of a failure arch at any time.

The discrete slip plane skewed mesh model was similar to the jointed media skewed mesh (Figures 13 a-d). The failure included a pyramid form, the formation of a failure arch at the third time step, failure to the surface at the fourth time step, and the final wide subsidence trough.

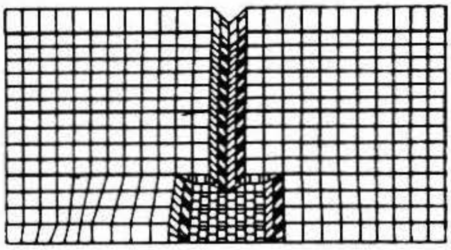
The combined straight mesh (Figures 14 a-d) resulted in a failure progression almost exactly like that of the jointed media straight mesh. A pipe failure to the surface occurred in the second time step, then a pyramid failure occurred from time step three to the final subsidence trough.

The combined skewed mesh (Figures 15 a-d) resulted in failure progression which was somewhat between that of the jointed media skewed model, and interestingly, the jointed media straight model. Piping failure occurred, although not quite as drastically or sudden as in the straight meshes. The final subsidence trough was comparable to the others.

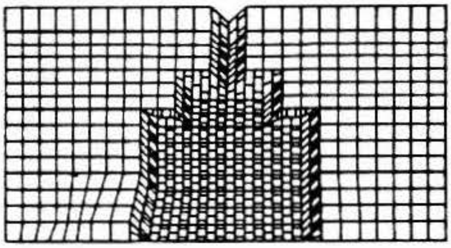
Some interesting variations of these numerical models are provided in appendix D.



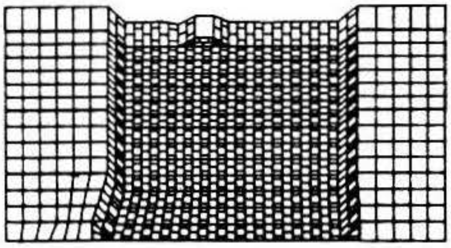
a. One support removed.



b. Three supports removed.

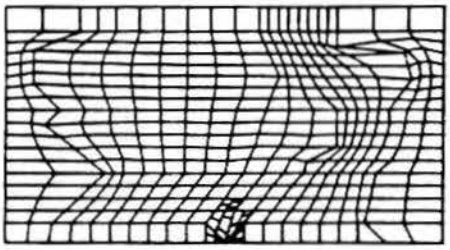


c. Five supports removed.

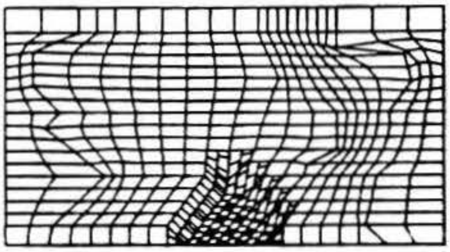


d. Seven supports removed.

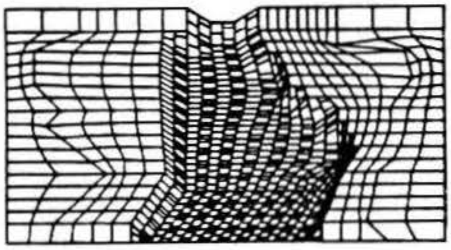
Figures 10 a-d. Jointed media straight mesh. Failure stress set to 0.1 10^7dynes/cm^2 (tension).



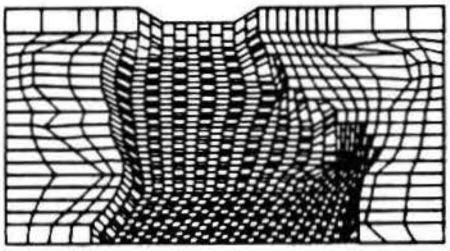
a. One support removed.



b. Three supports removed.

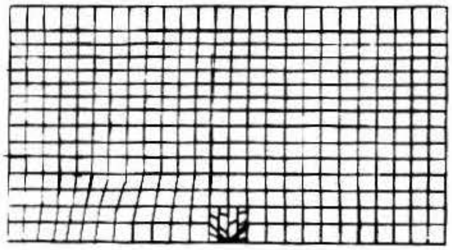


c. Five supports removed.

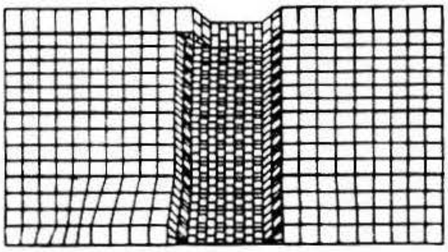


d. Seven supports removed.

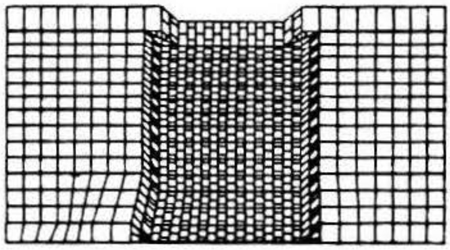
Figures 11 a-d. Jointed media skewed mesh. Failure stress set to zero.



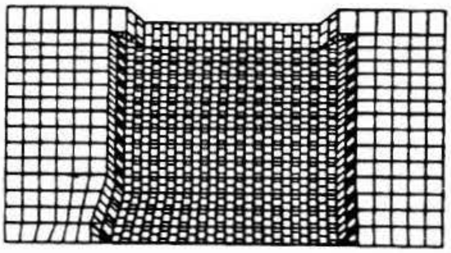
a. One support removed.



b. Three supports removed.

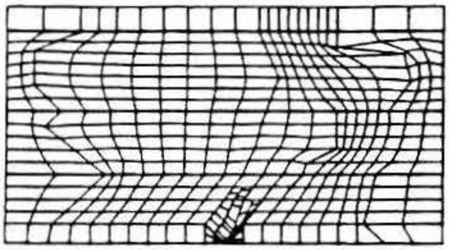


c. Five supports removed.

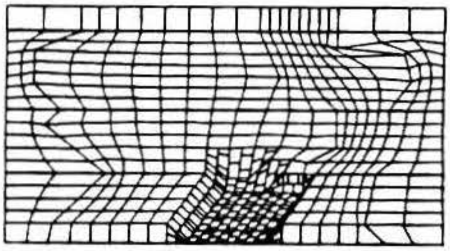


d. Seven supports removed.

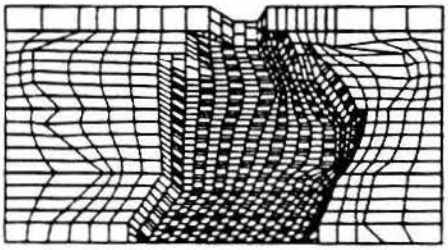
Figures 12 a-d. Discrete slip plane straight mesh. Failure stress set to 0.1 10^5dynes/cm^2 (tension).



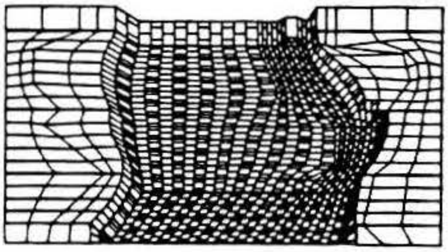
a. One support removed.



b. Three supports removed.

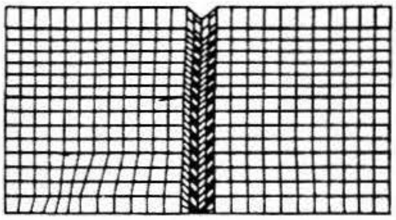


c. Five supports removed.

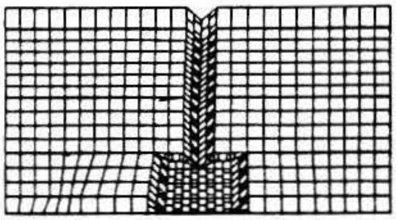


d. Seven supports removed.

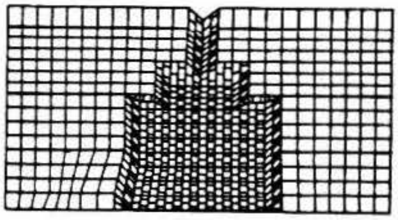
Figures 13 a-d. Discrete slip plane skewed mesh. Failure stress set to zero.



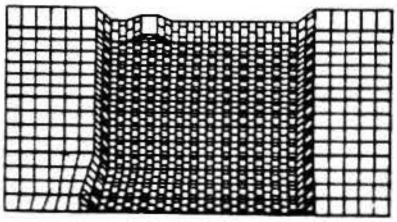
a. One support removed.



b. Three supports removed.

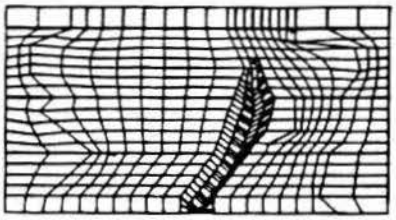


c. Five supports removed.

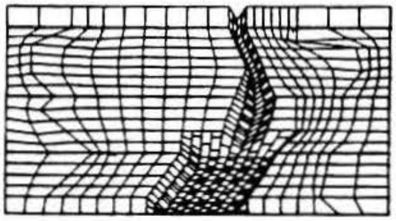


d. Seven supports removed.

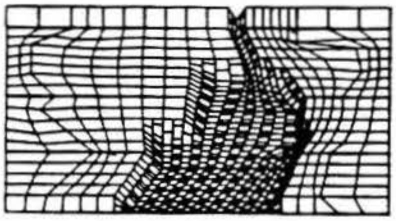
Figures 14 a-d. Combined straight mesh. Failure stress set to 0.1 10^7dynes/cm^2 (tension).



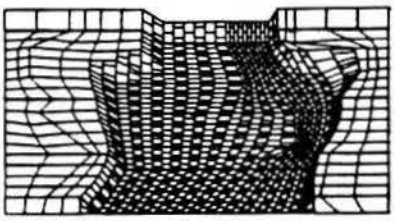
a. One support removed.



b. Three supports removed.



c. Five supports removed.



d. Seven supports removed.

Figures 15 a-d. Combined skewed mesh. Failure stress set to 0.1 10^7dynes/cm^2 (tension).

CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

Knowledge of fracture and joint modeling in rock mechanics problems has been increased by the contribution of this thesis. Modeling fractures as either low shear element slip planes, jointed media continua, or as a combination of both are all valid approaches. Since many geologic materials contain both faults, which are better modeled by discrete slip planes, and rock mass with somewhat regular joint spacing, which is best modeled by continuum theory, it would make sense at times to combine them. In any case, it is possible for the stress analyst to be able to choose between the two models, or use the combination of both.

Two important characteristics of the simulations were closely considered: first, the failure sequence, and second, the final subsidence troughs. The general failure progression of all models in the mine subsidence problem were very similar. This would seem to indicate that all the models used were valid. To some extent, this success was dependent on an understanding of the properties and geometry of the problem. The second important characteristic considered, the subsidence troughs, were not always similar. The maximum displacements of the troughs were comparable, but the trough edges were not. As noted in chapter four, the physical centrifuge model resulted in gradually sloping trough edges, while the numerical simulations contained very steep trough edges. This is caused by a weakness in the numerical stiffness formulation, which has difficulty in modeling the flow of the surface sand with solid finite elements. This difficulty is also indicated by the presence of isolated non-failed elements at the surface (refer to figures 10d, 13d, and 14d).

Problems and Solutions

The two different mesh types, skewed and straight, resulted in two consistent variations in the failure. First, the direction of failure was influenced, though only slightly, by the mesh type. The skewed mesh showed a tendency to fail diagonally, along the skewed elements, rather than in a more regular, vertical pattern like the straight mesh. The tendency for the skewed mesh to fail slightly diagonally implies that the finite-element formulation has some difficulty with the uneven elements. The failure was still quite good, however, and the final results seemed to be as good as those of the physical simulation. Second, the general form of failure varied between mesh types. The skewed mesh resulted in a pyramid form of failure, while the straight mesh contained more of a piping form. The varying forms of failure seem more pronounced than the slight directional tendencies.

For this problem, the skewed mesh seemed to be slightly more accurate. This is most likely due to the presence of major not-quite vertical slip planes in the centrifuge model (refer to Figure 4). While we were not able to use vertical slip planes (there is a problem with combining both vertical and horizontal discrete slip planes in a single model in

SCRUBS.BYU), the skewed mesh introduced diagonal failure tendencies. The diagonal failure may have reduced the piping failure, as the piping usually occurred vertically.

We used the least number of elements we could and still get results that did not seem effected by the element sizes. The run time for this kind of formulation and the size of the problem seemed reasonable. In our formulation, linear elements were used, whereas quadratic elements may have given better results. If quadratic elements were needed, fewer elements would be used, but the increased computational requirements for the quadratic formulation would offset the lowered computational requirements for the fewer elements.

As previously mentioned (chapter four), the failure stress was set at zero for the jointed media and discrete slip skewed meshes, at 0.1 10^7dynes/cm^2 for the jointed media straight mesh model and the combined models, and at 0.1 10^5dynes/cm^2 for the discrete slip plane straight mesh model. This criteria seemed to give the best results. Other criteria attempted included the separate horizontal and vertical stresses by themselves and the maximum principal stress. More study of the failure stress possibilities would be useful.

Although at present the use of the low shear element capability is currently limited to elastic deformations along the plane of slip, the incorporation of a plastic failure model for the slip plane material is possible. The Drucker-Prager yield criteria [6] would be especially applicable for this purpose and would be recommended for any further study in this area.

It should be noted here that the jointed media formulation in SCRUBS.BYU at present is only valid for a plane strain or axisymmetric condition. In the mine model, a plane strain assumption was made because of this, even though a plane stress situation may have been more realistic.

As previously mentioned, the poor profile of the subsidence trough at the edges in our numerical simulations were most likely a result of an incorrect stiffness formulation for the sand.

Summary

A summary of the recommendations made above are listed as follows:

1) Test the effectiveness of using quadratic elements in a model.
2) Study in detail the different failure criteria, and determine those which would be most accurate and in which situations.
3) Apply the Drucker-Prager yield conditions to a model in which the required parameters are available.
4) Do a plastic analysis of a subsidence problem, preferably using the Drucker-Prager yield condition.
5) Adjust the jointed media formulation for plane stress states.
6) Correct the stiffness formulation to allow for sandy materials.
7) Correct SCRUBS.BYU to allow combined horizontal and vertical discrete slip planes in a single model.

BIBLIOGRAPHY

SELECTED BIBLIOGRAPHY

1. Advani, S.H. and Lin, Y.T. "Subsidence and Roof Response Studies Related to Underground Coal Gasification," *Proceedings, Third Annual Underground Coal Conversion Symposium.* Fall Leaf Lake, CA, June 9 (1977).

2. Benzley, S.E. "SCRUBS.BYU, A Finite Element Computer Program for Subsidence Modeling," Engineering Research, College of Engineering Sciences and Technology. Brigham Young University, Provo, UT (1982).

3. Brauner, G. "Subsidence Due to Underground Mining, Part I," *I.C. 8571,* U.S. Bureau of Mines. Washington, D.C. (1973).

4. Clough, R. W. and Woodward, R.J. III "Analysis of Embankment Stresses and Deformations," *Journal of the Soil Mechanics and Foundations Division,* ACSE, Vol. 93, No. SM4. July (1967), pp.529-549.

5. Cook, Robert D *Concepts and Applications of Finite Element Analysis (Second Edition),* John Wiley and Sons Inc. NY (1981).

6. Drucker, D.C. and Prager, W. "Soil Mechanics and Plastic Analysis or Limit Design," *Quarterly of Applied Mechanics, 10.* (1952), pp.157-165.

7. Heuze, F.G. "Stability Analysis for Rock Structures," *19th Symposium on Rock Mechanics.* Stateline, NE (1978).

8. Jones, R. K. "User's Manual for QMESH, a Self-Organizing Mesh Generation Program," *SLA-74-02139,* Sandia Laboratories. Albuquerque, NM, July (1974).

9. Long, Michael G. "SCRUBS.BYU: A Two-Dimensional Finite Element Package for Continuum Analysis Using Quadratic Isoparametric Elements," Unpublished Master's Thesis. Brigham Young University, Provo, UT (1983).

10. Moreland, L.W. "Elastic Response of Regularly Jointed Media," *Geophysical Journal, Vol. 37.* (1974), pp.435-446.

11. Moreland, L.W. "Continuum Model of Regularly Jointed Mediums," *Journal of Geophysical Research, Vol. 79, No. 2.* (1974), pp.357-362.

12. Moreland, L.W. "Plane Wave Propagation in Anisotropic Jointed Media," *Quarterly Journal of Mechanics and Applied Mathematics, Vol. 30.* (1977), pp.1-21.

13. Nair, K. "Analytical Methods for Predicting Subsidence," *Proceedings, Lawn Subsidence Symposium.* Tokyo, Sept (1969).

14. National Coal Board *Subsidence Engineer's Handbook.* Hobart House, Grosvenor Square, London (1975), 111 pp.

15. Nayak, G.C. and Zienkiewicz, O.C. "Elasto-Plastic Stress Analysis. A Generalization for Various Constitutive Relations Including Strain Softening," *International Journal for Numerical Methods in Engineering, Vol. 5.* (1972), pp 113-135.

16. Nelson, Royd R. "Two-Dimensional Computer Modeling of Joints and Fractures in Continua", Unpublished Master's Thesis. Brigham Young University, Provo, UT, December (1983).

17. Peng, S. S. *Coal Mine Ground Control.* Wiley, NY (1978).

18. Sutherland, H.J., Hommert, P.J., Taylor, L.M., and Benzley, S.E. "Subsidence Prediction for the Forthcoming Tono UCG Project", *Proceedings, Ninth? Annual Underground Coal Conversion Symposium*, ?? IL, ? (1983).

19. Sutherland, H.J., Schuler, K.W., and Benzley, S.E. "Numerical and Phisical Simulations of Strata Movement Above Idealized Mine Structures," *SAND83-0701.* Sandia National Laboratories, Albuquerque, NM (1983).

20. Sutherland, H.J., Heckes, A.A., and Taylor, L.M. "Physical and Numerical Simulations of Subsidence above High Extraction Coal Mines," Sandia National Laboratories, Albuquerque, New Mexico (1983).

21. Thomas, Robert K. "A Continuum Description for Jointed Media," *SAND81-2615.* Sandia National Laboratories, Albuquerque, NM (1980), p 24.

22. Voight, B. and Dahl, H.D. "Numerical Continuum Approaches to Analysis of Nonlinear Rock Deformation," *Canadian Journal Earth Science, Vol. 7* (1970).

23. Witte, W., editor. *Numerical Methods in Geomechanics, Vol. 1, 2, and 3,* Proceedings of 3rd 1st Conference on Numerical Methods in Geomechanics. Balkena, Rotterdam (1979).

APPENDICES

APPENDIX A

SCRUBS.BYU SUBROUTINE MAP AND DESCRIPTIONS

The following appendix provides subroutine descriptions and subroutine maps for the finite element code SCRUBS. The subroutine descriptions (the main is included) are listed alphabetically.

The subroutine descriptions:

AUX:	Creates coordinate jacobian 0 interpolate X; YOCOJ(I,J), XY(1),XY(2) are respectively the coordinate jacobian matrix, the X coordinate and the Y coordinate.
BSUB:	Conducts systematic back-substitution and solves for boundary displacements and reaction transformations.
CONRUB:	This subroutine checks for rubble/continuum interaction and computes the hydrostatic force in the rubble if a continuum element is in contact.
CONVG:	Determines the percent convergence.
CORNN:	This subroutine determines if node IX is a corner node.
DATALST:	Writes title, line numbers for files (presently not used).
DROP:	Checks for elements about to fail and adjusts geometry for failure.
FLAGE:	Makes impact node calculations and new node assignments.

GAUSSP: Initializes gaussian constants.

GDATA: Reads geometrical data, boundary conditions, and gaussian integration data.

INVAR: Calculates invariants and sets the yield stress accordingly.

JOINTM: Calculates incremental stresses and residuals.

LDATA: Reads load control data.

LINEAR: Helps to update stress and strain.

MAKEEL: This subroutine adds a new element to the connectivity containing corner nodes I and K.

MOD: Forms D matrix for different stress/strain cases.

NEWNOD: This subroutine finds the new node number of node N.

NFLOW: Calculates the amount of yield that results.

NODEXY: Sets x,y values of new intermediate coordinates of quadratic or cubic elements.

OUTPUT: Writes displacements, reactions, stresses and strains to SCRUBS.LIS

PDISP: Initializes the displacement vector according to number of loading increments.

READF: Reads the mining parameters: which elements, and when.

REMLD: Removes gravity load from failed elements.

RESIDUE: Determines element failure, rubbleization, and residual.

RESOLV: Updates stiffness equation solution.

ROSB: Adjusts stiffness matrix.

ROTATE: This subroutine rotates 2D stress or strain vector by angle THETA.

RUBDRAW: Writes the rubble file RUB.MOV.

RZBASE: This subroutine finds the base R and Z dimensions for fall calculations.

SCRUBS: Program designed to compute the failure, collapse, and resulting subsidence of geologic materials. Performs elastic/plastic analysis of plane stress/strain and axisymmetric problems for linear, parabolic, and cubic elements.

SCRUBSV: This subroutine opens and closes the files used by the main program, SCRUBS.

SET: Sets element spacing and determines bandwidth and resulting stiffness storage required.

SFR: Sets shape functions according to element type.

SLIPP: This subroutine reads the slip plane data, adjusts the geometry and elements accordingly, and then records the adjusted information.

SOLVE: Solves stiffness equations, modifies loads.

STIFM: Forms and updates stiffness matrix, and calculates body forces.

SWITCH: Switches coordinates.

TFORM: Makes adjustments for inclined boundary situations.

ZONE: This subroutine reads meshing cards as done in the CHILES2 program. It is provided here only as an interim capability. This subroutine is called by the main program.

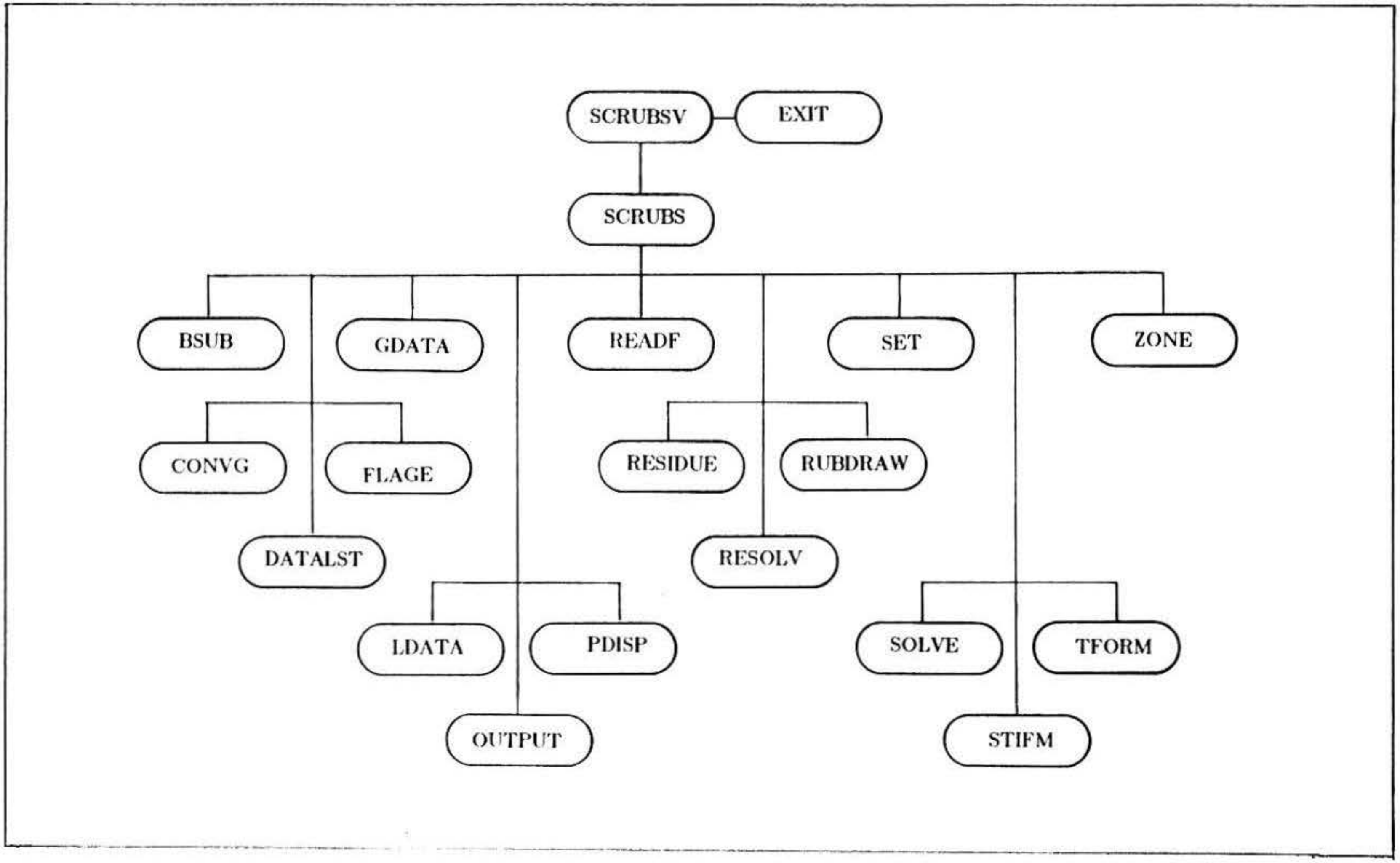


Figure 16 a. Subroutine map -- main.

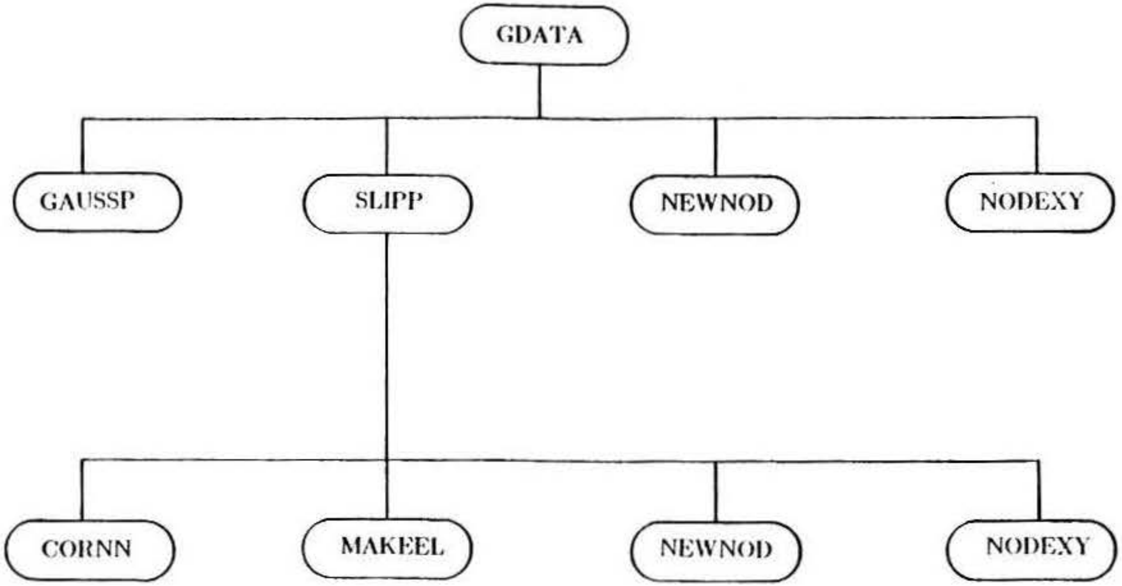


Figure 16 b. Subroutine map -- GDATA.

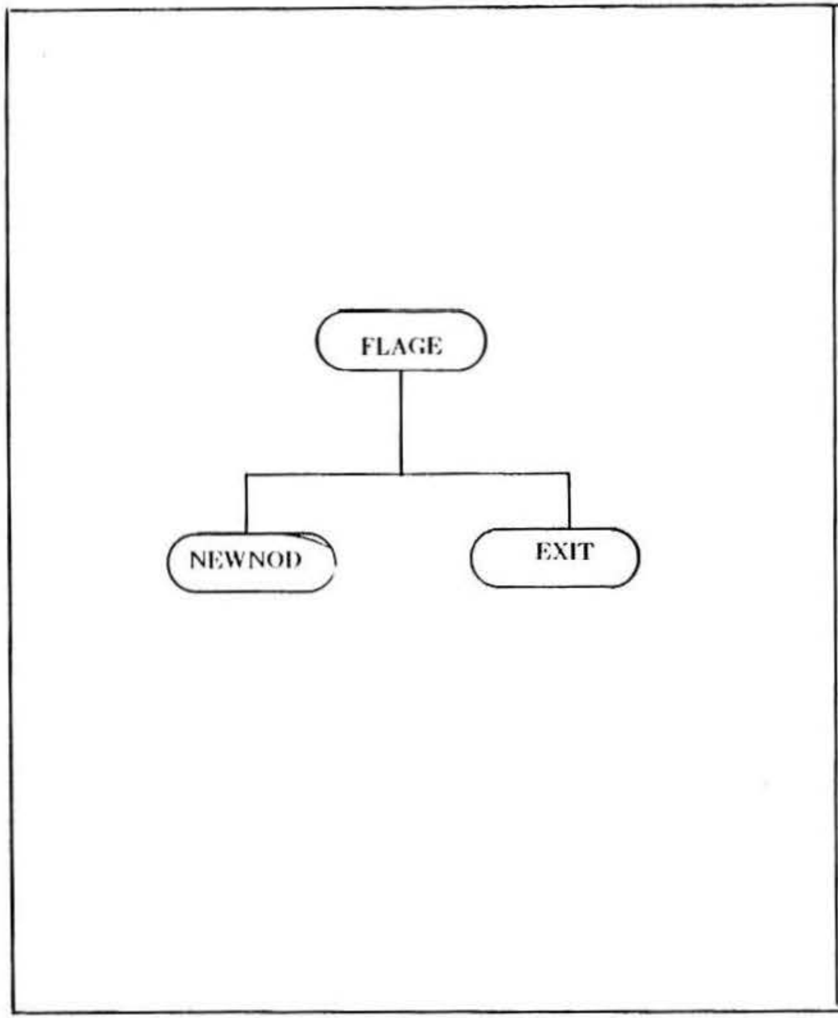


Figure 16 c. Subroutine map -- FLAGE.

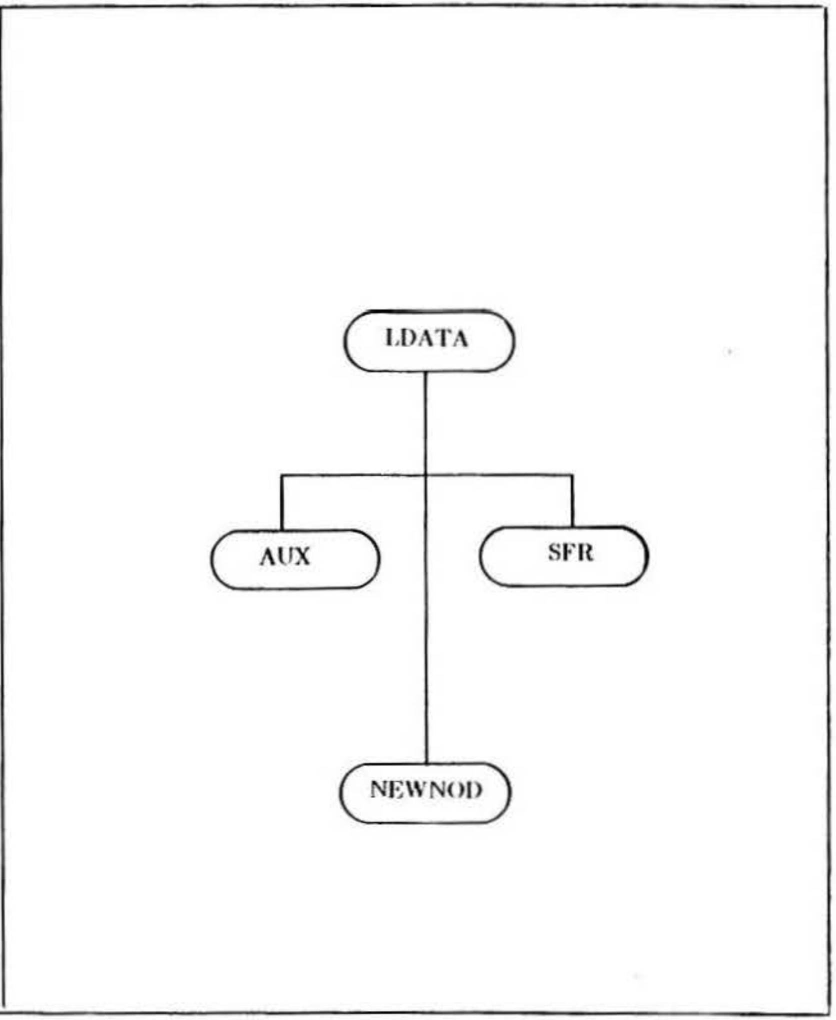


Figure 16 d. Subroutine map -- LDATA.

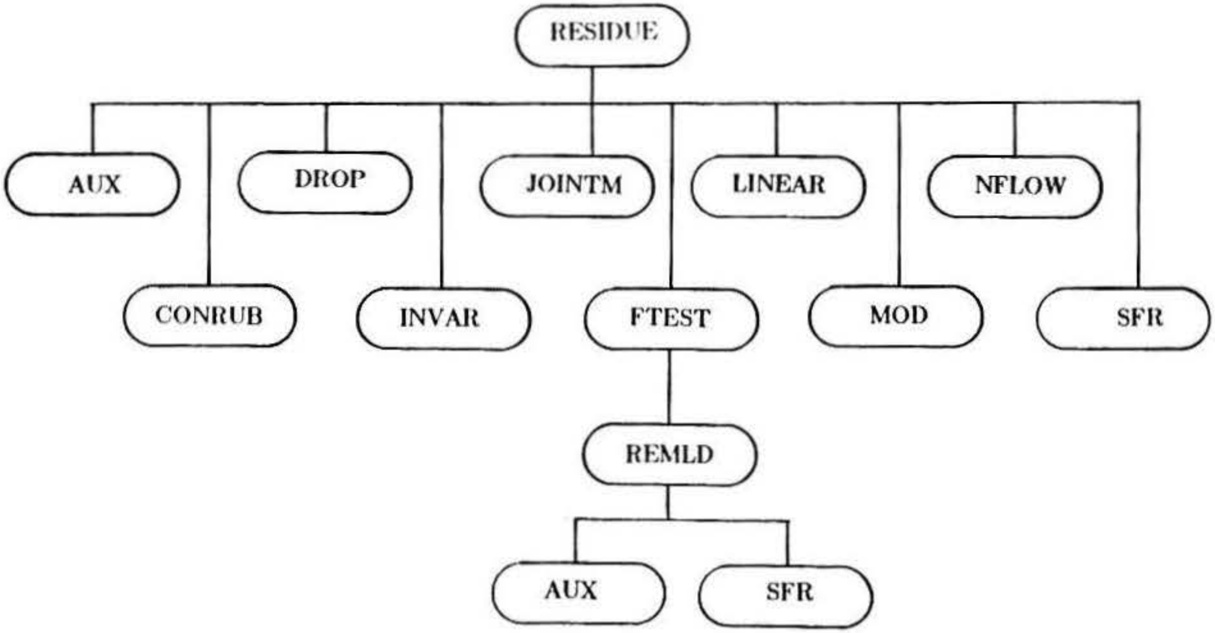


Figure 16 e. Subroutine map -- RESIDUE.

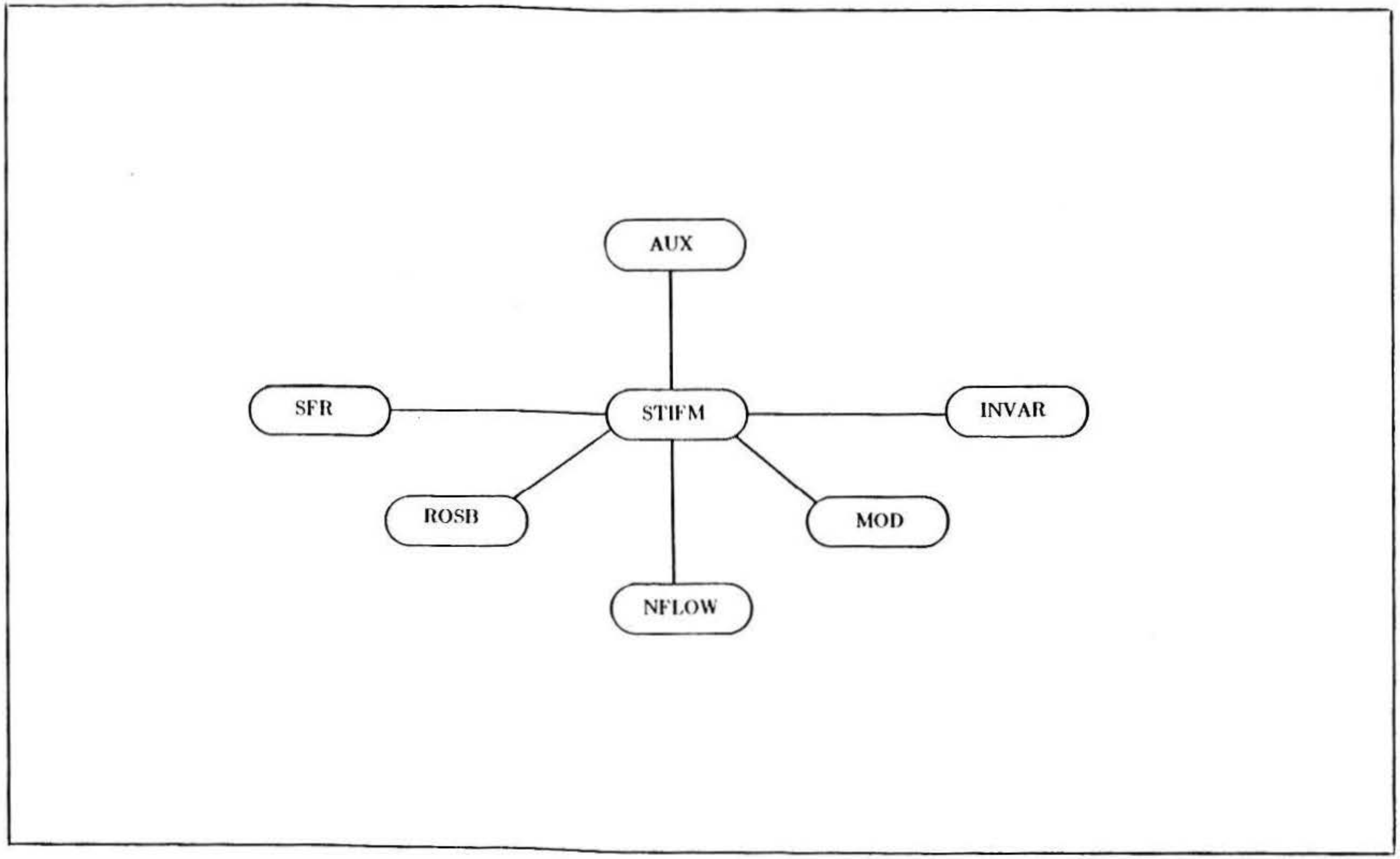


Figure 16 f. Subroutine map -- STIFM.

APPENDIX B

SCRUBS.BYU USER'S MANUAL

The following appendix provides: first, a short description of the program SCRUBS and its capabilities; second, detailed descriptions of the three files which are read by SCRUBS and the four files that are created by SCRUBS.

SCRUBS.DOC

SCRUBS is a FORTRAN computer program designed to compute the failure, collapse, and resulting subsidence of geologic materials. The uniqueness of this program is it's 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, quadradic 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 interation 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.

Three specific files are read and four other 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 four 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

SCRUBS.LIS = List of failure sequence

The contents of these six files are now explained in detail.

QMESH9.DAT

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 COMMEN 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.

End-of-File mark.

SCRUBS.DAT

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 minumum 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 (not more than 1000)
  - 6 - 10 Total number of elements, NE (not more that 1000)
  - 11 - 15 Total number of restrained boundary points, NB
  - 16 - 20 Total number of load cases/problem, NLD
  - 21 - 25 Number of d.o.f. per node, NDF
  - 26 - 30 Number of different materials, NMAT (not more than 10)
  - 31 - 35 Element type 1=linear, 2=parabolic, 3=cubic, NSFR
  - 36 - 40 Number of gauss points for stiffness calc. 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 of gauss points for nodal force residual calculation and stress storage (MGAUS) result even for elastic solution MGAUS defaults for NGAUS.

* * *

* * *

GROUP 3 - MATERIAL PROPERTIES

Line 1. Material data (I10, 7F10.2) NMAT Lines, limited to NMAT= 10 (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,3)
41 - 50	Hardening modulus	ORT(N,4)
51 - 60	Conical yield surface angle	ORT(N,5)
61 - 70	Thickness (leave as zero if NPP = 1)	ORT(N,6)

* 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)

** A negative value in this position activates the jointed media material description. For this case, one more line is read, for that material only, defined as follows: (2G12.5)

1-12 Fracture spacing ORT(N,9)

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

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 each increment, NRFF total, omit if NRFF=0. (16I5)
1 - 5 Element to be mined on this increment, MINES(INC,1)
6 - 10 Element to be mined on this increment, MINES(INC,2)
11 - 15 Element to be mined on this increment, MINES(INC,3)
16 - 20 Element to be mined on this increment, MINES(INC,4)
. .
. .
. .
etc.

* *

* *

GROUP 5 BOUNDARY CONDITIONS

Line 1. Ancillary control data (16I5)
1 - 5 Number of slip planes NNP
6 - 10 Number of singular points NSING
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 when counterclockwise.
```

Lines 3 and 4 are repeated in NSP blocks, each block representing a new slip plane. If NSP = 0, omit lines 3 and 4.

```
     Line 3.  Slip plane control data (2I5,G10.3)
               1 -  5 Number of nodes on slip plane boundary  NSLP(N)
               6 - 10 Angle flag of slip plane                IANG(N)
              11 - 20 Thickness of slip plane                 STHIK(N)

     Line 4.  Slip plane node list (16I5) NSLP(N)/16 cards
               1 -  5 Node number
               6 - 10 Node number        ISLP(I,1), I = 1,NSLP(N)
              11 - 15 Node number
                 .      .
                 .      .
                 .      .
                 etc.

     Line 6.  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.
```

* *
* *

* *

* *

* *

```
****************************************
*  NOTE!!!  THE FOLLOWING LINES OF     *
*  GROUP 6 ARE REPEATED IN NLD BLOCKS  *
*  EACH BLOCK REPRESENTING A NEW LOAD  *
*  CASE                                *
****************************************
```

GROUP 6 LOADING DEFINITION

Line 1. Ancillary load control data (2I5)
1 - 5 Initial stress counter ISTS (not used)
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 fac array $\{X_{inc}\} = FAC \, \{X_{total}\}$
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 this system (Only 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)
. . .
. .
. .

Line 7. Pressure element control, if no pressure loads (i.e. NRPRS=0) leave out and go to line 10. (8I5)
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
                     0-------------0
        6 - 10 Node 2
                     0------0------0
       11 - 15 Node 3
                     0----0---0----0
       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 element
         1 - 10 Normal pressure (Pn)                 |
                                                     |Pn
                                                     |
        11 - 20 Tangential pressure (Pt)             V
                                                  ________
        21 - 30 Zero                             |        ||
                                                 |        ||Pt
                                                 |        ||
                                                 |________|V
                                                  ------->

        (b) Different loads at each node of line element Pn
         1 - 10  Pn |
                    | Node 1
        11 - 20  Pt |

        21 - 30     |
                    | Node 2
        31 - 40     |

         etc.
```

Return to 8 for next line element.

Note (a) does not apply for linear elements

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 displadements 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,F10.3)
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 7 REDUCED PRINTOUT CONTROLS

Line 1. Print control (2I5)
1 - 5 Number of elements to be printed. If negative, 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.

* *

SCRUBS.MOV

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,

```
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.

```
(CORD(I,1),I=1,NP),(CORD(I,2),I=1,NP),
((NOP(4*(K-1)+M),K=1,NE),M=1,4),
(IMAT(I),I=1,NE)
```

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.

```
(TDIS(1,I),TDIS(2,I),I=1,NP),
(TDIS(1,I),TDIS(2,I),I=1,NP),
(TDIS(1,I),TDIS(2,I),I=1,NP)
```

Record 3'. (4xNE words) Average element stresses

```
((SPLT(I,J),J=1,NE),I=1,4)
```

Record 4'. (4xNE words) Average element strains

```
((SPLT(I,J),J=1,NE),I=1,4)
```

RUB.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.

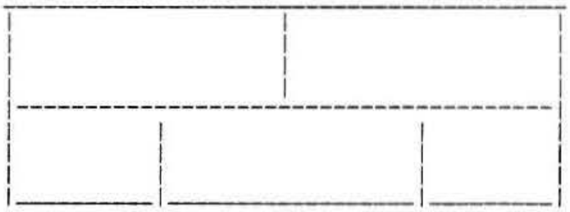
Record 3. (2 words, 2I5) Begining element number,NP1; Ending element number, NP2.

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 NP1 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)

SCRUBS.RST

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
3. Group 6, lines 10,11,12,and 13.

Note: Group 6, line 10 (NINC) must be changed to the number of remaining load increments!**

SCRUBS.LIS

This file contains input and output data formatted into a readable form. The input data is first written in its entirety, and includes a nodal mapping for the node numbers changed due to the slip plane addition.

After the input data is written, convergence information is output for each iteration. This information includes run times at various points in the program, and information on current loads, residuals, displacements, and plastic work. Convergence is reached when the normal of the residual sum ratio reaches the convergence percentage specified by the user. The flag NCHECK will be zero when convergence is attained.

The last portion of this file contains the output information. Output includes displacements and reactions for all boundary nodes, nodal displacements for all nodes if specified, and all stresses at each gauss point of any elements specified.

SCRUBS also has several checks for errors or incomplete solution which halt execution before the solution is obtained. For each of these an error message is output to the SCRUBS.LIS file. The following is a list of each of these stops, including a description and possible cause of the error.

NO CONVERGENCE ON INCREMENT I

Solution could not be reached in the number of iterations specified. This is usually due to too much plastic flow and can often be corrected by reducing the loads.

NEGATIVE DET STOPPED AT ELEMENT NO = I
X1,Y1= x , y
X2,Y2= x , y
X3,Y3= x , y
X4,Y4= x , y

This indicates a element is improperly deformed as shown

```
 1    3          1    4
 -----           -----
 !   !            )(
 !   !           -----
 -----
 2    4          2    3
```

Usually caused by too large of deformations.

PROGRAM HALTED IN SET STFFNESS SPACE EXCEEDED

Bandwidth is too high and stiffness dimensioning is exceeded, bandwidth must be decreased.

PROGRAM HALTED IN SOLVE
NEGATIVE OR ZERO DIAGONAL STIFFNESS

Stiffness matrix is not positive definite, error in material parameters or problem has rigid body modes.

ERROR-- FRACTURE SCPACING (DELTA) MUST BE POSITIVE & NONZERO

Fracture spacing has been input improperly

ERROR-- HALTED IN SLIPP, ERROR IN CONNECTIVITY

When slip plane is being added, improper slip node numbering leads to this error. Caused by improper mesh generation or adding of a slip plane along a free edge.

ERROR-- NODE I NOT FOUND IN CONNECTIVITY ARRAY

Program cannot find new node number of a node in the nodal mapping. Caused by error in mesh generation.

****
****

APPENDIX C

SCRUBS CODE LISTING

The following appendix contains the program listing of the computer code SCRUBS. This listing is included since SCRUBS was greatly updated in both logic and programming as this thesis was conducted. The subroutines will be found in alphabetical order. A summary of these subroutines with their descriptions can be found in appendix A.

```
*------------------------------------------------------------
*------------------------------------------------------------
*
*         Subroutine BSUB:
*         Conducts systematic back-substitution and solves for boundary
*         displacements and reaction transformations.
*
*         Written:
*         Last modified:JUL 24 1984
*
*------------------------------------------------------------
*------------------------------------------------------------
*
*         Called by:  SCRUBS
*
*------------------------------------------------------------
*------------------------------------------------------------

          SUBROUTINE BSUB

          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          DIMENSION DIS(2,500)
          COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1    NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2    NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3    TPWORK
          COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
          COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
     1    US(400,2)
          COMMON/LDS/R1(2000),RL(2000),RT(2000)
          COMMON/SCR/SK(20000) ,R
          EQUIVALENCE (SK(1),DIS(1,1))

          REWIND 3
          BACKSPACE 4
          ND=ND-1
          KJ=NDF*NP
          DIS(NDF,NP)=R1(KJ)
          R1(KJ)=0.
          I=NDF
          L=NP
          NBUF=NBUF-2
          IF(NNP.EQ.1) BACKSPACE 4
          GO TO 40
```

```
*-----------------------------------------------------------------
*
*               Start systematic back substitution.
*
*-----------------------------------------------------------------

10              L=L-1
20              I=I-1
                NNP=SK(NBUF)
                IF(NNP.EQ.1) BACKSPACE 4
                NBUF=NBUF-1
                NSZ=SK(NBUF)
                IA=NBUF-NSZ
                NBUF=IA-3
                KJ=NDF*(L-1)+I
                GASH=R1(KJ)
                R1(KJ)=0.
                DO 30 J=1,NSZ
                  LJ=L+(J+I-1)/NDF
                  K=I+J-(LJ-L)*NDF
                  GASH=GASH-SK(IA)*DIS(K,LJ)
30              IA=IA+1
                DIS(I,L)=GASH
40              IF(SK(NBUF+2).GT.0.) GO TO 50
                R1(KJ)=DIS(I,L)
                DIS(I,L)=U(ND )
                ND=ND-1
50              CONTINUE
                IF(I+L-2) 60,90,60
60              IF(NNP.NE.1) GO TO 70
                READ(4) NBUF,(SK(II),II=NSK,NBUF)
                BACKSPACE 4
70              IF(I.NE.1) GO TO 20
80              I=NDF+1
                GO TO 10
90              CONTINUE

*-----------------------------------------------------------------
*
*               Boundary displacements and reaction transformation.
*
*-----------------------------------------------------------------

                NOSH=0
                DO 110 M=1,NB
```

```
            MOSH=NOSH+1
            NOSH=MOSH+1
            N=NBC(M)
            NASH=2•N
            MASH=NASH-1
            RX=R1(MASH)
            RY=R1(NASH)
            IF(ANG(M).EQ.0.) GOTO 100
            GASH=ANG(M)•.017453292
            CS=COS(GASH)
            TN=TAN(GASH)
            DXC=DIS(1,N)•CS
            DYC=DIS(2,N)•CS
            DIS(1,N)=DXC-DYC•TN
            DIS(2,N)=DYC+DXC•TN
            RXC=RX•CS
            RYC=RY•CS
            RX=RXC-RYC•TN
            RY=RYC+RXC•TN
100         TRC(MOSH)=TRC(MOSH)+RX
            TRC(NOSH)=TRC(NOSH)+RY
            RT(MASH)=RT(MASH)+RX
            RT(NASH)=RT(NASH)+RY
110       CONTINUE

          RETURN
          END
```

```
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*           Subroutine CONRUB:
*           This subroutine checks for rubble/continuum interaction and
*           computes the hydrostatic force in the rubble if a
*           continuum element is in contact.
*
*           Written:
*           Last modified:JUL 19 1984
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*           Called by:  RESIDUE
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------

            SUBROUTINE CONRUB(SIGRES,M)

            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,IPS,
     1      NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2      NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3      TPWORK,KNE
            COMMON/FG/ FLAG(1000),BULK(1000,2),MINE(10,50),NME(10),GAMMA
            COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1      ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)
            COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)

            SIGRES=.0001

*-----------------------------------------------------------------
*
*           Computes area of rubble element.
*
*-----------------------------------------------------------------

           K1=(M-1)*4*NSFR+1
           K2=K1+NSFR
           K3=K2+NSFR
           K4=K3+NSFR
           J1=NOP(K1)
           J2=NOP(K2)
```

```
          J3=NOP(K3)
          J4=NOP(K4)
          COE1=CORD(J2,1)-CORD(J4,1)
          COE2=CORD(J3,2)-CORD(J1,2)
          COE3=CORD(J3,1)-CORD(J1,1)
          COE4=CORD(J4,2)-CORD(J2,2)
          AREA=.5*(COE1*COE2+COE3*COE4)
          IF(AREA.GE.BULK(M,2)) RETURN

*-------------------------------------------------------------
*
*         Computes hydrostatic compaction force since contact is established.
*
*-------------------------------------------------------------

          BSTN=(AREA-BULK(M,2))/BULK(M,2)
          SIGRES=BSTN*ABS(ORT(NMAT,1))
          WRITE(6,10) M,AREA,BULK(M,2),SIGRES
10        FORMAT('  M,AREA,BULK(M,2),SIGRES  !!!!!!!!',I5,3E13.3)

          RETURN
          END


*-------------------------------------------------------------
*-------------------------------------------------------------
*
*         Subroutine CONVG:
*         Determines the percent convergence.
*
*         Written:
*         Last modified:JUL 19 1984
*
*-------------------------------------------------------------
*-------------------------------------------------------------
*
*         Called by:  SCRUBS
*
*-------------------------------------------------------------
*-------------------------------------------------------------

          SUBROUTINE CONVG(NCHECK)

          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          DIMENSION RAB(2)
```

```
          COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1    NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2    NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,MALGO,PWORK,
     3    TPWORK
          COMMON/LDS/R1(2000),RL(2000),RT(2000)
          COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1    ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)
     2    ,STRS(4),STRN(4),R(3),RMAX(2),RTOT(3)
          COMMON/FG/ FLAG(1000),BULK(1000,2),MINES(10,50),NME(10),GAMMA

C         MASH=NCHECK=0
          MASH=0.0
          NCHECK=0.0
C         R(1)=R(2)=RMAX(1)=RMAX(2)=RTOT(1)=RTOT(2)=SUMD=SUMTD=0.
          R(1)=0.0
          R(2)=0.0
          RMAX(1)=0.0
          RMAX(2)=0.0
          RTOT(1)=0.0
          RTOT(2)=0.0
          SUMD=0.0
          SUMTD=0.0
          DO 20 I=1,NP
            DO 20 K=1,2
              MASH=MASH+1
              GISH=RT(MASH)
              GOSH=DIS(K,I)
              GESH=DISN(K,I)
              GASH=GISH-R1(MASH)
C       IF (I.GE.1.AND.I.LE.7) WRITE (6,10) I,K,RT(MASH),
C    1    R1(MASH),GASH
C             IF (I.GE.12.AND.I.LE.18) WRITE (6,10) I,K,RT(MASH),
C    1    R1(MASH),GASH
10      FORMAT('  ••• I,K,RT(MASH),R1(MASH),GASH.••' 2I5,3E15.5)
             R1(MASH)=GASH
             R(K)=R(K)+GASH•GASH
             RTOT(K)=RTOT(K)+GISH•GISH
             SUMD=SUMD+GOSH•GOSH
             SUMTD=SUMTD+GESH•GESH
             GASH=ABS(GASH)
             IF(GASH.GT.RMAX(K))RMAX(K)=GASH
20        CONTINUE
          R(3)=R(1)+R(2)
          RTOT(3)=RTOT(1)+RTOT(2)
          DO 30 K=1,3
```

```
                R(K)=SQRT(R(K))
                RTOT(K)=SQRT(RTOT(K))
                IF(RTOT(K).LT. 1E-10) RTOT(K)=1.
30            R(K)=100.•R(K)/RTOT(K)
              SUMD=SQRT(SUMD)
              SUMTD=SQRT(SUMTD)
        IF(SUMTD.LT. 1E-20) SUMTD=1.
              IF(SUMD.LT.  1E-20) SUMD=1.
              SUM=100.•SUMD/SUMTD
              IF(SUM.GT.CONFAC)  NCHECK=1
              WRITE(6,40)(RTOT(K),K=1,3),(R(I),I=1,3)
40            FORMAT(/5X,'X-NORM OF TOTAL LOADS =',E18.6/
    1         5X,'Y-NORM OF TOTAL LOADS =',E18.6/
    2         5X,'NORM OF SUM OF TOTAL LOADS =',E13.6/
    3         5X,'X-NORM RESIDUAL RATIO =',E18.6/
    4         5X,'Y-NORM RESIDUAL RATIO =',E18.6/
    5  5X,'NORM OF RESIDUAL SUM RATIO =',E13.6,5X)
              WRITE(6,50)SUMD,SUMTD,SUM,NCHECK
50            FORMAT(/5X,'NORM OF CURRENT INCREMENTED DISPLACEMENT =',E18.6/
    1         5X,'NORM OF CURRENT TOTAL INCREMENTED DISPLACEMENT =',E12.6/
    2         5X,'NORM OF DISPLACEMENT RATIO =',E18.6,5X,'NCHECK =',I2)
              IF(IT.GT.1) GO TO 60
              RAB(1)=RMAX(1)
              RAB(2)=RMAX(2)
              GO TO 80
60            DO 70 I=1,2
70            RMAX(I)=100.•RMAX(I)/RAB(I)
80            WRITE(6,90)(RMAX(K),K=1,2)
90            FORMAT(/5X,'MAXIMUM X-RESIDUAL =',E18.6/
    1         5X,'MAXIMUM Y-RESIDUAL =',E18.6)
              PRATIO=-100.
              TPWORK=TPWORK+PWORK
              IF(TPWORK.NE.0.)PRATIO=100.•PWORK/TPWORK
              WRITE(6,100)PWORK,TPWORK,PRATIO
100           FORMAT(/5X,'ITERATIVE PLASTIC WORK =',E18.6,2X,'INCREMENTAL'
    1         ' PLASTIC WORK =',E18.6/
    2         5X,'PERCENTAGE PLASTIC WORK =',F7.2)

              RETURN
              END
```

```
*------------------------------------------------------------
*------------------------------------------------------------
*
*          Subroutine CORNN:
*          This subroutine determines if node IX is a corner node.
*
*          Written:
*          Last modified:JUL 24 1984
*
*------------------------------------------------------------
*------------------------------------------------------------
*
*          Called by:  SLIPP
*
*------------------------------------------------------------
*------------------------------------------------------------

          SUBROUTINE CORNN(NADD,NSFR,NP,IFLAG,IX)

          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
   1      ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000),DUMMY(1000)
          REAL CORN

          J=0
10        J=J+1
          IF(J.GT.4000) THEN
            WRITE(6,20)IX
20          FORMAT(' ERROR -- NODE ',I3,' NOT FOUND IN '
   1      'CONNECTIVITY ARRAY')
            STOP
          ELSE
            IF(NOP(J).NE.IX)GOTO 10
            CORN=FLOAT(J-1)/FLOAT(NSFR)
      CORN=CORN-IFIX(CORN)
            IF(CORN.NE.0.0)RETURN
            NADD=NSFR

*------------------------------------------------------------
*
*          If it is a corner node, IFLAG = 1.
*
*------------------------------------------------------------

            IFLAG=1
```

```
      RETURN
     ENDIF

        END

*-------------------------------------------------------------
*-------------------------------------------------------------
*
*          Subroutine DATALST:
*          Writes title, line numbers for files (presently not used).
*
*          Written:
*          Last modified:JUL 24 1984
*
*-------------------------------------------------------------
*-------------------------------------------------------------
*
*          Called by:  SCRUBS
*
*-------------------------------------------------------------
*-------------------------------------------------------------

           SUBROUTINE DATALST(IOIN,IOSC,IOPR)

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/DLIST/ LN,LNCT,TITL(20)
C          DIMENSION IA(8)
           DIMENSION IA(20)

10         LN=LNCT
           WRITE(IOPR,20) TITL
C20        FORMAT(6X,8A10//
20         FORMAT(6X,20A4//
    1  14X,'10',8X,'20',8X,'30',8X,'40',8X,'50',8X,'60',8X,'70',8X,
    2  '80',/,6X,'12345678901234567890123456789012345678901234567 8'
    3          '901234567890123456789012345678 90')
C30        READ(IOIN,40) IA
30         READ(IOIN,40,END=70) IA
C40        FORMAT(8A10)
40         FORMAT(20A4)
C          IF(EOF(IOIN)) 70,50
50         WRITE(IOSC,40) IA
           WRITE(IOPR,60) IA
C60        FORMAT(6X,8A10)
60         FORMAT(6X,20A4)
```

```
            LN=LN-1
            IF(LN.LE.0) GO TO 10
            GO TO 30
70          ENDFILE IOSC
            REWIND IOSC

            RETURN
            END

*---------------------------------------------------------------
*---------------------------------------------------------------
*
*           Subroutine DROP:
*           Checks for elements about to fail and adjusts geometry for
*           failure.
*
*           Written:
*           Last modified:JUL 19 1984
*
*---------------------------------------------------------------
*---------------------------------------------------------------
*
*           Called by:  RESIDUE
*
*---------------------------------------------------------------
*---------------------------------------------------------------
*
*           Subroutines called:RZBASEEXIT
*
*---------------------------------------------------------------
*---------------------------------------------------------------

            SUBROUTINE DROP(NUMEL,NUMNP,AK)

            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1      NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2      NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3      TPWORK,KNE
            COMMON/FG/ FLAG(1000),BULK(1000,2),MINES(10,50),NME(10),GAMMA
            COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1      ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000),DUMMY(1000)
            COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
     1      US(400,2)
            COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
```

```
      DIMENSION IBTMS(4),MSN(2,4)

*-----------------------------------------------------------
*
*     Check for elements ready to fall.
*
*-----------------------------------------------------------

      DO 200 I=1,NUMNP
        FLG=FLAG(I)
        IF(FLG.EQ.4.) GO TO 20
        IF(FLG.EQ.8.) GO TO 10
        GO TO 200
10      CONTINUE
        NN=2*I
        ZOLD=CORD(I,2)-TDIS(2,I)
        ZNEW=AK
        GO TO 100
20      CONTINUE

*-----------------------------------------------------------
*
*     Drop to znew.
*
*-----------------------------------------------------------

        ZOLD=CORD(I,2)-TDIS(2,I)

*-----------------------------------------------------------
*
*     Find bottom element numbers.
*
*-----------------------------------------------------------

        IBTMS(1)=0
        IBTMS(2)=0
        IBTMS(3)=0
        IBTMS(4)=0
        ICNT=0
        L=NSFR*4
        DO 40 N=1,NUMEL
          DO 30 K=1,4
            KK=(N-1)*NSFR*4+K
            IF(NOP(KK).NE.I) GO TO 30
            ICNT=ICNT+1
```

```
              IBTMS(ICNT)=N
30          CONTINUE
40        CONTINUE
          GO TO (50,60,70,70) ICNT

*------------------------------------------------------------------
*
*        ICNT=1, top corner node drop.
*
*------------------------------------------------------------------

50        IBOT=IBTMS(1)
          MX=IMAT(IBOT)
          IF(MX.GE.100) MX=MX-100
          AREA=BULK(IBOT,2)
          ALPH=ORT(MX,8)
          ANEW=AREA/ALPH
          IREF=(IBOT-1)*NSFR*4
          I1=NOP(IREF+1)
          I2=NOP(IREF+1+NSFR)
          I3=NOP(IREF+1+NSFR*2)
          I4=NOP(IREF+1+NSFR*3)
          CALL RZBASE(I,IBOT,I1,I2,I3,I4,R,Z,RLEN,ZBASE)
          ZLEN=ANEW/RLEN
          GO TO 90

*------------------------------------------------------------------
*
*        ICNT=2, side boundary element drop.
*
*------------------------------------------------------------------

60        CONTINUE
          DO 70 K=1,ICNT
            IBOT=IBTMS(K)
            IREF=(IBOT-1)*NSFR*4
            I1=NOP(IREF+1)
            I2=NOP(IREF+1+NSFR)
            I3=NOP(IREF+1+NSFR*2)
            I4=NOP(IREF+1+NSFR*3)
            ZAVG=(CORD(I1,2)+CORD(I2,2)+CORD(I3,2)+CORD(I4,2))*.25
            ZAVG=ZAVG-.25*(TDIS(2,I1)+TDIS(2,I2)+TDIS(2,I3)+
     1       TDIS(2,I4))
            MX=IMAT(IBOT)
            IF(MX.GE.100) MX=MX-100
```

```
             ALPH=ORT(MX,8)
             AREA=BULK(IBOT,2)
             IF(ZAVG.GT.ZOLD) GO TO 75
             MX=IMAT(IBOT)
             IF(MX.GE.100) MX=MX-100
             ALPH=ORT(MX,8)
             AREA=BULK(IBOT,2)
             ANEW=AREA/ALPH
             CALL RZBASE(I,IBOT,I1,I2,I3,I4,R,Z,RLEN,ZBASE)
             ZLEN=ANEW/RLEN
75         CONTINUE
           GO TO 90

*-----------------------------------------------------------------
*
*         ICNT=4,  interior node drop.
*
*-----------------------------------------------------------------

70         CONTINUE
           ZBASET=0.0
           RLENT=0.0
           ALPH=0.0
           AREA=0.0
           ZDST=0.0
           DO 80 K=1,ICNT
            IBOT=IBTMS(K)
            IREF=(IBOT-1)*NSFR*4
            I1=NOP(IREF+1)
            I2=NOP(IREF+1+NSFR)
            I3=NOP(IREF+1+NSFR*2)
            I4=NOP(IREF+1+NSFR*3)
            ZAVG=(CORD(I1,2)+CORD(I2,2)+CORD(I3,2)+CORD(I4,2))*.25
            ZAVG=ZAVG-.25*(TDIS(2,I1)+TDIS(2,I2)+TDIS(2,I3)+
  1       TDIS(2,I4))
            IF(ZAVG.GT.ZOLD) GO TO 80
            MX=IMAT(IBOT)
            IF(MX.GE.100) MX=MX-100
            ALPH=ALPH+ORT(MX,8)
            AREA=AREA+BULK(IBOT,2)
            CALL RZBASE(I,IBOT,I1,I2,I3,I4,R,Z,RLEN,ZBASE)
            RLENT=RLENT+RLEN
            ZBASET=ZBASET+ZBASE
            C1=CORD(I1,2)-TDIS(2,I1)
            C2=CORD(I2,2)-TDIS(2,I2)
```

```
              C3=CORD(I3,2)-TDIS(2,I3)
              C4=CORD(I4,2)-TDIS(2,I4)
              ZD1=ABS(C3-C1)
              ZD2=ABS(C2-C4)
              ZZZ=.5*(ZD1+ZD2)
              ZDST=ZDST+ZZZ
80          CONTINUE
            ALPH=.5*ALPH
            ANEW=AREA/ALPH
C           ZLEN=ANEW/RLENT

*----------------------------------------------------------------
*
*          ZLEN is recomputed here to circumvent errors when sloping elements
*          are used, ZLEN is computed as the average of the bulked vertical
*          distance between diagonals of one of the bottom elements.
*
*----------------------------------------------------------------

            ZLEN=.5*ZDST
            ZLEN=ZLEN/ALPH
            ZBASE=.5*ZBASET
90          CONTINUE
            ZNEW=ZBASE+ZLEN
            IF(ZNEW.GT.ZOLD) ZNEW=ZOLD
100         CONTINUE
            FLAG(I)=10.
            DRP=ZNEW-ZOLD
            CORD(I,2)=ZNEW
            TDIS(2,I)=DRP
            WRITE(6,110) I,ZNEW,ZOLD,DRP,AK,ZLEN,ZBASE,ANEW,RLEN,
  1          ALPH,AREA
110         FORMAT(' I,ZNEW,ZOLD,DRP,AK,ZLEN,ZBASE,ANEW,RLEN,'
  1          'ALPH,AREA',/I5,5E15.5/5E15.5)
            IF(NSFR.NE.2)GO TO 140
            DO 130 IE=1,4
              N=(IBTMS(IE)-1)*8
              DO 120 M=1,7,2
                MM=M+N
                IF(NOP(MM).NE.I)GO TO 120
                MSN(1,IE)=NOP(MM+1)
                IF(MM.EQ.7)MSN(2,IE)=NOP(MM-6)
120           CONTINUE
              IM=MSN(1,IE)
              II=MSN(2,IE)
```

```
                 CORD(IM,2)=(CORD(I,2)+CORD(II,2))/2.0
130            TDIS(2,IM)=(TDIS(2,I)+TDIS(2,II))/2.0

•---------------------------------------------------------------
•
•             Until bulking is defined, fix nodes after fall.
•
•---------------------------------------------------------------

140            NBTOP=NB
               DO 160 KK=1,NB
                 ICHECK=NBC(KK)
                 IF(ICHECK.EQ.I) NFIX(KK)=11
                 IF(ICHECK.EQ.I) GO TO 170
                 IF(ICHECK.LE.I) GO TO 160
                 NBTOP=NB+1
                 NBCN=I
                 NFIXN=11
                 DO 150 JJ=KK,NBTOP
                   NBCMOV=NBC(JJ)
                   NFIXMOV=NFIX(JJ)
                   NBC(JJ)=NBCN
                   NFIX(JJ)=NFIXN
                   NBCN=NBCMOV
                   NFIXN=NFIXMOV
150              CONTINUE
                 GO TO 170
160            CONTINUE
               NBTOP=NB+1
               NBC(NBTOP)=I
               NFIX(NBTOP)=11
170            NB=NBTOP
               IF(NBTOP.LE.400) GO TO 190
               WRITE(6,180)
180            FORMAT(' ••••••••• NB EXCEEDS DIMENSIONED VALUE IN '
   1            'DROP•••••')
               CALL EXIT
190            CONTINUE
200          CONTINUE

•---------------------------------------------------------------
•
•             Find bulked elements, set BULK(N,1))=2.0
•
•---------------------------------------------------------------
```

```
          DO 210 I=1,NUMEL
            IREF=(I-1)*NSFR*4
            I1=NOP(IREF+1)
            I2=NOP(IREF+1+NSFR)
            I3=NOP(IREF+1+NSFR*2)
            I4=NOP(IREF+1+NSFR*3)
            SUM=FLAG(I1)+FLAG(I2)+FLAG(I3)+FLAG(I4)
            IF(SUM.NE.40.) GO TO 210
            BULK(I,1)=2.0
210       CONTINUE

          RETURN
          END

*-------------------------------------------------------
*-------------------------------------------------------
*
*         Subroutine FLAGE:
*         Makes impact node calculations and new node assignments.
*
*         Written:
*         Last modified:JUL 19 1984
*
*-------------------------------------------------------
*-------------------------------------------------------
*
*         Called by:  SCRUBS
*
*-------------------------------------------------------
*-------------------------------------------------------
*
*         Subroutines called:NEWNOD        EXIT
*
*-------------------------------------------------------
*-------------------------------------------------------

          SUBROUTINE FLAGE (NUMNP,NUMEL,NTAPE,IIF)

          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1    NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2    NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3    TPWORK,KNE
          COMMON/FG/ FLAG(1000),BULK(1000,2),MINES(10,50),NME(10),GAMMA
          COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
```

```
     1          ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000),DUMMY(1000)
                DIMENSION IBIE(100)

*-----------------------------------------------------------
*
*               Reads number of impact boundary nodes and writes it to
*               SCRUBS.LIS.
*
*-----------------------------------------------------------

                READ(NTAPE,10) NBIE
10              FORMAT(I5)
                WRITE(6,20) NBIE
20              FORMAT('  NUMBER OF BOUNDARY IMPACT NODES IS',I5)
                IF(NBIE.GT.100) GO TO 140
                IF(NBIE.EQ.0) GO TO 80
                NCARD=1+NBIE/16

*-----------------------------------------------------------
*
*               Writes impact node list after updating.
*
*-----------------------------------------------------------

                WRITE(6,30)
30              FORMAT('  IMPACT NODE LIST ')
                DO 80 K=1,NCARD
                  KSTART=1+(K-1)*16
                  KEND=KSTART+15
                  READ(NTAPE,40) (IBIE(KK),KK=KSTART,KEND)
40                FORMAT(16I5)
                  IF(IIF.GE.0) GOTO 60
                  DO 50 KK=KSTART,KEND
50                CALL NEWNOD(IBIE(KK))
60                WRITE(6,70)(IBIE(KK),KK=KSTART,KEND)
70                FORMAT(1X,16I5)
80              CONTINUE

*-----------------------------------------------------------
*
*               Makes impact node calculations.
*
*-----------------------------------------------------------

                DO 100 I=1,NUMNP
```

```
              ICOUNT=0
              FLAG(I)=0.0
              L=NSFR•4
              DO 90 J=1,NUMEL
                DO 90 K=1,L,NSFR
                  KK=NSFR•4•(J-1)+K
                  IF(NOP(KK).EQ.I) ICOUNT=ICOUNT+1
90            CONTINUE
              IF(ICOUNT.EQ.2) FLAG(I)=2.0
              IF(ICOUNT.EQ.1) FLAG(I)=3.0
100         CONTINUE
            IF(NSFR.NE.2) GOTO 120
            NOP8=NUMEL•8
            DO 110 I=2,NOP8,2
              MDSD=NOP(I)
110         FLAG(MDSD)=20.0
120         CONTINUE
            DO 130 I=1,NBIE
              IFIX=IBIE(I)
              IF(FLAG(IFIX).EQ.2.) FLAG(IFIX)=6.
              IF(FLAG(IFIX).EQ.3.) FLAG(IFIX)=7.
130         CONTINUE
            RETURN

•------------------------------------------------------------
•
•           Error trap for excessive number of boundary nodes.
•
•------------------------------------------------------------

140         WRITE(6,150) NBIE
150         FORMAT(' NBIE IS INPUT AS',I5,'BUT CANNOT BE LARGER
    +       'THAN 100')
            CALL EXIT

            END
```

```
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*          Subroutine GAUSSP: Initializes gaussian constants.
*
*          Written:
*          Last modified:MAR 19 1984
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*          Called by: GDATA
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------

           SUBROUTINE GAUSSP(M,X,W)

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           DIMENSION X(6),W(6)

           GO TO (10,20,30,40,50,60),M
10         X(1)=0.
           W(1)=1.
           RETURN
20         X(1)=-0.577350269189626
           W(1)=1.
           GO TO 70
30         X(1)=-0.774596669241483
           X(2)=0.
           W(1)=0.555555555555556
           W(2)=0.888888888888889
           GO TO 70
40         X(1)=-0.861136311594053
           X(2)=-0.339981043584856
           W(1)=0.347854845137454
           W(2)=0.652145154862546
           GO TO 70
50         X(1)=-0.906179845938664
           X(2)=-0.538469310105683
           X(3)=0.
           W(1)=0.236926885056189
           W(2)=0.478628670499366
           W(3)=0.568888888888889
           GO TO 70
```

```
60          X(1)=-0.932469354203152
            X(2)=-0.661209386466265
            X(3)=-0.238619186083197
            W(1)=0.171324492379170
            W(2)=0.360761573048139
            W(3)=0.467913934572691
70          N=M/2
            DO 80 J=1,N
              NASH=M+1-J
              X(NASH)=-X(J)
80          W(NASH)=W(J)

            RETURN
            END

*----------------------------------------------------------
*----------------------------------------------------------
*
*           Subroutine GDATA:
*           Reads geometrical data, boundary conditions, and gaussian
*           integration data.
*
*           Written:
*           Last modified:JUL 24 1984
*
*----------------------------------------------------------
*----------------------------------------------------------
*
*           Called by:SCRUBS
*
*----------------------------------------------------------
*----------------------------------------------------------
*
*           Subroutines called:SLIPPNODEXY
*           NEWNOD  GAUSSP
*
*----------------------------------------------------------
*----------------------------------------------------------

            SUBROUTINE GDATA

            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
   1 NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
   2 NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO, PWORK,
```

```
   3 TPWORK
          COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
          COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
   1      US(400,2)
          COMMON/GID/XG(6),CG(6),XMG(6),CMG(6),NGAUS,MGAUS,MCN
          COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
   1       NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)
          DIMENSION NOPE(12),DX(3),QUID(3),IB(12),NOPD(12)
          REAL CORDDUM(1000,2)

*---------------------------------------------------------
*
*         Reads number of slip planes (NNP), singular points (NSING),
*         boundary points (NNBB), summed reaction nodes (MPR),
*         indexing vector (IB), and writes to SCRUBS.LIS.
*
*---------------------------------------------------------

          READ(5,5) NNP,NSING,NNBB,MPR,(IB(I),I=1,12)
5         FORMAT(16I5)
          IF(NB.GT.0.AND.NNBB.EQ.0) NNBB=NB
          WRITE(6,7)NNP,NSING,MPR,(IB(I),I=1,12)
7         FORMAT(1X,/
   1      '        NUMBER OF SLIP PLANES---------',I3//
   2      '        NUMBER OF SINGULAR POINTS-----',I3//
   3      '        NO OF SUMMED REACTION NODES---',I3//
   4      '        IB VECTOR---------------------',12I4)
          IF(IB(1).NE.0) GO TO 40
          DO 10 I=1,12
10        IB(I)=NSFR
          IF(NSFR-2)40,30,20
C20       IB(5)=IB(6)=IB(11)=IB(12)=1
20        IB(5)=1
          IB(6)=1
          IB(11)=1
          IB(12)=1
          GO TO 40
C30       IB(4)=IB(8)=1
30        IB(4)=1
          IB(8)=1
40        CONTINUE
          DO 50 II=1,NP
            DO 50 JJ=1,NCORD
50        CORD(II,JJ) = 0.
```

```
*------------------------------------------------------------
*
*              Reads nodal point data from QMESH9.DAT.
*
*------------------------------------------------------------

               READ(9) (CORDDUM(N,1),CORDDUM(N,2),N=1,NP)
               DO 60 N=1,NP
                 CORD(N,1)=CORDDUM(N,1)
60             CORD(N,2)=CORDDUM(N,2)

*------------------------------------------------------------
*
*              Reads element data from QMESH9.DAT.
*
*------------------------------------------------------------

               IF (NCN.EQ.8) GO TO 70
               READ(9)(((NOP(NCN*(K-1)+M),M=1,NCN),IMAT(K)),K=1,NE)
               GO TO 80
70             READ(9)((NOP(8*(K-1)+1),NOP(8*(K-1)+3),NOP(8*(K-1)+5),
     *     NOP(8*(K-1)+7),NOP(8*(K-1)+2),NOP(8*(K-1)+4),
     *     NOP(8*(K-1)+6),NOP(8*(K-1)+8),IMAT(K)),K=1,NE)

*------------------------------------------------------------
*
*              Reads boundary data from QMESH9.DAT.
*
*------------------------------------------------------------

80             IF(NNBB.LE.0) GOTO 100
               DO 90 I=1,NNBB
90             READ (5,95) NBC(I),NFIX(I), (US(I,J),J=1,NDF) ,ANG(I)
95             FORMAT(2I5,4F10.3)
100            CONTINUE


*------------------------------------------------------------
*
*              Adds slip plane if needed.
*
*------------------------------------------------------------

               IF(NNP.GT.0)CALL SLIPP
```

```
*------------------------------------------------------------
*
*          Generates coordinates for intermediate nodes.
*
*------------------------------------------------------------

C          IF(NSFR.GT.1) CALL NODEXY
           IF(NSING.EQ.0) GOTO 140
           IF(NNP.GT.0) CALL NEWNOD(NSING)

*--------------------------------------------------------
*
*          Singular point calculations.
*
*--------------------------------------------------------

           DO 130 K=1,NE
             DO 110 I=1,8
               ICK=NOP((K-1)*8+I)
               IF(ICK.EQ.NSING) GOTO 120
110          CONTINUE
             GOTO 130
120          IFP=I-2
             IF(IFP.EQ.-1) IFP=7
             IF(IFP.EQ.0) IFP=8
             NFAR=NOP((K-1)*8+IFP)
             XQP=0.25*CORD(NFAR,1)+0.75*CORD(NSING,1)
             ZQP=0.25*CORD(NFAR,2)+0.75*CORD(NSING,2)
             IQP=I-1
             IF(IQP.EQ.0) IQP=8
             NQP=NOP((K-1)*8+IQP)
             CORD(NQP,1)=XQP
             CORD(NQP,2)=ZQP
             IFP=I+2
             IF(IFP.EQ.9) IFP=1
             IF(IFP.EQ.10) IFP=2
             NFAR=NOP((K-1)*8+IFP)
             XQP=0.25*CORD(NFAR,1)+0.75*CORD(NSING,1)
             ZQP=0.25*CORD(NFAR,2)+0.75*CORD(NSING,2)
             IQP=I+1
             IF(IQP.EQ.9) IQP=8
             NQP=NOP((K-1)*8+IQP)
             CORD(NQP,1)=XQP
             CORD(NQP,2)=ZQP
130        CONTINUE
```

```
140         CONTINUE
            IF(NB.EQ.0) GO TO 180
*-----------------------------------------------------------------
*
*           Writes fixed boundary conditions to SCRUBS.LIS.
*
*-----------------------------------------------------------------


            WRITE(6,145)
145         FORMAT(' BOUNDARY CONDITIONS'/
  1         ' NODE  FIX  X VALUE   Y VALUE     ANGLE')
            DO 150 I=1,NB
150         WRITE (6,95) NBC(I),NFIX(I), (US(I,J),J=1,NDF),ANG(I)
            IF(MPR.EQ.0) GO TO 180
            READ (5,5)(NOPR(I),I=1,MPR)
            IF(IIF.GE.0) GOTO 170
            DO 160 I=1,MPR
160         CALL NEWNOD(NOPR(I))
170         WRITE(6,5)(NOPR(I),I=1,MPR)
180         CONTINUE

*-----------------------------------------------------------------
*
*           Reads gaussian integration data and writes constants to SCRUBS.LIS.
*
*-----------------------------------------------------------------

            WRITE(6,190)
190         FORMAT('  GAUSSIAN INTEGRATION CONSTANTS')
            CALL GAUSSP(NGAUS,XG,CG)
            DO 210 I=1,NGAUS
              WRITE(6,200)XG(I),CG(I)
200           FORMAT(2F15.8)
210         CONTINUE
            CALL GAUSSP(MGAUS,XMG,CMG)
            DO 220 I=1,MGAUS
220         WRITE(6,200)XMG(I),CMG(I)

            RETURN
            END
```

```
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*          Subroutine INVAR:
*          Calculates invariants and sets the yield stress accordingly.
*
*          Written:
*          Last modified:JUL 24 1984
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*          Called by:  RESIDUESTIFM
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------

      SUBROUTINE INVAR   (SSS)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1 NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2 NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3 TPWORK
      COMMON/INV/SMEAN,STJ2,STJ3,SIGMA,PHI,APNI,STN,CPHI,YIELD
      COMMON/MATP/D(4,4),YM,PR,GM,HM,PPR,CLAME,ALPHA,YST,
     1 TABSTN(10),TABSTR(10,15)
      DIMENSION SSS(4)

*-------------------------------------------------------------
*
*          Calculates invariants.
*
*-------------------------------------------------------------

      X=SSS(1)
      Y=SSS(2)
      Z=SSS(4)
      XY=SSS(3)
      SMEAN=(X+Y+Z)/3.
C     SSS(1)=X=X-SMEAN
      TEMP=X-SMEAN
      X=TEMP
      SSS(1)=TEMP
C     SSS(2)=Y=Y-SMEAN
```

```
            TEMP=Y-SMEAN
            Y=TEMP
            SSS(2)=TEMP
C           SSS(4)=Z=Z-SMEAN
            TEMP=Z-SMEAN
            Z=TEMP
            SSS(4)=TEMP
            STJ2=XY*XY+.5*(X*X+Y*Y+Z*Z)
            IF(STJ2.EQ.0.0) STJ2=.00001
            STJ3=Z*(Z*Z-STJ2)
            SIGMA=SQRT(STJ2)
            STN=-2.5980762113*STJ3/(STJ2*SIGMA)
            IF(STN.LT.1.0.AND.STN.GT.-1.0) GO TO 10
            IF(STN.LE.-.9999) STN=-.9999
            IF(STN GE..9999) STN=.9999
10          CONTINUE
            PHI=ASIN(STN)/3.
            CPHI=COS(PHI)
            CON=1.73205080756888
            MYIELD=NYIELD
            IF(NYIELD.GT.100)MYIELD=NYIELD-100
            GO TO (20,30,40,50),MYIELD

*------------------------------------------------------------
*
*           Von Mises yield condition.
*
*------------------------------------------------------------

20          YIELD=CON*SIGMA
            RETURN

*------------------------------------------------------------
*
*           Tresca yield condition.
*
*------------------------------------------------------------

30          YIELD=2.*SIGMA*CPHI
            RETURN

*------------------------------------------------------------
*
*           Drucker yield condition.
*
```

```
*----------------------------------------------------------------

40          YIELD=3.*ALPHA*SMEAN+SIGMA
            RETURN

*----------------------------------------------------------------
*
*           Beltrami yield condition.
*
*----------------------------------------------------------------

50          YIELD=SQRT((3.+6.*PR)*SMEAN*SMEAN+2.*(1.-PR)*STJ2)

            RETURN
            END

*----------------------------------------------------------------
*----------------------------------------------------------------
*
*           Subroutine JOINTM:
*           Calculates incremental stresses and residuals.
*
*           Written:
*           Last modified:JUL 19 1984
*
*----------------------------------------------------------------
*----------------------------------------------------------------
*
*           Called by:  RESIDUE
*
*----------------------------------------------------------------
*----------------------------------------------------------------
*
*           Subroutines called:ROTATE
*
*----------------------------------------------------------------
*----------------------------------------------------------------

            SUBROUTINE JOINTM(L,STRN,STRS,MOSH,YIELD,PSTRN)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION K,MU,LAMB
      COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1      NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2      NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
```

```
     3          TPWORK
      COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
      COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1          ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)
     2          ,SIG(4),PSIG(4),DSIG(4),EP(4),NG(16),NGSTP(16)
                DIMENSION STRN(4),STRS(4),T(4),TT(4),E(4)

*-----------------------------------------------------------------
*
*               Assigns material parameters.
*
*-----------------------------------------------------------------

                K=-ORT(L,1)
                G=-ORT(L,2)
      CO=ORT(L,3)
      MU=ORT(L,4)
                TH=ORT(L,5)
      FAIL=ORT(L,7)
      DEL=ORT(L,9)
      GS=ORT(L,10)

*-----------------------------------------------------------------
*
*               Checks for delta and Gs equal to zero.
*
*-----------------------------------------------------------------

      IF(GS.EQ.0.0)GS=1.0
                IF(DEL.GT.0) GOTO 20
      WRITE(6,10)
10              FORMAT(/,' ERROR-- FRACTURE SPACING (DELTA) MUST BE'
    &           ' POSITIVE & NONZERO')
                STOP

*-----------------------------------------------------------------
*
*               Rotates incremental strain vector to local system.
*
*       z or y/   /   /
*          ^ /   /   / <- crack array
*          |/   /   /
*          |/   /   /
*          |   /   /
*          |  /   /
```

```
*        |/  /
*        |/  / <- angle theta
*        ----------------> r or x
*
*         Coordinate system for crack array
*
*-----------------------------------------------------------------

20           CONTINUE
      STRN(3)=STRN(3)/2.
             CALL ROTATE(STRN,E,TH)

*-----------------------------------------------------------------
*
*            Gets total stress vector and rotates.
*
*-----------------------------------------------------------------

      DO 30 I=1,4
        STRS(I)=TSTS(I,MOSH)
30           CONTINUE
      CALL ROTATE(STRS,TT,TH)

*-----------------------------------------------------------------
*
*            Solves for incremental stresses.
*
*-----------------------------------------------------------------

      TRTDOT=(K-2.*G/3.)*(E(1)+E(2)+E(4))
             T(1)=2.*G*E(1)+TRTDOT
             T(2)=2.*G*E(2)+TRTDOT
C       T1TOT=TT(1)+T(1)
             T2TOT=TSTS(2,MOSH)+T(2)

*-----------------------------------------------------------------
*
*            Tests for tensile failure of crack.
*            However, for this application allows tensile strength.
*
*-----------------------------------------------------------------

C            FAIL=0.0
C       IF(T1TOT.GT.FAIL) THEN
C         WRITE(6,*)' FAILURE STRESS= ',T1TOT
```

```
C          T(1)=0.0
C        ENDIF
C                FAIL=0.0
C                IF(T2TOT.GT.FAIL) THEN
C                   WRITE(6,*)' FAILURE STRESS= ',T2TOT
C                   TSTS(2,MOSH)=FAIL
C                   T(2)=0.0
C                ENDIF
      T(3)=2.*G*E(3)/(1.+G/DEL/GS)
      T(4)=2.*G*E(4)+TRTDOT

*------------------------------------------------------------
*
*               Checks for plastic slip.
*
*------------------------------------------------------------

C        F=DABS(TT(3)+T(3))+MU*(TT(1)+T(1))-CO
                F=DABS(TT(3)+T(3))+MU*(TT(2)+T(2))-CO
      IF(F.LE.0) GOTO 50
      XX=(2.*G/(1.+G/DEL/GS))
      Y=XX*E(3)+TT(3)

*------------------------------------------------------------
*
*               Solves for plastic strain.
*
*------------------------------------------------------------

C        IF (Y.LE.0) LAMB=E(3)+(CO-MU*(TT(1)+T(1))+TT(3))/XX
C        IF (Y.GT.0) LAMB=E(3)-(CO-MU*(TT(1)+T(1))-TT(3))/XX
                IF (Y.LE.0) LAMB=E(3)+(CO-MU*(TT(2)+T(2))+TT(3))/XX
                IF (Y.GT.0) LAMB=E(3)-(CO-MU*(TT(2)+T(2))-TT(3))/XX

*------------------------------------------------------------
*
*               Determines shear stress for plastic condition.
*
*------------------------------------------------------------

40              CONTINUE
      ELAS=E(3)-LAMB
      T(3)=2.*G*ELAS/(1.+G/DEL/GS)
      EPSTN(5,MOSH)=EPSTN(5,MOSH)+LAMB
```

```
50          CONTINUE
C     YIELD=DABS(TT(1)+T(1))*MU
            YIELD=DABS(TT(2)+T(2))*MU

*-----------------------------------------------------------
*
*           Rotates incremental stress matrix to global system.
*
*-----------------------------------------------------------

      CALL ROTATE(T,STRS,-TH)
      STRN(3)=STRN(3)*2.

      RETURN
      END

*-----------------------------------------------------------
*-----------------------------------------------------------
*
*           Subroutine LDATA:  Reads load control data.
*
*           Written:
*           Last modified:JUL 14 1984
*
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*           Called by:  SCRUBS
*
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*           Subroutines called:SFRAUX
*           NEWNOD
*
*-----------------------------------------------------------
*-----------------------------------------------------------

            SUBROUTINE LDATA(IIF)

            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            DIMENSION R(3)
            DIMENSION NOPL(4),PS(4,2),PQ(2),DJ(2),DENS(10)
            COMMON/FG/ FLAG(1000),BULK(1000,2),MINES(10,50),NME(10),GAMMA
            COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
```

```
     1      NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2      NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO, PWORK,
     3      TPWORK
            COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
            COMMON/GID/XG(6),CG(6),XMG(6),CMG(6),NGAUS,MGAUS,MCN
            COMMON/MATP/D(4,4),YM,PR,GM,HM,PPR,CLAME,ALPHA,YST
     1      ,TABSTN(10,15),TABSTR(10,15)
            COMMON/LDS/R1(2000),RL(2000),RT(2000)
            COMMON/NUME/P(12),DEL(2,12),DET,DI(2,12),DIJ(2,2),XYE(3,12),
     1      NOFE(12),DISE(2,12),RP,RPB,XYP(3)
            COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1      ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)
            COMMON/GRAV/GRAV

*-----------------------------------------------------------
*
*           Reads and writes load control data.
*
*-----------------------------------------------------------

            READ(5,10)TITLE
10          FORMAT(20A4)
            READ(5,20)NSTRS,NRE,NRG,NRPRS,NRT,NRC
20          FORMAT(8I5)
            WRITE(6,30)TITLE,ILD,NSTRS
30          FORMAT(1X,20A4,5X, 'LOAD CASE=',I3,I5)
            WRITE(6,40)NRE,NRG ,NRPRS,NRT,NRC
40          FORMAT(' NRE=',I3,2X,'NRG=',I2,2X,'NRPRS=',I2,2X,'NRT=',
     1       I2,2X,'NRC=',I2)
            IF(NRE.EQ.0) GO TO 80

*---------------------------------------------------------------
*
*           Reads external loads at nodes.
*
*---------------------------------------------------------------

            WRITE(6,50)
50          FORMAT('  LOADS FROM CARDS ')
            DO 70 I=1,NRE
              READ(5,60) NQ,(R(K),K=1,NDF)
60            FORMAT(I10,3F10.3)

*---------------------------------------------------------------
*
```

```
*           Changes node no. to new node no. if slip plane has been added.
*
*-----------------------------------------------------------------

              IF(IIF.LT.0)CALL NEWNOD(NQ)
              WRITE(6,60) NQ,(R(K),K=1,NDF)
              DO 70 K=1,NDF
                IC=(NQ-1)*NDF+K
                RL(IC)=RL(IC)+R(K)
70     CONTINUE

*-----------------------------------------------------------------
*
*           Reads gravity, centrifugal and density data.
*
*-----------------------------------------------------------------

80          CONTINUE
            IF((NRC+NRG).EQ.0) GOTO 230
            READ(5,90)THETA,GRAV,ANGVEL
90          FORMAT(8E10.3)
            WRITE(6,100)THETA,GRAV,ANGVEL
100         FORMAT(' ANGLE=',F7.2,' GRAVITY CONSTANT=',E10.4,' ANGULAR VELOC'
   1         'ITY=',F10.4)
            READ (5,90)(DENS(I),I=1,NMAT)
            WRITE(6,90)(DENS(I),I=1,NMAT)
            DENSAV=0.0
            DO 110 IA=1,NMAT
110         DENSAV=DENSAV+DENS(IA)
            GAMMA=DENSAV*GRAV/NMAT
            GUSH=0.
            IF(NRC.EQ.0) GO TO 120
            GUSH=ANGVEL*ANGVEL/GRAV
120         THETA=THETA/57.295779514
C      WRITE(6,130)
130         FORMAT(' DENSITY CONSTANTS FOR VARIOUS ELEMENTS')
            DO 220 M=1,NE
              L=IMAT(M)
              IF(L.GT.NMAT)L=L-100
              THICK=ORT(L,6)
              DENSE=DENS(L)
C             PQ(1)=PQ(2)=0.
              PQ(1)=0.
              PQ(2)=0.
              IF(DENSE.EQ.0.)GO TO 220
```

```
          IF(NRC.NE.0)GOTO 150
C         DO 140 IJ=1,NMAT
C140      DENS(IJ)=DENS(IJ)•GRAV
150       IF(NRG.EQ.0) GO TO 170
          PQ(1)=DENSE•SIN(THETA)•GRAV
          PQ(2)=-DENSE•COS(THETA)•GRAV
C     WRITE(6,160)M,L,PQ(1),PQ(2),DET
160       FORMAT(2I10,2F15.5,3X,D12.3)
170       NASH=NCN•(M-1)
          DO 190 K=1,NCN
            MASH=NOP(NASH+K)
            DO 180 I=1,NCORD
180         XYE(I,K)=CORD(MASH,I)
190       NOPE(K)=MASH
          DO 210 IGAUS=1,NGAUS
            DO 210 JGAUS=1,NGAUS
              G=XG(IGAUS)
              H=XG(JGAUS)
              CALL SFR(G,H)
              CALL AUX(M)
              DV=DET•CG(IGAUS)•CG(JGAUS)
              IF(NPP.EQ.2)DV=DV•RP•6.28318530718
              IF(NCORD.EQ.3)DV=DV•XYP(3)
              IF(THICK.NE.0.)DV=DV•THICK
              PQ(1)=PQ(1)+DENSE•GUSH•XYP(1)
              DO 200 I=1,NCN
                IC=(NOPE(I)-1)•2
                DO 200 K=1,2
                  IC=IC+1
200           RL(IC)=RL(IC)+PQ(K)•P(I)•DV
210       CONTINUE
220     CONTINUE
230     CONTINUE
        IF(NRPRS.EQ.0) GO TO 330
        READ(5,20)LNE
        WRITE(6,20)LNE
        NLN=NSFR+1

•------------------------------------------------------------
•
•       Reads pressure data.
•
•------------------------------------------------------------

        DO 320 LNEW=1,LNE
```

```
              READ (5,20)(NOPL(J),J=1,NLN)

*---------------------------------------------------------------
*
*            Changes node no. to new node no. if slip plane has been added.
*
*---------------------------------------------------------------

              DO 240 J=1,NLN
240           IF(IIF.LT.0)CALL NEWNOD(NOPL(J))
              WRITE(6,20)(NOPL(J),J=1,NLN)

*-----------------------------------------------------------
*
*            Reads pressure load.
*
*-----------------------------------------------------------

              READ (5,250)((PS(I,J),I=1,NLN),J=1,NDF)
250           FORMAT(8F10.3)
              IF(PS(3,1).NE.0.)GO TO 270
              PRT=PS(2,1)
              DO 260 I=1,NLN
                PS(I,1)=PS(1,1)
260           PS(I,2)=PRT
270           WRITE(6,250)((PS(I,J),I=1,NLN),J=1,NDF)
              H=-1.
              DO 280 J=1,NLN
                MASH=NOPL(J)
                DO 280 K=1,NCORD
                  XYE(K,J)=CORD(MASH,K)
280           CONTINUE
              DO 310 IGAUS=1,NGAUS
                G=XG(IGAUS)
                CALL SFR(G,H)
                DO 300 I=1,NCORD
C                 GASH=GOSH=GISH=0.
                  GASH=0.
                  GOSH=0.
                  GISH=0.
                  DO 290 K=1,NLN
                    GASH=GASH+XYE(I,K)*P(K)
                    IF(I.GT.2) GO TO 290
                    GOSH=GOSH+PS(K,I)*P(K)
                    GISH=GISH+XYE(I,K)*DEL(1,K)
```

```
290               CONTINUE
                  XYP(I)=GASH
                  IF(I.GT.2) GO TO 300
                  PQ(I)=GOSH
                  DJ(I)=GISH
300             CONTINUE
                DV=CG(IGAUS)
                IF(NCORD.EQ.3)DV=DV•XYP(3)
                IF(NPP.EQ.2)DV=6.2831853072•DV•XYP(1)
                DV1=DV•DJ(1)
                DV2=DV•DJ(2)
                AY=DV1•PQ(1)+DV2•PQ(2)
                AX=DV1•PQ(2)-DV2•PQ(1)
                DO 310 I=1,NLN
                  NASH=2•NOPL(I)
                  MASH=NASH-1
                  RL(MASH)=RL(MASH)+P(I)•AX
                  RL(NASH)=RL(NASH)+P(I)•AY
310           CONTINUE
320         CONTINUE
330         CONTINUE
C           WRITE(6,340)
340         FORMAT(' TOTAL NON-ZERO LOADS'/2X,
   1   '  NODE   X VALUE   Y VALUE')
            I=1
350         IF(RL(I).EQ.0.) GO TO 360
            K=(10•I/NDF+9) / 10
            J=(K-1)•NDF+1
            JND=J+NDF-1
C           WRITE(6,60) K, (RL(I), I=J,JND)
            I=JND
360         CONTINUE
            I=I+1
            IF(I.LE.NSZF) GO TO 350

            RETURN
            END
```

```
*------------------------------------------------------------
*------------------------------------------------------------
*
*        Subroutine LINEAR:
*        Helps to update stress and strain.
*
*        Written:
*        Last modified:JUL 19 1984
*
*------------------------------------------------------------
*------------------------------------------------------------
*
*        Called by:  RESIDUE
*
*------------------------------------------------------------
*------------------------------------------------------------

         SUBROUTINE LINEAR(STRN,STRS,W)

         IMPLICIT DOUBLE PRECISION (A-H,O-Z)
         COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
    1    NCORD
         COMMON/NUME/F(12),DEL(2,12),DET,DI(2,12),DIJ(2,2),XYE(3,12),
    1    NOPE(12),DISE(2,12),RP,RPB,XYP(3)
         COMMON/MATP/D(4,4),YM,PR,GM,HM,PPR,CLAME,ALPHA,YST,
    1    TABSTN(10),TABSTR(10,15)
         DIMENSION STRN(4),STRS(4)

         DO 20 I = 1,2
           DO 20 J = 1,2
             GASH=0.
             DO 10 K = 1,NCN
10           GASH=GASH+DISE(I,K)*DI(J,K)
20       DIJ(I,J)=GASH

*------------------------------------------------------------
*
*        Calculate strain at a point.
*
*------------------------------------------------------------

         STRN(1)=DIJ(1,1)
         STRN(2)=DIJ(2,2)
         STRN(3)=(DIJ(1,2)+DIJ(2,1))
         STRN(4)=0.
```

```
*----------------------------------------------------------------
*
*          Calculate rotation, W.
*
*----------------------------------------------------------------

           W=.5*(DIJ(2,1)-DIJ(1,2))
           IF(NPP.NE.2) GO TO 40
           GISH=0.
           DO 30 K=1,NCN
30         GISH=GISH+DISE(1,K)*P(K)
           IF(RP.EQ.0.) RP=.000001
           STRN(4)=GISH/RP
40         IF(NPP.EQ.1)STRN(4)=-PPR*(STRN(1)+STRN(2))
           GASH=CLAME*(STRN(1)+STRN(2)+STRN(4))
           DO 50 I=1,4
50         STRS(I)=GM*STRN(I)+GASH
           STRS(3)=(STRS(3)-GASH)*.5

           RETURN
           END

*----------------------------------------------------------------
*----------------------------------------------------------------
*
*          Subroutine MAKEEL.
*          This subroutine adds a new element to the connectivity containing
*          corner nodes I and K.
*
*          Written:
*          Last modified:JUL 19 1984
*
*----------------------------------------------------------------
*----------------------------------------------------------------
*
*          Called by:  SLIPP
*
*----------------------------------------------------------------
*----------------------------------------------------------------

           SUBROUTINE MAKEEL(MMM,I,K,IANG)

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
  1        NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
```

```
   2        NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
   3        TPWORK,KNE
            COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
   1        ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000),DUMMY(1000)

            JSTART=I
            IF(IANG.EQ.2)GOTO 10
            IF(CORD(I,2).LT.CORD(K,2))JSTART=K
            GOTO 20
 10         IF(CORD(I,1).GT.CORD(K,1))JSTART=K
 20         CONTINUE
            L=NSFR•4
            DO 30 M=1,L,NSFR
        MM=MMM+M
        IF(NOP(MM).NE.JSTART)GOTO 30
        JSTRT=MM
        IFLAG=M
 30         CONTINUE
            NEW=NE•L+1
            NOP(NEW)=NOP(JSTRT)
            DO 40 N=1,NSFR
        NEW=NEW+1
        IF(IFLAG.EQ.1)JSTRT=JSTRT+L
        JSTRT=JSTRT-1
 40         NOP(NEW)=NOP(JSTRT)
            DO 50 N=1,NSFR
        NOP(NEW+1)=NOP(NEW)+1
 50         NEW=NEW+1
            IX=NSFR-1
            DO 60 N=1,IX
 60         NOP(NEW+N)=NOP(NEW-NSFR-N)+1
            NEW=NEW+IX
            DO 70 N=NSFR,1,-1
        NEW=NEW+1
 70         NOP(NEW)=JSTART+N
            NE=NE+1
            IMAT(NE)=NMAT-NL

            RETURN
            END
```

```
*----------------------------------------------------------------
*----------------------------------------------------------------
*
*          Subroutine MOD:
*          Forms D matrix for different stress/strain cases.
*
*          Written:
*          Last modified:JUL 14 1984
*
*----------------------------------------------------------------
*----------------------------------------------------------------
*
*          Called by:  RESIDUE STIFM
*
*----------------------------------------------------------------
*----------------------------------------------------------------

           SUBROUTINE MOD(L,M)

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     &     NCORD
           COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
           COMMON/MATP/D(4,4),YM,PR,GM,HM,PPR,CLAME,ALPHA,YST

           YM=ORT(L,1)
           PR=ORT(L,2)
           YST=ORT(L,3)
           HM=ORT(L,4)
           ALPHA=ORT(L,5)
           DELTA=ORT(L,9)
           GS=ORT(L,10)

*----------------------------------------------------------------
*
*          Allows for bulk properties.
*
*----------------------------------------------------------------

           IF(YM.LT.0.0) GO TO 30
           PPR=PR/(1.-PR)
           GM=YM/(1.+PR)

*----------------------------------------------------------------
*
```

```
•             Note!!!!!!••••••• GM=2•SHEAR MODULUS ••••••••
•
•------------------------------------------------------------

              CLAME=PR•GM/(1.-2.•PR)
              IF(M.EQ.1)RETURN
              D(3,3)=GM•.5
C             D(1,3)=D(3,1)=D(2,3)=D(3,2)=0.
              D(1,3)=0.0
              D(3,1)=0.0
              D(2,3)=0.0
              D(3,2)=0.0
              IF(NPP.EQ. 1) GO TO 10
              YM=YM/(1.-PR•PR)
              PR=PR/(1.-PR)
10            CONTINUE

•------------------------------------------------------------
•
•             For plane stress/strain cases.
•
•------------------------------------------------------------

              CD=YM/(1.-PR•PR)
              CDU=CD•PR
C             D(1,1)=D(2,2)=CD
              D(1,1)=CD
              D(2,2)=CD
C             D(1,2)=D(2,1)=CDU
              D(1,2)=CDU
              D(2,1)=CDU
              IF(NPP.NE.2) GO TO 20

•------------------------------------------------------------
•
•             D matrix for axisymmetric case.
•
•------------------------------------------------------------

              D(4,4)=CD
C             D(1,4)=D(4,1)=D(2,4)=D(4,2)=CDU
              D(1,4)=CDU
              D(4,1)=CDU
              D(2,4)=CDU
              D(4,2)=CDU
```

```
C            D(3,4)=D(4,3)=0.
             D(3,4)=0.
             D(4,3)=0.
20           CONTINUE
             PR=ORT(L,2)
             RETURN
30           CONTINUE
             BM=-YM
             GM=ABS(PR)*2.

*-----------------------------------------------------------------
*
*            Note!!!!!!******* GM=2*SHEAR MODULUS ********
*
*-----------------------------------------------------------------

             CLAME=BM-GM/3.
             PR=.5
             PPR=PR/(1.-PR)
             IF(M.EQ.1) RETURN
             D(1,1)=CLAME+GM
             D(2,2)=CLAME+GM
             D(3,3)=GM/2.
             IF(ORT(L,2).LT.0) D(3,3)=D(3,3)/(1.+GM/2./DELTA/GS)
             D(1,2)=CLAME
             D(1,3)=0.0
             D(2,1)=CLAME
             D(2,3)=0.0
             D(3,1)=0.0
             D(3,2)=0.0
             IF(NPP.NE.2) GO TO 40
             D(4,4)=CLAME+GM
             D(1,4)=CLAME
             D(2,4)=CLAME
             D(3,4)=0.0
             D(4,1)=CLAME
             D(4,2)=CLAME
             D(4,3)=0.0

40           RETURN
             END
```

```
*----------------------------------------------------------------
*----------------------------------------------------------------
*
*          Subroutine NEWNOD:
*          This subroutine finds the new node number of node N.
*
*          Written:
*          Last modified:JUL 19 1984
*
*----------------------------------------------------------------
*----------------------------------------------------------------
*
*          Called by:  FLAGEGDATALDATA
*          SLIPP
*
*----------------------------------------------------------------
*----------------------------------------------------------------

           SUBROUTINE NEWNOD(N)

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1     NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2     NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3     TPWORK,KNE
           COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2),
     1     NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000),DUMMY(1000)
           COMMON/SLP/DCOR(1000,2)

           K=0
           DO 10 I=1,NP
10         IF(DCOR(N,1).EQ.CORD(I,1).AND.DCOR(N,2).EQ.CORD(I,2)) K=I
           N=K
           IF(K.NE.0) RETURN
           WRITE(6,20) N
20         FORMAT(' ****** ERROR IN MAPPING: NO NEW NODE FOUND FOR OLD '
     +     'NODE ',I3,' ******')

           STOP
           END
```

```
*---------------------------------------------------------------
*---------------------------------------------------------------
*
*          Subroutine NFLOW:
*          Calculates the amount of yield which results.
*
*          Written:
*          Last modified:MAR 19 1984
*
*---------------------------------------------------------------
*---------------------------------------------------------------
*
*          Called by:  RESIDUESTIFM
*
*---------------------------------------------------------------
*---------------------------------------------------------------

           SUBROUTINE NFLOW(SSS)

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1     NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2     NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3     TPWORK
           COMMON/MATP/D(4,4),YM,PR,GM,HM,PPR,CLAME,ALPHA,YST,
     1     TABSTN(10,15),TABSTR(10,15)
           COMMON/INV/SMEAN,STJ2,STJ3,SIGMA,PHI,APHI,STN,CPHI,YIELD
           COMMON/FLW/ABETA,AA(4),DD(4)
           DIMENSION SSS(4),A1(4),A2(4),A3(4)

*---------------------------------------------------------------
*
*          Initializes vector A1.
*
*---------------------------------------------------------------

C          A1(1)=A1(2)=A1(4)=1.
           A1(1)=1.
           A1(2)=1.
           A1(4)=1.
           A1(3)=0.

*---------------------------------------------------------------
*
*          Calculates vector A2.
```

```
•
•------------------------------------------------------------

            SSS(3)=2.•SSS(3)
            GASH=2.•SIGMA
            DO 10 I=1,4
10          A2(I)=SSS(I)/GASH

•------------------------------------------------------------
•
•           Calculates vector A3.
•
•------------------------------------------------------------

            CON=STJ2/3.
            A3(1)=CON+SSS(2)•SSS(4)
            A3(2)=CON+SSS(1)•SSS(4)
            A3(3)=-SSS(4)•SSS(3)
            A3(4)=CON+ SSS(1)•SSS(2)-.25•SSS(3)•SSS(3)
            CON=1.73205080756888
            APHI=PHI•57.29577951308232
            MYIELD=NYIELD
            IF(NYIELD.GT.100)MYIELD=NYIELD-100
            GO TO (20,30,50,60),MYIELD

•------------------------------------------------------------
•
•           Von Mises yield criteria.
•
•------------------------------------------------------------

20          C1=0.
            C2=CON
            C3=0.
            GO TO 70

•------------------------------------------------------------
•
•           Tresca yield criteria.
•
•------------------------------------------------------------

30          C1=0.
            AAPHI=ABS(APHI)
            IF(AAPHI.LT.29.0) GO TO 40
```

```
              C2=CON
              C3=0.
              GO TO 70
40            SN=SIN(PHI)
              CTS=SQRT(1.-STN•STN)
              C2=2.•(CPHI+SN•STN/CTS)
              C3=CON•SN/(STJ2•CTS)
              GO TO 70

•------------------------------------------------------
•
•             Drucker yield criteria.
•
•------------------------------------------------------

50            C1=ALPHA
              C2=1.
              C3=0.
              GO TO 70

•------------------------------------------------------
•
•             Beltrami yield criteria.
•
•------------------------------------------------------

60            C1=(1.+2.•PR)•SMEAN/YIELD
              C2=(1.-PR)•GASH/YIELD
              C3=0.
70            CONTINUE
              DO 80 I=1,4
80            AA(I)=C1•A1(I)+C2•A2(I)+C3•A3(I)
              IF(NPP.NE.1) GO TO 90
              CON=AA(4)
              AA(4)=-PPR•(AA(1)+AA(2))
90            GASH=CLAME•(AA(1)+AA(2)+AA(4))
              GISH=HM
              DO 100 I=1,4
                DD(I)=GM•AA(I)+GASH
                IF(I.EQ.3)DD(I)=.5•AA(I)•GM
100           GISH=GISH+AA(I)•DD(I)
              ABETA=1./GISH
              IF(NPP.EQ.1)AA(4)=CON

              RETURN
```

```
          END

•------------------------------------------------------------
•------------------------------------------------------------
•
•         Subroutine NODEXY:
•         Sets x,y values of new intermediate coordinates of quadratic or
•         cubic elements.
•
•         Written:
•         Last modified:MAR 19 1984
•
•------------------------------------------------------------
•------------------------------------------------------------
•
•         Called by:  GDATASLIPP
•
•------------------------------------------------------------
•------------------------------------------------------------

          SUBROUTINE NODEXY(ISTART)

          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1    NCORD
          COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2),
     1    NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)

          ANSFR=NSFR
          NQ=NSFR-1
          DO 30 N=ISTART,NE
            NASH=NCN•(N-1)
            DO 30 NLV=1,NCN,NSFR
              NV1=NOP(NASH+NLV)
              NL1=NLV+NSFR
              IF(NL1.GT.NCN) NL1=1
              NV2=NOP(NASH+NL1)
              DO 20 I=1,NQ
                M=NLV+I
                J=NOP(NASH+M)
                GISH=I
                GOSH=ANSFR-GISH
                DO 10 K=1,2
10                CORD(J,K)=(GOSH•CORD(NV1,K)+CORD(NV2,K)•GISH)/ANSFR
20            CONTINUE
```

```
30          CONTINUE

            RETURN
            END

•-----------------------------------------------------
•-----------------------------------------------------
•
•           Subroutine OUTPUT:
•           Writes displacements, reactions, stresses and strains to SCRUBS.LIS
•
•           Written:
•           Last modified:JUL 19 1984
•
•-----------------------------------------------------
•-----------------------------------------------------
•
•           Called by: SCRUBS
•
•-----------------------------------------------------
•-----------------------------------------------------

            SUBROUTINE OUTPUT

            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON/ELPRNT/ NELP,NPEL(50),NONODE
            DIMENSION SAVG(6)
            COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
    1       NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
    2       NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
    3       TPWORK
            COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
            COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
    1       US(400,2)
            COMMON/LDS/R1(2000),RL(2000),RT(2000)
            COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2),
    1       NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)

            MOUT=NOUT(INC)
            IF(MOUT.LT.100)GO TO 10
            M=MOUT/100
            MOUT=MOUT-M•100
            IF(IT.EQ.1)MOUT=M
10          IF(MOUT.EQ.0)RETURN
            IF(NONODE.GT.0) GO TO 50
```

```
*-----------------------------------------------------------
*
*           Writes x and y displacements.
*
*-----------------------------------------------------------

            WRITE(6,20)
20          FORMAT('  DISPLACEMENTS',5X,'X',15X,'Y',13X,'TOTAL X',11X,
  1         'TOTAL Y')
            DO 30 N=1,NP
30          WRITE(6,40)N,(DIS(I,N),I=1,NDF),(TDIS(I,N),I=1,NDF)
40          FORMAT(I10,6E16.6)
50          CONTINUE
            IF(MOUT.EQ.1)RETURN

*-----------------------------------------------------------
*
*           Writes node numbers and corresponding displacements and
*           reactions.
*
*-----------------------------------------------------------

            WRITE(6,60)
60          FORMAT(6X,'NODE NO',10X,'DISPLACEMENTS',16X,'REACTIONS')
            NASH=0
            DO 70 I=1,NB
              N=NBC(I)
              MASH=NASH+1
              NASH=MASH+1
70          WRITE(6,80)N,(TDIS(J,N),J=1,2),(TRC(J),J=MASH,NASH)
80          FORMAT(I10,6E16.6)
            IF(MPR.EQ.0)GO TO 130
C           SUMX=SUMY=0.0
            SUMX=0.0
            SUMY=0.0
            DO 110 I=1,MPR
              NASH=NOPR(I)
              DO 90 J=1,NB
                N=NBC(J)
                IF(N.EQ.NASH) GO TO 100
90            CONTINUE
              GO TO 110
100           MASH=2*J
              SUMX=SUMX+TRC(MASH-1)
              SUMY=SUMY+TRC(MASH)
```

```
110         CONTINUE

*------------------------------------------------------------
*
*           Writes total reactions at specified points.
*
*------------------------------------------------------------

            WRITE(6,120)SUMX,SUMY
120         FORMAT(' TOTAL REACTIONS AT SPECIFIED POINTS   ALONG X=',
    1        E16.6,' ALONG Y=', E16.6)
130         IF(MOUT.EQ.2)RETURN

*------------------------------------------------------------
*
*           Writes stresses at sampling points.
*
*------------------------------------------------------------

            WRITE(6,140)
140         FORMAT('    STRESSES AT SAMPLING POINTS '/
    1       ' GAUSS P',1X,'SIG X',6X,'SIG Y',6X,'TAU XY',6X,' SIG Z',6X,
    2       ' YIELD',6X,'PL. STRN')
            MCN=NSTORE/NE
            DO 290 M=1,NE
              IF(NELP.LT.0) GO TO 160
              DO 150 K=1,NELP
                IF(M.EQ.NPEL(K)) GO TO 160
150           CONTINUE
              GO TO 290
160           CONTINUE
              WRITE(6,170) M
170           FORMAT(' EL NO ',I3)
              K=MCN*(M-1)
              EPS=0.
              DO 190 L=1,MCN
                J=K+L
                WRITE(6,180)L,(TSTS(I,J),I=1,5),EPSTN(5,J)
180             FORMAT(I2,6(1PE12.3))
                EPS=EPS+EPSTN(5,J)
190           CONTINUE
              XMCN=MCN
              EPS=EPS/XMCN
              DO 200 I=1.6
200           SAVG(I)=0.0
```

```
            DO 210 I=1,5
              DO 210 L=1,MCN
                J=K+L
210         SAVG(I)=SAVG(I)+TSTS(I,J)
            DO 220 I=1,5
220         SAVG(I)=SAVG(I)/XMCN
            SMEAN=(SAVG(1)+SAVG(2)+SAVG(4))/3.
            IF(SAVG(5).EQ.0)GOTO 230
            RATIO=SMEAN/SAVG(5)
230         WRITE(6,240) (SAVG(I),I=1,5),EPS,SMEAN,RATIO
240         FORMAT(' AV',6(1PE12.4)/' SIG MEAN='1PE10.3,' R=',1PE10.3)
            DO 250 I=1,4
250         SAVG(I)=0.0
            DO 260 I=1,4
              DO 260 L=1,MCN
                J=K+L
260         SAVG(I)=SAVG(I)+EPSTN(I,J)
            DO 270 I=1,4
270         SAVG(I)=SAVG(I)/XMCN

*-----------------------------------------------------------
*
*          Writes plastic strains.
*
*-----------------------------------------------------------

            WRITE(6,280) (SAVG(I),I=1,4)
280         FORMAT(3X,4E12.4,' (PLASTIC STRAINS) ')
290       CONTINUE
          IF(MOUT.EQ.3)RETURN

*-----------------------------------------------------------
*
*          Writes residual forces at the nodes.
*
*-----------------------------------------------------------

          WRITE(6,300)
300       FORMAT(' RESIDUAL  FORCES AT NODAL POINTS'/
   1       5X,'NODE NO',9X,'X VALUE',9X,'Y VALUE')
          DO 310 J=1,NP
            N=2*J
310       WRITE(6,320)J,R1(N-1),R1(N)
320       FORMAT(I10,3E20.6)
```

```
           RETURN
           END

*-----------------------------------------------------------
*-----------------------------------------------------------
*
*          Subroutine PDISP:
*          Initializes the displacement vector according to number of
*          loading increments.
*
*          Written:
*          Last modified:MAR 19 1984
*
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*          Called by:  SCRUBS
*
*-----------------------------------------------------------
*-----------------------------------------------------------

           SUBROUTINE PDISP(L)

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
  1        NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
  2        NSTORE
           COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
           COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
  1        US(400,2)

           IF(L-1)10,60,30
10         IC=NB*NDF
           DO 20 I=1,IC
20         U(I)=0.
           RETURN
30         FACT=FAC(L)
           IC=1
           NZ=10**(NDF-1)
           DO 50 I=1,NB
             JZ=NFIX(I)
             IZ=NZ
             DO 50 J=1,NDF
               IF(JZ.LT.IZ) GO TO 40
               U(IC)=US(I,J)*FACT
```

```
                IC=IC+1
                JZ=JZ-IZ
40              IZ=IZ/10
50           CONTINUE
             RETURN
60           FACT=FAC(INC)
             IC=0
             DO 70 I=1,NB
               DO 70 J=1,NDF
                 IC=IC+1
70           U(IC)=US(I,J)*FACT

             RETURN
             END

*-----------------------------------------------------------
*-----------------------------------------------------------
*
*            Subroutine READF:
*            Reads the mining parameters:  which elements, and when.
*
*            Written:
*            Last modified:JUL 24 1984
*
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*            Called by:  SCRUBS
*
*-----------------------------------------------------------
*-----------------------------------------------------------

             SUBROUTINE READF(NRFF)

             IMPLICIT DOUBLE PRECISION (A-H,O-Z)
             COMMON/FG/ FLAG(1000),BULK(1000,2),MINES(10,50),NME(10)

*-------------------------------------------------------
*
*            Reads number of elements to be mined per increment.
*
*-------------------------------------------------------

             READ (5,10)(NME(I),I=1,NRFF)
10           FORMAT(25I5)
```

```
          WRITE(6,20)
20        FORMAT('   LIST OF FAILURE(MINING) SEQUENCE '/
   1      ' INCREMENT  NUMBER OF ELEMENTS  ',15X,'ELEMENT LIST')

*-----------------------------------------------------------
*
*         Reads the element numbers.
*
*-----------------------------------------------------------

          DO 50 I=1,NRFF
            NL=NME(I)
            IF(NL.GT.25)GOTO 30
            READ(5,10) (MINES(I,K),K=1,NL)
            WRITE(6,25) (I,NL,(MINES(I,K),K=1,NL))
25          FORMAT(16I5)
            GOTO 50
30          READ(5,10) (MINES(I,K),K=1,25)
            READ(5,10) (MINES(I,K),K=26,NL)
            WRITE(6,25) (I,NL,(MINES(I,K),K=1,14))
            IDUM=(NL+2)/16
            IY=1
            DO 40 IZ=1,IDUM
              IY=IY+14
              IYY=IY+13
              IF(IYY.GT.NL)IYY=NL
40          WRITE(6,45) (MINES(I,K),K=IY,IYY)
45        FORMAT(10X,14I5)
50        CONTINUE

          RETURN
          END
```

```
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*           Subroutine REMLD:
*           Removes gravity load from failed elements.
*
*           Written:
*           Last modified:JUL 19 1984
*
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*           Called by:  FTEST
*
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*           Subroutines called:SFRAUX
*
*-----------------------------------------------------------
*-----------------------------------------------------------

      SUBROUTINE REMLD(M)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(3)
      DIMENSION NOPL(4),PS(4,2),PQ(2),DJ(2),DENS(10)
      COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1 NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2 NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO, PWORK,
     3 TPWORK
      COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
      COMMON/GID/XG(6),CG(6),XMG(6),CMG(6),NGAUS,MGAUS,MCN
      COMMON/MATP/D(4,4),YM,PR,GM,HM,PPR,CLAME,ALPHA,YST
     1 ,TABSTN(10,15),TABSTR(10,15)
      COMMON/LDS/R1(2000),RL(2000),RT(2000)
      COMMON/NUME/P(12),DEL(2,12),DET,DI(2,12),DIJ(2,2),XYE(3,12),
     1 NOPE(12),DISE(2,12),RP,RPB,XYP(3)
      COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1 ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)
      COMMON/GRAV/GRAV

      GUSH=0.0
      L=IMAT(M)
      IF(L.GT.NMAT)L=L-100
```

```
          THICK=ORT(L,6)
          DENSE=DENS(L)
C         PQ(1)=PQ(2)=0.
          PQ(1)=0.
          PQ(2)=0.
          IF(DENSE.EQ.0.)GO TO 60
10        NASH=NCN•(M-1)
          DO 30 K=1,NCN
            MASH=NOP(NASH+K)
            DO 20 I=1,NCORD
20          XYE(I,K)=CORD(MASH,I)
30        NOPE(K)=MASH
          DO 50 IGAUS=1,NGAUS
            DO 50 JGAUS=1,NGAUS
              G=XG(IGAUS)
              H=XG(JGAUS)
              CALL SFR(G,H)
              CALL AUX(M)
              DV=DET•CG(IGAUS)•CG(JGAUS)
              IF(NPP.EQ.2)DV=DV•RP•6.28318530718
              IF(NCORD.EQ.3)DV=DV•XYP(3)
              IF(THICK.NE.0.)DV=DV•THICK
              PQ(1)=PQ(1)+DENSE•GUSH•XYP(1)•GRAV•FACTOR
              DO 40 I=1,NCN
                IC=(NOPE(I)-1)•2
                DO 40 K=1,2
                  IC=IC+1
C                 RL(IC)=RL(IC)-PQ(K)•P(I)•DV
                  RT(IC)=RT(IC)-PQ(K)•P(I)•DV
40      CONTINUE
50        CONTINUE
60        CONTINUE

          RETURN
          END
```

```
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*           Subroutine RESIDUE:
*           Determines element failure, rubbleization, and residual.
*
*           Written:
*           Last modified JUL 19 1984
*
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*           Called by:  SCRUBS
*
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*           Subroutines called:MODSFR
*           AUXLINEARJOINTM
*           INVARNFLOWFTEST
*           CONRUBDROP
*
*-----------------------------------------------------------
*-----------------------------------------------------------

      SUBROUTINE RESIDUE (TIME1,IFCOUNT,MTGPY)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1 NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2 NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3 TPWORK
      COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
      COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
     1 US(400,2)
      COMMON/LDS/R1(2000),RL(2000),RT(2000)
      COMMON/BOTTOM/ AK
      COMMON/FG/ FLAG(1000),BULK(1000,2),MINES(10,50),NME(10),GAMMA
      COMMON/GID/XG(6),CG(6),XMG(6),CMG(6),NGAUS,MGAUS,MCN
      COMMON/MATP/D(4,4),YM,PR,GM,HM,PPR,CLAME,ALPHA,YST
     1 ,TABSTN(10,15),TABSTR(10,15)
      COMMON/NUME/P(12),DEL(2,12),DET,DI(2,12),DIJ(2,2),XYE(3,12),
     1 NOPE(12),DISE(2,12),RP,RPB,XYP(3)
      COMMON/INV/SMEAN,STJ2,STJ3,SIGMA,PHI,APHI,STN,CPHI,YIELD
      COMMON/FLW/ABETA,AA(4),DD(4)
```

```
      COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1 ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)
     2 ,SIG(4),PSIG(4),DSIG(4),EP(4),NG(16),NGSTP(16)
      DIMENSION STRN(4),STRS(4),TSIG(4)
      DIMENSION SAVG(5)
      REAL TIME1,TIME2,TIME3

      NECN=NE*NCN
      READ (8)(NOP(J),J=1,NECN),(IMAT(I),I=1,NE),(((DISN(K,I),
     1 TDIS(K,I)),K=1,2),(CORD(I,J),J=1,NCORD),I=1,NP),(((TSTS(I,J),
     2 EPSTN(I,J)),I=1,5),J=1,NSTORE)
      REWIND 8
      IFCOUNT=0
      IF(IT.GT.1) GO TO 20

*----------------------------------------------------------------
*
*           For initial iteration step set increment equal to zero.
*
*----------------------------------------------------------------

      DO 10 I=1,2
        DO 10 J=1,NP
10    DISN(I,J)=0.
20    CONTINUE

*----------------------------------------------------------------
*
*           Update coordinates, total displacements, and displacement increment.
*
*----------------------------------------------------------------

      DO 30 I = 1,NDF
        DO 30 J=1,NP
          GASH=DIS(I,J)
          CORD(J,I)=CORD(J,I)+GASH
          TDIS(I,J)=TDIS(I,J)+GASH
          DISN(I,J)=DISN(I,J)+GASH
30    CONTINUE
      DO 40 J=1,NSZF
40    R1(J)=0.
      PWORK=0.
      WRITE(6,50)FACTOR,IT
50    FORMAT('  LFACTOR   =',F7.3,2X,'ITERATION NO.=',I3)
      MTGPY=0
```

```
*-----------------------------------------------------------------
*
*            Control loop on all elements.
*
*-----------------------------------------------------------------

             DO 360 MNEW=1,NE
               MASH=MNEW-1
               NOSH=MASH*NCN
               MOSH=MASH*MCN

*-----------------------------------------------------------------
*
*            Find coordinates and connectivity.
*
*-----------------------------------------------------------------

               DO 70 K=1,NCN
                 MASH=NOP(NOSH+K)
                 DO 60 I=1,NCORD
                   XYE(I,K)=CORD(MASH,I)
             IF(I.GT.2) GO TO 60
             DISE(I,K)= DIS(I,MASH)
60               CONTINUE
                 NOPE(K)=MASH
70             CONTINUE
               L=IMAT(MNEW)
               IF(L.GT.NMAT)L=L-100

*-----------------------------------------------------------------
*
*            Get elasticity matrix.
*
*-----------------------------------------------------------------

               CALL MOD(L,1)
               THICK=ORT(L,6)
               IF(THICK.EQ.0.)THICK=1.
               NPLST=0
               IJG=0

*-----------------------------------------------------------------
*
*            Loop on integration points.
*
```

```
*-------------------------------------------------------------

        DO 280 IGAUS=1,MGAUS
          DO 280 JGAUS=1,MGAUS
            IJG=IJG+1
            NG(IJG)=0
            NGSTP(IJG)=0
            G=XMG(IGAUS)
            H=XMG(JGAUS)

*-------------------------------------------------------------
*
*       Compute shape functions and derivatives.
*
*-------------------------------------------------------------

            CALL SFR(G,H)
            CALL AUX(MNEW)
            DV=DET*CMG(IGAUS)*CMG(JGAUS)*THICK
            IF(NPP.EQ.2)DV=DV*RP*6.283185307179586
            IF(NCORD.EQ.3)DV=DV*XYP(3)
            MOSH=MOSH+1

*-------------------------------------------------------------
*
*       Compute elastic strains and stresses in element.
*
*-------------------------------------------------------------

            CALL LINEAR(STRN,STRS,W)
            BRING=1.
            IF(NL.NE.1) GO TO 90

*-------------------------------------------------------------
*
*       If element rubbleizes, set stresses to zero.
*
*-------------------------------------------------------------

            IF(BULK(MNEW,1).EQ.0.0) GO TO 90
            DO 80 J=1,4
80          STRS(J)=0.000001
90          CONTINUE
            IF(BULK(MNEW,1).NE.0.0) GO TO 100
```

```
•-----------------------------------------------------------------
•
•           For jointed media material find incremental stresses and residuals
•           in subroutine JOINTM.
•
•-----------------------------------------------------------------

                 IF(ORT(L,2).GE.0) GOTO 100
                 CALL JOINTM(L,STRN,STRS,MOSH,YIELD,PSTRN)

•-----------------------------------------------------------------
•
•           Set incremental stresses equal to elastic strains.
•
•-----------------------------------------------------------------

100              DO 110 I=1,4
                   DSIG(I)=STRS(I)
                   EP(I)=EPSTN(I,MOSH)
C110           TSIG(I)=SIG(I)=TSTS(I,MOSH)
                   TEMP=TSTS(I,MOSH)
                   TSIG(I)=TEMP
                   SIG(I)=TEMP
110              CONTINUE
                 IF (NALGO.EQ.0) GO TO 240
                 PSTRN=EPSTN(5,MOSH)
                 PSTY=TSTS(5,MOSH)

•-----------------------------------------------------------------
•
•           Add incremental stress to total stress.
•
•-----------------------------------------------------------------

                 DO 120 I=1,4
120              TSIG(I)=TSIG(I)+STRS(I)
                 IF(ORT(L,2).LT.0) GOTO 240
                 CALL INVAR    (TSIG)
                 RD=1.
                 DF=YST-PSTY
                 IF(DF.LE.0.0) GO TO 130
                 DF1=YIELD-YST
                 IF(DF1.LE.0.0) GO TO 240
                 RD=DF1/(DF+DF1)
                 GO TO 140
```

```
130                DF1=YIELD-PSTY
                   IF(DF1.LE.0.0) GO TO 240

•----------------------------------------------------------------
•
•          Compute plastic step.
•
•----------------------------------------------------------------

140                MSTEP=DF1•8.0/YST+1.
                   ASTEP=MSTEP
                   RD1=1.-RD
                   DO 150 I=1,4
C                    TSIG(I)=PSIG(I)=SIG(I)+RD1•STRS(I)
                     TEMP=SIG(I)+RD1•STRS(I)
                     TSIG(I)=TEMP
                     PSIG(I)=TEMP
150          STRS(I)=RD•STRS(I)/ASTEP
                   TLEMDA=0.
                   NG(IJG)=MOSH
                   NGSTP(IJG)=MSTEP
                   NPLST=NPLST+1
                   DO 200 ISTEP=1,MSTEP
                     CALL INVAR (TSIG)

•----------------------------------------------------------------
•
•          Find position on effective curve.
•
•----------------------------------------------------------------

                     MASH=0.
160              MASH=MASH+1
                     IF(PSTRN.LT.TABSTN(L,MASH)) GO TO 170
                     GO TO 160
170              NASH=MASH
                     MASH=NASH-1
                     ETAB=TABSTN(L,NASH)-TABSTN(L,MASH)
                     HM=(TABSTR(L,NASH)-TABSTR(L,MASH))/ETAB

•----------------------------------------------------------------
•
•          Compute AA,ABETA,PD,TSIG.
•
•----------------------------------------------------------------
```

```
            CALL NFLOW(TSIG)
                GASH=0.
                DO 180 I=1,4
180         GASH=GASH+AA(I)•STRS(I)
                DLEMDA=GASH•ABETA
                IF(DLEMDA.LT.0.) DLEMDA=0.
                GASH=0.
                DO 190 I=1,4
                  GASH=GASH+AA(I)•PSIG(I)
                  EP(I)=EP(I)+AA(I)•DLEMDA
C190          TSIG(I)=PSIG(I)=PSIG(I)+STRS(I)-DLEMDA•DD(I)
                  TEMP=PSIG(I)+STRS(I)-DLEMDA•DD(I)
                  PSIG(I)=TEMP
                  TSIG(I)=TEMP
190         CONTINUE
                GASH=GASH•DLEMDA
                PSTRN=PSTRN+GASH/YIELD
                PWORK=PWORK+GASH•DV
                TLEMDA=TLEMDA+DLEMDA
200            CONTINUE
               DO 210 I=1,4
                 GASH=TSIG(I)-SIG(I)
                 PSIG(I)=DSIG(I)-GASH
210            DSIG(I)=GASH
               CALL INVAR (TSIG)
               MASH=0
220            MASH=MASH+1
               IF(PSTRN.LT.TABSTN(L,MASH)) GO TO 230
               GO TO 220
230         NASH=MASH
               MASH=NASH-1
               ETAB=TABSTN(L,NASH)-TABSTN(L,MASH)
               GASH=(PSTRN-TABSTN(L,MASH))/ETAB
               MASH=MASH-1
               CYIELD=TABSTR(L,MASH+1)•(1.-GASH)+GASH•
  1          TABSTR(L,MASH+2)
               IF(YIELD.GT.CYIELD)BRING=CYIELD/YIELD


•-----------------------------------------------------------
•
•           Add Jaumann stress to DSIG.
•
•-----------------------------------------------------------


240            CONTINUE
```

```
                   DSIG(1)=DSIG(1)+2.*W*TSTS(1,MOSH)
                   DSIG(2)=DSIG(2)-2.*W*TSTS(2,MOSH)
                   DSIG(3)=DSIG(3)+W*(TSTS(2,MOSH)-TSTS(1,MOSH))

*----------------------------------------------------
*
*            Add increment to total stress.
*
*----------------------------------------------------

                   DO 250 I=1,4
                     GASH=BRING*(SIG(I)+DSIG(I))
                     EPSTN(I,MOSH)=EP(I)
C250            STRS(I)=TSTS(I,MOSH)=GASH
                     STRS(I)=GASH
                     TSTS(I,MOSH)=GASH
250                CONTINUE
                   YIELD=BRING*YIELD
                   EPSTN(5,MOSH)=PSTRN
                   TSTS(5,MOSH)=YIELD
                   DO 260 I=1,ICS
260                STRS(I)=DV*STRS(I)

*----------------------------------------------------
*
*            Compute residual.
*
*----------------------------------------------------

                   DO 270 K=1,NCN
                     GASH=0.
                     IF(NPP.EQ.2)GASH=P(K)*STRS(4)/RP
                     MASH=2*NOPE(K)-1
                     NASH=MASH+1
                     R1(MASH)=R1(MASH)+DI(1,K)*STRS(1)+
   1               DI(2,K)*STRS(3)+GASH
                     R1(NASH)=R1(NASH)+DI(1,K)*STRS(3)+
   1               DI(2,K)*STRS(2)
270         CONTINUE
280            CONTINUE
               IF(NL.NE.1) GO TO 350

*----------------------------------------------------
*
*            Rubble failure logic.
```

```
•
•------------------------------------------------------------

              DO 290 I=1,5
290           SAVG(I)=0.0
              JJ=MCN•(MNEW-1)
              DO 300 I=1,MCN
                JJ1=JJ+1
                DO 300 J=1,5
300           SAVG(J)=SAVG(J)+TSTS(J,JJ1)
              DO 310 I=1,5
310           SAVG(I)=SAVG(I)•.25
              MATNO=IMAT(MNEW)
              IF(MATNO.GT.100) MATNO=MATNO-100
              CALL FTEST(MNEW,SAVG,MATNO,IFLFG)
              IFCOUNT=IFCOUNT+IFLFG
              IF(IFLFG.EQ.0) GO TO 350

•------------------------------------------------------------
•
•            If element has failed, adjust R1 array and set coordinates,
•            displacements and stresses.
•
•            First check for continuum/rubble interaction.
•
•------------------------------------------------------------

              CALL CONRUB(SIGRES,MNEW,NMAT)
              INDEX=(MNEW-1)•MCN
              DO340 IGAUS=1,MGAUS
                DO340 JGAUS=1,MGAUS
                  G=XMG(IGAUS)
                  H=XMG(JGAUS)
                  CALL SFR(G,H)
                  CALL AUX(MNEW)
                  DV=DET•CMG(IGAUS)•CMG(JGAUS)•THICK
                  IF(NPP.EQ.2)DV=DV•RP•6.283185307179586
                  IF(NCORD.EQ.3)DV=DV•XYP(3)
                  INDEX=INDEX+1
                  DO 320 I=1,ICS
                    STRS(I)=DV•TSTS(I,INDEX)
320               TSTS(I,INDEX)=SIGRES
                  TSTS(3,INDEX)=0.0
                  TSTS(5,INDEX)=1.73205•SIGRES
                  DO 330 K=1,NCN
```

```
                GASH=0.
                IF(NPP.EQ.2)GASH=P(K)•STRS(4)/RP
                MASH=2•NOPE(K)-1
                NASH=MASH+1
                R1(MASH)=R1(MASH)-DI(1,K)•STRS(1)-
  1         DI(2,K)•STRS(3)-GASH
                R1(NASH)=R1(NASH)-DI(1,K)•STRS(3)-DI(2,K)•STRS(2)
330           CONTINUE
340         CONTINUE
350         CONTINUE
            MTGPY=MTGPY+NPLST
360       CONTINUE
          IF(NL.EQ.1) CALL DROP(NE,NP,AK)
          WRITE(8)(NOP(J),J=1,NECN),(IMAT(I),I=1,NE),(((DISN(K,I),
  1       TDIS(K,I)),K=1,2),(CORD(I,J),J=1,NCORD ),I=1,NP),(((TSTS(I,J),
  2       EPSTN(I,J)),I=1,5),J=1,NSTORE)
          REWIND 8
          WRITE(6,370)MTGPY,IFCOUNT
370       FORMAT('  TOTAL GAUSS POINTS YIELDED DURING ITERATION =',I5/
  1    ' TOTAL NUMBER OF ELEMENTS RUBBLEIZED DURING ITERATION =',I5)
          TIME4=SECNDS(TIME1)
          WRITE(6,380)TIME4
380       FORMAT('  RESULTANT FORCES OBTAINED AT',F10.3)

          RETURN
          END

•-------------------------------------------------------
•-------------------------------------------------------
•
•         Subroutine RESOLV:
•         Updates stiffness equation solution.
•
•         Written:
•         Last modified:JUL 18 1984
•
•-------------------------------------------------------
•-------------------------------------------------------
•
•         Called by:  SCRUBS
•
•-------------------------------------------------------
•-------------------------------------------------------

          SUBROUTINE RESOLV
```

```
           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
    1      NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
    2      NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
    3      TPWORK
           COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
    1      US(400,2)
           COMMON/LDS/R1(2000),RL(2000),RT(2000)
           COMMON/SCR/SK(20000) ,R

           READ(4) NTP ,(SK(I),I=NSK,NTP )
           NBUF=NSK-1
           ND=1
           NNP=1
           DO 60 L=1,NP
             DO 50 I=1,NDF
               IC=I+NDF•(L-1)
               R=R1(IC)
               NBUF=NBUF+1
               NSZ=SK(NBUF)
               NBUF=NBUF+1
               IF(SK(NBUF) GT.0.) GO TO 10
               RS=-R
               R=-U(ND)
               ND=ND+1
               R1(IC)=SK(NBUF)•R+RS
               GO TO 20
   10          CONTINUE
               R1(IC)=SK(NBUF)•R
   20          CONTINUE
               IF(L+I-NP-NDF) 30,70,30
   30          NBUF=NBUF+1
               DO 40 J=1,NSZ
                 IK=IC+J
                 R1(IK)=R1(IK)-SK(NBUF)•R
   40          NBUF=NBUF+1
               NBUF=NBUF+1
               IF(NBUF.LT.NTP) GO TO 50
               READ(4) NTP ,(SK(II),II=NSK,NTP)
               NBUF=NSK-1
               NNP=0
   50        NNP=NNP+1
   60      CONTINUE

   70      RETURN
```

```
          END

*---------------------------------------------------------
*---------------------------------------------------------
*
*         Subroutine ROSB:
*         Adjusts stiffness matrix.
*
*         Written:
*         Last modified:JUL 14 1984
*
*---------------------------------------------------------
*---------------------------------------------------------
*
*         Called by:  STIFM
*
*---------------------------------------------------------
*---------------------------------------------------------

          SUBROUTINE ROSB

          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1    NCORD
          COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
     1    US(400,2)
          COMMON/NUME/P(12),DEL(2,12),DET,DI(2,12),DIJ(2,2),XYE(3,12),
     1    NOPE(12),DISE(2,12),RP,RPB,XYP(3)
          COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1    ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)
     2    ,COJ(2,2),COJIN(2,2),B(4,24),ESTIFM(24,24),DVD(4,4),DVDB(4)

          DO 60 I=1,NCN
            DO 10 M=1,NB
              IF(NOPE(I).EQ.NBC(M)) GO TO 20
10          CONTINUE
            GO TO 60
20          IF(ANG(M).EQ.0.) GOTO 60
            GASH=ANG(M)*.017453292
            CS=COS(GASH)
            TN=TAN(GASH)
            MASH=2*I-1
            NASH=MASH+1
            DO 30 K=1,LV
              STFUC=ESTIFM(MASH,K)*CS
```

```
              STFVC=ESTIFM(NASH,K)•CS
              ESTIFM(MASH,K)=STFUC+STFVC•TN
              ESTIFM(NASH,K)=STFVC-STFUC•TN
30          CONTINUE
            DO 40 K=MASH,NASH
              STFUC=ESTIFM(K,MASH)•CS
              STFVC=ESTIFM(K,NASH)•CS
              ESTIFM(K,MASH)=STFUC+STFVC•TN
              ESTIFM(K,NASH)=STFVC-STFUC•TN
40          CONTINUE
            DO 50 J=MASH,NASH
              DO 50 K=1,LV
                ESTIFM(K,J)=ESTIFM(J,K)
50          CONTINUE
60        CONTINUE

          RETURN
          END

•------------------------------------------------------------
•------------------------------------------------------------
•
•         Subroutine ROTATE:
•         This subroutine rotates 2D stress or strain vector by angle THETA.
•
•         Written:
•         Last modified:MAR 19 1984
•
•------------------------------------------------------------
•------------------------------------------------------------
•
•         Called by:  JOINTM
•
•------------------------------------------------------------
•------------------------------------------------------------

          SUBROUTINE ROTATE(A,B,THETA)

    IMPLICIT DOUBLE PRECISION (A-H,O-Z)
    DIMENSION A(4),B(4)

    C=DCOS(THETA)
    S=DSIN(THETA)
    B(1)=A(1)•C•C+A(2)•S•S-2.•A(3)•C•S
    B(2)=A(1)•S•S+A(2)•C•C+2.•A(3)•C•S
```

```
      B(3)=A(1)•C•S+A(2)•C•S+A(3)•C•C+A(3)•S•S
      B(4)=A(4)

      RETURN
      END

•-----------------------------------------------------------------
•-----------------------------------------------------------------
•
•             Subroutine RUBDRAW:
•             Writes the rubble file RUB.MOV.
•
•             Written:
•             Last modified:JUL 19 1984
•
•-----------------------------------------------------------------
•-----------------------------------------------------------------
•
•             Called by:  SCRUBS
•
•-----------------------------------------------------------------
•-----------------------------------------------------------------

              SUBROUTINE RUBDRAW(NUMEL,TT,NUMNP)

              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
   1          NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
   2          NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
   3          TPWORK,KNE
              DIMENSION EPX4(20,1),R(1),Z(1),IX(5,1)
              COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
              COMMON/FG/ FLAG(1000),BULK(1000,2),MINES(10,50),NME(10),GAMMA
              COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
   1          ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000),DUMMY(1000)
              DIMENSION IXAR(4000)
              DIMENSION IXN(20),RN(12),ZN(12),ZETA(12),ETA(12)
              REAL TIME1,TIME2,TIME3
              DIMENSION IDUM(12)
              REAL CORDDUM(1000,2),R1,R2,R3,R4,Z1,Z2,Z3,Z4,ZERO

              DATA IXN/1,2,6,-4,2,3,8,-6,4,5,10,-9,5,7,11,-10,7,8,12,-11/
              DATA ZETA/-1., 0., 1.,-1.,-.5, 0., .5, 1.,-1.,-.5, .5, 1./
              DATA  ETA/-1.,-1.,-1., 0., 0., 0., 0., 0., 1., 1., 1., 1./
              DATA IDUM/8,16,24,31,39,47,55,62,70,78,86,94/
```

```
          NP1=1
          NP2=NUMEL
          IBEGIN=NUMEL
          ICNT=1

*---------------------------------------------------------------
*
*         Eliminates plotting of boundary around rubble.
*
*---------------------------------------------------------------

          IR1=0
          DO 20 I=1,NUMEL
            IF(BULK(I,1).EQ.0.0) GO TO 10
            NP2=NP2-1
            ICNT=ICNT+1
            GO TO 20
10          CONTINUE
            IR2=4*NSFR*(I-1)
            IXAR(IR1+1)=NOP(IR2+1)
            IXAR(IR1+2)=NOP(IR2+1+NSFR)
            IXAR(IR1+3)=NOP(IR2+1+NSFR*2)
            IXAR(IR1+4)=-NOP(IR2+1+NSFR*3)
            IR1=IR1+4*NSFR
20        CONTINUE
          WRITE(14,30) TT,ICNT
30        FORMAT (E15.5,I5)
          NONE=1
          NCON=4*NP2*NSFR
          WRITE(14,60) NONE,NUMNP,NP2,NCON
          WRITE (14,60) NP1,NP2
          ZERO=0.0
          DO 40 I=1,NUMNP
            CORDDUM(I,1)=SNGL(CORD(I,1))
40        CORDDUM(I,2)=SNGL(CORD(I,2))
          WRITE(14,50) ((CORDDUM(I,1),CORDDUM(I,2),ZERO),I=1,NUMNP)
50        FORMAT (6E12.5)
          WRITE(14,60) (IXAR(I),I=1,NCON)
60        FORMAT (16I5)
          ICNT=0
          NP1=NP2
          DO 100 I=1,NUMEL
            IF (BULK(I,1).EQ.0.0) GO TO 100
            ICNT=ICNT+1
            NP1=NP2+1
```

```
              NP2=NP1+4*NSFR
              INDEX=4*(I-1)*NSFR
              I1=NOP(INDEX+1)
              I2=NOP(INDEX+1+NSFR)
              I3=NOP(INDEX+1+NSFR*2)
              I4=NOP(INDEX+1+NSFR*3)
              Z1=SNGL(CORD(I1,2))
              Z2=SNGL(CORD(I2,2))
              Z3=SNGL(CORD(I3,2))
              Z4=SNGL(CORD(I4,2))
              R1=SNGL(CORD(I1,1))
              R2=SNGL(CORD(I2,1))
              R3=SNGL(CORD(I3,1))
              R4=SNGL(CORD(I4,1))
              ISWSW=0
              IF(ISWSW.EQ.0) GO TO 80
              IF(BULK(I,1).EQ.2.0) GO TO 80
              DO 70 KK=1,12
                IF(I.EQ.IDUM(KK)) ISWSW=1
70            CONTINUE

*-------------------------------------------------------------------
*
*            Leaves gap for unconsolidated rubble.
*
*-------------------------------------------------------------------

              ZMAX=AMAX1(Z1,Z2,Z3,Z4)
              ZMIN=AMIN1(Z1,Z2,Z3,Z4)
              IELE=IMAT(I)
              ALPH=ORT(IELE,8)
              RMAX=AMAX1(R1,R2,R3,R4)
              RMIN=AMIN1(R1,R2,R3,R4)
C             RDEL=RMAX-RMIN
C             AREA=BULK(I,2)
C             ANEW=AREA/ALPH
C             ZDEL=ANEW/RDEL
C             ZNEW=ZMIN+ZDEL
C             ZAVG=(Z1+Z2+Z3+Z4)*.25
              C1=CORD(I1,2)-TDIS(2,I1)
              C2=CORD(I2,2)-TDIS(2,I2)
              C3=CORD(I3,2)-TDIS(2,I3)
              C4=CORD(I4,2)-TDIS(2,I4)
              ZD1=ABS(C3-C1)
              ZD2=ABS(C4-C2)
```

```
          ZDST=.5•(ZD1+ZD2)
          ZLEN=ZDST/ALPH
          ZNEW=ZMIN+ZLEN
          ZNEW=SNGL(ZNEW)
          IF(Z1.GT.ZAVG.AND.FLAG(I1).GT.1.0) Z1=ZNEW
          IF(Z2.GT.ZAVG.AND.FLAG(I2).GT.1.0) Z2=ZNEW
          IF(Z3.GT.ZAVG.AND.FLAG(I3).GT.1.0) Z3=ZNEW
          IF(Z4.GT.ZAVG.AND.FLAG(I4).GT.1.0) Z4=ZNEW
80        CONTINUE

•-----------------------------------------------------------------
•
•        Computes nodal coordinates for intra-element rubble.
•
•-----------------------------------------------------------------

          DO 90 INDEX=1,12
            F1=1.-ZETA(INDEX)
            F2=1.+ZETA(INDEX)
            G1=1.-ETA(INDEX)
            G2=1.+ETA(INDEX)
            RN(INDEX)=.25•(F1•G1•R1+F2•G1•R2+F2•G2•R3+F1•G2•R4)
            ZN(INDEX)=.25•(F1•G1•Z1+F2•G1•Z2+F2•G2•Z3+F1•G2•Z4)
90        CONTINUE
          NJ=12
          NPT=5
          NCON=20
          WRITE(14,60) NONE,NJ,NPT,NCON
          WRITE(14,60) NP1,NP2
          WRITE (14,50) ((RN(J),ZN(J),ZERO),J=1,12)
          WRITE (14,60) (IXN(J),J=1,20)
100      CONTINUE

         RETURN
         END
```

```
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*            Subroutine RZBASE:
*            This subroutine finds the base R and Z dimensions for fall
*            calculations.
*
*            Written:
*            Last modified:MAR 19 1984
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*            Called by:  DROP
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*            Subroutines called:EXIT
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------

            SUBROUTINE RZBASE(I,IBOT,I1,I2,I3,I4,R,Z,RLEN,ZBASE)

            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1      ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000),DUMMY(1000)

            R1=CORD(I1,1)
            R2=CORD(I2,1)
            R3=CORD(I3,1)
            R4=CORD(I4,1)
            Z1=CORD(I1,2)
            Z2=CORD(I2,2)
            Z3=CORD(I3,2)
            Z4=CORD(I4,2)
            RAVG=(R1+R2+R3+R4)*.25
            ZAVG=(Z1+Z2+Z3+Z4)*.25
            RMX=0.0
            IF(R1.GT.RAVG)RMX=RMX+R1
            IF(R2.GT.RAVG)RMX=RMX+R2
            IF(R3.GT.RAVG)RMX=RMX+R3
            IF(R4.GT.RAVG)RMX=RMX+R4
            RMX=RMX*.5
```

```
            RMN=0.0
            IF(R1.LT.RAVG)RMN=RMN+R1
            IF(R2.LT.RAVG)RMN=RMN+R2
            IF(R3.LT.RAVG)RMN=RMN+R3
            IF(R4.LT.RAVG)RMN=RMN+R4
            RMN=RMN*.5
            RLEN=RMX-RMN

*-----------------------------------------------------------------
*
*           Determines ZBASE node.
*
*-----------------------------------------------------------------

            IF(I.NE.I1) GO TO 10
            ITEST1=I2
            ITEST2=I4
            GO TO 60
10          IF(I.NE.I2) GO TO 20
            ITEST1=I1
            ITEST2=I3
            GO TO 60
20          IF(I.NE.I3) GO TO 30
            ITEST1=I2
            ITEST2=I4
            GO TO 60
30          IF(I.NE.I4) GO TO 40
            ITEST1=I3
            ITEST2=I1
            GO TO 60
40          WRITE(6,50) I,IBOT
50          FORMAT('   NODE NUMBER ',I5,' IS NOT OF ELEMENT ',I5/)
            CALL EXIT
60          CONTINUE
            IF(CORD(ITEST1,2).LT.CORD(ITEST2,2)) GO TO 70

*-----------------------------------------------------------------
*
*           ITEST2 is ZBASE node.
*
*-----------------------------------------------------------------

            ZBASE=CORD(ITEST2,2)
            RETURN
```

```
*------------------------------------------------------------
*
*            ITEST1 is ZBASE node.
*
*------------------------------------------------------------

70           ZBASE=CORD(ITEST1,2)

             RETURN
             END
```

```
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*          Main SCRUBS: Program designed to compute the failure, collapse,
*          and resulting subsidence of geologic materials.
*
*          Elastic Plastic analysis of plane stress/strain and axisymmetric
*          problems for linear, parabolic, and cubic elements.
*
*          Written by:
*          G. NAYAK..........University College of Swansea
*
*          Compiled for CDC 7600 by:
*          C. ANDERSON.......Los Alamos Scientific Laboratory
*          RV BROWNING.......Los Alamos Scientific Laboratory
*
*          Modified to treat finite strain, large deformation, and
*          geotechnical rubble formation by:
*          S.E. BENZLEY......Brigham Young University
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*          Called by:SCRUBSV (a file driver)
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*          Subroutines called:SOLVEREADF
*          ZONEOUTPUTLDATA
*          GDATASETPDISP
*          FLAGETFORMRESIDUE
*          RESOLVBSUBSTIFM
*          CONVGRUBDRAWDATALST (not used at present)
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------

           SUBROUTINE SCRUBS

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/ELPRNT/ NELP,NPEL(50),NONODE
           COMMON/DUMM/ SAVG(4),SPLT(4,1000)
           COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1     NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
```

```
     2       NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3       TPWORK,KNE
             COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
             COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
     1       US(400,2)
             COMMON/LDS/R1(2000),RL(2000),RT(2000)
             COMMON/GID/XG(6),CG(6),XMG(6),CMG(6),NGAUS,MGAUS,MCN
             COMMON/MATP/D(4,4),YM,PR,GM,HM,PPR,CLAME,ALPHA,YST
     1       ,TABSTN(10,15),TABSTR(10,15)
             COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),
     1       CORD(1000,2)
     1       ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000),
     2       DUMMY(1000)
             COMMON/DLIST/ LN,LNCT,TITL (20)
             COMMON/FG/ FLAG(1000),BULK(1000,2),MINES(10,50),NME(10),
     1       GAMMA
             COMMON/BOTTOM/ AK
             COMMON/QMESH/NEL
             DIMENSION IRAY(2)
             REAL TINC,UDUM(2,1000),SPLTDUM(4,1000),CORDDUM(1000,2)
             REAL TIME1,TIME2,TIME3,TITLD(20)
             DATA LNCT/55/
             DATA IRAY(1),IRAY(2)/10,12/

C            CALL FTNBIN(1,2,IRAY)
C            REWIND 12
C            CALL DATALST(12,5,6)
             NRS=0

*-------------------------------------------------------------
*
*            Opens and reads data from QMESH9.DAT file in subroutine ZONE.
*
*-------------------------------------------------------------

      CALL ZONE(NRS)
             IF(NRS.NE.1) GO TO 20

*-------------------------------------------------------------
*
*            Reads restart file, SCRUBS.RST.
*
*-------------------------------------------------------------

             REWIND(2)
```

```
      READ(2) (TITLE(I),I=1,57),(FAC(I),I=1,1210),(NBC(I),I=1,1800),
     1 (R1(I),I=1,6000),(XG(I),I=1,27),((D(I,J),I=1,4),J=1,4),
     2 ((DIS(I,J),I=1,2),J=1,1000),LN,LNCT,(TITL (I),I=1,20),
     3 (FLAG(I),I=1,3510),AK,NELP,(NPEL(I),I=1,51)
      NECN=NE*NCN
      READ(2) (NOP(J),J=1,NECN),(IMAT(I),I=1,NE),(((DISN(K,I),
     1 TDIS(K,I)),K=1,2),(CORD(I,J),J=1,NCORD ),I=1,NP),(((TSTS(I,J),
     2 EPSTN(I,J)),I=1,5),J=1,NSTORE)
     3 ,((TABSTN(I,J),I=1,10),J=1,15),((TABSTR(I,J),I=1,10),J=1,15)
      REWIND (8)
      WRITE(8)(NOP(J),J=1,NECN),(IMAT(I),I=1,NE),(((DISN(K,I),
     1 TDIS(K,I)),K=1,2),(CORD(I,J),J=1,NCORD ),I=1,NP),(((TSTS(I,J),
     2 EPSTN(I,J)),I=1,5),J=1,NSTORE)
      REWIND (8)

*----------------------------------------------------------
*
*     Writes the solution controls to SCRUBS.LIS.
*
*----------------------------------------------------------

      WRITE(6,10) (TITLE(I),I=1,20)
10    FORMAT(' ',20A4,//' ****  THIS PROBLEM HAS BEEN RESTARTED  '
     1 '****'/)
      WRITE(6,15) NP,NE,NB,NLD,NDF,NMAT,NSFR,NGAUS,NALGO,NPP,
     1 NYIELD,NT,NL,NSIZ,NCORD,MGAUS
15    FORMAT('             CONTROL PARAMETERS'/
     1 '        NUMBER OF NODAL POINTS--------',I3//
     2 '        NUMBER OF ELEMENTS-----------',I3//
     3 '        NUMBER OF BOUNDARY POINTS----',I3//
     4 '        NUMBER OF LOAD CASES----------',I2//
     5 '        NUMBER OF DEGREES OF FREEDOM--',I2//
     6 '        NUMBER OF MATERIALS-----------',I2//
     7 '        ORDER OF THE ELEMENTS---------',I2//
     8 '        NUMBER OF GAUSS POINTS--------',I2//
     9 '        SOLUTION ALGORITHM------------',I2//
     A '        GEOMETRY NUMBER---------------',I2//
     B '        YIELD CONDITION NUMBER--------',I2//
     C '        NUMBER OF ARBITRARY STIFF.----',I2//
     D '        BLANK-------------------------',I2//
     E '        BAND WIDTH (NOT NEEDED)-------',I2//
     F '        NUMBER OF COORDINATES---------',I2//
     G '        GAUSS PTS FOR. RES. CALC.-----',I2)
      CALL OUTPUT
      GO TO 30
```

```
20          CONTINUE

*-----------------------------------------------------------------
*
*           Reads NPROB, the number of problems, from SCRUBS.DAT, and
*           writes NPROB and the starting time to SCRUBS.LIS.
*
*-----------------------------------------------------------------

            READ(5,25) NPROB
25          FORMAT(16I5)
            DO 760 NPR=1,NPROB
              TIME1=SECNDS(0.0)
              WRITE(6,26) NPR,TIME1
26            FORMAT('  PROBLEM NO.',I3,3X,'EXECUTION STARTED AT ',
     1         F10.3)

*-----------------------------------------------------------------
*
*           Reads the title from SCRUBS.DAT and writes it to SCRUBS.LIS.
*
*-----------------------------------------------------------------

              READ(5,27) (TITLE(I),I=1,20)
27      FORMAT(20A4)
              WRITE(6,27) (TITLE(I),I=1,20)

*-----------------------------------------------------------------
*
*           Reads solution control data from SCRUBS.DAT.
*
*-----------------------------------------------------------------

              FACTOR=0.
              READ (5,25)NP,NE,NB,NLD,NDF,NMAT,NSFR,NGAUS,NALGO,NPP,
     1      NYIELD,NT,NL,NSIZ,NCORD,MGAUS

*-----------------------------------------------------------------
*
*           Opens RUB.MOV file if rubble flag is set equal to 1.
*
*-----------------------------------------------------------------

              IF(NL.EQ.1)OPEN(UNIT=14,TYPE='NEW',ACCESS='SEQUENTIAL',
     1      NAME='RUB.MOV')
```

```
*----------------------------------------------------------
*
*            Reads comment header from first line of QMESH9.DAT,
*            then the number of elements (NE) and the
*            number of nodal points (NP).
*
*----------------------------------------------------------

              REWIND 9
C             READ(9) KOMT
              IF(NEL.EQ.0) READ(9) KOMT
              READ(9) NE,NP

*----------------------------------------------------------
*
*            Writes the solution controls to SCRUBS.LIS.
*
*----------------------------------------------------------

              IF(NCORD.EQ.0) NCORD=2
              IF(MGAUS.EQ.0)MGAUS=NGAUS
              WRITE(6,15)NP,NE,NB,NLD,NDF,NMAT,NSFR,NGAUS,NALGO,NPP,
    1        NYIELD,NT,NL,NSIZ,NCORD,MGAUS

*-------------------------------------------------------------
*
*            Reads material properties.
*
*-------------------------------------------------------------

30            CONTINUE
              DO 40 L=1,NMAT
                READ (5,35)N,(ORT(N,I),I=1,6)
35              FORMAT(I10,6G10.4)
40            IF(ORT(N,2).LT.0.0) READ (5,50) (ORT(N,I),I=9,10)
50            FORMAT(2G12.5)
              WRITE(6,60)
60            FORMAT(' MATERIAL PROPERTIES'/
    1         '     MATL   YOUNGS MOD    POIS RAT      YIELD'
    2    '   HARD MOD  ANGLE  THICK ')
              DO 90 N=1,NMAT
                IF(NYIELD.NE.3) GO TO 70

*----------------------------------------------------------
```

```
*
*          Adjusts the yield stress and the yield angle for Von Mises
*          failure criteria.
*
*------------------------------------------------------------------

             C=ORT(N,3)
             THETA=ORT(N,5)
             SN=SIN(THETA)
             DNOM=1.73205080757*(3.-SN)
             ORT(N,3)=6.*C*COS(THETA)/DNOM
             ORT(N,5)=2.*SN/DNOM
70           WRITE(6,80)N,(ORT(N,I),I=1,6)
80           FORMAT(I10,G12.4,G12.4,2G12.4,3G9.4)
90          IF(ORT(N,2).LT.0.0)WRITE(6,100) (ORT(N,I),I=9,10)
100         FORMAT('     JOINTED MEDIA MATERIAL-- DELTA=',F8.4,
   1        'GS=',E10.4)

*---------------------------------------------------------------------
*
*          Tabulated stress/plastic-strain data.
*
*---------------------------------------------------------------------

            WRITE(6,110)
110         FORMAT(' TABULATED PLASTIC STRAIN-STRESS DATA')
            DO 180 N=1,NMAT
             TABSTR(N,1)=ORT(N,3)
             IF(ORT(N,4).LT.0.) GO TO 120
             NTAB=2
             TABSTN(N,2)=1.
             TABSTR(N,2)=TABSTR(N,1)+ORT(N,4)
             GO TO 140
*------------------------------------------------------------------
*
*          Reads tabulated stress/strain data from SCRUBS.DAT.
*
*------------------------------------------------------------------


120          READ(5,35) NTAB
             READ(5,130)(TABSTR(N,I),I=2,NTAB)
130          FORMAT(10F8.0)
             READ(5,130) (TABSTN(N,I),I=2,NTAB)
140          CONTINUE
```

```
                TABSTN(N,1)=0.0

•------------------------------------------------------------
•
•          Writes tabulated stress/strain data to SCRUBS.LIS.
•
•------------------------------------------------------------

                WRITE(6,150)
150             FORMAT(' TABULATED STRESS VALUES')
                WRITE(6,160) (TABSTR(N,I),I=1,NTAB)
160             FORMAT(' ',15E9.3)
                WRITE(6,170)
170             FORMAT(' TABULATED STRAIN VALUES')
180           WRITE(6,190) (TABSTN(N,I),I=1,NTAB)
190           FORMAT(1X,15E9.3)
              IF(NRS.EQ.1) GO TO 290

•------------------------------------------------------------
•
•          Reads and sets bulking parameters.
•
•------------------------------------------------------------

              IF(NL.EQ.0) GO TO 260
              DO 200 L=1,NMAT
200           READ(5,35) N,(ORT(N,I),I=7,8)
              WRITE(6,210)
210           FORMAT('   BULKING PROPERTIES'/' MATL FAILURE '
   1          'STRESS ALPHA'/)
              DO 220 N=1,NMAT
220           WRITE(6,80),N,(ORT(N,I),I=7,8)
              READ(5,230) AK
230           FORMAT(E10.3)
              WRITE(6,240) AK
240           FORMAT(' ++++++++ LOWER BOUNDARY INDICATOR, AK =',E15.4)

•------------------------------------------------------------
•
•          Reads failure parameters, NRFF = number of load steps with defined
•          element removal.
•
•------------------------------------------------------------

              READ(5,25) NRFF
```

```
          WRITE(6,250) NRFF
250       FORMAT(' FAILURE READ FLAG (I.E. NUMBER OF LOAD '
  1       'STEPS) = ',I5)
          IF(NRFF.NE.0) CALL READF(NRFF)
260       CONTINUE

*-----------------------------------------------------------
*
*        Preset discs.
*
*-----------------------------------------------------------

          REWIND 3
          REWIND 4
          REWIND 8
          ICS=3
          IF(NPP.EQ.2)ICS=4
          NCN=4*NSFR
          MCN=MGAUS*MGAUS
          NSZF=NP*NDF
          LV=NCN*NDF
          NSTORE=MCN*NE
          WRITE(6,270)NCN,MCN,LV,NSZF,ICS,NSTORE
270       FORMAT(' NODES/ELEMENT=',I2,2X,' GAUSSIAN POINTS=',I3,
  1        2X,'UNKNOWNS/ELEMENT=',I3,2X,'TOTAL UNKNOWNS=',I4,2X,
  2        'DMATRIX SIZE=',I2,2X,'STRESS STORE=',I4)

*-----------------------------------------------------------
*
*        Recomputes NSTORE if slip plane is added.
*
*-----------------------------------------------------------

          IIF=1
          IF(NNP.GT.0)IIF=-1

*-------------------------------------------------------
*
*        Reads geometrical data.
*
*-------------------------------------------------------

          CALL GDATA (IIF)

*-----------------------------------------------------------
```

```
•
•          Sets initial areas.
•
•------------------------------------------------------------

           IF(NL.EQ.0) GO TO 290
           DO 280 N=1,NE
             K1=(N-1)•NSFR•4+1
             K2=K1+NSFR
             K3=K2+NSFR
             K4=K3+NSFR
             J1=NOP(K1)
             J2=NOP(K2)
             J3=NOP(K3)
             J4=NOP(K4)
             COE1=CORD(J2,1)-CORD(J4,1)
             COE2=CORD(J3,2)-CORD(J1,2)
             COE3=CORD(J3,1)-CORD(J1,1)
             COE4=CORD(J4,2)-CORD(J2,2)
             AREA=.5•(COE1•COE2+COE3•COE4)
             BULK(N,1)=0.
             BULK(N,2)=AREA
280        CONTINUE
290        CONTINUE

•------------------------------------------------------------
•
•          Writes first record for plot tape output (SCRUBS.MOV).
•
•------------------------------------------------------------

           DO 300 I=1,20
300        TITLD(I)=TITLE(I)
           WRITE(10) TITLD,NE,NP
           DO 310 I=1,NP
             CORDDUM(I,1)=SNGL(CORD(I,1))
310        CORDDUM(I,2)=SNGL(CORD(I,2))
           WRITE(10)(CORDDUM(I,1),I=1,NP),(CORDDUM(I,2),I=1,NP),
    1((NOP(NCN•(K-1)+M),K=1,NE),M=1,NCN),(IMAT(I),I=1,NE)
           IF(NRS.EQ.1) GO TO 360

•------------------------------------------------------------
•
•          Sets element spacing and checks bandwith and stiffness dimensioning.
•
```

```
•----------------------------------------------------------

            CALL SET

•-------------------------------------------------------------
•
•         Inputs loads.
•
•-------------------------------------------------------------

            DO 750 ILD=1,NLD
              TTWORK=0.
              READ(5,25)ISTS,LDTYPE
              WRITE(6,320)ISTS,LDTYPE
320           FORMAT(' INITIAL STRESS COUNTER=',I2,2X,'LOAD TYPE '
    1          'COUNTER=',I2)
              NASH=0
              DO 330 J=1,NP
                DO 330 I=1,NDF
                  NASH=NASH+1
C                 330 TDIS(I,J)=RL(NASH)=DISN(I,J)=RT(NASH)=R1(NASH)=0>
                  TDIS(I,J)=0.
                  RL(NASH)=0.0
                  DISN(I,J)=0.0
                  RT(NASH)=0.0
                  R1(NASH)=0.0
330           CONTINUE
              DO 340 J=1,NSTORE
                DO 340 I=1,5
                  EPSTN(I,J)=0.
340           TSTS(I,J)=0.
              NECN=NE•NCN
              WRITE(8)(NOP(J),J=1,NECN),(IMAT(I),I=1,NE),(((DISN(K,I),
    1       TDIS(K,I)),K=1,2),(CORD(I,J),J=1,NCORD ),I=1,NP),
    2     (((TSTS(I,J),EPSTN(I,J)),I=1,5),J=1,NSTORE)
              REWIND 8
              NASH=0
              DO 350 I=1,NB
                DO 350 K=1,2
                  NASH=NASH+1
350           TRC(NASH)=0.
              IF(LDTYPE.EQ.2) GO TO 410
              CALL LDATA(IIF)
360           CONTINUE
              READ (5,25)NINC
```

```
              WRITE(6,370) NINC
370           FORMAT(' NO. OF INCREMENTS =',I2/
   1           ' INCREMENT     OUTPUT  LD FACTOR')
              READ (5,380)(FAC(I),I=1,NINC)
380           FORMAT(16F5.3)
              READ (5,25)(NOUT(I),I=1,NINC)
              DO 390  I=1,NINC
390           WRITE(6,400)I,NOUT(I),FAC(I)
400           FORMAT(2I10,F10.3)
              GO TO 420
410           FACTOR=1.
420           CONTINUE
              READ (5,35)NIT,CONFAC
              WRITE(6,430)NIT,CONFAC
430           FORMAT(' NUMBER ITERATIONS=',I2,5X,'CONVERGENCE '
   1           'FACTOR =',F10.2)
              IF(NL.EQ.1.AND.NRS.NE.1) CALL FLAGE(NP,NE,5,IIF)

*---------------------------------------------------------------
*
*           Sets print controls.
*
*---------------------------------------------------------------

              READ(5,440) NELP,NONODE
440           FORMAT(16I5)
              IF(NELP.LE.0) GO TO 470
              READ(5,440) (NPEL(K),K=1,NELP)
              WRITE(6,450)
450           FORMAT('     ELEMENTS TO BE PRINTED')
              WRITE(6,460) (NPEL(K),K=1,NELP)
460           FORMAT(1X,26I5/26I5)
470           CONTINUE

*---------------------------------------------------------------
*
*           Begins loop for each increment.
*
*---------------------------------------------------------------

              DO 740 INC=1,NINC
                TPWORK=0.
                WRITE(6,480)INC
480         FORMAT('  LOAD INCREMENT NO. =',I3)
                IF(LDTYPE.EQ.2) GO TO 500
```

```
            GASH=FAC(INC)
            FACTOR=FACTOR+GASH
            NASH=0
            DO 490 J=1,NP
              DO 490 I=1,NDF
                NASH=NASH+1
                GOSH=RL(NASH)*GASH
                RT(NASH)=RT(NASH)+GOSH
                R1(NASH)=R1(NASH)+GOSH
490         CONTINUE
            CALL TFORM
            GO TO 510
500         CALL LDATA(IIF)
510         IF(INC.GT.1 .AND. NALGO.NE.1) GO TO 530
            CALL PDISP(1)
            CALL STIFM
            TIME3=SECNDS(TIME1)
            WRITE(6,520)TIME3
520     FORMAT(' STIFFNESS FORMULATION FINISHED AT',F10.3)
            CALL SOLVE
            GO TO 540
530         CALL PDISP(INC)
            CALL RESOLV
540         CALL BSUB
        IF(NALGO.EQ.0) CALL RESIDUE (TIME1,IFCOUNT,MTGPY)
            IF(NALGO.EQ.0) GO TO 600
            CALL PDISP(0)

*------------------------------------------------------------
*
*       Begins loop for each iteration.
*
*------------------------------------------------------------

            DO 580 IT=1,NIT
              CALL RESIDUE (TIME1,IFCOUNT,MTGPY)
              CALL CONVG(NCHECK)
              CALL TFORM
              IF(IT.EQ.1 .AND. NOUT(INC).GT.100)CALL OUTPUT
              IF(NCHECK.EQ.0) GO TO 600
              IF(NALGO.LT.2)GO TO 560
              IF(NALGO.GT.2)GO TO 550
              IF(IT.NE.1)GO TO 560
550           CONTINUE
```

```
*-----------------------------------------------------------------
*
*             If no elements have yielded or rubblized, do not reform stiffness
*     matrix.
*
*-----------------------------------------------------------------

              IICK=IFCOUNT+MTGPY
              IF(IICK.EQ.0) GO TO 560
                 CALL STIFM
                 TIME3=SECNDS(TIME1)
                 WRITE(6,520)TIME3
                 CALL SOLVE
                 GO TO 570
560              CALL RESOLV
570              CALL BSUB
580            CONTINUE
               IT=NIT+1
               CALL RESIDUE (TIME1,IFCOUNT,MTGPY)
               CALL CONVG(NCHECK)
               WRITE(6,590) INC
590         FORMAT('        NO CONVERGENCE ON INCREMENT NO',I3)
               GO TO 760
600            CONTINUE
               CALL OUTPUT

*-----------------------------------------------------------------
*
*             Write restart tape (SCRUBS.RST).
*
*-----------------------------------------------------------------

               REWIND(2)
               WRITE(2) (TITLE(I),I=1,57),(FAC(I),I=1,1210),(NBC(I),
     1      I=1,1800),(R1(I),I=1,6000),(XG(I),I=1,27),((D(I,J),
     2      I=1,4),J=1,4),((DIS(I,J),I=1,2),J=1,1000),LN,LNCT,
     3      (TITL (I),I=1,20),(FLAG(I),I=1,3510),AK,NELP,
     4         (NPEL(I),I=1,51)
               NECN=NE*NCN
               WRITE(2)(NOP(J),J=1,NECN),(IMAT(I),I=1,NE),
     1         (((DISN(K,I),TDIS(K,I)),K=1,2),(CORD(I,J),J=1,NCORD ),
     2          I=1,NP),(((TSTS(I,J),EPSTN(I,J)),I=1,5),J=1,NSTORE),
     3         ((TABSTN(I,J),I=1,10),J=1,15),((TABSTR(I,J),I=1,10),
     4          J=1,15)
               WRITE(6,610)
```

```
610             FORMAT(' * * * *  RESTART FILE HAS BEEN '
   1          'WRITTEN  * * * *')
                IF(NALGO.EQ.0) GO TO 740
                TTWORK=TTWORK+TPWORK
                WRITE(6,620)INC,TTWORK
620           FORMAT(' INC NO ',I2,5X,'TOTAL PLASTIC WORK =',E12.4)
                TIME2=SECNDS(TIME1)
                WRITE(6,630)  TIME2
630           FORMAT(4X,'EXECUTION FINISHED AT ',F10.3)

*---------------------------------------------------------------
*
*             Computes average stress in each element.
*
*---------------------------------------------------------------

                XMCN=MCN
                DO 670 M=1,NE
                  K=MCN*(M-1)
                  DO 640 I=1,4
640               SAVG(I)=0.0
                  DO 650 I=1,4
                    DO 650 L=1,MCN
                      J=K+L
650               SAVG(I)=SAVG(I)+TSTS(I,J)
                  DO 660 I=1,4
660               SAVG(I)=SAVG(I)/XMCN
                  SPLT(1,M)=SAVG(1)
                  SPLT(2,M)=SAVG(2)
                  SPLT(3,M)=SAVG(4)
                  SPLT(4,M)=SAVG(3)
670             CONTINUE

*---------------------------------------------------------------
*
*             Writes displacements, stresses, and strains for plot tape
*             SCRUBS.MOV.
*
*---------------------------------------------------------------

                TINC=INC
                WRITE(10) TINC
                DO 680 I=1,NP
                  UDUM(1,I)=SNGL(TDIS(1,I))
680             UDUM(2,I)=SNGL(TDIS(2,I))
```

```
                DO 690 J=1,NE
                  DO 690 I=1,4
690             SPLTDUM(I,J)=SNGL(SPLT(I,J))
                WRITE(10) (UDUM(1,I),UDUM(2,I),I=1,NP),
     1    (UDUM(1,I),UDUM(2,I),I=1,NP),
     2    (UDUM(1,I),UDUM(2,I),I=1,NP)
                WRITE(10) ((SPLTDUM(I,J),J=1,NE),I=1,4)

*-----------------------------------------------------------
*
*          Writes rubble file (RUB.MOV).
*
*-----------------------------------------------------------

                IF(NL.EQ.1) CALL RUBDRAW(NE,TINC,NP)

*-----------------------------------------------------------
*
*          Computes average strain in each element.
*
*-----------------------------------------------------------

                DO 730 M=1,NE
                  K=MCN*(M-1)
                  DO 700 I=1,4
700               SAVG(I)=0.0
                  DO 710 I=1,4
                    DO 710 L=1,MCN
                      J=K+L
710           SAVG(I)=SAVG(I)+EPSTN(I,J)
                  DO 720 I=1,4
720           SAVG(I)=SAVG(I)/XMCN
                  SPLTDUM(1,M)=SNGL(SAVG(1))
                  SPLTDUM(2,M)=SNGL(SAVG(2))
                  SPLTDUM(3,M)=SNGL(SAVG(4))
                  SPLTDUM(4,M)=SNGL(SAVG(3))
730             CONTINUE
                WRITE(10) ((SPLTDUM(I,J),J=1,NE),I=1,4)
740           CONTINUE
750         CONTINUE
760       CONTINUE

          RETURN
          END
```

```
*------------------------------------------------------------
*------------------------------------------------------------
*
*           Subroutine SCRUBSV:  This subroutine opens and closes the files
*           used by the main program, SCRUBS.  The files:
*           UNITNAME
*             2SCRUBS.RST
*             3scratch file
*             4scratch file
*             5SCRUBS.DAT
*             6SCRUBS.LIS
*             8scratch file
*             9QMESH9.DAT
*            10SCRUBS.MOV
*            14RUB.MOV
*
*           Written:
*           Last modified:MAR 19 1984
*
*------------------------------------------------------------
*------------------------------------------------------------
*
*           Subroutines called:SCRUBSEXIT
*
*------------------------------------------------------------
*------------------------------------------------------------

            PROGRAM SCRUBSV

            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1      NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2      NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3      TPWORK,KNE
            COMMON/QMESH/NEL

*--------------------------------------------------------
*
*           The files are opened.
*
*--------------------------------------------------------

            OPEN(UNIT=2,TYPE='UNKNOWN',ACCESS='SEQUENTIAL',
     1  NAME='SCRUBS.RST',FORM='UNFORMATTED')
            OPEN(UNIT=3,TYPE='SCRATCH',ACCESS='SEQUENTIAL',
```

```
     1           FORM='UNFORMATTED')
                 OPEN(UNIT=4,TYPE='SCRATCH',ACCESS='SEQUENTIAL',
     1           FORM='UNFORMATTED')
      OPEN(UNIT=5,TYPE='OLD',ACCESS='SEQUENTIAL',NAME='SCRUBS.DAT')
      OPEN(UNIT=6, TYPE='NEW',ACCESS='SEQUENTIAL',NAME='SCRUBS.LIS')
      OPEN(UNIT=10,TYPE='NEW',ACCESS='SEQUENTIAL',NAME='SCRUBS.MOV',
     1     FORM='UNFORMATTED')
      OPEN(UNIT=9,TYPE='UNKNOWN',ACCESS='SEQUENTIAL',
     1     NAME='QMESH9.DAT',FORM='UNFORMATTED')
      OPEN(UNIT=8,TYPE='SCRATCH',ACCESS='SEQUENTIAL',
     1     FORM='UNFORMATTED')

*------------------------------------------------------------
*
*                The main program, SCRUBS, is called.
*
*------------------------------------------------------------

                 CALL SCRUBS

*------------------------------------------------------------
*
*                The files are closed, and the system call EXIT is used.
*
*------------------------------------------------------------

      CLOSE(UNIT=2)
      CLOSE(UNIT=3)
      CLOSE(UNIT=4)
      CLOSE(UNIT=5)
      CLOSE(UNIT=6)
      IF(NEL.EQ.0) CLOSE(UNIT=9)
      IF(NEL.NE.0) CLOSE(UNIT=9,DISPOSE='DELETE')
      IF(NL.EQ.1)CLOSE(UNIT=10)
      CLOSE(UNIT=8)
      CALL EXIT

                 STOP
      END
```

```
•------------------------------------------------------------------
•------------------------------------------------------------------
•
•          Subroutine SET:
•          Sets element spacing and determines bandwidth and resulting
•          stiffness storage required.
•
•          Written:
•          Last modified:MAR 19 1984
•
•------------------------------------------------------------------
•------------------------------------------------------------------
•
•          Called by:  SCRUBS
•
•------------------------------------------------------------------
•------------------------------------------------------------------

           SUBROUTINE SET

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS
   1       ,NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
   2       NSTORE
           COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
           COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
   1       ,NOP(4000),IMAT(1000),ION(6,1000)

           DO 10 J=1,6
             DO 10 I=1,NP
10         ION(J,I)=0

•------------------------------------------------------------------
•
•          Selects spacing.
•
•------------------------------------------------------------------

           DO 50 L=1,NE
             NASH=NCN•(L-1)
             IST=NOP(NASH+1)
             MAX=IST
             MIN=IST
             DO 20 N=2,NCN
               NTH=NOP(NASH+N)
```

```
              IF (NTH .EQ. 0) GO TO 30
              IF ( MAX .LT. NTH) MAX=NTH
              IF ( MIN .LE. NTH) GO TO 20
              MIN=NTH
20          CONTINUE
30          IF (NSIZ .LT. (MAX-MIN)*NDF) NSIZ=(MAX-MIN)*NDF
            I=MIN
            DO 40 M=1,6
              IF(ION(M,I).NE.0) GO TO 40
              ION(M,I)=L
              GO TO 50
40          CONTINUE
50        CONTINUE
          L=1
          DO 70 I=1,NP
            DO 60 M=1,6
              IF(ION(M,I).EQ.0) GO TO 70
              IONARY(L)=ION(M,I)
60          L=L+1
70        CONTINUE
          IONARY(L)=0

*-----------------------------------------------------------
*
*         Determines if bandwidth is too high and stiffness dimensioning
*         is exceeded.
*
*-----------------------------------------------------------

          NSK=((NSIZ+NDF)*(NSIZ+NDF+1))/2+1
          IF(NSK.LT.NSZF+1) NSK=NSZF+1
          IF(NSK.LE.9000) RETURN
          WRITE(6,80)
80        FORMAT(' PROGRAM HALTED IN SET, '
   1 'STIFFNESS SPACE EXCEEDED')

          STOP
          END
```

```
*---------------------------------------------------------------
*---------------------------------------------------------------
*
*          Subroutine SFR:
*          Sets shape functions according to element type.
*
*          Written:
*          Last modified:JUL 14 1984
*
*---------------------------------------------------------------
*---------------------------------------------------------------
*
*          Called by:  LDATAREMLDRESIDUE
*          STIFM
*
*---------------------------------------------------------------
*---------------------------------------------------------------

           SUBROUTINE SFR(G,H)

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1     NCORD
           COMMON/NUME/P(12),DEL(2,12),DET,DI(2,12),DIJ(2,2),XYE(3,12),
     1     NOPE(12),DISE(2,12),RP,RPB,XYP(3)

*---------------------------------------------------------------
*
*          G and H denote the xi and eta values at the point considered.
*
*---------------------------------------------------------------

           GG = G*G
           GH = G*H
           HH = H*H
           GGH = GG*H
           GHH = G*HH
           IF(NSFR-2)10,20,30

*---------------------------------------------------------------
*
*          Linear shape functions and their first derivatives for
*          four noded arbitrary quadrilateral.
*
*---------------------------------------------------------------
```

```
10        P(1)=(1.-G-H+GH)/4.
          P(2)=(1.+G-H-GH)/4.
          P(3)=(1.+G+H+GH)/4.
          P(4)=(1.-G+H-GH)/4.
          DEL(1,1)=(-1.+H)/4.
          DEL(1,2)=-DEL(1,1)
          DEL(1,3)=(1.+H)/4.
          DEL(1,4)=-DEL(1,3)
          DEL(2,1)=(-1.+G)/4.
          DEL(2,2)=(-1.-G)/4.
          DEL(2,3)=-DEL(2,2)
          DEL(2,4)=-DEL(2,1)
          RETURN

*-----------------------------------------------------------------
*
*         Parabolic shape functions and their first derivatives for
*         curved 8-nodal arbitrary quadrilateral.
*
*-----------------------------------------------------------------

20        G2 = G*2.
          H2 = H*2.
          GH2 = GH*2.
          P( 1) = (-1.+GH+GG+HH-GGH-GHH)/4.
          P( 2) = ( 1.-H-GG+GGH)/2.
          P( 3) = (-1.-GH+GG+HH-GGH+GHH)/4.
          P( 4) = ( 1.+G-HH-GHH)/2.
          P( 5) = (-1.+GH+GG+HH+GGH+GHH)/4.
          P( 6) = ( 1.+H-GG-GGH)/2.
          P( 7) = (-1.-GH+GG+HH+GGH-GHH)/4.
          P( 8) = ( 1.-G-HH+GHH)/2.
          DEL(1,1)=( H+G2-GH2-HH)/4.
          DEL(1,2)= -G+GH
          DEL(1,3)=(-H+G2-GH2+HH)/4.
          DEL(1,4)=( 1.-HH)/2.
          DEL(1,5)=( H+G2+GH2+HH)/4.
          DEL(1,6)= -G-GH
          DEL(1,7)=(-H+G2+GH2-HH)/4.
          DEL(1,8)=(-1.+HH)/2.
          DEL(2,1)=( G+H2-GG-GH2)/4.
          DEL(2,2)=(-1.+GG)/2.
          DEL(2,3)=(-G+H2-GG+GH2)/4.
          DEL(2,4)= -H-GH
          DEL(2,5)=( G+H2+GG+GH2)/4.
```

```
          DEL(2,6)=( 1.-GG)/2.
          DEL(2,7)=(-G+H2+GG-GH2)/4.
          DEL(2,8)= -H+GH
          RETURN
30        CONTINUE
          GGG = GG*G
          HHH = H*HH
          GGGH = GGG*H
          GHHH = G*HHH

*-----------------------------------------------------------
*
*         Cubic shape functions and their first derivatives for
*         curved 12-nodal arbitrary quadrilateral.
*
*-----------------------------------------------------------

          G9 = G*9.
          G10 = G*10.
          G18 = G*18.
          G27 = G*27.
          H9 = H*9.
          H10 = H*10.
          H18 = H*18.
          H27 = H*27.
          GG9 = GG*9.
          GG27 = GG*27.
          GG81 = GG*81.
          GH10 = GH*10.
          GH18 = GH*18.
          GH27 = GH*27.
          HH9 = HH*9.
          HH27 = HH*27.
          HH81 = HH*81.
          GGG9 = GGG*9.
          GGG27 = GGG*27.
          GGH9 = GGH*9.
          GGH27 = GGH*27.
          GGH81 = GGH*81.
          GHH9 = GHH*9.
          GHH27 = GHH*27.
          GHH81 = GHH*81.
          HHH9 = HHH*9.
          HHH27 = HHH*27.
          GGGH9 = GGGH*9
```

```
           GGGH27 = GGGH•27.
           GHHH9 = GHHH•9.
           GHHH27 = GHHH•27.
           P(1)= -10.+G10+H10+GG9-GH10+HH9-GGG9-GGH9-GHH9-HHH9+GGGH9+GHHH9
           P(2)=  9.-G27-H9-GG9+GH27+GGG27+GGH9-GGGH27
           P(3)=  9.+G27-H9-GG9-GH27-GGG27+GGH9+GGGH27
           P(4)= -10.-G10+H10+GG9+GH10+HH9+GGG9-GGH9+GHH9-HHH9-GGGH9-GHHH9
           P(5)=  9.+G9-H27-GH27-HH9-GHH9+HHH27+GHHH27
           P(6)=  9.+G9+H27+GH27-HH9-GHH9-HHH27-GHHH27
           P(7)= -10.-G10-H10+GG9-GH10+HH9+GGG9+GGH9+GHH9+HHH9+GGGH9+GHHH9
           P(8)=  9.+G27+H9-GG9+GH27-GGG27-GGH9-GGGH27
           P(9)=  9.-G27+H9-GG9-GH27+GGG27-GGH9+GGGH27
           P(10)=-10.+G10-H10+GG9+GH10+HH9-GGG9+GGH9-GHH9+HHH9-GGGH9-GHHH9
           P(11)=  9.-G9+H27-GH27-HH9+GHH9-HHH27+GHHH27
           P(12)=  9.-G9-H27+GH27-HH9+GHH9+HHH27-GHHH27
           DEL(1, 1)= 10.+G18-H10-GG27-GH18-HH9+GGH27+HHH9
           DEL(1, 2)=-27.-G18+H27+GG81+GH18-GGH81
           DEL(1, 3)= 27.-G18-H27-GG81+GH18+GGH81
           DEL(1, 4)=-10.+G18+H10+GG27-GH18+HH9-GGH27-HHH9
           DEL(1, 5)= 9.-H27-HH9+HHH27
           DEL(1, 6)=  9.+H27-HH9-HHH27
           DEL(1, 7)=-10.+G18-H10+GG27+GH18+HH9+GGH27+HHH9
           DEL(1, 8)= 27.-G18+H27-GG81-GH18-GGH81
           DEL(1,10)= 10.+G18+H10-GG27+GH18-HH9-GGH27-HHH9
           DEL(1, 9)=-27.-G18-H27+GG81-GH18+GGH81
           DEL(1,11)=-9.-H27+HH9+HHH27
           DEL(1,12)=-9.+H27+HH9-HHH27
           DEL(2, 1)= 10.-G10+H18-GG9-GH18-HH27+GGG9+GHH27
           DEL(2, 2)=-9.+G27+GG9-GGG27
           DEL(2, 3)=-9.-G27+GG9+GGG27
           DEL(2, 4)= 10.+G10+H18-GG9+GH18-HH27-GGG9-GHH27
           DEL(2, 5)=-27.-G27-H18-GH18+HH81+GHH81
           DEL(2, 6)= 27.+G27-H18-GH18-HH81-GHH81
           DEL(2, 7)=-10.-G10+H18+GG9+GH18+HH27+GGG9+GHH27
           DEL(2, 8)= 9.+G27-GG9-GGG27
           DEL(2, 9)= 9.-G27-GG9+GGG27
           DEL(2,10)=-10.+G10+H18+GG9-GH18+HH27-GGG9-GHH27
           DEL(2,11)= 27.-G27-H18+GH18-HH81+GHH81
           DEL(2,12)=-27.+G27-H18+GH18+HH81-GHH81
           DO 50 J=1,12
             DO 40 I=1,2
               DEL(I,J)=DEL(I,J)/32.
40           CONTINUE
             P(J)=P(J)/32.
50         CONTINUE
```

```
      RETURN
      END

*------------------------------------------------------------
*------------------------------------------------------------
*
*     Subroutine SLIPP:
*     This subroutine reads the slip plane data, adjusts the geometry
*     and elements accordingly, and then records the adjusted
*     information.
*
*     Written:
*     Last modified:MAR 19 1984
*
*------------------------------------------------------------
*------------------------------------------------------------
*
*     Called by:  GDATA
*
*------------------------------------------------------------
*------------------------------------------------------------
*
*     Subroutines called:NEWNODCORNN
*     MAKEELNODEXY
*
*------------------------------------------------------------
*------------------------------------------------------------

      SUBROUTINE SLIPP

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1 NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2 NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3 TPWORK,KNE
      COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
     1 US(400,2)
      COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1 ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000),DUMMY(1000)
      COMMON/SLP/DCOR(1000,2)
      DIMENSION MAP(1000,2)
      INTEGER SLIP(500,2),CORNER(500),EL

      NSP=IABS(NNP)
```

```
•-----------------------------------------------------------
•
•             Saves old information.
•
•-----------------------------------------------------------

              NEOLD=NE
              NPOLD=NP
              DO 10 N=1,NP
         DCOR(N,1)=CORD(N,1)
10      DCOR(N,2)=CORD(N,2)

              WRITE(6,20)
20            FORMAT(///,1X,10('•'),' SLIP PANE BOUNDARY DATA ',10('•'))
              WRITE(6,30) NSP
30            FORMAT(//,' NUMBER OF SLIP PLANES = ',I2)

•-----------------------------------------------------------
•
•             Loops on number of slip planes.
•
•-----------------------------------------------------------

              DO 340 INSP=1,NSP

•-----------------------------------------------------------
•
•             Reads in slip plane data.
•
•-----------------------------------------------------------

                READ(5,40) NSLIP,IANG,DELTA
40              FORMAT(2I5,G10.3)

                K=(NSLIP-1)/16.+1
                J=1
                DO 60 I=1,K
          JJ=J+15
          IF(JJ.GE.NSLIP)JJ=NSLIP
          READ(5,50) (SLIP(N,1),N=J,JJ)
50                FORMAT(16I5)
60              J=J+16
                IF(DELTA.LE.O)DELTA=(DABS(CORD(1,1)-CORD(2,1))+
   +             DABS(CORD(1,2)-CORD(2,2)))/50.
```

```
•-------------------------------------------------------------
•
•            Changes slip node array to new node numbers.
•
•-------------------------------------------------------------

              IF(INSP.LE.1) GOTO 80
              DO 70 N=1,NSLIP
70        CALL NEWNOD(SLIP(N,1))
80            CONTINUE
              L=4•NSFR
              NECN=NCN•NE

•-------------------------------------------------------------
•
•            Loops on number of slip plane nodes.
•
•-------------------------------------------------------------

              DO 210 I=1,NSLIP
         NADD=1
         IX=SLIP(I,1)
                SLIP(I,2)=0
         CORDN=CORD(IX,IANG)+DELTA

•-------------------------------------------------------------
•
•            Determines if node is a corner node.
•
•-------------------------------------------------------------

         CALL CORNN(NADD,NSFR,NP,SLIP(I,2),IX)

•-------------------------------------------------------------
•
•            Adds NADD to all numbers in connectivity greater than IX.
•
•-------------------------------------------------------------

         DO 120 J=1,NECN
                   IXX=NOP(J)-IX
                   IF(IXX)120,90,110
90                 EL=(J+L-1)/L
                   X=0.0
                   MMM=L•(EL-1)
```

```
          DO 100 M=1,L,NSFR
            MM=MMM+M
100         X=X+CORD(NOP(MM),IANG)/4.
                IF(X.LT.CORDN)GOTO 120
110         NOP(J)=NOP(J)+NADD
120             CONTINUE

*------------------------------------------------------------
*
*           Updates remaining slip plane node numbers.
*
*------------------------------------------------------------

          DO 130 N=1,NSLIP
130        IF(SLIP(N,1).GT.SLIP(I,1))SLIP(N,1)=SLIP(N,1)+NADD

*------------------------------------------------------------
*
*           Updates coordinate array numbers (add NADD to those < IX).
*
*------------------------------------------------------------

                DO 140 J=NP,IX,-1
                  IJ=J+NADD
           CORD(IJ,1)=CORD(J,1)
140       CORD(IJ,2)=CORD(J,2)

*------------------------------------------------------------
*
*           Assigns coordinate to new node created.
*
*------------------------------------------------------------

                CORD(IJ,IANG)=CORDN
                NP=NP+NADD
          NUMDUM=0
                JJ=NB

*------------------------------------------------------------
*
*           Updates boundary arrays, giving new nodes same boundary conditions
*           as corresponding slip plane nodes.
*
*------------------------------------------------------------
```

```
              DO 200 JJJ=1,JJ
         J=JJJ+NUMDUM
                IXX=NBC(J)-IX
                IF(IXX)200,150,190
150       DO 180 II=1,NADD
           NB=NB+1
                  IJ=J+II
                  DO 170 N=NB,IJ,-1
              NBC(N)=NBC(N-1)
              NFIX(N)=NFIX(N-1)
              DO 160 NN=1,NDF
160           US(N,NN)=US((N-1),NN)
170               ANG(N)=ANG(N-1)
180       NBC(IJ)=NBC(IJ)+1
         NUMDUM=NADD
         GOTO 200
190             NBC(J)=NBC(J)+NADD
200     CONTINUE
210          CONTINUE

*-------------------------------------------------------------------
*
*          Adds row of slip elements to connectivity.
*
*          Sets up array of corner nodes.
*
*-------------------------------------------------------------------

             NCORN=0
             DO 220 I=1,NSLIP
               IF(SLIP(I,2).EQ.0)GOTO 220
               NCORN=NCORN+1
               CORNER(NCORN)=SLIP(I,1)
220          CONTINUE
             IQUIT=1
             K=0
             I=1

*-------------------------------------------------------------------
*
*          Finds corner node in connectivity array.
*
*-------------------------------------------------------------------

230          J=0
```

```
240         J=J+1
            IF(J.GT.NECN)GOTO 290
            IF(NOP(J).NE.CORNER(I))GOTO 240

*------------------------------------------------------
*
*           Determines element number of element containing node.
*
*------------------------------------------------------

            EL=(J+L-1)/L

*------------------------------------------------------
*
*           Finds other corner node contained in same element.
*
*------------------------------------------------------

            MMM=L*(EL-1)
            DO 260 M=1,L,NSFR
              MM=MMM+M
        DO 250 N=1,NCORN
                IF(N.EQ.I)GOTO 250
                IF(NOP(MM).EQ.CORNER(N))K=N
250       CONTINUE
260         CONTINUE

*------------------------------------------------------
*
*           Prints error if no such node exists.
*
*------------------------------------------------------

            IF(K.GT.0)GOTO 280
            WRITE(6,270)
270         FORMAT(' ERROR-- HALTED IN SLIPP, ERROR IN CONNECTIVITY')
            STOP

*------------------------------------------------------
*
*           Makes new element.
*
*------------------------------------------------------

280         IF(K.EQ.KK)GOTO 240
```

```
              CALL MAKEEL(MMM,CORNER(I),CORNER(K),IANG)

*------------------------------------------------------------
*
*             Checks to see how many elements are made.
*
*------------------------------------------------------------

              IQUIT=IQUIT+1
              IF(IQUIT.GE.NCORN)GOTO 300
              KK=I
              IF(I.EQ.1)KKK=K
              I=K

*------------------------------------------------------------
*
*             Branches to next element.
*
*------------------------------------------------------------

              GOTO 230
290           I=1
              KK=KKK
              GOTO 230


*------------------------------------------------------------
*
*             Writes slip plane data.
*
*------------------------------------------------------------

300           CONTINUE
         WRITE(6,310) INSP,DELTA,IANG
310           FORMAT(//,' SLIP PLANE ',I2,
   +       /,'   THICKNESS=',G10.3,
   +       /,'   ANGLE FLAG=',I2,
   +       /,' NODE LIST: ')
         KNUM=(NSLIP-1)/16.+1
              KSTART=1
              DO 330 ITT=1,KNUM
                KEND=KSTART+15
                IF(KEND.GT.NSLIP)KEND=NSLIP
           WRITE(6,320) (SLIP(J,1),J=KSTART,KEND)
320         FORMAT(16I5)
```

```
330     KSTART=KSTART+16
340             CONTINUE

•----------------------------------------------------------------
•
•               Writes mapping of old to new nodes into scrubs.lis.
•
•----------------------------------------------------------------

                J=0
                DO 350 I=1,NPOLD
                  N=I
                  CALL NEWNOD(N)
                  IF(N.EQ.I)GOTO 350
                  J=J+1
                  MAP(J,1)=I
                  MAP(J,2)=N
350             CONTINUE
                WRITE(6,360)
360             FORMAT(/,' •••••••• NODE NUMBERS WERE CHANGED DUE TO SLIP PLANE'
    +   ' ADDITION ••••••••',//,' NODAL MAPPING OF NODES CHANGED:',//
    +           ,4X.4('ORIGINAL   NEW     '),/,5X,4('NODE#    NODE#     '))
                K=J/4
                KK=K•4-J
                K1=0
                IF(KK.LE.-1)K1=1
                K2=K1
                IF(KK.LE.-2)K2=K2+1
                K3=K2
                IF(KK.LE.-3)K3=K3+1
                DO 380 M=1,K
                WRITE(6,370) MAP(M,1),MAP(M,2),MAP(M+K+K1,1),MAP(M+K+K1,2),
    +           MAP(M+2•K+K2,1),MAP(M+2•K+K2,2),MAP(M+3•K+K3,1),MAP(M+3•K+K3,2)
370             FORMAT(6X,4(I3,6X,I3,7X))
380             CONTINUE
                WRITE(6,370) (MAP(N,1),MAP(N,2),N=K+1,K3•(K+1),K+1)
                WRITE(6,390)
390             FORMAT(//,' SLIP PLANE ELEMENTS CREATED:',/,7X,'ELEMENT #'
    +           ,6X,'CONNECTIVITY',/)
                ISTART=L•NEOLD+1
                DO 410 N=NEOLD+1,NE
                IEND=ISTART+L-1
                WRITE(6,400) N,(NOP(I),I=ISTART,IEND)
400             FORMAT(10X,I3,4X,12I5)
410             ISTART=IEND+1
```

```
*-----------------------------------------------------------------
*
*          Numbers new mid-side nodes.
*
*-----------------------------------------------------------------

           N=NEOLD+1
           IF(NSFR.GT.1)CALL NODEXY(N)

           RETURN
           END

*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*          Subroutine SOLVE:
*          Solves stiffness equations, modifies loads.
*
*          Written:
*          Last modified:JUL 14 1984
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------
*
*          Called by:  SCRUBS
*
*-----------------------------------------------------------------
*-----------------------------------------------------------------

           SUBROUTINE SOLVE

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1     NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2     NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO, PWORK,
     3     TPWORK
           COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
     1     US(400,2)
           COMMON/LDS/R1(2000),RL(2000),RT(2000)
           COMMON/SCR/SK(20000) ,R
     1     ,ESTIFM(24,24),NN(12),SKS(1200),NOPS(24),ACT(48)

           NBF=10000
           NSIZ=0
           NNP=0
```

```
            NBUF=NSK-1
            NZ=1
            ND=1
            L=0
            IF (NT) 10,60,10
10          WRITE (6,20)
20          FORMAT(' ELASTIC STIFFNESSES READ IN')
            READ(5,30)NPSP
30          FORMAT(16I5)
            NACT=NDF•NPSP
            NSKS=NACT•(NACT+1)/2
            READ(5,30)(NOPS(I),I=1,NPSP)
            READ(5,40)(ACT(I),I=1,NACT)
40          FORMAT(4E20.13)
            READ(5,40)(SKS(I),I=1,NSKS)
            MASH=0
            DO 50 I=1,NPSP
              NASH=(NOPS(I)-1)•NDF
              DO 50 J=1,NDF
                MASH=MASH+1
                NASH=NASH+1
50          R1(NASH)=R1(NASH)+ACT(MASH)
60          CONTINUE
            DO 70 N=1,NBF
70          SK(N)=0.
            NC=1
            READ(3)
   1        N,(NN(I),I=1,NCN),((ESTIFM(I,J),J=1,LV),I=1,LV)
80          L=L+1
            DO 140 M=1,6
              IF(N.EQ.0) GO TO 150
              DO 90 I=1,NCN
                IF (NN(I) .EQ. L) GO TO 100
90            CONTINUE
              GO TO 150
100           NC=NC+1
              DO 130 I=1,NCN
                IF(NN(I).EQ.0) GO TO 130
                II=(NN(I)-L)•NDF
                DO 120 J=1,NCN
                  IF(NN(J).EQ.0) GO TO 120
                  JJ=(NN(J)-L)•NDF
                  DO 110 IL=1,NDF
                    IA=II+IL
                    IC=(I-1)•NDF+IL
```

```
                DO 110 JM=1,NDF
                  JB=JJ+JM
                  IF(IA.GT.JB) GO TO 110
                  JD=(J-1)*NDF+JM
                  IAB=((JB-1)*JB)/2+IA
                  SK(IAB)=SK(IAB)+ESTIFM(IC,JD)
110           CONTINUE
120         CONTINUE
            NX=(NN(I)-L)*NDF
            IF(NSIZ.LT.NX) NSIZ=NX
130       CONTINUE
          READ(3)
   1    N,(NN(I),I=1,NCN),((ESTIFM(I,J),J=1,LV),I=1,LV)
140     CONTINUE

*-----------------------------------------------------------------
*
*       Sets up load.
*
*-----------------------------------------------------------------

150     CONTINUE
        IF (NT) 200,200,160
160     DO 170 I1=1,NPSP
          IF(L.EQ.NOPS(I1))GO TO 180
170     CONTINUE
        GO TO 200
180     IJ=(I1-1)*NDF
        DO 190 I2=1,NDF
          II=IJ+I2
          DO 190 JJ=II,NACT
            I3=(10*JJ/NDF+9)/10
            I5=JJ*(JJ-1)/2+II
            JB=NDF*(NOPS(I3)-L-I3+1)+JJ
            IA=JB*(JB-1)/2+I2
            SK(IA)=SK(IA)+SKS(I5)
190     CONTINUE
200     CONTINUE
        NSZ=NSIZ+NDF
        JZ=0
        IF(NZ.EQ.NB+1) GO TO 210
        IF(L.NE. NBC(NZ)) GO TO 210
        JZ=NFIX(NZ)
        IZ=10**(NDF-1)
        NZ=NZ+1
```

```
210         DO 390 I=1,NDF
              NSZ=NSZ-1
              IC=I+NDF*(L-1)
              R=R1(IC)
              NBD=0
              IF(JZ.EQ.0) GO TO 230
              IF(JZ.LT.IZ) GO TO 220
              LZ=(NZ-2)*NDF+I
              U(ND) =U(LZ)
              RS=-R
              R=U(ND)
              ND=ND+1
              NBD=1
              JZ=JZ-IZ
220           IZ=IZ/10
230           CONTINUE
              IF (SK(1) .GE. 0.) GO TO 250
              WRITE (6,240) L,I,SK(1)
240           FORMAT(' NODE NO.',I5,I10,' D.F. ','SK(1) =',E15.6)
              SK(1) = 0.
250           CONTINUE
              NBLK=NSZ+4
              IF((NBUF+NBLK).LE.NBF) GO TO 260
              WRITE(4) NBUF,(SK(J),J=NSK,NBUF)
              NBUF=NSK-1
              NNP=0
260           NNP=NNP+1
              NBUF=NBUF+1
              SK(NBUF)=NSZ
              NBUF=NBUF+1
              IA=NBUF+1

*----------------------------------------------------------------
*
*           Invert diagonal term.
*
*----------------------------------------------------------------

              IF(NBD.EQ.1) GO TO 300
              IF(SK(1).NE.0.) GO TO 280
              WRITE(6,270)L,I
270           FORMAT(' PROGRAM HALTED IN SOLVE'/
   1           ' NEGATIVE OR ZERO DIAGONAL STIFFNESS'/
   2           '       NODE NO.',I4,I10,' D.F.')
              STOP
```

```
280       CONTINUE
          XK=1./SK(1)
          SK(NBUF)=XK

*---------------------------------------------------------------
*
*         Modify loads and off-diagonal terms.
*
*---------------------------------------------------------------

290       R1(IC)=XK*R
          GO TO 310
300       SK(NBUF)=-SK(1)
          R1(IC)=SK(1)*R+RS
          XK=1.
          R=-R
310       CONTINUE
          IF(L+I-NP-NDF) 320,400,320
320       NBUF=NBUF+1
          KJ=1
          DO 330 J=1,NSZ
            KJ=KJ+J
            SK(NBUF)=XK*SK(KJ)
            IF(NBD.EQ.1) SK(NBUF)=-SK(NBUF)
330       NBUF=NBUF+1
          SK(NBUF)=NSZ
          NBUF=NBUF+1
          SK(NBUF)=NNP
          DO 360 J=1,NSZ
            IF(SK(IA).EQ.0.) GO TO 360
            IF(NBD.EQ.1) GO TO 350
            JB=(J*(J+1))/2+1
            KJ=1
            DO 340 K=1,J
              KJ=KJ+K
              IJ=JB+K
340         SK(IJ)=SK(IJ)-SK(KJ)*SK(IA)
350         JB=IC+J
            R1(JB)=R1(JB)-SK(IA)*R
360       IA=IA+1
          DO 370 J=1,NSZ
            IK=(J*(J-1))/2
            IJ=IK+J+1
            DO 370 K=1,J
            IA=IK+K
```

```
              JB=IJ+K
370           SK(IA)=SK(JB)
              IJ=(NSZ*(NSZ+1))/2+1
              IK=IJ+NSZ
              DO 380 K=IJ,IK
380           SK(K)=0.
390         CONTINUE
            NSIZ=NSIZ-NDF
            GO TO 80
400         WRITE(4) NBUF,(SK(J),J=NSK,NBUF)
            IF(NSIZ.EQ.0) NSIZ=NDF

            RETURN
            END

*--------------------------------------------------------------
*--------------------------------------------------------------
*
*           Subroutine STIFM:
*           Forms and updates stiffness matrix, and calculates body forces.
*
*           Written:
*           Last modified:JUL 14 1984
*
*--------------------------------------------------------------
*--------------------------------------------------------------
*
*           Called by:  SCRUBS
*
*--------------------------------------------------------------
*--------------------------------------------------------------
*
*           Subroutines called:MODSFR
*           AUXINVARNFLOW
*           ROSB
*
*--------------------------------------------------------------
*--------------------------------------------------------------

            SUBROUTINE STIFM

            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1      NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2      NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
```

```
     3      TPWORK,LNEW
            COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
            COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
     1      US(400,2)
            COMMON/GID/XG(6),CG(6),XMG(6),CMG(6),NGAUS,MGAUS,MCN
            COMMON/MATP/D(4,4),YM,PR,GM,HM,PPR,CLAME,ALPHA,YST
     1      ,TABSTN(10,15),TABSTR(10,15)
            COMMON/NUME/P(12),DEL(2,12),DET,DI(2,12),DIJ(2,2),XYE(3,12),
     1      NOPE(12),DISE(2,12),RP,RPB,XYP(3)
            COMMON/INV/SMEAN,STJ2,STJ3,SIGMA,PHI,APHI,STN,CPHI,YIELD
            COMMON/FLW/ABETA,AA(4),DD(4)
            COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1      ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)
     2      ,COJ(2,2),COJIN(2,2),B(4,24),ESTIFM(24,24),DVD(4,4),DVDB(4)
            COMMON/FG/ FLAG(1000),BULK(1000,2),MINES(10,50),NME(10),GAMMA
            DIMENSION STRS(4)

            NECN=NE*NCN
            READ (8)(NOP(J),J=1,NECN),(IMAT(I),I=1,NE),(((DISN(K,I),
     1      TDIS(K,I)),K=1,2),(CORD(I,J),J=1,NCORD ),I=1,NP),(((TSTS(I,J),
     2      EPSTN(I,J)),I=1,5),J=1,NSTORE)
            REWIND 8
            DO 170 M=1,NE
              LNEW=IONARY(M)
              NASH=NCN*(LNEW-1)
              NG=MCN*(LNEW-1)
              L=IMAT(LNEW)

*-----------------------------------------------------------------
*
*           Uses new stiffness for failed material.
*
*-----------------------------------------------------------------

              IF(NL.NE.1) GO TO 10
              IF(BULK(LNEW,1).NE.0.) L=NMAT
  10          CONTINUE

*-----------------------------------------------------------------
*
*           Note material no. of element along sloping boundary is greater
*           than NMAT by 100.
*
*-----------------------------------------------------------------
```

```
          IF(L.GT.NMAT) L=L-100
          DO 30 K=1,NCN
            MASH=NOP(NASH+K)
            DO 20 I=1,NCORD
              XYE(I,K)=CORD(MASH,I)
20          CONTINUE
            NOPE(K)=MASH
30        CONTINUE
          CALL MOD(L,0)

*------------------------------------------------------------
*
*         Sets initial values as zeroes.
*
*------------------------------------------------------------

          DO 40 I=1,LV
            DO 40 J=1,LV
              ESTIFM(I,J)=0.
              IF(I.LE.4) B(I,J)=0.
40        CONTINUE
          THICK=ORT(L,6)
          IF(THICK.EQ.0.)THICK=1.

*------------------------------------------------------------
*
*         Formulates stiffness.
*
*------------------------------------------------------------

          DO 160 IGAUS=1,NGAUS
            DO 160 JGAUS=1,NGAUS
              G=XG(IGAUS)
              H=XG(JGAUS)
              CALL SFR(G,H)
              CALL AUX(M)
              DV=DET*CG(IGAUS)*CG(JGAUS)*THICK
              IF(NPP.EQ.2)DV=DV*RP*6.283185307179586
              IF(NCORD.EQ.3)DV=DV*XYP(3)

*------------------------------------------------------------
*
*         Calculates B matrix from DI matrix.
*
*------------------------------------------------------------
```

```
                NASH=0
                DO 50 I=1,NCN
                  MASH=NASH+1
                  NASH=MASH+1
                  B(1,MASH)=DI(1,I)
                  B(2,NASH)=DI(2,I)
                  B(3,MASH)=DI(2,I)
                  B(3,NASH)=DI(1,I)
                  IF(NPP.EQ.2) B(4,MASH)=P(I)/RP
50              CONTINUE

*---------------------------------------------------------
*
*           Calculates DVD matrix.
*
*---------------------------------------------------------

                DO 60 I=1,ICS
                  DO 60 J=1,ICS
                    DVD(I,J)=DV*D(I,J)
60            CONTINUE

*---------------------------------------------------------
*
*           Tangential stiffness calculations.
*
*---------------------------------------------------------

                NG=NG+1
                PSTY=TSTS(5,NG)
                PSTRN=EPSTN(5,NG)
                IF(ORT(L,2).LT.0.0) GO TO 110
                IF(PSTRN.EQ.0.) GO TO 110
                MASH=0
70            MASH=MASH+1
                IF(PSTRN.LT.TABSTN(L,MASH)) GO TO 80
                GO TO 70
80            NASH=MASH
                MASH=NASH-1
                ETAB=TABSTN(L,NASH)-TABSTN(L,MASH)
                GASH=(PSTRN-TABSTN(L,MASH))/ETAB
                CYIELD=TABSTR(L,MASH  )*(1.-GASH)+GASH*TABSTR(L,NASH)
                IF(PSTY.LT..9999*CYIELD) GO TO 110
                HM=(TABSTR(L,NASH)-TABSTR(L,MASH))/ETAB
                DO 90 I=1,4
```

```
90             STRS(I)=TSTS(I,NG)
               CALL INVAR   (STRS)
               CALL NFLOW(STRS)
               CON=DV•ABETA
               DO 100 I=1,ICS
                 DO 100 J=1,ICS
100            DVD(I,J)=DV-CON•DD(I)•DD(J)
110            DO 150 J=1,LV
                 DO 130 K=1,ICS
                   GASH=0.
                   DO 120 I=1,ICS
120          GASH=GASH+DVD(K,I)•B(I,J)
130              DVDB(K)=GASH
                 DO 150 I=J,LV
                   GASH=ESTIFM(I,J)
                   DO 140 K=1,ICS
140          GASH=GASH+B(K,I)•DVDB(K)
                   ESTIFM(J,I)=GASH
150         ESTIFM(I,J)=GASH
160        CONTINUE
           IF(IMAT(LNEW).GT.NMAT) CALL ROSB
           WRITE(3)
  1         LNEW,(NOPE(I),I=1,NCN),((ESTIFM(I,J),J=1,LV),I=1,LV)
170       CONTINUE
          A=0.
          N=0
          WRITE(3)N,(N,I=1,NCN),((A,J=1,LV),I=1,LV)
          REWIND 3

          RETURN
          END
```

```
*------------------------------------------------------------
*------------------------------------------------------------
*
*              Subroutine SWITCH:
*              Switches coordinates.
*
*              Written:
*              Last modified:JUL 19 1984
*
*------------------------------------------------------------
*------------------------------------------------------------
*
*              Called by:  FTEST
*
*------------------------------------------------------------
*------------------------------------------------------------

               SUBROUTINE SWITCH(J,K)

               I=K
               K=J
               J=I

               RETURN
               END

*------------------------------------------------------------
*------------------------------------------------------------
*
*              Subroutine TFORM:
*              Makes adjustments for inclined boundary situations.
*
*              Written:
*              Last modified:JUL 14 1984
*
*------------------------------------------------------------
*------------------------------------------------------------
*
*              Called by:  SCRUBS
*
*------------------------------------------------------------
*------------------------------------------------------------

               SUBROUTINE TFORM
```

```
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1    NCORD
          COMMON/BOUN/NBC(400),NFIX(400),U(800),ANG(400),TRC(800),
     1    US(400,2)
          COMMON/LDS/R1(2000),RL(2000),RT(2000)

          DO 10 M=1,NB
            IF(ANG(M).EQ.0.) GOTO 10
            N=NBC(M)
            NASH=2•N
            MASH=NASH-1
            GASH=ANG(M)•.017453292
            CS=COS(GASH)
            TN=TAN(GASH)
            RXC=R1(MASH)•CS
            RYC=R1(NASH)•CS
            R1(MASH)=RXC+RYC•TN
            R1(NASH)=RYC-RXC•TN
 10       CONTINUE

          RETURN
          END
```

```
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*          Subroutine AUX:
*          Creates coordinate jacobian 0 interpolate X; YOCOJ(I,J),XY(1),XY(2)
*          are respectively the coordinate jacobian matrix, the X
*          coordinate and the Y coordinate.
*
*          Written:
*          Last modified:MAR 16 1984
*
*-----------------------------------------------------------
*-----------------------------------------------------------
*
*          Called by:  LDATAREMLDRESIDUE
*          STIFM
*
*-----------------------------------------------------------
*-----------------------------------------------------------

           SUBROUTINE AUX(M)

           IMPLICIT DOUBLE PRECISION (A-H,O-Z)
           COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1     NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2     NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3     TPWORK,LNEW
           COMMON/NUME/P(12),DEL(2,12),DET,DI(2,12),DIJ(2,2),XYE(3,12),
     1     NOPE(12),DISE(2,12),RP,RPB,XYP(3)
           COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1     ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000)
     2     ,COJ(2,2),COJIN(2,2)

           DO 20 I = 1,2
           DO 20 J = 1,2
           GASH = 0.
           DO 10 K = 1,NCN
             GASH = GASH+XYE(I,K)*DEL(J,K)
10         CONTINUE
           COJ(I,J) = GASH
20         CONTINUE
           DO 40 I = 1,NCORD
             GASH = 0.
             DO 30 K = 1,NCN
               GASH = GASH+XYE(I,K)*P(K)
```

```
30            CONTINUE
              XYP(I) = GASH
40          CONTINUE
            RP = XYP(1)

*----------------------------------------------------------------
*
*           Calculates the determinant and inverse of the coordinate Jacobian.
*           DET and COJIN(I,J) are determinant and inverse coordinate
*           jacobian, respectively.
*
*----------------------------------------------------------------

            DET = COJ(1,1)*COJ(2,2)-COJ(1,2)*COJ(2,1)
            IF(DET)50,70,90
50          CONTINUE
C           WRITE(6,60)M,((XYE(JJ,II),JJ = 1,2),II = 1,4),DET,COJ(1,1),
C    + COJ(2,2),COJ(1,2),COJ(2,1)
60          FORMAT('  NEGATIVE DET STOPPED AT ELEMENT NO =',
   +         I3,/,' X1,Y1',2F8.2,/,' X2,Y2',2F8.2,/,' X3,Y3',2F8.2,
   +         /,' X4,Y4',2F8.2,/,' DET',D12.4,/,' COJ(1,1),COJ(2,2)'
   +  ,2D12.4,/,' COJ(1,2),COJ(2,1)',2D12.4)
c           STOP
70          DET = .000001
            WRITE(6,80)M
80          FORMAT(' ZERO DETERMINENT EL NO ',I3)
90          CONTINUE
            COJIN(1,1) = COJ(2,2)/DET
            COJIN(2,2) = COJ(1,1)/DET
            COJIN(1,2) = -COJ(1,2)/DET
            COJIN(2,1) = -COJ(2,1)/DET

*----------------------------------------------------------------
*
*           Calculates cartesian derivatives of shape functions DI(I,K), I=X,Y.
*
*----------------------------------------------------------------

            DO 120 I = 1,2
              DO 110 K = 1,NCN
                GASH = 0.
                DO 100 J = 1,2
                  GASH = GASH+DEL(J,K)*COJIN(J,I)
100             CONTINUE
                DI(I,K) = GASH
```

```
110          CONTINUE
120         CONTINUE

            RETURN
            END

*--------------------------------------------------------------
*--------------------------------------------------------------
*
*           Subroutine ZONE:
*           This subroutine reads meshing cards as done in the CHILES2
*           program.  It is provided here only as an interim capability.
*           This subroutine is called by the main program.
*
*           Written:
*           Last modified:JUL 19 1984
*
*--------------------------------------------------------------
*--------------------------------------------------------------
*
*           Called by:  SCRUBS
*
*--------------------------------------------------------------
*--------------------------------------------------------------

            SUBROUTINE ZONE (NRS)

            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON/SCR/ R(2000),Z(2000),IX(5,2000),CODE(2000)
            COMMON/QMESH/NEL

*--------------------------------------------------------------
*
*           Reads element properties from SCRUBS.DAT.
*
*--------------------------------------------------------------

            READ(5,10) NEL,NODES,NUMPC
10          FORMAT  (6I5)
            IBOMB=0
            IF(NEL.NE.0) GO TO 30
            WRITE(6,20)
20          FORMAT(' *****MESH GENERATOR QMESH USED FOR GRID INPUT*****')
            RETURN
30          CONTINUE
```

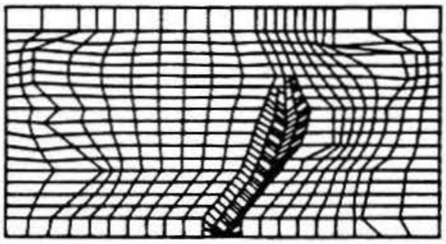
```
            WRITE(6,40) NEL,NODES,NUMPC
40          FORMAT (4X,'NUMBER OF ELEMENTS (NEL)----------------'
  1         '-------',I5/4X,'NUMBER OF NODAL POINTS (NODES)----'
  2         '----------------',I5/4X,'NUMBER OF PRESSURE CARDS '
  3         '(NUMPC)-----------------',I5)

•------------------------------------------------------------
•
•           NEL = 0 for mesh generator, NEL = -1 for restart.
•
•------------------------------------------------------------

            IF (NEL.LT.0) NRS=1
            IF(NRS.EQ.1) RETURN
            NFORCE=NUMPC
            NUMSC=NUMPC
            N=0
50          READ(5,10) M,(IX(I,M),I=1,5)
60          N=N+1
            IF (M-N) 100,80,70
70          IX(1,N)=IX(1,N-1)+1
            IX(2,N)=IX(2,N-1)+1
            IX(3,N)=IX(3,N-1)+1
            IX(4,N)=IX(4,N-1)+1
            IX(5,N)=IX(5,N-1)
80          IF (M-N) 100,90,60
90          IF (NEL-N) 120,120,50
100         WRITE(6,110) M
110         FORMAT (' •••FATAL ERROR   ELEMENT CARD, M=',I5)
            IBOMB=1
120         CONTINUE

•------------------------------------------------------------
•
•           Read nodal point data.
•
•------------------------------------------------------------

            N=0
130         READ (5,140) M,CODE(M),R(M),Z(M)
140         FORMAT (I5,F5.0,4E10.0)
            NNL=N+1
            IF (NNL.EQ.1) GO TO 150
            ZX=FLOAT(M-N)
            DR=(R(M)-R(N))/ZX
```

```
          DZ=(Z(M)-Z(N))/ZX
150       N=N+1
          IF (M-N) 180,170,160
160       IF (CODE(NNL-1).EQ.CODE(M)) CODE(N)=CODE(M)
          R(N)=R(N-1)+DR
          Z(N)=Z(N-1)+DZ
          GO TO 150
170       IF (NODES-M) 180,200,130
180       WRITE (6,190) M
190       FORMAT (' ***FATAL ERROR   NODAL POINT CARD, M=',I5)
          IBOMB=1
200       CONTINUE
          WRITE(9) NEL,NODES
          WRITE(9) (R(I),Z(I),I=1,NODES)
          WRITE(9)((IX(I,M),I=1,5),M=1,NEL)
          REWIND 9

*--------------------------------------------------------------
*
*         Output mesh.
*
*--------------------------------------------------------------

          MPRINT=0
          DO 230 I=1,NEL
            IF (MPRINT.NE.0) GO TO 210
            MPRINT=50
210         MPRINT=MPRINT-1
            WRITE(6,220) I,(IX(J,I),J=1,5)
220         FORMAT(1I13,4I6,I7)
230       CONTINUE
          MPRINT=0
          DO 260 I=1,NODES
            IF(MPRINT.NE.0) GO TO 240
            MPRINT=50
240         MPRINT=MPRINT-1
            WRITE(6,250) I,R(I),Z(I)
250         FORMAT(6X,I6,2F12.3)
260       CONTINUE

          RETURN
          END
```

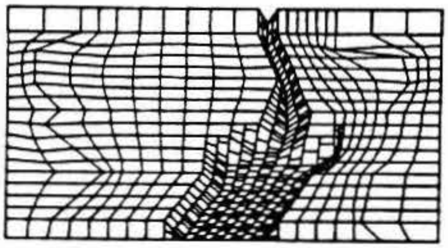
APPENDIX D

ADDITIONAL SUBSIDENCE RESULTS

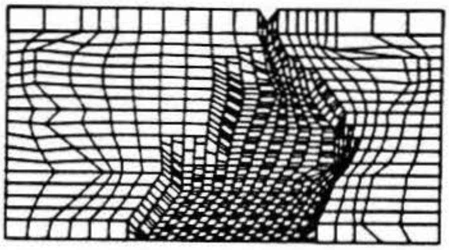
The following appendix contains additional results from running SCRUBS on the subsidence prediction problem found in chapter four. These three runs have been left out of the main report since they were results obtained during the course of trying to find good test parameters and were not considered to be as important to the conclusions as the tests included in the body of the report. However, it was thought that some readers may obtain valuable insights and information from them, so they have been included here. The test parameters varied in these tests were the failure stress. The failure stress used is listed in the figure heading.



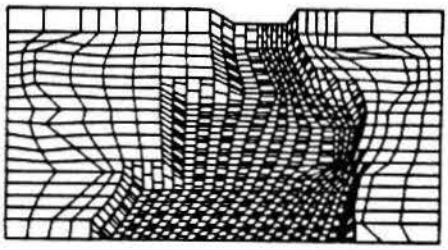
a. One support removed.



b. Three supports removed.

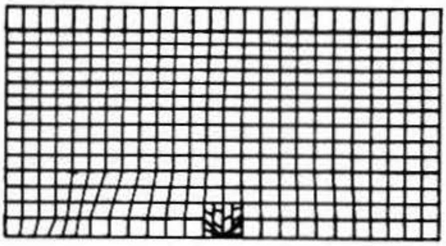


c. Five supports removed.

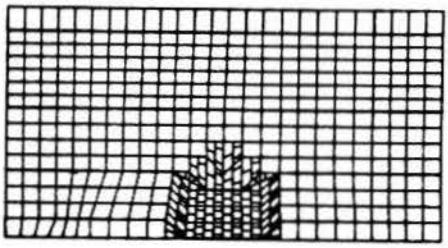


d. Seven supports removed.

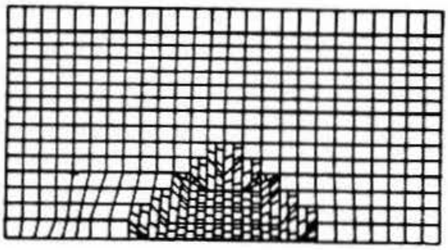
Figure 17 a-d. Jointed media skewed mesh. Failure stress was 0.1 10^5dynes/cm^2.



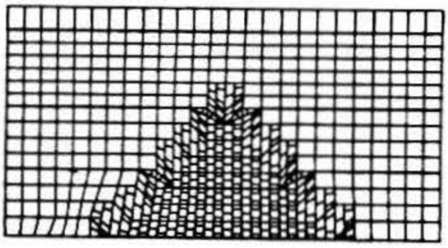
a. One support removed.



b. Three supports removed.

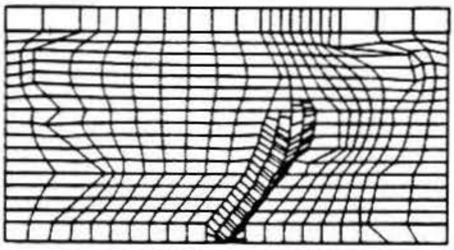


c. Five supports removed.

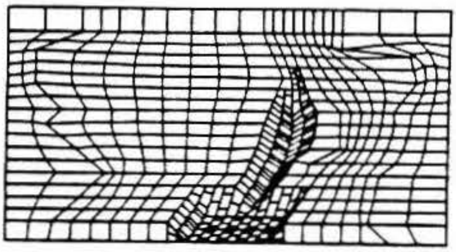


d. Seven supports removed.

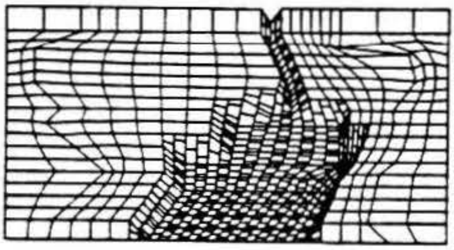
Figure 18 a-d. Discrete slip plane straight mesh. Failure stress was 0.1 10^9dynes/cm^2.



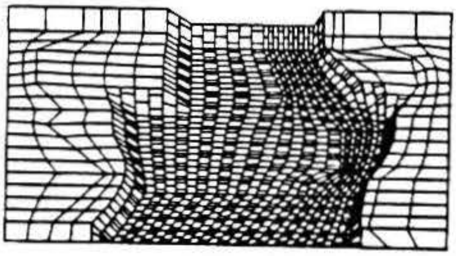
a. One support removed.



b. Three supports removed.



c. Five supports removed.



d. Seven supports removed.

Figure 19 a-d. Discrete slip plane skewed mesh. Failure stress was 0.1 10^5dynes/cm^2.

APPENDIX E

SELECTED FILES

The following appendix contains typical files used in subsidence prediction with SCRUBS.BYU. First is the file FTEST, which is compiled with scrubs but is user-supplied (the one in this appendix is the one used to obtain our results), next are the two QMESH card input files used to create the meshes, and last is a typical SCRUBS.BYU data file used to obtain the combined skewed mesh model. To read the QMESH card files, the reader should consult reference 8. To read the SCRUBS.BYU data file, the user should consult appendix B, the SCRUBS.BYU user's manual. The SCRUBS.BYU subroutine FTEST is written in standard fortran and contains comments to show what is done.

CARD DESCRIPTIONS

The subroutine FTEST:

```
*--------------------------------------------------------------
*--------------------------------------------------------------
*
*           Subroutine FTEST:
*           This is a user supplied subroutine to determine the failure of
*           element M.
*
*           Written:
*           Last modified:JUL 19 1984
*
*--------------------------------------------------------------
*--------------------------------------------------------------
*
*           Called by:  RESIDUE
*
*--------------------------------------------------------------
*--------------------------------------------------------------
*
*           Subroutines called:REMLDSWITCH
*
*--------------------------------------------------------------
*--------------------------------------------------------------

            SUBROUTINE FTEST(M,SAVG,MATNO,IFLFG)

            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON/CONTR/TITLE(20),NP,NE,NB,NDF,NCN,NSFR,NSZF,LV,NPP,ICS,
     1 NCORD,NSK,NSIZ,NL,NBUF,NT,ND,NNP,ILD,NLD,NMAT,IT,NIT,INC,NINC,
     2      NSTORE,MPR,LDTYPE,FACTOR,OMEGA,CONFAC,NYIELD,ISTS,NALGO,PWORK,
     3      TPWORK,KNE
            COMMON/FG/ FLAG(1000),BULK(1000,2),MINE(10,50),NME(10)
            COMMON/GEP/FAC(40),NOUT(40),ORT(10,10),NOPR(30),IONARY(1000)
            DIMENSION SAVG(5),NM(4),MB(1000)
            COMMON/SCR/DIS(2,1000),TDIS(2,1000),DISN(2,1000),CORD(1000,2)
     1 ,NOP(4000),IMAT(1000),TSTS(5,4000),EPSTN(5,4000),DUMMY(1000)
            COMMON/GID/XG(6),CG(6),XMG(6),CMG(6),NGAUS,MGAUS,MCN

            IFLFG=0

*--------------------------------------------------------------
*
```

```
*           On first iteration, cycle 1, determines if element below
*           element M has failed and creates MB(M) where MB(M)
*           is the element number of the element below M.
*
*------------------------------------------------------------

            IF (INC.GT.1) GO TO 70
            IF (IT.GT.1) GO TO 70

*------------------------------------------------------------
*
*           Finds four nodes associated with element M.
*
*------------------------------------------------------------

            N1=NCN*(M-1)
            DO 10 I=1,NCN,NSFR
              IX=I+N1
10          NM(I)=NOP(IX)

*------------------------------------------------------------
*
*           Finds two lowest nodes of element M.
*
*------------------------------------------------------------

            DO 30 I=1,3
              DO 20 I1=1,3
                I2=I1+1
20            IF(CORD(NM(I1),2).GT.CORD(NM(I2),2))CALL SWITCH(NM(I1),
   &        NM(I2))
30          CONTINUE

*------------------------------------------------------------
*
*           Finds element below element M (this is the element which has the
*           two bottom nodes of element M as its two top nodes).
*
*------------------------------------------------------------

            MB(M)=-1
            DO 60 I=1,NE
              IF(M.EQ.I) GO TO 60
              KKK=NCN*(I-1)
              IM=0
```

```
            DO 50 IG=1,NCN,NSFR
              KK=KKK+IG
              IF(NOP(KK).EQ.NM(1).OR.NOP(KK).EQ.NM(2))GOTO 40
              GOTO 50
40            IM=IM+1
              IF(IM.EQ.2)MB(M)=1
50          CONTINUE
60        CONTINUE
70        CONTINUE

*---------------------------------------------------------------
*
*         For first iteration, checks for mined out or specified
*         element failure.
*
*---------------------------------------------------------------

          IF(IT.NE.1) GO TO 90
          NMEL=NME(INC)
          IF(NMEL.EQ.0) GO TO 90
          DO 80 I=1,NMEL
            IF(MINE(INC,I).EQ.M) GO TO 170
80        CONTINUE

*---------------------------------------------------------------
*
*         Sets the failure criteria value to sigfail.
*
*---------------------------------------------------------------

90        SIGFAIL=ORT(MATNO,7)

          IF(BULK(M,1).NE.0.0) RETURN

*---------------------------------------------------------------
*
*         If bottom element MB(M) has failed, and element (M) is a slide line,
*         then fails the slide line directly.
*
*---------------------------------------------------------------

          MBTEST=MB(M)
          IF (NNP.EQ.0) GO TO 110
          IF (MBTEST.LT.0) GO TO 110
          IF(BULK(MBTEST,1).EQ.0) GO TO 110
```

```
           IF (MATNO.EQ.NMAT-1) GO TO 170

110        CONTINUE

           MSTRT=(M-1)*MCN
           DO 160 I=1,MCN
             INDEX=MSTRT+I

*------------------------------------------------------------
*
*          Compares the failure criteria with the current state of
*          stress at this point in the logic for a failure test.
*
*          In this specific application, the failure measure is
*          taken as either the vertical stress or the horizontal stress.
*
*          Also, no element is allowed to fail if the element below
*          it has not failed, nor if it has no elements below it.
*          if the element is a slide line, fail with no stress check.
*
*          TSTS(1,INDEX)=HORIZONTAL STRESS
*          TSTS(2,INDEX)=VERTICAL STRESS
*
*          BULK(M,1)=0.0ELEMENT HAS NOT FAILED
*          BULK(M,1)=1.0ELEMENT HAS FAILED
*
*------------------------------------------------------------

             STR1=TSTS(1,INDEX)
             STR2=TSTS(2,INDEX)
             IF(MBTEST.LT.0) RETURN
             IF((STR1.GT.SIGFAIL.AND.BULK(MBTEST,1).NE.0.0).OR.
   &       (STR2.GT.SIGFAIL.AND.BULK(MBTEST,1).NE.0.0)) GOTO 170

160        CONTINUE
           RETURN

*------------------------------------------------------------
*
*          States that element has failed, and increases value of flag
*          by 1 on each node.
*
*------------------------------------------------------------

170        CONTINUE
```

```
      IFLFG=1
      WRITE(6,180) M
180   FORMAT(' * * * * * ELEMENT ',I5,' HAS BECOME RUBBLE '
   &  '* * * * *')
      DO 190 I=1,4
        KK = 4*(M-1)+I
        NODE=NOP(KK)
190   FLAG(NODE)=FLAG(NODE)+1.0

*------------------------------------------------------------
*
*     Sets BULK(M,1) = 1.0 for failed element.
*
*------------------------------------------------------------

      BULK(M,1) = 1.0

*------------------------------------------------------------
*
*     Removes gravity loading from failed element.
*
*------------------------------------------------------------

      CALL REMLD(M)

*------------------------------------------------------------
*
*     Computes element area just prior to failure.
*
*------------------------------------------------------------

      K1=(M-1)*4+1
      K2=K1+1
      K3=K2+1
      K4=K3+1
      J1=NOP(K1)
      J2=NOP(K2)
      J3=NOP(K3)
      J4=NOP(K4)
      COE1=CORD(J2,1)-CORD(J4,1)
      COE2=CORD(J3,2)-CORD(J1,2)
      COE3=CORD(J3,1)-CORD(J1,1)
      COE4=CORD(J4,2)-CORD(J2,2)
      AREA=.5*(COE1*COE2+COE3*COE4)
      AOLD=BULK(M,2)
```

```
          BULK(M,2)=AREA
200       WRITE(6,210) AOLD,AREA
210       FORMAT(' AOLD  ANEW ',2E15.5)

          RETURN
          END
```

This is the QMESH input card file used to create the skewed mesh (again, for those readers not familar with QMESH, see reference 8) :

```
COMMEN   SKEWED MESH
POINT     1/   1.6500  62.4500        0
POINT    38/  21.2000  62.4500        0
POINT    45/  30.8000  62.4500        0
POINT    39/  68.9000  62.4500        0
POINT    40/  90.0000  62.4500        0
POINT     2/ 118.3500  62.4500        0
POINT     3/   1.6500  56.1000        0
POINT     4/  21.2000  56.1000        0
POINT    46/  30.8000  56.1000        0
POINT     5/  68.9000  56.1000        0
POINT     6/  90.0000  56.1000        0
POINT     7/ 118.3500  56.1000        0
POINT     8/  90.0000  52.1000        0
POINT     9/   9.0000  51.1000        0
POINT    10/ 103.5000  50.1000        0
POINT    11/   9.0000  42.1000        0
POINT    12/  74.5000  44.1000        0
POINT    13/ [illegible]  43.1000        0
POINT    14/   1.6500  35.1000        0
POINT    15/  16.0000  35.1000        0
POINT    16/  63.0000  35.1000        0
POINT    17/  71.0000  35.1000        0
POINT    18/  90.0000  35.1000        0
POINT    19/ 104.0000  35.1000        0
POINT    20/ 118.3500  35.1000        0
POINT    21/  12.0000  31.1000        0
POINT    22/  90.0000  25.1000        0
POINT    23/  82.0000  21.1000        0
POINT    24/   1.6500  18.1000        0
POINT    25/  28.0000  18.1000        0
POINT    47/  33.0000  18.1000        0
POINT    26/  60.0000  18.1000        0
POINT    27/  65.0000  18.1000        0
```

```
POINT      28/  81.0000   18.1000             0
POINT      29/ 100.5000   18.1000             0
POINT      30/ 118.3500   18.1000             0
POINT      31/   1.6500    5.1000             0
POINT      32/  14.1000    5.1000             0
POINT      41/  24.3000    5.1000             0
POINT      33/  54.9000    5.1000             0
POINT      42/  95.7000    5.1000             0
POINT      34/  97.5000    5.1000             0
POINT      35/ 118.3500    5.1000             0
POINT      36/   1.6500    0.0000             0
POINT      43/  24.3000    0.0000             0
POINT      44/  95.7000    0.0000             0
POINT      37/ 118.3500    0.0000             0
LINE  STR  138/  1   38   0   2   1.0000    0
LINE  STR   13/  1    3   0   1   1.0000  100
LINE  STR  345/ 38   45   0   2   1.0000    0
LINE  STR  439/ 45   39   0   6   1.0000    0
LINE  STR  340/ 39   40   0   8   1.0000    0
LINE  STR  402/ 40    2   0   3   1.0000    0
LINE  STR   27/  2    7   0   1   1.0000  100
LINE  STR   34/  3    4   0   2   1.0000   40
LINE  STR  314/  3   14   0   7   1.0000  100
LINE  STR  446/  4   46   0   2   1.0000   40
LINE  STR   49/  4    9   0   2   1.0000   50
LINE  STR  465/ 46    5   0   6   1.0000   40
LINE  STR   56/  5    6   0   8   1.0000   40
LINE  STR  512/  5   12   0   4   1.0000   60
LINE  STR   67/  6    7   0   3   1.0000   40
LINE  STR   68/  6    8   0   1   1.0000   90
LINE  STR  720/  7   20   0   7   1.0000  100
LINE  STR  810/  8   10   0   1   1.0000   90
LINE  STR  911/  9   11   0   3   1.0000   50
LINE  STR  113/ 10   13   0   2   1.0000   90
LINE  STR  115/ 11   15   0   2   1.0000   50
LINE  STR  117/ 12   17   0   3   1.0000   60
LINE  STR  119/ 13   19   0   3   1.0000   90
LINE  STR  415/ 14   15   0   2   1.0000   30
LINE  STR  124/ 14   24   0   5   1.0000  100
LINE  STR  116/ 15   16   0   7   1.0000   30
LINE  STR  121/ 15   21   0   1   1.0000   50
LINE  STR  617/ 16   17   0   1   1.0000   30
LINE  STR  126/ 16   26   0   5   1.0000   70
LINE  STR  118/ 17   18   0   3   1.0000   30
LINE  STR  819/ 18   19   0   5   1.0000   30
```

```
LINE STR  122/ 18  22  0   3   1.0000 110
LINE STR  120/ 19  20  0   3   1.0000  30
LINE STR  230/ 20  30  0   5   1.0000 100
LINE STR  225/ 21  25  0   4   1.0000  50
LINE STR  223/ 22  23  0   1   1.0000 110
LINE STR  228/ 23  28  0   1   1.0000 110
LINE STR  425/ 24  25  0   2   1.0000  20
LINE STR  231/ 24  31  0   4   1.0000 100
LINE STR  247/ 25  47  0   2   1.0000  20
LINE STR  232/ 25  32  0   4   1.0000  50
LINE STR  426/ 47  26  0   5   1.0000  20
LINE STR  227/ 26  27  0   1   1.0000  20
LINE STR  728/ 27  28  0   3   1.0000  20
LINE STR  233/ 27  33  0   4   1.0000  80
LINE STR  229/ 28  29  0   6   1.0000  20
LINE STR  930/ 29  30  0   2   1.0000  20
LINE STR  234/ 29  34  0   4   1.0000 120
LINE STR  335/ 30  35  0   4   1.0000 100
LINE STR  332/ 31  32  0   2   1.0000  10
LINE STR  336/ 31  36  0   1   1.0000 100
LINE STR  341/ 32  41  0   2   1.0000  10
LINE STR  433/ 41  33  0   6   1.0000  10
LINE STR  342/ 33  42  0   8   1.0000  10
LINE STR  434/ 42  34  0   1   1.0000  10
LINE STR  435/ 34  35  0   2   1.0000  10
LINE STR  337/ 35  37  0   1   1.0000 100
LINE STR  343/ 36  43  0   4   1.0000 200
LINE STR  444/ 43  44  0  14    1.0000 200
LINE STR  437/ 44  37  0   3   1.0000 200
SIDE        12/ 138  345  439  340  402
SIDE        37/  34  446  465   56   67
SIDE        45/ 446  465
SIDE       415/  49  911  115
SIDE       517/ 512  117
SIDE       619/  68  810  113  119
SIDE       117/ 116  617
SIDE       118/ 617  118
SIDE       119/ 118  819
SIDE       120/ 819  120
SIDE       125/ 121  225
SIDE       128/ 122  223  228
SIDE       226/ 247  426
SIDE       227/ 247  426  227
SIDE       228/ 227  728
SIDE       229/ 728  229
```

```
SIDE      230/ 229  930
SIDE      333/ 341  433
SIDE      334/ 342  434
SIDE      335/ 332  341  433  342  434  435
SIDE      337/ 343  444  437
REGION  1   1/  12  -27   37  -13
REGION  2   2/  45  517  117  415
REGION  2   3/-116 -126  226  125
REGION  2   4/ 227 -233  333 -232
REGION  2   5/ -56  619  119  517
REGION  2   6/ 118  128  228 -126
REGION  2   7/ 229 -234  334 -233
REGION  3   8/ -34  415 -415 -314
REGION  3   9/-415  125 -425 -124
REGION  3  10/-425 -232 -332 -231
REGION  3  11/ -67 -720 -120  619
REGION  3  12/ 120 -230  230  128
REGION  3  13/-930 -335 -435 -234
REGION  4  14/ 335 -337  337 -336
SCHEME      1M
SCHEME      2MS2S
SCHEME      3MS2S5S
SCHEME      4MS
SCHEME      5MS
SCHEME      6MS2S5S
SCHEME      7MS
SCHEME      8MS
SCHEME      9MS2S
SCHEME     10MS
SCHEME     11MS
SCHEME     12MS
SCHEME     13MS
SCHEME     14MS
END       4
P-L-      36  343  43  444  44  437  37
END
```

This is the QMESH input card file used to create the straight mesh (again, for those readers not familar with QMESH, see reference 8) :

```
COMMEN    STRAIGHT MESH
POINT       1/  1.6500  62.4500        0
POINT       8/ 15.0000  62.4500        0
POINT      20/ 70.0000  62.4500        0
```

```
POINT       9/ 100.0000   62.4500          0
POINT       2/ 118.3500   62.4500          0
POINT       3/   1.6500   56.1000          0
POINT       4/  15.0000   56.1000          0
POINT       5/  70.0000   56.1000          0
POINT       6/ 100.0000   56.1000          0
POINT       7/ 118.3500   56.1000          0
POINT      10/   1.6500   35.1000          0
POINT      11/  15.0000   35.1000          0
POINT      12/  60.0000   35.1000          0
POINT      13/  70.0000   35.1000          0
POINT      14/  85.0000   35.1000          0
POINT      15/ 100.0000   35.1000          0
POINT      16/ 118.3500   35.1000          0
POINT      17/   1.6500   18.1000          0
POINT      18/  15.0000   18.1000          0
POINT      19/  60.0000   18.1000          0
POINT      21/  85.0000   18.1000          0
POINT      22/ 100.0000   18.1000          0
POINT      23/ 118.3500   18.1000          0
POINT      24/   1.6500    5.1000          0
POINT      25/  15.0000    5.1000          0
POINT      26/  24.3000    5.1000          0
POINT      27/  60.0000    5.1000          0
POINT      28/  95.7000    5.1000          0
POINT      29/ 100.0000    5.1000          0
POINT      30/ 118.3500    5.1000          0
POINT      31/   1.6500    0.0000          0
POINT      32/  24.3000    0.0000          0
POINT      33/  95.7000    0.0000          0
POINT      34/ 118.3500    0.0000          0
LINE STR   18/   1   8   0   3   1.0000    0
LINE STR  820/   8  20   0  12   1.0000    0
LINE STR  209/  20   9   0   6   1.0000    0
LINE STR   92/   9   2   0   4   1.0000    0
LINE STR   13/   1   3   0   1   1.0000  100
LINE STR   27/   2   7   0   1   1.0000  100
LINE STR   34/   3   4   0   3   1.0000   40
LINE STR  310/   3  10   0   6   1.0000  100
LINE STR   45/   4   5   0  12   1.0000   40
LINE STR  411/   4  11   0   6   1.0000   50
LINE STR   56/   5   6   0   6   1.0000   40
LINE STR  513/   5  13   0   6   1.0000   60
LINE STR   67/   6   7   0   4   1.0000   40
LINE STR  615/   6  15   0   6   1.0000   90
```

```
LINE STR  716/  7  16  0   6  1.0000 100
LINE STR  111/ 10  11  0   3  1.0000  30
LINE STR  117/ 10  17  0   4  1.0000 100
LINE STR  112/ 11  12  0  10  1.0000  30
LINE STR  113/ 12  13  0   2  1.0000  30
LINE STR  119/ 12  19  0   4  1.0000  80
LINE STR  114/ 13  14  0   3  1.0000  30
LINE STR  115/ 14  15  0   3  1.0000  30
LINE STR  121/ 14  21  0   4  1.0000 130
LINE STR  116/ 15  16  0   4  1.0000  30
LINE STR  123/ 16  23  0   4  1.0000 100
LINE STR  118/ 11  18  0   4  1.0000  50
LINE STR  718/ 17  18  0   3  1.0000  20
LINE STR  124/ 17  24  0   3  1.0000 100
LINE STR  819/ 18  19  0  10  1.0000  20
LINE STR  125/ 18  25  0   3  1.0000  50
LINE STR  921/ 19  21  0   5  1.0000  20
LINE STR  127/ 19  27  0   3  1.0000  80
LINE STR  222/ 21  22  0   3  1.0000  20
LINE STR  223/ 22  23  0   4  1.0000  20
LINE STR  229/ 22  29  0   3  1.0000 140
LINE STR  230/ 23  30  0   3  1.0000 100
LINE STR  225/ 24  25  0   3  1.0000  10
LINE STR  226/ 25  26  0   3  1.0000  10
LINE STR  227/ 26  27  0   7  1.0000  10
LINE STR  231/ 24  31  0   1  1.0000 100
LINE STR  228/ 27  28  0   7  1.0000  10
LINE STR  829/ 28  29  0   1  1.0000  10
LINE STR  930/ 29  30  0   4  1.0000  10
LINE STR  334/ 30  34  0   1  1.0000 100
LINE STR  332/ 31  32  0   6  1.0000 200
LINE STR  333/ 32  33  0  14  1.0000 200
LINE STR  433/ 33  34  0   5  1.0000 200
SIDE        12/  18 820 209  92
SIDE        37/  34  45  56  67
SIDE       113/ 112 113
SIDE       114/ 113 114
SIDE       115/ 114 115
SIDE       116/ 115 116
SIDE       122/ 921 222
SIDE       223/ 222 223
SIDE       230/ 225 226 227 228 829 930
SIDE       227/ 226 227
SIDE       229/ 228 829
SIDE       334/ 332 333 433
```

```
REGION  1   1/  12 -27   37 -13
REGION  2   2/ -45 -513  113 -411
REGION  2   3/-112 -119 -819 -118
REGION  2   4/-819 -127  227 -125
REGION  2   5/ -56 -615  115 -513
REGION  2   6/ 114 -121 -921 -119
REGION  2   7/ 122 -229  229 -127
REGION  3   8/ -34 -411 -111 -310
REGION  3   9/-111 -118 -718 -117
REGION  3  10/-718 -125 -225 -124
REGION  3  11/ -67 -716 -116 -615
REGION  3  12/ 116 -123  223 -121
REGION  3  13/-223 -230 -930 -229
REGION  4  14/ 230 -334  334 -231
SCHEME       1M
SCHEME       2M
SCHEME       3M
SCHEME       4M
SCHEME       5M
SCHEME       6M
SCHEME       7M
SCHEME       8M
SCHEME       9M
SCHEME      10M
SCHEME      11M
SCHEME      12M
SCHEME      13M
SCHEME      14M
END       4
P-L-     31  332   32  333   33  433   34
END
```

This is the SCRUBS.BYU data file used to create the failure and subsidence of Figure 15 a-d:

```
  0
  1
JOINTED MEDIA WITH HORIZONTAL SLIP - SKEWED MESH
  0   0  58   1   2   6   1   2   3   0   1   0   1   0   2   2
     1-.6000E+110.6000E+060.1000E+21 1.000    0.0000E+000.0000E+00
     2-.5500E+12-.2700E+120.1000E+21 1.000    0.0000E+000.0000E+00
 10.000    0.10000E+09
     3-.5500E+12-.2700E+120.1000E+21 1.000    0.0000E+000.0000E+00
 10.000    0.10000E+09
```

```
      40.2000E+130.3000    0.1000E+21 1.000    0.0000E+000.0000E+00
      5-.3450E+120.3450E+120.1000E+32 1.000    0.0000E+000.0000E+00
      6-.6000E+080.6000E+060.1000E+21 1.000    0.0000E+000.0000E+00
      10.1000E+070.9800
      20.1000E+070.9800
      30.1000E+070.9800
      40.0000E+00 60.00
      50.0000E+00 1.000
      60.0000E+000.0000E+00
0.0000E+00
   5
   0   4   8   8   8

 224 228  368  369
 140 144  232  236  366  367  370  371
 132 136  240  244  364  365  372  373
 124 128  248  252  362  363  374  375
   4   0  58    0
   1   11 0.000E+00 0.000E+00 0.000E+00
   2    1 0.000E+00 0.000E+00 0.000E+00
   3    1 0.000E+00 0.000E+00 0.000E+00
   4    1 0.000E+00 0.000E+00 0.000E+00
   5    1 0.000E+00 0.000E+00 0.000E+00
   6    1 0.000E+00 0.000E+00 0.000E+00
   7    1 0.000E+00 0.000E+00 0.000E+00
   8    1 0.000E+00 0.000E+00 0.000E+00
   9    1 0.000E+00 0.000E+00 0.000E+00
  10    1 0.000E+00 0.000E+00 0.000E+00
  11    1 0.000E+00 0.000E+00 0.000E+00
  12    1 0.000E+00 0.000E+00 0.000E+00
  13    1 0.000E+00 0.000E+00 0.000E+00
  14    1 0.000E+00 0.000E+00 0.000E+00
  15    1 0.000E+00 0.000E+00 0.000E+00
  16    1 0.000E+00 0.000E+00 0.000E+00
  17    1 0.000E+00 0.000E+00 0.000E+00
  18    1 0.000E+00 0.000E+00 0.000E+00
  19    1 0.000E+00 0.000E+00 0.000E+00
  20    1 0.000E+00 0.000E+00 0.000E+00
  21    1 0.000E+00 0.000E+00 0.000E+00
  22   11 0.000E+00 0.000E+00 0.000E+00
  23   10 0.000E+00 0.000E+00 0.000E+00
  44   10 0.000E+00 0.000E+00 0.000E+00
  45   10 0.000E+00 0.000E+00 0.000E+00
  66   10 0.000E+00 0.000E+00 0.000E+00
  67   10 0.000E+00 0.000E+00 0.000E+00
```

```
 88  10 0.000E+00 0.000E+00 0.000E+00
 89  10 0.000E+00 0.000E+00 0.000E+00
110  10 0.000E+00 0.000E+00 0.000E+00
111  10 0.000E+00 0.000E+00 0.000E+00
132  10 0.000E+00 0.000E+00 0.000E+00
133  10 0.000E+00 0.000E+00 0.000E+00
154  10 0.000E+00 0.000E+00 0.000E+00
155  10 0.000E+00 0.000E+00 0.000E+00
176  10 0.000E+00 0.000E+00 0.000E+00
177  10 0.000E+00 0.000E+00 0.000E+00
198  10 0.000E+00 0.000E+00 0.000E+00
199  10 0.000E+00 0.000E+00 0.000E+00
220  10 0.000E+00 0.000E+00 0.000E+00
221  10 0.000E+00 0.000E+00 0.000E+00
242  10 0.000E+00 0.000E+00 0.000E+00
243  10 0.000E+00 0.000E+00 0.000E+00
264  10 0.000E+00 0.000E+00 0.000E+00
265  10 0.000E+00 0.000E+00 0.000E+00
286  10 0.000E+00 0.000E+00 0.000E+00
287  10 0.000E+00 0.000E+00 0.000E+00
308  10 0.000E+00 0.000E+00 0.000E+00
309  10 0.000E+00 0.000E+00 0.000E+00
330  10 0.000E+00 0.000E+00 0.000E+00
331  10 0.000E+00 0.000E+00 0.000E+00
352  10 0.000E+00 0.000E+00 0.000E+00
353  10 0.000E+00 0.000E+00 0.000E+00
374  10 0.000E+00 0.000E+00 0.000E+00
375  10 0.000E+00 0.000E+00 0.000E+00
396  10 0.000E+00 0.000E+00 0.000E+00
397  10 0.000E+00 0.000E+00 0.000E+00
418  10 0.000E+00 0.000E+00 0.000E+00
 22   2 0.100
 27  28  29  30  31  32  33  25  26  41  42  34  35  36  37  38
 39  40  23  24  43  44
 22   2 0.100
115 116 117 118 119 120 113 114 121 122 123 124 125 126 127 128
129 130 111 112 131 132
 22   2 0.100
230 231 223 224 225 226 227 228 229 234 235 236 237 238 239 232
233 221 222 240 241 242
 22   2 0.100
393 394 395 396 385 386 387 388 389 390 391 392 379 380 381 382
383 384 377 378 375 376
  0   1
GRAVITY LOAD
```

```
   0   0   1   0   0   0
0.0000E+00.14715E+06.00000E+00
1.2500    2.6400    1.6500    7.8000    3.3350    2.6400
   5
1.0000.0000.0000.0000.000
   0   0   0   0   0
      352.5000
   0
   0   1
```

APPLICATION OF COMBINED JOINTED MEDIA AND DISCRETE SLIP PLANE CHARACTERISTICS TO SUBSIDENCE PREDICTIONS

David W. Basinger

Department of Civil Engineering

M.S. Degree, December 1984

ABSTRACT

This thesis presents an application of a numerical formulation incorporating the effects of joints, cracks, and fractures to a soil subsidence prediction problem, and the extension of that formulation to combined discrete slip planes and jointed media continua formulations. The results obtained are compared to each other and to a physical centrifuge simulation performed previously on the same problem.

COMMITTEE APPROVAL:

Steven E Benzley
Steven E. Benzley
Committee Chairman

Reese J. Goodwin
Reese J. Goodwin
Committee Member

Henry N. Christiansen
Graduate Coordinator