INFORMATION TO USERS

This dissertation was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.
2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.
3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.
4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

University Microfilms

300 North Zeeb Road
Ann Arbor, Michigan 48106
A Xerox Education Company

72-26,460

PARRISH, David Keith, 1944-
A NONLINEAR FINITE ELEMENT FOLD MODEL.

Rice University, Ph.D., 1972
Geology

University Microfilms, A XEROX Company, Ann Arbor, Michigan

RICE UNIVERSITY

A NONLINEAR FINITE ELEMENT FOLD MODEL

by

David Keith Parrish

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

DOCTOR OF PHILOSOPHY

Thesis Director's signature:

A handwritten signature in dark ink, appearing to read "B. Burchfiel", is written over a horizontal line.

Houston, Texas

May, 1972

PLEASE NOTE:

Some pages may have
indistinct print.

Filmed as received.

University Microfilms, A Xerox Education Company

ACKNOWLEDGEMENTS

Dr. B.C. Burchfiel provided excellent advice, guidance and moral support during the work on this thesis. Drs. H.G. Ave Lallemant, J.C. DeBremaecker, and W.P. Vann were also always available for consultation when I needed them. Mr. Robert G. Vos kindly permitted me to use some computer programs which he had developed, and our discussions about the finite element method provided much of the insight which I needed for this project.

I wish to thank Mr. Joseph Forrest for allowing me to illustrate a specimen from his collection.

This thesis was completed during the tenure of a National Science Foundation Fellowship. In addition, financial support was derived from National Science Foundation Grant Nos. GA-1079 and GA-21375 awarded to Dr. B.C. Burchfiel.

My wife, Sharon, deserves a very special word of appreciation for the many hours she spent drafting and typing.

TABLE OF CONTENTS

CHAPTER 1	Introduction.	1
	Statement of Problem.	1
	Significance.	1
	Conclusions	5
	Outline of Paper	5
CHAPTER 2	The Finite Element Method and Its Application to the Nonlinear Problem	6
	Introduction.	6
	Review.	6
	Application of Finite Element Method to a Nonlinear Problem	15
CHAPTER 3	Assumptions, Boundary, and Initial Conditions	18
CHAPTER 4	Results of the Model.	25
	Geometry.	25
	Stresses.	26
	Viscosities	31
	Summary	37
CHAPTER 5	Conclusions and Discussion.	39
REFERENCES.		45
APPENDIX A	Generalized Flow Chart of Finite Element Computer Program	47

APPENDIX B	Description of Computer Program	48
B-1	General Description.	49
B-2	Description of Major Subprograms	58
B-3	Symmetry Used in Program	97
B-4	Young's Overrelaxation Method.	99
B-5	Catalog of Major Variables Used in Program . . .	102
APPENDIX C	List of Computer Program and Sample Output.	107.

ILLUSTRATIONS

Figure 1.	Plot of differential stress versus $-\log$ strain rate for marble.	3
Figure 2.	Planar continuum divided into finite elements, i, j, k are nodal points defining an element, u and v are x and y components of velocity of each nodal point.	7
Figure 3.	Typical element of continuum body defined by nodes i, j, k whose coordinates are $X_i, Y_i, X_j, Y_j, X_k, Y_k$	9
Figure 4.	Geometrical interpretation of instantaneous viscosity, ϕ , which must be computed for each element for each time interval of deformation.	16
Figure 5.	Initial configuration for fold model	19
Figure 6.	Finite element net used to approximate section ACODF in figure 5.	20
Figure 7.	Fold geometry and orientations of principal compressive stress in initial (0%) second (5%) shortening	27
Figure 8.	Fold geometry and orientation of principal compressive stresses at 25% and 30% shortening	27
Figure 9.	Fold geometry and orientation of principal compressive stresses at 5% increments from 35% to 60% shortening	28
Figure 10.	Fold geometry and orientations of principal compressive stresses at 5% increments from 65% to 90% shortening.	29
Figure 11.	Plot of differential stress, $(\sigma_1 - \sigma_3)$ versus shortening (% average strain) for selected elements in the fold model	30
Figure 12.	Distribution of equivalent viscosities at 25% and 30% shortening.	32
Figure 13.	Distribution of equivalent viscosities at 35% through 60% shortening.	33
Figure 14.	Distribution of equivalent viscosities at 65% through 90% shortening.	34
Figure 15.	Plot of equivalent viscosity versus shortening (% average strain)	35

Figure 16. Naturally deformed fold collected from Mineral Bluff formation, Murphy, North Carolina	.42
Figure 17. Final deck configuration for finite element program	.57
Figure 18. Flow chart for subprogram NODESONBOUN	.62
Figure 19. Flow chart (continued) for subprogram NODESONBOUN	.63

CHAPTER I

INTRODUCTION

Statement of Problem

This study examines a finite element computer model of a folding viscous layer as it deforms according to a nonlinear viscous flow law of the form:

$$(1.1) \quad \dot{\epsilon} = A \exp(-Q^*/RT) \sigma^n$$

where

$\dot{\epsilon}$ = strain rate.

A = diffusion coefficient characteristic of the material.

Q^* = activation energy for creep.

R = gas law constant.

T = temperature ($^{\circ}$ K).

σ = differential stress = $\sigma_1 - \sigma_3$, σ_1 and σ_3 = maximum and minimum principal compressive stresses.

n = empirical constant.

The geologic materials which are used in the model are quartzite for the viscous layer and marble for the matrix in which the layer is embedded. These materials were chosen because of the available experimental data on them.

Significance

This problem was first suggested by the apparent success with which Dieterich (1969) applied the finite element method to the deformation of geologic materials. Dieterich analysed the deformation of a Newtonian

viscous layer embedded in a less viscous matrix, i.e. the viscosity of the materials remained constant throughout deformation. Dieterich was able to use the linear model to interpret the origin of cleavage in rocks, and Dieterich and Carter, (1969) used the stress orientations which the model provided for the interpretation of some natural folds.

Although the linear model is a useful first approximation to natural folding, geologic materials probably deform nonlinearly. In analyzing natural folds, Chapple (1968) suggested that nonlinear material behavior might be the reason most natural folds deviate from those resulting from ideal linear calculations. Dieterich and Carter (1969) stated (p. 142) that a simple viscous model represents only a "crude approximation of the behavior of natural materials undergoing folding at low rates of strain".

This study is an attempt to extend Dieterich's linear model to a nonlinear model on the presumptions that such a model will more realistically describe the natural deformation of geologic materials.

The most promising insight to the actual nonlinear flow laws which govern geologic deformation has come from extensive steady state creep experiments which have been conducted recently on metals and ceramics (Sherby and Burke, 1967; Chang, 1962) and also on rock forming minerals, viz, halite (Carter and Heard, 1970; Heard, 1970), olivine (Carter and Ave Lallemant, 1970), marble (Heard, 1963; Heard and Raleigh, 1971), and quartz (Heard and Carter, 1968; Ave Lallemant and Carter, 1971). All these workers find evidence that the strain rate of deformation in the steady state region is a power function of stress which has the form of equation (1.1).

However, as figure 1 illustrates, the available experimental data only extend to strain rates of 10^{-8} /sec., whereas geologic strain

Figure 1. Plot of differential stress versus $-\log$ strain rate for marble. Heavy lines represent experimental data. Straight lines are extrapolation of data to geologic strain rates. After Heard, 1963.

rates are of the order 10^{-14} - 10^{-16} /sec. (Whitten, 1956; Crittendon, 1963). Although nonlinear flow laws such as equation (1.1) provide the best extrapolation of this experimental data to geologic strain rates, (Heard, 1968; Heard and Raleigh, 1971), actual experimental determination of material behavior at such low strain rates is not possible.

Computer model studies such as this provide a means of examining extrapolated flow laws at geologic strain rates. Once a computer program for the structural analysis of a continuum is developed, the rapid computation speed of the computer can be used to calculate the effect of the flow law acting over long time periods. By computing successive increments of deformation hundreds of times, deformations which would take millions of years can be calculated in minutes.

A computer program for the method of the structural analysis of nonlinear deformations was developed for this study. The program was written for the Burroughs B5700 digital computer at Rice University, Houston, Texas, and a complete description of it is included in the appendices.

The finite element method for analysis of a continuum is the basis of the computer program. The finite element method was originally introduced by Turner et. al. (1956) for the analysis of engineering structures. Later, using techniques summarized by Zienkiewicz and Cheung (1967) the method was extended to the analysis of a continuum. Dieterich (1969) applied the finite element method to the analysis of geologic deformations. The program used for this study performs the computations necessary for a finite element analysis of a continuum whose material behavior is nonlinear. The method used to apply the finite element method to a nonlinear material is described in chapter 2.

Conclusions

The following points summarize the findings of this study.

1. The results corroborate and are similar to Dieterich's linear model.
2. The power law function has a very strong "feedback" mechanism which causes very high viscosity contrasts to develop in the fold.
3. The fold produced in this model study has a very broad open, nearly concentric geometry. This geometry differs from most natural folds in that they usually have a more "similar" type geometry.
4. The materials used in this model displayed a viscosity contrast such that a great deal of thickening took place in the folding layer in the early compression stages. If such thickening takes place during natural fold development, original layer thicknesses and dominant fold wavelengths will be extremely difficult to determine from geometry alone. Complete finite strain analysis will be necessary to reproduce original bed thickness.
5. This model provides the basis for future modification for a more realistic nonlinear fold model.

Outline of Paper

The following chapters will describe the finite element method and its application to a nonlinear problem, describe the assumptions, initial conditions, and boundary conditions which were used in the model, describe the results of the study, and discuss the results and their geologic implications.

CHAPTER 2

THE FINITE ELEMENT METHOD AND ITS APPLICATION TO THE NONLINEAR PROBLEM

Introduction

The finite element method was used as the numerical technique to idealize the viscous solids in this model. This technique is well suited for simulation of flow laws because the equations of motion in viscous solids are replaced by sets of relatively simple relations between forces and velocities at a finite number of nodal points specified throughout a continuous body. Further, the technique simulates finite deformation by a series of essentially infinitesimal deformations. Therefore, two of the problems which arise in using analytical methods for modeling are avoided; first, involved differential equations need not be solved, and second, analytical methods which usually apply only to infinitesimal deformations can be extended to finite deformations.

Review

The finite element method is routinely used in engineering analysis (Clough, 1965), and because complete descriptions of its application to a continuum are available (Zienckiewicz and Cheung, 1967), only a review of basic relations are presented here.

The finite element method divides a continuum into a finite number of discrete elements, (fig. 2). Thus, the total continuous structure, which consists of an infinite number of points, is approximated by finite number of points (nodes which connect the elements). By analyzing each

Figure 2. Planar continuum divided into finite elements, i, j, k are nodal points defining an element, u and v are x and y components of velocity of each nodal point.

element and appropriately combining the results, the total structure is analyzed.

The total analysis of the continuum is thus reduced to a set of linear simultaneous equations of the form:

$$(2.1) \quad \{F\} = [K]\{u\}$$

where

$\{F\}$ = set of forces acting at the nodal points of the structure,

$\{u\}$ = set of nodal velocities throughout the structure, and

$[K]$ = transformation matrix relating $\{F\}$ and $\{u\}$.

Thus, the velocity at each nodal point in the idealized structure is related to the forces on that point by a transformation matrix $[K]$ which is called the stiffness matrix. The key tasks in the analysis lie in forming the stiffness matrix $[K]$ and solving the resulting set of equations.

The finite element analysis proceeds as follows:

1. The continuum is assumed to be two-dimensional and is subjected to plane stress or plane strain.

2. A number of imaginary lines are chosen to separate the body into finite elements interconnected at nodes (fig. 2). The velocities of these nodal points are the basic unknown parameters in the problem. They are designated by u = horizontal velocity of each node and v = vertical velocity of each node. Each element is defined by a set of three nodes, i, j, k , (fig. 3). The coordinates of each node are x_i, y_i, x_j, y_j , and x_k, y_k . The set of nodal velocities of each element can be represented as a vector of the form:

Figure 3. Typical element of continuum body defined by nodes i , j , k whose coordinates are X_i , Y_i , X_j , Y_j , X_k , Y_k .

$$\{u\}^e = \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_k \\ v_k \end{Bmatrix}^e$$

where the superscript denotes the element velocity vector. The set of all nodal velocities in the total structure form the vector $\{u\}$ in equation 2.1.

The set of forces acting at nodal points producing the nodal velocities can be represented as a vector for each element of the form

$$\{F\}^e = \begin{Bmatrix} X_i \\ Y_i \\ X_j \\ Y_j \\ X_k \\ Y_k \end{Bmatrix}^e$$

and the set of forces acting on all nodal parts in the structure forms the vector $\{F\}$ in equation 2.2. $\{F\}^e$ and $\{u\}^e$ must be related by determining the coefficients of the stiffness matrix, $[K]^e$. Coefficients for the stiffness matrix of the total structure are obtained by appropriately combining the stiffness coefficients from each element stiffness matrix, which is determined by the following analysis. This analysis assumes constant rate of strain within each element.

3. The two-dimensional rate tensor defining the rate of strain within each element is written in one-dimensional form as equation 2.3a. The rate of strain at a given point in each element is a function of the

element nodal velocities. That functional relationship is expressed by equation 2.3b.

$$(2.3a) \quad \{\dot{\epsilon}\}^e = \begin{Bmatrix} \dot{\epsilon}_x \\ \dot{\epsilon}_y \\ \dot{\gamma}_{xy} \end{Bmatrix}^e = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix}^e$$

$$(2.3b) \quad \{\dot{\epsilon}\}^e = [B]\{u\}^e$$

where the transformation matrix $[B]$ is obtained by appropriate differentiation of a velocity function which satisfies compatibility conditions within and along the boundaries of adjacent elements. In this analysis, strain rate is assumed constant within each element. The assumption of constant strain rate within each element leads to the selection of linear velocity functions within each element. (See Zienkiewicz and Cheung, 1967, for a full derivation of $[B]$). The linear velocity functions used are $u = a + bx + cy$, and $v = d + ex + fy$, where u and v are velocities of a general point in the element, x , y are the coordinates of a general point, and a , b , c , d , e , and f are constants.

4. A stress tensor defining the state of stress within the element can be written in one-dimensional form and defined by equations 2.4a - 2.4c.

$$(2.4a) \quad \{\sigma\}^e = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}^e$$

$$(2.4b) \quad \{\sigma\}^e = [D] \{\dot{\epsilon}\}^e$$

$$(2.4c) \quad \{\sigma\}^e = [D] [B] \{u\}^e$$

Matrix $[D]$ reflects the physical properties of the material through the parameter η = viscosity and ν = Poisson's ratio. (See Polakowski and Ripling, 1966, for full derivation of $[D]$). Equation 2.4c is obtained by use of equation 2.2.

The nodal forces which are statically equivalent to the actual boundary stresses can be calculated by imposing a set of arbitrary (virtual) nodal velocities. This is accomplished by calculating the external and internal work rates due to the forces and stresses resulting from the virtual velocities. The external and internal work rates can then be equated and the equivalent nodal forces solved for.

The strain rates due to a set of virtual velocities $\{u^*\}$ are calculated using equation 2.2.

$$(2.5) \quad \{\dot{\epsilon}\}^e = [B] \{u^*\}^e$$

The external energy dissipation, which is the external work rate, W_e , produced by nodal forces acting through the virtual velocities is given by equation 2.6.

$$(2.6) \quad W_e = \left(\{u^*\}^e \right)^T \{F\}^e$$

The internal work rate, W_i , is the product of the stress and strain rates integrated over the volume of the element,

$$(2.7) \quad W_i = \int_v \left(\{\dot{\epsilon}\}^e \right)^T \{\sigma\}^e d \text{ vol}$$

or, using equations (2.3) and (2.4) (matrix brackets have been dropped in the following equations):

$$(2.8) \quad W_i = \int_V \left\{ \{u^*\}^e \right\}^T B^T D B \{u\}^e d \text{ vol.}$$

By the virtual work principle, W_e must be equal to W_i , so

$$\left\{ \{u^*\}^e \right\}^T \{F\}^e = \int_V \left\{ \{u^*\}^e \right\}^T B^T D B \{u\}^e d \text{ vol.}$$

Because B , D and $\{u\}^e$ are not functions of position within the element, they can be moved outside the integral. Therefore,

$$\{F\}^e = \int_V B^T D B d \text{ vol} \{u\}^e = B^T D B \{u\}^e \int_V d \text{ vol},$$

which is valid for any set of velocities applied to the element.

The set of forces acting on each nodal point can be determined by summing the contributions from all elements, so the equilibrium of the total structure is expressed by

$$(2.9) \quad \{F\} = \sum B^T D B \int d \text{ vol} \{u\}^e$$

The summation is over all elements.

Equation 2.9 is a set of simultaneous equations which has the same form as 2.1 where

$$[K] = \sum B^T D B \int d \text{ vol.}$$

Because $\int d \text{ vol}$ can be computed as $\int d \text{ vol} = \text{one half the thickness of the element times its area}$, $[K]$ can be computed by a digital computer using a series of matrix and summation operations.

6. After having determined equation 2.9, boundary conditions are specified as a set of velocities. Specifying these velocities removes the singularity of the stiffness matrix. The stiffness matrix is singular up to this point because there is no unique solution to the system of equations which it defines. The analysis of each element provides coefficients for the virtual displacement of each node with respect to the other nodes, but K does not define a set of independent relations between forces and velocities because there are more equations than unknowns. This corresponds to the physical situation of permitting rigid-body translation and rotation when a set of forces are applied. If no boundary velocities are specified, an infinite displacement may result from a finite force. Therefore, some of the equations must be removed from the system. This is done by specifying some velocities as known quantities so that the corresponding equations for those quantities need not be solved. After the boundary conditions are specified a set of applied forces or nodal velocities may be imposed and the corresponding unknown reaction forces and unknown velocities can be solved for.

7. Having determined a velocity distribution of points throughout the body, equations (2.3) and (2.4) can be used to compute the strain-rates and stresses within each element.

8. Finally, the velocities can be allowed to operate for a small period of time, Δt , to yield a displacement of each nodal point which will define a new set of elements for another finite element analysis of the body. Thus the behavior of the body through time can be observed by a series of successive finite element analyses.

Application of the Finite Element Method to a Nonlinear Problem.

The finite element method reduces the analysis of a structure to the solution of a set of linear simultaneous equations (2.1). Therefore in order to apply the finite element method to nonlinear flow, the problem must be reduced to a series of linear problems. This is accomplished by allowing each element to behave linearly for a short period of time and adjusting the linear flow law for each period.

because the flow law is nonlinear, the parameter of viscosity, η , defined by the linear behavior of a Newtonian fluid $\sigma = \eta \dot{\epsilon}$ must be replaced by a parameter, ϕ , which is the instantaneous viscosity at any point in the deformational history of a viscous solid. ϕ is a function of the differential stress, σ , to which the viscous solid is subjected. Geometrically, ϕ corresponds to the slope of a line from the origin to a point on a stress vs. strain rate curve (see fig. 4), or

$$(2.11) \quad \phi = \frac{\sigma}{\dot{\epsilon}}$$

$$\sigma = \sigma_1 - \sigma_3$$

$$\dot{\epsilon} = \text{rate of strain in material.}$$

Ideally, in a nonlinear material, ϕ changes each instant in response to infinitesimal changes in the differential stress, σ . An approximation to this phenomenon, which is the artifice used for the nonlinear problem, is to divide the duration of the deformation into small time intervals during which the viscosity in each element, ϕ_e , remains constant. However, ϕ_e is allowed to have a new value in each succeeding time interval, depending on the state of stress (or equivalently, the strain rate) in the preceeding interval. During each k th time interval, Δt_k , the

Figure 4. Geometrical interpretation of instantaneous viscosity, ϕ , which must be computed for each element for each time interval of deformation.

differential stress, σ , is computed for each element. This differential stress is used in equation 2.11 to compute ϕ_{k+1} , the viscosity for the next time interval Δt_{k+1} . Because viscosity can be specified for every element in the finite element approximation, the viscosity will vary not only with time but also with position in the body.

Except for the modification of the viscosity of each element for each time interval, the analysis proceeds in the same manner for a non-linear flow law as for a linear flow law discussed earlier.

Young's overrelaxation method (Froberg, 1965) was used as the iterative technique to solve the simultaneous equations (2.1) which result from the finite element formulation. An explanation of the technique is given in appendix D.

CHAPTER 3

Chapter 3 describes the assumptions, boundary conditions and initial conditions used in the fold model. The fold model is discussed in chapter 4.

The model consists of a single viscous layer embedded within a less viscous matrix. Figure 5 is a diagram of the layer and matrix. The layer is given an initial sine wave perturbation so that it will fold when compressed laterally. The initial wave length and thickness of the viscous layer were calculated on the basis of the viscosity contrast between the layer and the matrix. The equivalent viscosities (Griggs, 1939) shown in table 3.1 were used in the Biot (1961) formula for infinitesimal fold amplification to determine a dominant wavelength to thickness ratio of 9 to 1. The Biot formula is

$$L_D = 2\pi h \sqrt[3]{\frac{\eta}{6\eta_1}}$$

where

L_D = dominant wavelength.

h = layer thickness.

η = layer viscosity.

η_1 = matrix viscosity.

Because of the symmetry inherent in the original configuration the computational size of the problem is reduced to only 1/4 of the total problem shown in Figure 5. First, the sine wave is symmetrical about

Figure 5. Initial configuration of fold model. See text for explanation.

Figure 6. Finite element net used to approximate section ACODF in figure 5.

line ABC. Hence, only half the total wavelength is necessary to describe the whole fold. A second symmetry arises when the point O in figure 5 is held fixed. Point O becomes a center of symmetry such that a point at P' is subjected to stresses which are equal but opposite to the stresses at point P. Consequently, only the deformation in the half of the layer above line COD must be computed. Because of these two symmetries, the portion of the fold deformation which must be solved is enclosed by ACODF in figure 5 and other portions of the fold are determined by symmetry.

Figure 6 shows the finite element nodal point net which was used to approximate section ACODF of figure 5. The viscous layer is section ACODE. Note that the nodal point distribution is denser in the viscous region, especially near the crest and trough of the fold. These are the areas where the greatest variations in geometry and stress distribution are anticipated. Therefore, the elements are arranged to give greatest resolution in those areas.

The flow law, i.e. the stress-strain rate relationship for both the layer and the matrix, is the power function flow law (eqn. 1.1) which was adopted by Heard (1968), Heard and Raleigh (1971) and Heard and Carter (1968) as the best extrapolation of their data to geologic strain rates. The activation energy for creep (Q^* in equation 1.1) and the material constant A in equation 1.1 are assumed to remain constant throughout the deformation experiment. Temperature is assumed to be constant at 500°C throughout the experiment.

The basic assumption necessary for applying a power law to geologic material is that the creep rate be dependent on a self diffusion process, such as dislocation climb, or on the motion of jogged screw

dislocations (Sherby and Burke, 1967, p.332). Such processes are proposed by Heard (1968) as well as by Heard and Carter (1968). Heard proposes dislocation climb and Heard and Carter propose trapped dislocation arrays of some type to explain subbasal deformation lamellae in quartz.

Because the power function flow law is operative in the steady state region of flow (Sherby and Burke, 1967), implicit in a model of such a function is the assumption of steady state deformation. This assumption is reasonable since the elastic component of strain is confined to the first 1-2% strain in geologic materials (see Heard, 1960 for stress-strain data typical of geologic materials). At the low strain rates which are specified in the model, the assumption that the material is in the steady state region of deformation seems valid as a first approximation.

Materials used in this model were quartzite and marble. Quartzite functions as the viscous layer and marble functions as the less viscous matrix. Experimental data for these two materials were obtained from Heard (1963) and Heard (1968) for marble and from Heard and Carter (1968) for quartzite. From these data representative material constants (n , Q^* in eqn. 1.1) were selected for computation of the equivalent viscosities (Griggs, 1939) for the flow of quartzite and marble at a strain rate of 10^{-14} /sec. These computed values are given in table I and they were used as the starting material values in the model.

To make the materials nearly incompressible, Poisson's ratio of 0.4 was assigned to every element.

Material	A ($\text{Kb}^{-n} \text{sec}^{-1}$)	Q^* (Kcal/mole)	n	Equivalent Viscosity (Poises)
marble	$1.949 \times 10^{+12}$	62	8	1.0×10^{21}
quartzite*	6.49×10^{-2}	45	5	1.2×10^{22}

*Recalculated from data of Heard and Carter, 1968; only steady state data in the subbasal field were used (No.s 639, 630, 651, 659, 660, 622, 662, 653).

Table 1. Experimental constants and resulting initial equivalent viscosities for materials in the fold model. A = diffusion coefficient, Q^* = activation energy for creep, n = stress exponent, equivalent vis-

$$\text{cosity} = \frac{\sigma}{3\epsilon}.$$

The initial boundary conditions used for each step of the incremental analysis were:

1. Point 0 in figure 6 was held stationary, i.e. $u = 0$, $v = 0$.

2. Frictionless walls ABC and FED in figure 6 converged at a representative geologic rate of shortening of 10^{-14} /sec. Nodal points along wall ABC were given specified velocities $u = U$. Nodal points along wall FED were given specified velocities $u = -U$. The actual velocities at which the walls converged in each increment of deformation depended on the total width of the model at that time. Because a constant natural strain rate was maintained, the velocities of the walls decreased as deformation proceeded.

The walls were allowed to converge for a time interval Δt , sufficient to shorten the model width, L, by 5% ($\Delta L/L = .05$). The amount of shortening is given by $\Delta L = 2u\Delta t$, where u = the velocity of each

wall. Therefore, the time interval for each deformation increment is $\Delta t = L \times (0.1u)$.

Because deformation is a function of the stress deviator, body forces are not included in the computations.

CHAPTER 4

RESULTS OF THE MODEL

Geometry

The fold geometries generated by the power law model as well as the maximum principal compressive stress orientations produced in each layer configuration are shown in figures 7, 8, 9, and 10. Figure 7 illustrates the early stages of folding in which the viscous layer is subhorizontal. Figure 8 illustrates the early stages of fold amplification, and figures 9 and 10 show, during 5% increments up to 90%, natural strain.

As can be observed in figures 7-10, the thickness of the viscous layer increases during deformation. The total amount of thickening is more than 35% at 90% shortening. Two separate thickening processes can be distinguished. In the early stages, when the layer is subhorizontal, a homogeneous uniform thickening increases the layer thickness by about 18% at 25% shortening. However, after the fold has amplified to about 10°-15° limb dip, compression begins to cause flow in the inner hinge, (i.e. the concave surface of the anticline and syncline). However, the flow in the outer hinge, (i.e. convex surface of anticline and syncline) does not take place. Therefore, the second stage of thickening takes place in the inner hinge of the fold. (Note that the inner hinge of the anticline is the same surface as the outer hinge of the syncline.)

The final geometry is that of a broad open, apparently "concentric" fold. This concentric shape, however, was not produced by any of the modes of deformation usually associated with concentric folds. First,

this is definitely not a flexural slip fold. Flexural slip is precluded by the finite element method, because the boundaries of elements across the layer-matrix interface must remain compatible. There can be no slip between the layer and the matrix. Further, a true buckling phenomenon has not taken place, because the layer has not maintained uniform thickness, but has, in fact, thickened uniformly. Most of the thickening has taken place in the early stages of deformation when the layer was sub-horizontal. This thickening during the production of a concentric geometry may have implications about the traditional interpretation of "concentric" folds. These implications are discussed in the next chapter.

Stresses

The short lines in figures 7-10 are parallel to the orientations of σ_1 , the maximum principal compressive stress. The lines are not proportional to stress magnitudes. Variation of maximum principal compressive stress magnitudes for selected portions of the fold are shown in figure 11. The principal compressive stress axes are perpendicular to principal axes of infinitesimal elongation.

The orientation of stresses with respect to the fold limb changes during the fold process. In the outer fold hinge, σ_1 is initially parallel to the layer, but as the fold amplitude increases, the limbs rotate such that σ_1 becomes subnormal to the layer. At about 25-30% shortening, a zone of extension develops in the outer hinge. Between this extension zone and the lower part of the layer, which is under compression, a neutral zone ($\sigma_1 = \sigma_3$) develops. This neutral zone migrates from the outer hinge toward the inner hinge. An extensional region also developed in the linear models of Dieterich, (1969). In fact, Dieterich (pers. commun., 1971) has also noted a similar stress distribution in an unpublished nonlinear model ($\dot{\epsilon} = \frac{\sigma^3}{\eta}$).

Figure 7. Fold geometry and orientations of principal compressive stress in initial (0%) and second (5%) shortening.

Figure 8. Fold geometry and orientations of principal compressive stresses at 25% and 30% shortening.

Figure 9. Fold geometry and orientation of principal compressive stresses at 5% increments from 35% to 60% shortening.

Figure 10. Fold geometry and orientations of principal compressive stresses at 5% increments from 65% to 90% shortening.

Figure 11. Plot of differential stress, $(\sigma_1 - \sigma_3)$ versus shortening (% average strain) for selected elements in the fold model. The location, symbol, and number for each selected element is shown in the insert.

As shown in figure 11, the stress magnitudes in the extensional area (elements 115, 116) become very small. Near 50% shortening, a state of tension develops within one element (small squares) in the outer hinge. In the inner fold hinge (triangles of figure 11) the compressive stresses increase steadily from approximately 1.24 Kb to 2.25 Kb as the fold is shortened 90%. Throughout the deformation, the stresses in this region remain parallel to the layer. The stresses in the regions represented by solid squares, open circles and x's in figure 11 experience rotations and changes in magnitude which are intermediate between the extremes represented by the outermost and innermost portions of the hinge. In the limb of the fold (solid circles, figure 11) stresses remain relatively parallel to the shortening direction, while the limb increases in dip. At the end of deformation, stresses are inclined at approximately 65° to the limb. This angle is slightly lower than the 70° angle given by Dieterich, (1969) for the same percent shortening.

Viscosities

Equivalent viscosities at each strain episode are contoured in figures 12, 13, and 14. The contour units are Poises $\times 10^{21}$. Viscosities are indicated by dark shading (<3), white (3-5), lines (5-10), and white with contour lines (>10, each contour line = 10 units). The figures show a progressive development of a high viscosity region in the outer hinge of the fold. This "stiff" region migrates downward as folding proceeds. Finally, a very viscous band forms across the crest of the fold. The highest viscosity computed in the band is nearly two orders of magnitude greater than the viscosities in the inner hinge.

The high viscosities that form in the crest are a result of the decrease in stress magnitudes in the region of extension discussed above. Because viscosity is computed as $\eta = \frac{\sigma}{3\dot{\epsilon}}$ where $\dot{\epsilon} \propto \sigma^5$, as strain rate

Figure 12. Distribution of equivalent viscosities at 25% and 30% shortening. See text for description.

Figure 13. Distribution of equivalent viscosities at 35% through 60% shortening. See text for description.

Figure 14. Distribution of equivalent viscosities at 65% through 90% shortening. See text for description.

Figure 15. Plot of equivalent viscosity versus shortening (% average strain). The location, symbol, and number of each selected element is shown in the insert.

decreases, the viscosity increases. The effect of decreasing strain rate can be observed by comparing figures 12-14. As the extensional zone migrates through the layer, the viscosities increase.

This migration of high viscosity through the hinge can also be seen in figure 15, in which viscosities in selected portions of the fold are plotted against percent shortening. Elements are represented by the same symbols as were used in figure 11. Note that the viscosity of the layer remains nearly uniform during the first 20% shortening.

Element 116 (open squares) is the outermost element on the upper part of the anticlinal crest. At 35-40% average strain, the viscosity of element 116 increases sharply above all other elements in the fold. Element 115 (solid squares) is the next innermost element. At 50% shortening, element 115 develops a high viscosity while 116 has decreased in viscosity. Element 149 is the next innermost element to show a sharp increase; it is the next innermost element selected. Finally, the viscosity of element 205 which is the innermost element, starts to increase during the last deformation phases. The progressive movement of the high viscosities through the fold mark the movement of the "neutral zone" separating the extensional area from the compressional area.

Although the model does not show development of structures within the folding layer, the contour of viscosities in the 55% shortening stage (figure 14) shows a pattern which shows a marked resemblance to second order folds. This pattern suggests a mechanism by which some drag folds may originate.

The pattern represents a viscosity contrast created by the stress distribution in the fold. These viscosity contrasts are likely areas for differential flow. This model, therefore, demonstrates that inhomogeneities,

i.e. viscosity contrasts, are ideal areas for the development of secondary folds.

Summary

In summary, the features which were produced by this nonlinear fold model are:

1. The layer has thickened overall by nearly 35%. About half the thickening is due to uniform thickening of the layer in the early stages of deformation. The rest of the thickening is due to flow into the inner hinge.
2. A broad open fold developed. Although the final fold shape resembles a buckle or concentric fold, the deformation process was not buckling or flexural slip.
3. The pattern of maximum principal compressive stresses show the development of a region of extension in the outer fold hinge. The stress magnitudes in the extensional region are relatively low.
4. The maximum principal compressive stresses in the inner hinge increase in magnitude throughout folding and remain parallel to layering.
5. Maximum compressive stresses in the fold limb increase, then decrease in magnitude as the dip of the limb increases. They finally become oriented at a high angle to the layering.
6. The equivalent viscosity pattern shows two features: a) A highly viscous region develops in the outer hinge and migrates toward the inner hinge. The viscous region is aligned with, and is a result of, the neutral zone of low differential stresses which develops between the compressive and extensional

regions of the fold. b) At one point in the deformation history, the inhomogeneities produced by differing viscosities in the fold form a pattern which is suggestive of second order folding.

CHAPTER 5

CONCLUSIONS AND DISCUSSION

The purpose of this study was to construct a fold model which deformed according to the power flow law. This has been accomplished, and the following is a summary of the conclusions and geologic implications that can be drawn from the model. In addition, this chapter discusses possible improvements which should be made in future fold models.

The geometry and stress distributions computed in this model are similar to and therefore corroborate previous linear models of Dieterich, (1969). As in Dieterich's linear model, this model shows the development of an extensional region in the outermost hinge of the fold. This extensional region implies that a neutral zone separates the extensional region from the compressional regions of the fold. It should be pointed out that the neutral zone discussed here should not be confused with a neutral surface which arises in buckling phenomena. The neutral zone in this model does not extend throughout the length of the layer in the manner that a neutral surface does in a buckled beam. Rather, the neutral zone exists only in the anticlinal crest of the fold. This observation implies that contrary to elementary descriptions of flexural folds, if flow has taken place, it may be almost impossible to define a true neutral surface throughout the fold length.

Another feature of this model which is similar to the linear model is the rotation of the limbs relative to the stress directions within the limb so that when high dips are reached, the maximum principal stress axes are directed across the limb.

Finally, this model is generally similar in geometry to the linear model. That is, a very open, "concentric" fold developed.

There are basically two differences between this nonlinear model and Dieterich's linear model. First, the nonlinear model displayed more flow of material into the inner hinge. This is a direct consequence of the power law (eqn. 1.1). The inner hinge is a region of high stress and therefore high strain rate. As stress increases, the viscosities in that region are reduced. Therefore, the strain rates increase. There was less flow of material in the outer hinge. If any flow occurred, it was away from the outer hinge toward the limbs. Again the power law produced a change in viscosities to cause this effect. In this case, the viscosities became higher because the differential stress was reduced in the outer hinge. Both the flow in the inner hinge and the lack of flow in the outer hinge illustrate a very strong positive feedback characteristic of the power law. Once an area starts to become more viscous, such as the outer hinge in the early stages of fold amplification, that area continues to become more viscous with increased strain.

A second feature which can be observed in a nonlinear model, but not in a linear model is the distribution of viscosities as the fold develops. Mapping viscosities in this fold reveals how inhomogeneities i.e. viscosity differences, can develop within an originally homogeneous layer during the fold development. These inhomogeneities probably also develop in natural deformations, and lead to differential flow. These areas of differential flow are probably recorded in the geologic record as second order folds.

The large amount of thickening within the fold layer, especially in the early stages may affect the interpretation of the development of concentric folds as well as the interpretation of wavelength-thickness

ratios in natural folds. If folded layers undergo thickening in their early stages such as the layer in this model did, then caution must always be used in determining original thicknesses of folded layers. As a consequence, the measurement of dominant wavelength-thickness ratios of natural folds will be difficult to interpret, unless independent strain measurements are made to determine whether or not layer thickening has occurred. Geometry alone is not sufficient to determine fold mechanics.

The primary purpose in developing a model is to simulate reality. A successful model should generate a fold geometry which looks like real folded rocks. To an extent, this model did not simulate reality, at least not for the geologic materials selected. Although this model produced a fold which is much more open than most natural folds in quartzite and carbonate strata, fold geometries like this model are observed in geologic record. Figure 16 is a photograph of such a fold from the Mineral Bluff formation, Murphy, North Carolina. The specimen pictured in figure 16 was collected by Mr. Joseph Forrest. This fold occurs in a calc-silicate gneiss. The folded layer is composed primarily of quartz and plagioclase, with hornblende porphyroblasts and the matrix is quartz-sericite schist. This fold is very similar in geometry to the folds illustrated in this report. Furthermore, the hornblende porphyroblasts are aligned parallel to layering in the outer hinge and perpendicular to layering in the inner hinge. On the limbs the porphyroblasts form with no indication of a neutral fiber. These orientations are perpendicular to the maximum principal compressive stress directions in the nonlinear model.

The nonlinear model suggests modifications for future fold models which may permit the modeling of more realistic folds and provide more information about geologic deformation. The folds in this study lack a

Figure 16. Naturally deformed fold collected from Mineral Bluff formation, Murphy, North Carolina. See text for description.

"similar" type geometry. Although ideal similar folds rarely exist in nature, most folds exhibit a degree of differential thickening in their hinges relative to their limbs so that they have a similar rather than concentric geometry. This feature may be due to some sort of uniform flattening process (Ramsay, 1962; Desitter, 1964), or to flow from the limbs into the hinge (Donath and Parker, 1964). Neither of these processes occurred in this model. Material did not flow from the limbs to the outer fold hinge in the nonlinear model primarily because of the feedback nature of the power flow law. In the early stages of fold amplification, high viscosities began to develop in the outer hinge. Strain rates in the outer hinge were consequently low, which led to higher viscosities. By the time the fold limb had rotated so that stresses were directed across it so as to produce flow parallel to layering, two other effects had also occurred: (1) the stresses acting on the limb had decreased in magnitude so that they were not effective in flattening the limb, and (2) viscosities in the hinge had become too high to permit flow into that area.

Therefore, it seems reasonable, not only on the basis of this model, but also from the geologic reasoning that one flow law probably is not operative at all times and places during fold development, that perhaps different flow laws should be applied to different parts of the fold. A basis for assigning these different flow laws may be found in experiments on metals (Sherby and Burke, 1967), in which thermally activated processes determine the creep rate. As the stress and strain rate increase from low strain rates and stresses to faster strain rates and higher stresses, the flow laws change from Newtonian linear to a power function to an exponential function.

In terms of this fold model, the low strain rates and stresses occur in the outer hinge, indicating that either a linear law or a low

(<5) exponent for the power law should apply. Such a flow law would certainly prevent the high viscosities shown in figures 12-14 and 15 from developing. For the region of high strain rates and stresses in the inner hinge, it seems that an exponential law of the form $\dot{\epsilon} = A \exp(B\sigma)$ should be applied.

As a further modification, fracture criteria should be applied to fold models. Experimental data demonstrate that as strain rates increase, the mode of deformation changes from ductile flow to rupture. Such a deformation transition probably takes place in folds. Within the inner hinge, the strain rates probably increase to such a point that they shift the mode of deformation from the ductile regime to the rupture regime.

The flow laws which are suggested here for various parts of a fold imply differing mechanisms of creep in each of those areas. Unfortunately, at the present time we cannot distinguish these different mechanisms by either petrographic or field observations. Various thermally activated processes such as self diffusion, dislocation pile up, syntectonic recrystallization, etc. are observed in deformed rocks. However, the contributions of each cannot as yet be separated so that reasonable geologic assumptions can be made for flow laws in separate parts of a fold. Hopefully, results of future fold models will provide the insight needed to determine the mechanisms by which folds form.

REFERENCES

- Ave Lallemant, H.G., and Carter, N.L., 1971, Pressure dependence of quartz deformation lamellae orientations: *Am. Jour. Sci.*, v. 270, p. 218-235.
- Biot, M.A., 1961, Theory of folding of stratified viscoelastic media and its implication in tectonics and orogenesis: *Geol. Soc. America Bull.*, v. 72, p. 1595-1619.
- Carter, N.L. and Ave Lallemant, H.G., 1970, High temperature flow of dunite and peridotite: *Geol. Soc. America bull.*, v. 81, p. 2181-2702.
- _____, and Heard, H.C., 1970, Temperature and rate dependent deformation of halite: *Amer. Jour. Sci.*, v. 269, p. 193-249.
- Chang, R., 1962, Dislocation theories of the high temperature creep of crystalline solids, p. 275-285 in Kingsberg, C., (ed), *Physics and Chemistry of Ceramics*: New York, Gordon and Breach, 361 p.
- Clough, R.W., 1965, The finite element method in structural mechanics, p. 85-119 in Zienciewicz, O.C. and Holister, G.S., (ed), *Stress Analysis: Recent Developments in Numerical and Experimental Methods*: New York, John Wiley.
- Crittenden, M.D., Jr., 1963, Effective viscosity of the earth derived from isostatic loading of Pleistocene, Lake Bonneville: *Jour. Geophys. Research*, v. 68, p. 5517-5530.
- De Sitter, L.U., 1964, *Structural geology*, 2nd ed.: New York, McGraw-hill, 551 p.
- Dieterich, James H., 1969, Origin of cleavage in folded rocks: *Amer. Jour. Sci.*, v. 267, p. 155-165.
- _____, and Carter, N.L., 1969, Stress history of folding: *Amer. Jour. Sci.*, v. 267, p. 129-154.
- Donath, Fred A. and Parker, Donald B., 1964, Folds and folding: *Geol. Soc. of America Bull.*, v. 75, p. 45-62.
- Froberg, C.E., 1965, *Introduction to numerical analysis*: Reading, Mass. Addison-Wesley.

- Heard, H.C., 1960, Transition from brittle to ductile flow in Solenhofen limestone as function of temperature, confining pressure, and interstitial fluid pressure: Geol. Soc. of America Mem. 79, p. 193-226.
- _____, 1963, Effect of large changes in strain rate in the experimental deformation of Yule marble: Jour. of Geol. v. 71, p. 162-195.
- _____, 1968, (Abs), Steady state flow in Yule marble at 500°C to 800°C: Amer. Geophys. Union Trans.
- _____, 1970, (Abs), The flow law of polycrystalline halite: Amer. Geophys. Union Trans., v. 51, p. 424.
- _____, and Carter, N.L., 1968, Experimentally induced "natural" intergranular flow in quartz and quartzite: Amer. Jour. Sci., v. 266, p. 1-42.
- _____, and Raleigh, C.B., 1971, Steady state flow in marble at 500°C to 800°C: Bull. Geol. Soc. Am., (in press).
- Polakowski, N.H. and Rippling, E.J., 1966, Strength and structure of engineering materials: Englewood Cliffs, N.J., Prentice-Hall, Inc. 535 p.
- Ramberg, H., 1964, Selective buckling of composite layers with contrasted rheological properties; a theory for simultaneous formation of several orders of folds: Tectonophys., v.1, p. 307-341.
- Sherby, O.D. and Burke, P.M., 1967, Mechanical behavior of crystalline solids at elevated remperature: Prog. in Mat. Sci., v. 13, p. 324.
- Turner, M.J., Clough, R.W., Martin, H.C., and Topp, L.J., 1956, Stiffness and deflection analysis of complex structures: Jour of the Aeron. Sci., v. 23, no. 9, p. 805-823.
- Whitten, C.A., 1956, Crustal movement in California and Nevada: Amer. Geophys. Union Trans., v. 37, p. 393-398.
- Young, David, 1954, Iterative methods for solving partial difference equations of elliptic type: Trans Amer. Math. Soc., v. 76, p. 92-111.
- Zienkiewicz, O.C. and Cheung, T., 1967, Finite element method in structural and continuum mechanics: London, McGraw-Hill, 274 p.

Appendix A. Generalized flow chart of finite element computer program.

APPENDIX B

DESCRIPTION OF COMPUTER PROGRAM

APPENDIX B-1

GENERAL DESCRIPTION OF FINITE ELEMENT

COMPUTER PROGRAM

TITLE Compatible Algol Finite Element Program - FINITE/VERSI

DATE November 18, 1971

AUTHOR David Parrish, Department of Geology, Rice University,
Houston, Texas.

PURPOSE Perform the necessary computations for a finite element analysis of either an elastic solid or a linear or non-linear viscous solid. Subprograms are included which adapt the program for a nonlinear viscous solid whose flow law is $\dot{\epsilon} = A \exp(-Q^*/RT)\sigma^n$. (See text for discussion of flow law.) The following discussion is intended for the application of the program to a viscous problem. For elastic problems substitute "displacements" and "strains" for the terms "velocities" and "strain rates" respectively.

METHOD This program is a modification of the finite element method of analysis applied to a continuum described by Zienkiewicz and Cheung, 1967. The modifications have been made for the adaptation to a nonlinear material, for the implementation of a dynamic incremental analysis scheme, and for substitution of an iterative technique for solving the simultaneous equations (Young's Over-relaxation Method).

INPUT

1. Constants for constitutive law of the materials under analysis.
2. Datum coordinates x, y of nodal points in structure.
3. Nodal configuration of elements.
4. Boundary conditions - specified velocities of nodal points.
5. Partitions into which structure is divided.
6. Material parameters:
Viscosity or elasticity constant, and Poisson ratio for each material.
7. Concentrated forces to be applied to structure.
8. Body forces on structure.

OUTPUT

1. All input data
2. Equivalent viscosities for each element.
3. Stresses and strain-rates in each element.
4. Nodal velocities.
5. New coordinate positions after forces are applied.
6. Data file of new coordinates, stresses, and strain rates available for plotting.

ALGORITHM

The program consists of four major blocks. (See flow chart in Appendix A). The outermost block is the program block itself. The primary function of this block is to make most of the declaration which will be needed later in the program. It also reads the number of separate problems which will be solved.

The second block is entered once for every problem to be solved. This block reads the control parameters

necessary to declare the nonstatic arrays appropriate to the size of each problem. The third block is called the main calling block because it calls the subprograms declared previously. This block declares non-static arrays and reads all input data.

After the calling block has read data and computed from that data information necessary to solve the problem, the fourth major block is entered. This is the structure block. It declares the stiffness matrix and computes deformation due to the set of displacements (velocities) and/or forces which have been specified for the structure. If the problem requires a dynamic analysis, incremental deformations are solved repeatedly.

The technique used to solve the simultaneous equations which result from the finite element analysis is Young's Overrelaxation Method. (Young, 1954, see also Appendix B-4).

MACHINE AND LANGUAGE

The program is written in compatible Algol and has been run on the Rice University Burroughs B5700 computer using a Level XII compiler system.

AVAILABILITY

A copy of the program may be obtained by writing to the author, c/o Department of Geology, Rice University, Houston, Texas.

INSTRUCTIONS FOR OPERATION

A. Data preparation

1. Prepare problem control card.

Col 1-4 = NPROB (format 14) = number of data sets to be processed.

2. Prepare data sets.

a. Title card is first card of each data set.

Col 1-72 = any alphanumeric data.

b. Control card 1- input data parameters

Free field format: separate variables by commas and terminate with*: NPART, NPOIN, NELEM, NBOUN, NCOLN, NYM, NFREE, NCONC, FPVX, FPVY, NP, DELTEE, NTIMES, *.

c. Control card 2

1. Col. 1-5 = TRUE if problem is linear.

= False if problem is nonlinear.

2. Col. 6-8 = blank.

3. Col. 9 = 0 data will not be saved to restart program.
= 1 data will be saved so that program can be restarted from point at which analysis ceases.

4. Col. 10-12 = blank.

5. Col. 13 0. Coordinate data is on CARD FILE.
= 1. Coordinate data is on DISK FILE.
= 2. Coordinate data is on CARD FILE but other restart data is on DISK FILE.

6. Col. 14-16 = blank.

7. Col. 17 = 1 = element stresses will be printed on output and stored on tape for plot.

8. Col. 18-20 = blank.
9. Col. 21 = 1 = element strains will be printed on output and stored on tape for plot.

d. Control Card 3 - Iteration Parameters.

1. Col. 1-4 = RFACT = overrelaxation parameter (Format F5.3). $0 < \text{RFACT} < 2$.
2. Col. 5-17 = EPS = error tolerance allowed for displacement (velocity) residuals (Format R12.5).
3. Col. 18-21 = ITTER = maximum number of iterations allowed (Format I4).

e. Flow law data set - one card for each different material in the structure in the following format.

1. Col. 1-14 = A = diffusion constant (Format R14.8).
2. Col. 15-28 = Q^* = activation energy for creep. (Cal/mole) (Format R14.8).
3. Col. 29-42 = T = temperature ($^{\circ}\text{C}$) (Format R14.8).
4. Col. 43-46 = N = exponent of stress (Format I4).

f. Control card 4 - Deformation Parameters

1. Col. 1-14 = STRAINRATE = strain rate at which total structure is deformed by convergence of vertical walls. (Format R14.6).
2. Col. 15-28 = LOGSTRAIN = logarithmic strain allowed in each increment of deformation. (Format R14.6).

g. Coordinate data set.

1. Card 1

Col. 1-4 = NPOIN = number of nodal points (Format I4).

2. Coordinate cards

a. Col. 1-14 = x coordinate (Format R14.6).

b. Col. 15-28 = y coordinate (Format R14.6).

h. Element data set

1. Card 1

Col. 1-4 = NELEM = number of elements (Format I4).

2. Element data cards.

a. Col. 1-4 = i th nodal point	}	number each
b. Col. 5-8 = j th nodal point		element anti-
c. Col. 9-12 = k th nodal point		clockwise

3. Col. 13-28 = blank.

4. Col. 29-44 = thickness of element (Format R16.8).

5. Col. 45-48 = material constant data applicable to element.

i. Boundary condition data set - boundary conditions are specified as nodal displacements for elastic bodies or velocities for viscous bodies.

1. Card 1

Col. 1-4 = NBOUN (Format I4).

2. Boundary condition data cards.

a. Col. 1-4 = node at which boundary condition is specified.

b. Col. 5-7 = blank.

c. Col. 8 = 0 = velocity in x direction is specified.

= 1 = velocity in x direction is not specified.

- d. Col. 9-12 = 0 = velocity in y direction is specified.
 = 1 = velocity in y direction is not specified.
- e. Col. 13-28 = magnitude of displacement (velocity) in x direction (Format R16.8).
- f. Col. 29-44 = magnitude of velocity in y direction (Format R16.8).
- j. Partitioning data set. Structure may be divided into partitions. If the problem utilizes the embedded boundary feature, the structure must have at least two partitions. The last partition must consist of only the row of elements adjacent to the embedded boundary.
 - 1. Card 1
 - Col. 1-4 = NPART (Format I4).
 - 2. Partition data cards - one card for each partition.
 - a. Col. 1-4 = NSTART (Format I4)
 - b. Col. 5-8 = NEND (Format I4)
 - c. Col. 9-12 = NFIRST (Format I4)
 - d. Col. 13-16 = NLAST (Format I4).
- k. Elastic Property - data set.
 - 1. Card 1-4 = NYM (Format I4)
 - 2. Elastic property data cards - one card for each different material in structure.
 - a. Col. 1-14 = Young's modulus x-direction.
 - b. Col. 15-28 = Young's modulus in y-direction.
 - c. Col. 29-42 = Poisson's Ratio in x-direction (Format R14.8).
 - d. Col. 43-56 = Poisson's Ratio in y-direction (Format R14.8).

e. Col. 57-70 - Shear modulus in anorthotropic material.

1. Concentrated force data set. Concentrated forces may be specified, an applied force of zero at that nodal point is assumed. Concentrated forces and displacements parallel directions would not be specified for the same nodal point.

1. Card 1

Col. 1-4 = NCONC (Format 14).

2. Concentrated nodal force data cards - one for each node at which forces are specified.

a. Col. 1-4 = node at which force is specified.

b. Col. 4-18 = force in x-direction.

c. Col. 19-32 = force in y-direction.

If more than one set of forces are to be applied to structure, prepare separate sets of concentrated force data sets as specified above.

B. Desk assembly

1. Assemble the prepared data sets as shown in figure 17.

N.B. exception - if FROMFILE (control card 2, Col. 13) is specified as 1; omit coordinate data cards from nodal coordinate data set.

2. Precede data with program source deck.
3. Add necessary system control cards to assembled deck.

Figure 17. Final deck configuration for finite element program.

APPENDIX B-2: DESCRIPTION OF MAJOR SUBPROGRAMS

COUNTBLOCKS

PURPOSE Count number of non-zero coefficient blocks in stiffness matrix which will be stored in [BT], i.e. number of nodes greater than or equal to the *i*th node.

CALLED BY Main Program

VARIABLES J,K,L, minimum, intermediate, and maximum nodes of an element.

NPOIN = total nodal points.

[BULARA] = Boolean array used for flagging nodes.

[BZV] = number of non-zero blocks in each row of thickness matrix.

```

ALGORITHM
1. Declare Boolean array NPOIN x NPOIN.
2. Find minimum, intermediate, and maximum nodes of
   element.

```

3. Initialize elements $[J,K]$, $[J,L]$, and $[K,L]$ of

Boolean array to TRUE, e.g.

Elem 1 6 2

[illegible]

4. Go to 2 until all elements done.
5. Count number of TRUEs in each row counting from 2.
6. Accumulate total number of elements in each row in [BZV].

MAKEPT

PURPOSE Construct index array for the condensed stiffness matrix,
[BT].

CALLED BY Main Program

VARIABLES [PT] = index array for [BT].
[COLMWAT] = indicates what column in each row was
 last used for storage.
MUL = minimum nodal point in element.
DT[1],DT[2] = intermediate and maximum node in element.

ALGORITHM 1. Determine minimum, intermediate, and maximum node
 of each element.
 2. Insert minimum node (MUL) in first column of MULth
 row of PT.
 3. Insert intermediate and maximum nodal numbers in
 proper sequential order within MULth row.

NODESONBOUN

PURPOSE Place indices for points on embedded boundary into index matrix PT.

CALLED BY Main Program

VARIABLES PT[i,j] = index matrix for BT.
 BT[i,j] = condensed stiffness matrix.
 LKOPP = element opposing LKth element.
 LK = element index.
 NSTART = first element in last partition.
 NPARTMI = last element minus one.
 BAR[i] = FLAG, indicating node on embedded boundary.
 IMAGE[i] = vector of image nodes corresponding to LKth element.
 BUL = Boolean, FALSE = no center node.
 TRUE = center node.
 CNODE = center node which is held fixed during deformation.
 BZV[i,j] = vector whose ith element contains total number of non-zero block in BT.

METHOD See Appendix B-3.

ALGORITHM See flow chart.

Figure 18. Flow chart for subprogram NODESONBOUND.

Figure 19. Flow chart (continued) for subprogram NODESONBOUN.

NLCONSTANTS

PURPOSE Compute initial equivalent viscosities needed for non-linear problem.

CALLED BY Main Program

VARIABLES

ENERGY[i] = activation energy for creep = Q^* for
 ith material.

$:= A_0 \exp(-Q^*/RT)$.

A[i] = A_0 = diffusion constant for ith material.

TEMP[i] = temperature = $T(^{\circ}K)$ for ith material.

EI[i] = equivalent viscosity of ith material.

METHOD

The non-linear flow law under consideration is

$\dot{\epsilon}_0 = A_0 \exp(-Q^*/RT) \sigma^n$ where

$\dot{\epsilon}_0$ = strain rate at which deformation takes place.

A_0 = diffusion constant.

Q^* = activation energy for creep.

R = gas constant.

T = temperature ($^{\circ}K$).

$\sigma = \sigma_1 - \sigma_2$

n = constant exponent characteristic of material.

At a strain rate $\dot{\epsilon}_0$, the equivalent viscosity is

$\eta = (\dot{\epsilon}_0 / Q^*)^{1/n} / \dot{\epsilon}_0$.

ALGORITHM

1. Compute $A_0 \exp(-Q^*/RT)$.
2. Compute $\{\dot{\epsilon}_0 / A_0 \exp(-Q^*/RT)\}^{1/n} = \sigma$
3. Go to 1 until η_0 has been computed for each different material in the structure.

MAKEBT

PURPOSE Compute stiffness matrices for each element in structure and transfer them to total condensed stiffness matrix, $BT[i,j]$.

CALLED BY Main Program

CALLS TO PROCEDURE FEM
 FEMTRANS
 NONLINEAR
 RITAPE

VARIABLES II = current partition.
 NSTAR = first element of current partition.
 NEN = last element of current partition.
 NFIRS = first node of current partition.
 NLAS = last node of current partition.
 LK = current element.
 [XE2] = vector of nodal coordinates.
 TH = thickness of current element.
 PR1, PR2 = Poisson's ratios in x, y directions in current element.
 YM1, YM2 = viscosity of current element.

METHOD An element stiffness matrix $C[i,j]$ is first generated by PROCEDURE FEM. Then the upper triangular portion is stored in BT by FEMTRANS. BT is the condensed stiffness matrix because it stores only the non-zero entries

in the upper triangular portion of the total stiffness matrix K (see appendix B-2).

ALGORITHM

1. Fill [XE] with coordinates of nodes of LKth element.
2. Initialize:
 - Thickness,
 - Poisson's Ratio
 - Viscositiesfor LKth element (compute viscosities if non-linear problem).
3. Call FEM to compute finite element matrix for element.
4. Call FEMTRANS to transfer finite element matrix to total stiffness matrix.
5. Call RITAPE to store stress matrices.
6. Call RITAPE to store strain matrices.
7. Go to 1 until all elements of this partition done.

NONLINEAR

PURPOSE Call VISCOMP when certain conditions are specified
 and set up viscosities for FEM.

CALLED BY MAKEBT

CALLS TO VISCOMP

VARIABLES MUL = index for time steps of incremental
 deformation

 FROMFILE = Flag for source of data

 EI[i] = Viscosity of ith material

 YMI,YM2 = Viscosity parameter passed to FEM

ALGORITHM 1. Always call VISCOMP to compute new viscosities
 after the first increment of deformation.

 2. If data is restart data (FROMFILE = 1) and
 constants have been calculated for viscosity
 computation then call VISCOMP.

 3. Initialize parameters for FEM.

VISCOMP

PURPOSE Compute viscosities of each element for next deformation increment.

CALLED BY NONLINEAR

VARIABLES

$EI[i]$ = new viscosity to be used next increment of deformation.

 MINSTRAIN = minimum (principal compressive) strain of previous increment.

 ENERGY[i] = Diffusion coefficient = $A \exp(-Q^*/RT)$ for ith material.

METHOD The equivalent viscosity is given by $\eta = \frac{\sigma}{\dot{\epsilon}}$ where

$$\sigma = \left(\dot{\epsilon} / A \exp(-Q^*/RT) \right)^{1/n}.$$

ALGORITHM

1. Determine principal compressive strain.
2. Compute η according to equation given above.

LOCAL

PURPOSE Convert datum coordinates of element to local coordinates
 with origin at centroid of element.

CALLER BY PROCEDURE FEM

VARIABLES [XE2] = Vector containing coordinates of i, j, m
of element

$$XE2[0] = X(i)$$
$$XE2 [i] = Y(i)$$
$$XE2[2] = X(j)$$
$$XE2 [3] = Y(j)$$
$$XE2 [4] = X(m)$$
$$XE2 [5] = Y(m)$$

ORX,ORY = Coordinates of centroid

ALGORITHM

1. Compute centroid of element.
2. Subtract coordinates of centroid from nodes.

STORES

PURPOSE	Store matrices for later use.
CALLED BY	PROCEDURE FEM
CALLS TO	RITAPE
VARIABLES	<p>STORES1 = matrix to be stored.</p> <p>STORES2 = vector in which matrices are temporarily stored.</p> <p>STORES3 = file where matrices are to be stored.</p> <p>STORES4 = counter; points to word in STORES2 in which next word of STORES1 will be stored.</p>
METHOD	Matrices are stored in vector STORES2 until 540 words (30 eighteen-word matrices) have been stored. The vector is then written on file specified by STORES3.
ALGORITHM	<ol style="list-style-type: none">1. Increment counter (STORES4) by 1.2. Store word from matrix (STORES1) in storage vector (STORES2).3. If vector is full, call RITAPE to write vector on file (STORES3).4. Go to 1 until all words of matrix written in vector.

FEMTRANS

PURPOSE Place coefficients from upper triangular portion of
 element stiffness matrix in condensed stiffness matrix.

CALLED BY MAKEBT

CALLS TO FILLBT

VARIABLES BT[i,j] = condensed stiffness matrix.
 C[i,j] = element stiffness matrix computed by FEM.
 PT[i,j] = index array for BT.
 MROWBT = row of BT into which coefficient block
 will be placed.
 NCOLMBT = initial column of BT into which a coef-
 ficient will be placed.
 R,S = row and column of C, (initialized for
 FILLBT).
 CNODE = center node = origin of coordinate
 system in embedded boundary problem.

ALGORITHM 1. Determine row and column of BT into which coefficient
 block will be placed.
 a. Upper triangular portion is stored, i.e. store
 in minimum row, maximum column.
 b. Determine column by searching PT.
 2. Initialize row, column from which coefficients in
 C will come.

3. Call FILLBT to place coefficient block in BT.
4. Go to 1 until all upper triangular portion of C is stored.
5. If this is an embedded boundary problem call BOUNDARYFIX.

FILLBT

PURPOSE Place coefficients of coefficient block from element stiffness matrix into condensed stiffness matrix.

CALLED BY FEMTRANS

VARIABLES BT[i,j] = condensed stiffness matrix.

 C[i,j] = element stiffness matrix computed by FEM.

 MROWBT = row of BT into which coefficients will be placed.

 MJ = initial column of BT into which a coefficient will be placed.

 R,S, = Row and column of C (initialized in FEMTRANS).

METHOD Coefficients of C are added to coefficients in BT.

BOUNDARYFIX

PURPOSE Call LOCATEBT if necessary for embedded boundary problem.

CALLED BY MAKEBT

CALLS TO LOCATEBT

VARIABLES LKOPP = element opposite LKth element in last partition.

 NPARTMI = NPART - 1 = last partition in structure counting from 0.

METHOD If stiffness matrix being computed by MAKEBT is in last partition, LOCATEBT is called. When LOCATEBT is exited, LKOPP is decremented.

LOCATEBT

PURPOSE Search PT for proper location and store imabe nodal coefficients condensed stiffness matrix BT.

CALLED BY BOUNDARYFIX

CALLS TO FILLBTMINUS

VARIABLES

BAR[i] = flag indicating the ith point is on embedded boundary.

IMAGE[i] = image node corresponding to the ith nodal point.

L = indicator of coefficient relating internal node to boundary node.

NOD[LK,J] = nodal elements for element LK, J = 0, 1, 2.

MROWBT = row of BT.

NCOLMBT = column of BT.

PT[i,j] = index array for BT.

BT[i,j] = condensed stiffness matrix.

METHOD

When an element stiffness matrix is generated by FEM, it is stored in BT by FEMTRANS. If the element is in the last partition and if the boundary is embedded the coefficients in the element stiffness matrix are also needed as coefficients for an image element. This image element is located adjacent to an element opposite the LKth element, i.e. the LKOPPth element.

LOCATEBT determines which coefficients must be saved⁷⁷ and locates the place in BT where they must be stored.

ALGORITHM

1. Determine permutation of the nodes of the LKOPPth element. This permutation is a function of the orientation of the LKOPPth element. One node on the boundary, the permutation equals 210. Two nodes on the boundary, the permutation equals 021.
2. If a coefficient block K_{ij} in the stiffness matrix relates an internal node to a boundary node, then search PT to find the proper location for the coefficient. e.g.

Node 0 not affected.) coefficients in coefficient block K_{00} are not stored because they do not relate an internal node to a boundary node. Coefficient blocks K_{01} , K_{11} , K_{02} , K_{22} , K_{12} , K_{22} are stored.

3. Compute BT location.
 - a. Row of BT is the minimum of the two nodes which the coefficients relate.
 - b. Column of BT is four times the image of the large of the two nodes which the coefficient relates.
4. Call FILLBTMINUS to put negative coefficients in their location in BT.

FILLBTMINUS

PURPOSE Put negative stiffness coefficients in condensed stiffness matrix as coefficients for image nodes.

CALLED BY LOCATEBT

VARIABLES

BT[i,j] = condensed stiffness matrix.

C[i,j] = element stiffness matrix computed in FEM.

MROWBT = row of BT into which coefficients will be placed.

MJ = initial column into which coefficients will be placed.

R,S = row and column of C (initialized in LOCATEBT).

METHOD Coefficients of C are subtracted from coefficient in BT.

WIDE

PURPOSE Compute width of structure before each increment
 of deformation.

CALLED BY Main Program

VARIABLES WIDTH = width of structure.
 x[i,0] = x-coordinate of ith point.
 NLAST[i] = last nodal point in ith partition.
 NFIRST[i] = first nodal point in ith partition.
 NPARTMI = last partition counting from 0.

METHOD Subtract the x-coordinate of the first nodal point
 from the x-coordinate of the last nodal point.

FIRSTVEL

PURPOSE Compute the velocity at which the walls of the structure are converging, as specified initially by the input boundary velocities.

CALLED BY Main Program

VARIABLES INIVEL = Initial velocity of convergence as specified by boundary velocities.

 [BV] = specified boundary velocities.

 NBOUNMI = NBOUN - 1.

METHOD Sum the magnitudes of the boundary condition velocities specified for the first nodal point and the last nodal point.

FILLF

PURPOSE Specify load as $K(i,i) \times 10^{12} \times V(s)$ and multiply $K(i,i)$ by 10^{12} . This effectively removes the singularity of the stiffness matrix. K, i.e., specifies the boundary conditions of the problem.

CALLED BY Main Program

VARIABLES

NF[i] = vector of nodes for which velocities are specified as boundary conditions.

NB[i,0] = 0 - x velocity is specified.

NB[i,1] = 0 - y velocity is specified.

WIDTH = current width of structure.

INIVEL = initial velocities specified by input.

BT = condensed stiffness coefficient matrix.

BV[i,0] = specified x-velocity, ith node.

BV[i,1] = specified y-velocity, ith node.

f[i,0] = x-load at ith node.

f[i,1] = y-load at ith node.

VEL = current velocity for boundary conditions.

ALGORITHM

1. If velocity at boundary point is specified then
2. compute boundary velocity appropriate for that increment of deformation = $V(s) \times \frac{\text{width}}{\text{initial velocity}}$.
3. Multiply $K(i)$ by 10^{12} .
4. Initialize $f(i)$ to $K(i,i) \times 10^{12} \times V(s)$.

SOLVER

PURPOSE Solve simultaneous equations arising from finite element analysis which stores condensed stiffness matrix (see Appendix B-2).

CALLED BY Main Program

CALLS TO USL, USR, VSL, VSR, SOLVEX, SOLVEY

VARIABLES

COUNT = counts number of iterations.

COLMWAT[i] = column indicator = points to next column to be interrogated in searching upper triangular portion BT for coefficients in lower triangular portion of stiffness matrix.

PT[i,j] = index array for BT.

BT[i,j] = condensed stiffness matrix.

MJ = row pointer = points to initial row of BT to begin interrogation for coefficients for lower triangular portion of stiffness matrix.

BZV[i] = number of columns in PT.

BAR[i] = flag indicating ith point is on embedded boundary.

EPSILON = current residual error velocities.

EPS = error tolerance allowed.

ITTER = maximum number of iterations allowed.

METHOD Young's Overrelaxation Method

ALGORITHM

Since only the upper triangular portion of the stiffness matrix is stored, coefficients required from that portion are accessed in a straight forward manner, but the coefficients from the lower triangular portion of the matrix must be obtained by searching for their symmetrical counterparts in the upper triangular portion. Since $[K_{ij}] = [K_{ji}]$, a symmetric counterpart of a coefficient block from the i th row and the j th column in the lower triangular portion is stored in the j th row and the i th column of the stiffness matrix.

To find the column in which a coefficient block is actually stored, the index matrix must be searched in the j th row until the column containing the i th index is located.

The algorithm follows:

1. Initialize counter.
2. Increment counter.
3. Search index array for location of coefficients

$$K[i,j] \quad j = 1 \dots i - 1.$$

4. Form and sum the products;

$$\sum_{j=1}^{i-1} K_{ij}^{xu} \quad U_j \quad \sum_{j=1}^{i-1} K_{ij}^{xv} \quad V_j$$

(USL)

5. Search index array for location of coefficients

$$K[i,k] \quad k = i = 1 \dots N.$$

6. Form and sum the products:

$$\sum_{k=i+1}^N K_{ik}^{xu} U_k \quad \sum_{k=i+1}^N K_{ik}^{xv} V_k$$

(USR)

7. Solve for U_i (SOLVEX).
 8. Search index array for location of coefficients

$$K_{ij} \quad j = 1 \dots i - 1.$$

9. Form and sum the products:

$$\sum_{j=1}^{i-1} K_{ij}^{yu} U_j \quad \sum_{j=1}^{i-1} K_{ij}^{yv} V_j$$

(VSL)

10. Search index array for location of coefficients

$$K_{ik} \quad k = i + 1 \dots N.$$

11. Form and sum the products:

$$\sum_{k=i+1}^N K_{ik}^{yu} U_k \quad \sum_{k=i+1}^N K_{ik}^{yv} V_k$$

(VSR)

12. Solve for V_i (SOLVEY).
 13. Go to 3 until all ith row done.
 14. Go to 2 until error is within tolerance desired of
 until desired number of iterations have been done.
 15. Go to 1 until all sets of forces have been
 completed.

MODIFICATION For the embedded boundary problem, the preceding algorithm is modified by the addition of the following steps.

- 6a. If i th node is on embedded boundary, then search matrix for negative index.
- 6b. Form and sum the products

$$\sum -K_{ij}^{xu} -U_j$$

$$\sum -K_{ij}^{xv} -V_j$$

j ranges over all image nodes connected to i th node.

(USR)

- 11a. If i th node is on embedded boundary, then search index matrix for negative index.

12. Form and sum the products

$$\sum -K_{ij}^{yu} -U_j$$

$$\sum -K_{ij}^{yv} -V_j$$

j ranges over all image nodes connected to i th node.

(VSR)

USL, USR, VSL, VSR

PURPOSE

Form the summations needed for Young iteration scheme:

$$Q(i)^{n+1} = R[\{F(i) - K(i,k) Q(k)^{n+1} - K(i,m) Q(m)^n / K(i,i)\} - Q(i)^n] + Q(i)^n.$$

$$Q(i) = U(i) = \text{x-velocity of } i\text{th nodal point or}$$

$$= V(i) = \text{y-velocity of } i\text{th nodal point.}$$

$$R = \text{Overrelaxation factor.}$$

$$F(i) = X(i) = \text{x-force on } i\text{th nodal point or}$$

$$= Y(i) = \text{y-force on } i\text{th nodal point.}$$

$$i = 1, \dots, N.$$

$$k = 1, \dots, i - 1.$$

$$m = i + 1, \dots, N.$$

$$N = \text{total nodal points.}$$

$$n = \text{iteration.}$$

VARIABLES

$$BT[I,J] = \text{condensed matrix of non-zero elements of stiffness matrix.}$$

$$Q[I] = \text{vector of horizontal (u) and vertical (v) nodal velocities.}$$

METHOD

Rewrite Young iteration scheme as

$$U(i)^{n+1} = R[\{F(i) - (XUSUMLEFT + XVSUMLEFT) - (XUSUMRIGHT + XVSUMRIGHT) / K(i,i)\} - U(i)^n] + U(i)^n.$$

$$Y(i)^{n+1} = R[\{F(i) - (YUSUMLEFT + YVSUMLEFT) - (YUSUMRIGHT + YVSUMRIGHT) / K(i,k)\} - V(i)] + V(i)^n.$$

USL forms:

$$XUSUMLEFT = K(i,k)U(k)$$

$$XVSUMLEFT = K(i,k)V(k)$$

USR forms:

$$XUSUMRIGHT = k(i,m)U(m)$$

$$XVSUMRIGHT = k(i,m)V(m)$$

VSL forms:

$$YUSUMLEFT = k(i,k)U(k)$$

$$YVSUMLEFT = k(i,k)V(k)$$

VSR forms:

$$YUSUMRIGHT = K(i,m)U(m)$$

$$YVSUMRIGHT = K(i,m)V(m)$$

SOLVEX, SOLVEY

PURPOSE Solve for X and Y velocities in Young's overrelaxation method.

CALLED BY SOLVER

VARIABLES XUSUMLEFT YUSUMLEFT
 XVSUMLEFT YVSUMLEFT
 XUSUMRIGHT YUSUMRIGHT
 XVSUMRIGHT YVSUMRIGHT
 (see derivation of Young's overrelaxation method for definitions).

EPSILON = maximum error in U or V during this iteration.

ALGORITHM

1. Combine XUSUMLEFT and XVSUMLEFT or YUSUMLEFT and YVSUMLEFT.
2. Combine XUSUMRIGHT and XVSUMRIGHT and
3. Form $\Delta U(i) = F(i) - XUSUMLEFT - XUSUMRIGHT - U(i)$
 or $\Delta V(i) = F(i) - YUSUMLEFT - YVSUMRIGHT - V(i)$.
4. Form $U(i) = U(i) + R \Delta U(i)$ or
 $V(i) = V(i) + R \Delta V(i)$.
5. Determine error = EPSILON = $M \times (\text{EPSILON}, |\Delta Q(i)|)$

INCREMENT

PURPOSE Increment coordinates of nodal points to their new position traveling at velocity of deformation for a time interval sufficient to deform structure by a specified strain.

CALLED BY Main Program

VARIABLES LOGSTRAIN = true (logarithmic) strain to which structure is deformed in each time interval.

MUL = current increment of deformation.

YEARS = time in years required to achieve total strain to this point.

DELTEE = Δt = time interval during which nodal points move.

STRAINRATE = total shortening strain rate specified for structure.

ST[i,mm] = storage vector for displacement due to m^{th} load vector.

Q[i] = velocity of i^{th} nodal point.

X[i,j] = coordinate of i^{th} nodal point.

ALGORITHM

1. Compute time in years required to reach present deformation.
2. Compute displacement in x and y direction of each nodal point. (Displacement = velocity x time).
3. Add displacement to coordinates of each point.

COMPSTRESS

PURPOSE	Compute stresses or strains at centroid of each element.
CALLED BY	Main Program
CALLS TO	PROCEDURE STRESS PROCEDURE PRINC
VARIABLES	<p>[SAV] = vector into which stress or strain matrices are read from disk file.</p> <p>DELSIG[LK] = differential stress = $\sigma_1 - \sigma_2$ for LKth element.</p> <p>FM = counter, counts words used in [SAV].</p> <p>NELEM = number of elements.</p> <p>SE[i,j] = stress (strain) matrix.</p> <p>P[i] = velocity vector computed by main program.</p> <p>DT[i] = vector of stresses [strains] passed to PRINC.</p> <p>ELEMSTR = principal strains for LKth element.</p>
METHOD	<ol style="list-style-type: none"> 1. Read 540 words from file. 2. Check to see if 540 words have been used. If so, read 540 words from file. 3. Make 3 x 6 stress (strain) matrix. 4. Call procedure to compute stress (strain) tensor. 5. Compute differential stress (DELSIG). 6. Write $\sigma_1, \sigma_2, \theta, \sigma, \sigma_x, \sigma_y, \tau_{xy}$ ($\epsilon_1, \epsilon_2, \theta, \sigma_1 - \sigma_2, \epsilon_x, \epsilon_y, \gamma_{xy}$)

7. Write principal stresses (strains) on file for
Plotter program.
8. Ready principal strains for viscosity computation.

LOAD

PURPOSE Compute the concentrated nodal forces equivalent to body forces on structure.

CALLED BY Main Program

CALLS TO DET

VARIABLES VOL = volume of element.
 DELTA = $2 \times$ area of element.
 UT = concentrated equivalent force in x-direction.
 VT = concentrated equivalent force in y-direction.
 F[i,j] = concentrated forces on structure.
 X[i,j] = nodal coordinates.
 XE2[i] = nodal coordinates of element.

ALGORITHM 1. Zero force vector.
 2. Place nodal coordinates in vector form.
 3. Compute volume of element.
 4. Compute even distribution of force per unit volume at each node element.
 5. Add nodal forces to force vector.
 6. Go to 2 for all elements in structure.

FEM

PURPOSE Compute element stiffness matrix.

CALLED BY MAKEBT

CALLS TO LOCAL

 DET

 MATM

 STORES

VARIABLES DEN = $1/2 \times$ area of element.

 B = displacement matrix.

 XE2 = element nodal coordinates.

 DT = D = elasticity matrix.

 SE = BD = stress matrix.

 WANTSTRAIN = 1 = element strains wanted in output.

 WANTSTRESS = 1 = element stresses wanted in output.

 VOL = volume of element.

 C = element stiffness matrix.

METHOD Element stiffness matrix $C = B^T D B \times VOL$.

ALGORITHM 1. Convert global coordinates to local coordinates.

 2. Make displacement matrix B .

 3. Make elasticity matrix for plane strain or plane stress case.

 4. Make stress matrix.

 5. If strains are wanted in output, store B .

6. If stresses are wanted in output, store B.
7. Make stiffness matrix C.

STRESS

PURPOSE Compute elemental stresses (strains) from nodal velocities.

CALLED BY COMPSTRESS

VARIABLES SE = stress (strain) matrix, (see chapter 11).
 Q = nodal velocities.
 QT = element nodal velocities.
 ELEMSTR = elemental stresses (strains).

METHOD

$$\begin{aligned} \sigma_1 \\ \sigma_2 \end{aligned} &= [B] [D] \{U\}^e.$$

$$\tau_{xy}$$

where

$$[B] \quad = \quad \text{stress matrix.}$$

$$\{U\}^e \quad = \quad \text{element nodal velocities.}$$

$$\begin{aligned} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{aligned} &= [B] \{U\}^e.$$

$$[B] \quad = \quad \text{displacement matrix.}$$

PRINC

PURPOSE Compute principal stresses (strains) and principal angle from x, y, and shear stresses (strains).

CALLED BY COMPSTRESS

VARIABLES D = x, y, and shear stresses (strains).
 FLGG = 0 - stresses
 = 1 - strains.

$$\text{RUTE} = \left[\left(\frac{\sigma_x - \sigma_y}{2} \right)^2 + (\tau_{xy})^2 \right]^{1/2} \quad \text{or} \quad \left[\left(\frac{\epsilon_x - \epsilon_y}{2} \right)^2 + (\gamma)^2 \right]^{1/2}$$

$$\text{MEEN} = \frac{\sigma_x + \sigma_y}{2} \quad \text{or} \quad \frac{\epsilon_x + \epsilon_y}{2}$$

METHOD σ_1 = MEEN + RUTE
 σ_2 = MEEN - RUTE

$$\theta_{\tau xy} = 1/2 \tan^{-1} \left(\frac{2\tau_{xy}}{\sigma_1 - \sigma_2} \right)$$

$$\theta_{\gamma xy} = 1/2 \tan^{-1} \left(\frac{\gamma_{xy}}{\epsilon_1 - \epsilon_2} \right)$$

APPENDIX B-3: SYMMETRY USED IN PROGRAM

Due to the symmetry of the problem (see figure 5) only one fourth of the structure is given as data to the program. Two types of symmetry are used. First, only one half the fold wavelength must be studied since the other half will be symmetrical about line ABC.

Secondly, a point of symmetry exists at 0, the origin of the datum coordinate system. If this point is held fixed as a boundary condition and sides ABC and DEF are converging toward point 0 at the same rate, a point at P' will be affected by a set of forces equal but opposite to those affecting a point at P. Therefore, the only portion of the total field that must be studied is ABCODEF. However, to make the structure effectively behave as if the section below line COD in figure 5 were present, the stiffness matrix must be modified to include coefficients for nodal points below the boundary COD. Because line COD does not represent a physical boundary at all, but a surface embedded in the layer, it is referred to as an embedded boundary. The nodes below the embedded boundary are called image nodes since they are generated by a center of symmetry operation at 0, and the elements below the embedded boundary are called the image elements.

When the stiffness matrix is modified by the inclusion of image node coefficients, the coefficients are flagged by entering their index as a negative number in the index matrix. Thus, when nodal points on embedded boundary are encountered, the index matrix is searched not only for the location of actual nodal point coefficients but also for image nodal point coefficients.

As an example, suppose that in the figure below, the nodes in the structure are numbered as shown. The primed nodal points are image nodal points generated through the symmetry nodal point, 25.

The image nodes affecting node 28 are 15' and 16'.

The element stiffness matrix for image element 28 16' 15' is the same as that for element 15 22 16, except that the coefficients are opposite in sign because the corresponding nodal points are moving in opposite directions, e.g. if node 15 moves right and up, node 15' moves left and down. When the velocities for node 28 are computed, coefficients are used not only for nodes 20, 21, and 27, but also for nodes 16' and 15'. The nodal velocities which affect node 28 are Q_{20} , Q_{21} , Q_{27} , $-Q_{15}$, $-Q_{16}$.

The algorithms in the program which modify the stiffness matrix and the index matrix are LOCATEBT and NODEONBOUND respectively. The Young's Overrelaxation Method algorithm, SOLVER, has also been modified to search for the image coefficients when solving the simultaneous equations.

APPENDIX B-4: YOUNG'S OVERRELAXATION METHOD

The set of equations to be solved is:

$$(1) \quad \{F\} = [K] \{Q\}$$

$\{F\}$ = set of nodal forces.

$\{Q\}$ = set of nodal velocities.

$[K]$ = stiffness matrix.

$$\{F_i\} = \begin{Bmatrix} X_i \\ Y_i \end{Bmatrix} = \text{forces on the } i\text{th nodal point.}$$

$$\{Q_i\} = \begin{Bmatrix} U_i \\ V_i \end{Bmatrix} = \text{velocities of the } i\text{th nodal point.}$$

The forces on the i th nodal point are:

$$(2) \quad \{F_i\} = [K_{ij}] \{Q_j\}$$

$i = 1, \dots, N.$

N = total nodal points.

$[K_{ij}]$ = 2×2 submatrix of K .

Equation 2 can be rewritten as the sum of three terms:

$$(3) \quad \{F_i\} = \sum_{j=1}^{i-1} [K_{ij}] \{Q_j\} + [K_{ii}] \{Q_i\} + \sum_{k=i+1}^N [K_{ik}] \{Q_k\}$$

Substituting L and R for the left and right terms on the right side of the equation 3:

$$\{F_i\} = L + [K_{ii}] \{Q_i\} + R.$$

Solving for Q_i :

$$(4) \quad [K_{ii}] \{Q_i\} = \{F_i\} - L - R.$$

When written in its full form, the matrix equation (4) is:

$$K0 U_j + K1 V_i = X_i = \sum_{j=1}^{i-1} (K_{ij}^{xu} U_j + K_{ij}^{xv} V_j) - \sum_{k=i+1}^N (K_{ik}^{xu} U_k + K_{ik}^{xv} V_k)$$

$$K2 U_i + K3 V_i = Y_i - \sum_{j=1}^{i-1} (K_{ij}^{yu} U_j + K_{ij}^{yv} V_j) - \sum_{k=i+1}^N (K_{ik}^{yu} U_k + K_{ik}^{yv} V_k)$$

Where:

K0	K1
K2	K3

=

K_{ii}^{xu}	K_{ii}^{xv}
K_{ii}^{yu}	K_{ii}^{yv}

$$\text{Letting } XUSUMLEFT = \sum_{j=1}^{i-1} K_{ij}^{xu} U_j$$

$$XVSUMLEFT = \sum_{j=1}^{i-1} K_{ij}^{xv} V_j$$

$$XUSUMRIGHT = \sum_{k=i+1}^N K_{ik}^{xu} U_k$$

$$XVSUMRIGHT = \sum_{k=i+1}^N K_{ik}^{xv} V_k$$

and solving for U_i :

$$U_i = (X_i - (XUSUMLEFT + XVSUMLEFT) - (XUSUMRIGHT + XVSUMRIGHT) - K1 V_i / K0).$$

Using analogous notation and solving for V_i :

$$V_i = (Y_i - (YUSUMLEFT + YVSUMLEFT) - YUSUMRIGHT + YVSUMRIGHT) - K2 U_i / K3).$$

U_i and V_i can be solved for iteratively by starting with an initial estimate of U_i, V_i for all i and computing successively better estimates. Convergence on true values of U_i and V_i is achieved when the new estimates of U_i^{n+1} or V_i^{n+1} differ little from the old values,

U_i^n, V_i^n , i.e. when ΔU_i or ΔV_i approaches zero, where $\Delta U_i = U_i^{n+1} -$

$U_i^n = (X_i - (XUSUMLEFT + XVSUMRIGHT) - (XUSUMRIGHT + XVSUMRIGHT) - K1 V_i / K0) - U_i^n$, and ΔV_i is given by an analogous equation.

The rate of convergence is usually increased by using an overrelaxation factor, RFACT, in the following manner.

$$U_i^{n+1} = (RFACT \times \Delta U_i^n) + U_i^n$$

$$V_i^{n+1} = (RFACT \times \Delta V_i^n) + V_i^n$$

$$-2 < RFACT < +2$$

Using these equations in an iterative manner to solve for U_i and V_i , is Young's Overrelaxation Method. The relaxation factor used in the computer program was 1.73.

APPENDIX B-5

CATALOG OF MAJOR VARIABLES USED IN PROGRAM

INTEGERS

BOUNDARY	= FLAG, NEQ 0 indicates this nodal point is the origin of coordinates problem with an embedded boundary.
CNODE	= nodal point which is origin in embedded boundary problem.
DEFREE	= total degrees of freedom in structure.
DFRMI	= DEFREE - 1.
FM	= file mark, points to word in vector [SAV] (STORE_ STRESS COMPSTRESS).
FMARK	= counter.
FROMFILE	= FLAG, determining file from which data will come.
IMAGENODE	= image of node through center of symmetry.
ITTER	= maximum number of iterations to be run for Young's Overrelaxation Method.
LK	= usually used to index elements.
LKOPP	= element opposite LKth element in last partition.
NBOUN	= number of boundary conditions.
NBOUNMI	= NBOUN - 1.
NCOLN	= number of different sets of nodal forces applied to structure.
NCOLNMI	= NCOLN - 1.
NCONC	= number of nodes at which concentrated.
NCONCMI	= NCONC - 1.

NELEM = number of elements.
 NELEMMI = NELEM - 1.
 NEN = last element in partition (MAKEBT).
 NFIRS = first node in partition, (MAKEBT).
 NFREE = degrees of freedom per node.
 NLAS = last node in partition, (MAKEBT).
 NPART = number of partitions.
 NPARTMI = NPART - 1.
 NPOIN = number of nodal points.
 NPOINMI = NPOIN - 1.
 NPROB = number of problems.
 NSTAR = first element in partition (MAKEBT).
 NTIMES = number of deformation increments in problem.
 NYM = number of different materials in structure.
 NYMMI = NYM - 1.
 PARTWIDTH = width of maximum partition.
 PTMAX = maximum number of columns needed for PT.
 WHICHFIEL = FLAG, indicating restart data to be saved.

REALS

ENERGY = temporary storage of $\dot{Q}^*$.
 EPS = tolerance specified by user for error in velocities $\dot{Q}$.
 EPSILON = current error in velocities.
 DELTEE = Δt = time increment specified by user for each deformation.
 FPVX = force per unit volume in x-direction.
 FPVY = force per unit volume in y-direction.
 G = shear viscosity.

INIVEL = initial velocity of converging walls as specified by user.
 LOGSTRAIN = $\int \frac{dx}{x}$ = logarthemic strain allowed for each increment of deformation.
 PRI, PR2 = current value for Poisson ratio in x and y direction.
 RFACT = overrelaxation factor for Young's overrelaxation method.
 STRAINRATE = specified strain rate at which deformation is to take place.
 TEMP = temperature.
 TH = thickness of element.
 YMI, YM2 = current value for viscosity in x and y direction.
 XUSUMLEFT
 XUSUMRIGHT
 XVSUMLEFT
 XVSUMRIGHT
 YUSUMRIGHT
 YUSUMLEFT
 YVSUMLEFT
 YVSUMRIGHT

See Appendix B-4

ARRAYS

A = diffusion constant.
 AN = angle of anorthoropy width datum coordinate system.
 B = element displacement matrix.
 BAR = 1 = node of embedded boundary.
 BV = boundary condition velocities.
 BZV = number of non-zero coefficient blocks.

C	= element stiffness matrix.
DELSIG	= $\sigma = \sigma_1 - \sigma_2$ = differential stress.
E1,E2	= Young's modulus or viscosities in x, y.
ELEMSTR	= principal strains in element.
ENERGY	= Q^* = activation energy for creep.
F	= set of applied concentrated forces.
GE	= shear modules in anorthotropic material.
NB	= indicator of boundary condition specified in x or y direction.
NEND	= last element of each partition.
NEP	= integer vector containing indications of which material properties apply to each element.
NF	= nodes at which boundary velocity is specified.
NFIRST	= first node of each partition.
NLAST	= last node of each partition.
NOD	= nodal numbers of each element.
NSTART	= first element of each partition.
P, Q, LQ	= nodal velocities.
PI,P2	= Poisson ratios in x, y.
POWER	= exponent of differential stress = N.
QT	= nodal velocities stored in vector form.
SAV,SV	= storage vectors (buffers) for stress and displacement matrices.
SE	= element stress matrix.
ST	= storage vector for displacements due to each set of forces on system.
TEMP	= temperature.

THICK = thickness of element.
X = coordinates of nodal points.
XE2 = nodal coordinates stored in vector form.

APPENDIX C

LIST OF COMPUTER PROGRAM

AND

SAMPLE OUTPUT

 COMMENTS: *****
 A COMPATIBLE ALGOL PROGRAM WHICH PERFORMS COMPUTATIONS FOR
 FINITE ELEMENT ANALYSIS OF NON-NEWTONIAN VISCOUS SOLIDS.
 AUTHOR: DAVID PARRISH, DEPT. OF GEOLOGY, RICE UNIVERSITY
 REFERENCE: ZIENKIEWICZ, FINITE ELEMENT METHOD IN STRUCTURAL
 AND CONTINUUM ANALYSIS.

COMMENT: * * OUTERMOST BLOCK, DECLARATION OF FILES, INTEGERS, REALS,
RUNCLEANS, FORMATS, STATIC ARRAYS, AND DEFINES.

```

      LABEL
      FILE IN
      FILE OUT
      SAVE FILE
      FINISH!
      CASH (2,10);
      LINE 15 (2,17);
      PLUTS DISK SERIAL (20:352) *PLUTS*/MR2J001W
      (13,096,CASH 3);
      COMMENT: 33P REC = 996 WDS =

```

```

ONE RECORD FOR (NITEM,NPOIN,NTIMES)
ONE REC FOR (ICSTRAIN,STRAINRATE,DETECT)
PLUS 204 RECS FOR ELEMENTS
PLUS 125 RECS FOR CONJOINED POINTS
PLATE DISK SERIAL (20120611*PLATE*W*230011"
SAVE FILE

```

```

SAVE FILE
COURNF2 DISK SERIAL (2:100) "COURNF1"/"PRESTART"
(2,3,100,SAVE 3)
SAVE FILE
INCNORF2 DISK SERIAL "COURNF1"/"PRESTART"

```

SAVE FILE
INCDORUF2 DISK SERIAL "C77KDF1"/"RFESTART"

```
SAVE FILE
(23,300);
DELSTG RTSK SERIAL [1307] "DELSTGFW/RESTART"
(23,300,SAVE 3);
INDELSTG RTSK SERIAL [1307] "DELSTGFW/RESTART"
(23,300);
```

```
SAVE FILE      INDFLSIG NISK SERIAL "DEL.SIGF"/"RESTART"
                (2,3,300,SAVE 3))
SAVE FILE      WFILE DISK SERIAL (1:1) "WFILE"/"RESTART"
```

```

SAVE FILE
      (2,300,300)SAVE 3)
      INQFILE DISK SERIAL "JFILE"/"RESTART"
      (2,300,300):

```

```
SAVE FILE
(2,300,300);
VISCFL DISK SERIAL [11] "VISCFL"/"RESIART"
(2,300,300,SAVE 3);
SAVE FILE
```

```
SAVE FILE INVISCFI NISK SERIAL "VISCFI"/"RESTART"
(2,300,300)I
FILE STRSS DISK SERIAL (2011) (2,540)I
```

```

FILE
  STRAIN DISK SERIAL [2011] (2,540)
  FILE
  SWITCH
  SWITCH FILE
  STRAINORIN := STRESS, STRAIN;
  COORDIN := COORDP, INCOORDP;
  SWITCH FILE
  SWITCH FILE
  OSTIMATE := OELSLAP, INDELST;
  SWITCH FILE
  INQ := OFFICE, INQFILE;
  SWITCH FILE
  VISC := VISCDEF, INVISCDEF;
  SWITCH FILE
  PLATFILE := PLUTS, PLUTER;
  SWITCH FILE
  INTELFILE := INTEL, SCSUT;
  SWITCH FILE
  INTEGER

```

[illegible][illegible]

WPP, LINEAR;
AVYTTTLFC~811J;

CTURDFOPMT(2R14.6,X16,I4).
SGLVEPARAM(R5.3,R12.5,I4).

LEMFQRM 1314X16X16.8, 19).

ECCEFCRMJ(314,16,8,14),
RRUNFCRMJ(314,2,16,8,14),
PARTFCRMJ(414).

BARTEFFERT (414),
COLAFORT (14, 2K10.0),

COLNFRMT2(2R14.6),
SYMFRNRT(=R14.9),

IMPERIAL (1949),
IMPERIAL (1949),
IMPERIAL (1949),

```

-IWHFR(15,414),
-NCARD(14);

```

```

PROGRAM('PROBLEM NUMBER'14.03,1246 /),
PLANSTRESS('PLANE STRESS PROBLEM' /),

```

PLASTSTRAN, PLANE STRESS PROBLEM/),
PLASTSTRAN, PLANE STRAIN PROBLEM/),
INPPROB, LINEAR PROBLEM/).

```

      CALL PUR(1, LINEAR PROBLEM/),
      CALL INPPROB("NON-LINEAR PROBLEM/"),
      STRESS (2, 10.6).

```

```

PASS (R14,6),
ORDERFORMN(14,2R14-6,X16,14),

```

COORDINATES/NUMBER
Y=X26, RAR)

```
ESTRTC// "DATA COMING FROM DISK FILE."
"COORDINATE LISTING SUPPRESSED."
```

APPROXIMATE LISTING SUPPRESSED.
 ITIE1(X12,"NODAL",X14,"FUNDARY",X4,"
 D-D C- REF- Y3-BASIC Y2-RECORD

D.O.F. PFK", X3, "NUDAL", X3, "FORCE PER
PARTITIONS POINTS ELEMENTS CONDITI

3.5 IN Y² M NO. 1 POINT FORCES VOLUME I
1.5 IN Y² M

[illegible]

Q, X3, 2F170.61),
ITIEP(= DELTA TIME
NEDD... 2... 1...
4TIMES),

NFP2(E11.20X40IQ/),
IFHE3(CONSTANT AND X-AXIS-ENERGY)

TEMPERATURE POWER),
NF03(3E10.5,14),

LEFORMO(414,X16,H16.8,JA),

Q1, INFO(,X1, "CONCENTRATE" NODAL FORCE
FX",X12,"FY"),

YMINF06/Y417, "M A T F R I A L C D N
A, "V I S C O S I T I E S "X5, "F O I S

T T D SW/ NEP XR, XR, XI, XY, XI, X
 SHEAP WOTULUS.

```
TIMEFORMAT('PROCESSOR TIME = "F10.4"'
```

```
LEWFC4026/"ELEMENTS"/"ELEMENT",X5,"NODE
THICKNESS",X3,"WFL"/X7,"WFL",X3,"J",X3,
```

UNIFORM(314,2R16.8,14),
UNIFORM(2("BOUNDARY CONDITIONS"/"NOO

UNOFFICIAL/BOUNDARY CHANGES//NOO

ARTFORMN(2,I4,X1,I4,X3,Y4,X2,I4),
ARTFORMN(/"PARTITIONS"/"ELEMENTS"-

START FWD FIRST LAST,
ONLNFUWU(14,2K14.6),

DLKFFQRMN? (2R14.6),
YMFQRKOC (14.5SR14.5).

```

VELFORMC/MODE VLOCITIES/X8,"U",X14,"U"),
VELFORMZ(14,2R14,6),
DUMFORMC(14,6),
DUMFORMZ(14,6),
DEFORMATION/RELAXATION FACTOR=X3,"FPSILON",X5,
ITERATIONS/X6,R5,X3,X6,R12,X3,X14,/,
STRAINFORMC/X3,"STRAIN KAPPA",X4,"LOG STRAIN"/2014,6/,
FLURC("YOU DUMMY, INSUFFICIENT COORDINATE DATA."/),
FLUPC("YOU IDIOT, INSUFFICIENT ELEMENT DATA."/),
FLIFC("GOOD GRIFF, CHECK YOUR BOUNDARY CONDITIONS."/),
FLURC("INSUFFICIENT PARTITIONING DATA."/),
FLUPC("INSUFFICIENT MODULI, SHEAR GRIFF."/),
FLURC("INSUFFICIENT LOAD VECTOR DATA."/),
PRINC2(14,X1,X14,5),
VISFORMC/EQUIVALENT VISCOSITY FOR ELEMENT",
14,"=X14,6),
DELTIMEFORMC/"TIME FOR",X5,X2," PERCENT DEFORMATION",
"=X14,3," YEARS"/,/,
FRONTTILE ("FROM FILE IS",14),
PRINTSTRESS("STRESSES WILL BE COMPUTED."/),
PRINTSTRAIN("STRAINS WILL BE COMPUTED."/),
TIPROC("TIPERATION =14),
FINALC("AM A STUPID MACHINER."/),
STRITILEC("ELEMENT",X6,"SIGMA 1",X7,"SIGMA 2",X8,
"THETA",X9,"DELSIG",X7,"SIGMA-X",X7,"SIGMA-Y",X6,
"SIGMA-X",X6),
("ELEMENT",X6,"STRAIN=1",X8,"STRAIN=2",X10,"THETA",X5,
"STRAIN DIFF",X2,"STRAIN-V",X4,"STRAIN-Y",X6,
"STRAIN-X",X6),
FORMFORMATC ("CALL DATA READ FROM CARDS."/),
("CALL DATA READ FROM DISK FILE."/),
("COORDINATE DATA READ FROM CARDS."/),
("OTHER RESTART DATA READ FROM DISK."/),
("RESTART DATA WILL NOT BE SAVED."/),
("RESTART DATA WILL BE SAVED."/)

```

SWITCH FORMAT

SWITCH FORMAT

SWITCH FORMAT

COMMENT: ** * * * * *

INDEX TO DEFINES IN GENERAL ORDER IN WHICH THEY OCCUR.
 INDENTATION INDICATES NESTING.

COUNTBLOCKS
 MAKEPT
 NODESONHORN
 NLCONSTANTS
 MAKEBT
 NONLINEAR
 VISCOMP
 LOCAL (NESTED IN PROCEDURE FEM)
 STRESSSTRESS (NESTED IN PROCEDURE FEM)
 FEMTRANS

FLERT
 BOUNDARYFX
 LOCATER
 FILTHMINUS
 COMRINE
 WIDE
 FIRSTVEL
 FILLF
 SOLVER
 USE
 VSL
 USR
 VSR

SOLVEX
 SOLVEY
 INCREMENT
 COMPMSTRESS
 UTILITY PACKAGE

* * * * *


```

NODE FROM BEING ENTERED IN THE ... WHEN ... * * * * *
INSERT NODE: FORW(MMJJ,MHJ),PT(NROW,MHJ) NEQ 1023))
FORW(MMJJ,MHJ),-1,MHJ) PT(NROW,MHJ)
PT(NROW,MHJ) := -NCOLM;
BYVIMROW := BYVIMROW + 1;
END;
MOD NODE: LKOP := LKOPP-1;
END OF ALL ELEMENTS IN LAST PARTITION;
COMMENT: * PUT IMAGE NAMES OF BOUNDARY NODES IN LAST COLUMN OF [PT];
FORW(1,0,1,RAR(1) NEQ 1)
BOUNDARY := 1; L:=NROW; FORW(1,BOUNDARY,1,NROW);
BEGIN PT(1,PTMAX); L:=1; END;
END;
** IF AN EUL NODE IS FOUND, THEN SKIP TO NEXT NODE UNLESS
THE NODE IS THE ONE OF THE CENTER ELEMENTS. IN THAT CASE
CHECK TO SEE IF -NCOLM HAS ALREADY BEEN INSERTED INTO [PT].
IF SO, SKIP TO NEXT NODE TO BE INSERTED (I.E. GO TO MOD NODE).
OTHERWISE INSERT -NCOLM INTO [PT]. THIS IS TO KEEP CENTER

```



```

NONLINEAR = COMMENT: NONLINEAR BLOCK DECIDES WHETHER TO COMPUTE
              VISCOSITIES FOR NON-LINEAR PROBLEM.
              IF THIS IS THE FIRST INFORMATION INCREMENT,
              THEN DO NOT COMPUTE NEW VISCOSITIES.
              IF THIS RUN USES RESTART DATA AND AN(1)
              INDICATES CONSTANTS FOR VISCOSITY COMPUTA-
              TION HAVE BEEN COMPUTED, THEN COMPUTE
              VISCOSITIES. * * * * *
BEGIN
  IF M(1) * 1 OR (PROMFILE=1 AND AN(1)=NEW 0) THEN
    VISCOMP: COMMENT: COMPUTE VISCOSITIES. * * * * *
    YM1 := E1(J)
    YM2 := E2(J)
    VISCOMP = COMMENT: E1 IS NEW TENSILE VISCOSITY TO BE USED
              THIS INCREMENT OF DEFORMATION. SHEAR VISCOSITY IS
              TENSILE VISCOSITY/2*(1+POISSONS RATIO). * * * * *
    BEGIN
      MINSTRAIN:=ABS(MIN(ELEMS19(LK,0),ELEMS19(LK,1)))
      E1(J):=((MINSTRAIN/ENERGY(J))+(1/POWER(LJ)))
      MINSTRAIN:=
      WRITE(LINE, VISCFORMD,LK+1,E1(J)/(2*(1+PR1)))
    END
  END

```

```

FENTRANS = REAL*4
COMMENT: * * * PUT STIFFNESS COEFFICIENTS IN ARRAY [BT]. *
END*
ETV:
COMMENT: * * * IF CNOOF NEG 0 THEN BOUNDARY FIXI
      CAUSE IS CENTER OF SYMMETRY REQUIRING EXTRA COEF-
      FICIENTS TO BE STORED. * * * * * * * * * * * * * * * *
END OF FEM TRANSFER BLOCK.

```

```

STORES
      (STORE1,STORE2,STORE3,STORE4)
      COMMENT: STORE MATRICES FOR LATER USE.
      STORE1 = WITHIN TO BE STORED
      STORE2 = VECTOR IN WHICH TO STORE MATRIX TEMPORARILY
      STORE3 = FILE IN WHICH TO STORE MATRICES.
      STORE4 = VECTOR WORD COUNTER.
      = REF:4
      LABEL
      FORN(1,0,1,2)
      FORN(J,0,1,5)
      STORE4 := STORE4+1
      STORES2(STORE4) := STORES1(J)
      END*
      IF STORE4 = 540 THEN VECTOR IS FULL, OUNO TO DISK.
      BEGIN
      PUTAPF(STORE3,STORE4,STORE2) * DISK WRITE.
      END*
      END STORE STORES.

```

```

FILLET = COMMENT OUT 2x2 BLOCK OF (C) IN (RT) * * * * *
      BEGIN
      MJ := JXNCULH3T;
      FORUM(0,1,1) FORUM(0,1,1)
      BEGIN
      AT(MJ,RT,MJ) := RT(MJ,RT,MJ) + C(MJ,MJ,S+1);
      MJ := MJ + 1;
      END;
      ENDA;

      BEGIN
      COMMENT: CALL LOCATEH IF IN LAST PARTITION. * * * * *
      IF IT = NPARTM
      THEN
      BEGIN
      LOCATEH;
      KOPP := (KOPP-1);
      ENDA;

```

```

LOCATER =
COMMENT: * * DETERMINE LOCATION IN [BT] FOR PLACEMENT
          OF STIFFNESS FOR IMAGE POINTS * * * * *
BEGIN
COMMENT: * * PERMUTE NODES OF LKOPP, DEPENDING ON ITS ORIENTATION!
          FOR NODE ON BOUNDARY: PERMUTATION = 210
          TWO NODES ON BOUNDARY: PERMUTATION = 012 * * * * *
          IF PARINOD(LK+1,1) + PARINOD(LK+2,1) = 1
          THEN BEGIN IMAGE(0) = NOD(LKOPP,21) IMAGE(1) = NOD(LKOPP,11)
                   IMAGE(2) = NOD(LKOPP,01) END
          ELSE BEGIN IMAGE(0) = NOD(LKOPP,01) IMAGE(1) = NOD(LKOPP,21)
                   IMAGE(2) = NOD(LKOPP,11) END
          FOR(I=0,1,2) FOR(J=1,1,2)
          BEGIN
            L = PARINOD(LK,1) + PARINOD(LK,J) - 1
            IF L NEG 0
            COMMENT: * * * * * AT LEAST ONE OF THE NODES MUST BE ON BOUNDARY TO
                      NECESSITATE MODIFICATION OF BT. * * * * *
            THEN
              BEGIN
                IF MOD(LK,1) < MOD(LK,J)
                THEN COMMENT: ROW IS 1, COLUMN IS IMAGE OF J. * * * * *
                BEGIN COMMENT: C(J,1) IS STORED. * * * * *
                  R = NREF x JJ
                  S = NREF x JJ
                  WRMBT = MOD(LK,1) - 1
                  FOR(NCOLMBT=0,1,
                    PTIMROBT,NCOLMBT) NEW -IMAGE(J)
                  END
                ELSE COMMENT: ROW IS J, COLUMN IS IMAGE OF 1. * * * * *
                BEGIN COMMENT: C(1,J) IS STORED. * * * * *
                  R = NREF x JJ
                  S = NREF x JJ
                  WRMBT = MOD(LK,J) - 1
                  FOR(NCOLMBT=0,1,
                    PTIMROBT,NCOLMBT) NEW -IMAGE(1)
                  END
                COMMENT: * * COEFFICIENT BTIMROBT,NCOLMBT IS STORED AS COEFFICIENT
                          FOR STENCILMBT x Q(MRMBT), E.G. BT(28,15) IS
                          STORED AS BT(15,28) * * * * *
              END
            FILRTMINUS =
            COMMENT: PUT 2x2 BLOCK OF [C] IN [BT] * * * * *
            BEGIN
              M = 4*NCOLMBT
              BTIMROBT,MJ = BTIMROBT,MJ - CIR *S
              BTIMROBT,MJ+1 = BTIMROBT,MJ+1 - CIR *S+1
              BTIMROBT,MJ+2 = BTIMROBT,MJ+2 - CIR+1,S
              BTIMROBT,MJ+3 = BTIMROBT,MJ+3 - CIR+1,S+1
            END
          END
          COMMENT: PUT NEGATIVE COEFF. IN [BT].
          END
          END

```

```

CUMULATIVE      COMMENT: COMBINE COEFFICIENTS OF NODES ON BOUNDARY **
= BEGIN FORU(I,BOUNDARY,1,NP0IN-1) FORU(J,0,1,BZV(I)-1)
COMMENT: J IS COLN OF CROSS COEFF.  E.G. (27-23) LOCATED IN J=1. * * *
      IF PT(I,J) GT 0 THEN BEGIN IMAGENODE :=
      PT(PT(I,J)-1,PTMAX);
COMMENT: IMAGE NODE IS IMAGE OF CROSS COEF NODE E.G. IMAGE(23)=27 * * *
      FORU(MHT,0,1,PT(IMAGENODE-1,MHT) * PT(I,PTMAX) )
COMMENT: IMAGE OF ITH NDU IS LOCATED IN IMAGENODE ROW IN COLN MHT * * *
      FORU(MNJ,1,1,PT(I,MNJ) * PT(IMAGENODE) / FORU(K,0,1,3)
COMMENT: IMAGE OF CROSS COEFF. IS IN COLN MNJ OF ROW I. * * * * *
      BEGIN AT(IMAGENODE-1,(XNMH)+K1 :=
      AT(IMAGENODE-1,(XNMH)+K1)- AT(I,(XNMH)+K1);
      AT(I,(XNMH)+K1 := 0;END;      END;      END#;

```

```

WIDE =      COMMENT: COMPUTE THE WIDTH OF THE STRUCTURE **
WIDTH := (XINLASTINPARTM1)-1.03 - XINFIRST(0)-1.03);
FIRSTVEL =      COMMENT: COMPUTE THE INITIAL WALL CONVERGENCE **
INITVEL := (ARS(RV(MR0IN1),0)) + ABS(RV(0,0));

```

```

FILLF = COMMENT: SPECIFY LOAD AS #+12 X RT(I,1) X SPECIFIED
        VELOCITY. MULTIPLY RT(I,1) BY #+12;
        FORC(I,0,1,ABOUNM1)

BEGIN
  J := NRT(I,1);
  IF NB(I,0) = 0
  THEN
    BEGIN
      VEL:=AV(I,0) *WIDTH/INIVEL;
      RT(I,0) := RT(I,0) * #+12;
      F(2*XJ,0) := RT(I,0) * VEL;
    END;
  ELSE IF RT(I,1) = 0
  THEN
    BEGIN
      VEL:=AV(I,1) *WIDTH/INIVEL;
      RT(I,0) := RT(I,0) * #+12;
      F(2*XJ)+1,0 := RT(I,0) * VEL;
    END;
  END;
END;

```

```

SOLVER = COMMENT: SOLVE SIMULTANEOUS EQUATIONS USING
        GAUSS-SEIDEL WITH OVERRELAXATION FACTOR;
        BEGIN
          FORC(MM,0,1,NCOLNM1)
        BEGIN
          INTEGER COUNT;
          DO
            COUNT := 0;
          BEGIN
            COUNT:=COUNT+1; COMMENT: COUNT ITERATIONS. * * * *;
            COMMENT: INITIALIZE COLUMN POINTER TO COLUMN 1 FOR ALL ROWS. * *;
            FORC(I,0,1,NPOINM1) BEGIN COLMMATE(I):=1; WHILE
              PT(I,COLMMATE(I)) LSS 0 DO COLMMATE(I):=COLMMATE(I)+1;
              END;
            J := K := 0;
            M := 0; * INITIALIZE ROW POINTER TO ROW 0.
            ESTJON := 0;
            FORC(I,0,1,NPOINM1)
          BEGIN
            LABEL
              GIVEY;
              R := Z RT;
              XSUMLEFT := XSUMLEFT + YSUMLEFT + YSUMLEFT;
              XSUMRIGHT := XSUMRIGHT + YSUMRIGHT + YSUMRIGHT;
              I := C; * ZERO ACCUMULATORS.

            COMMENT: * * * SEARCH (PT) FOR LOCATION OF COEFF. K(I,N), N=1...NPOIN
              A COEFFICIENT K(I,N) WILL BE STOPPED AS K(N,1). IF THE
              NTH COLUMN OF PT CONTAINS I+1 THEN THAT COEFFICIENT IS
              REQUIRED IN ROW I. * * * * *
              FORC(NCOLM,MJ,1,I)=1;
              IF (PT(NCOLM,COLMMATE(NCOLM))) = I+1
            THEN
              BEGIN
                MJ:=XCOLMMATE(NCOLM);
                USL;
              END;
            FORC(NCOLM, 1,1,RZV(I)=1)

            COMMENT: * * SEARCH (PT) FOR LOCATION OF COEFF. K(I,K), K=I+1...N. * * *;
            IF PT(I,NCOLM) GTR 0 THEN

            COMMENT: NEGATIVE INDICES INDICATE COEFFICIENTS STORED FOR
              LATER ROWS NEEDING IMAGE NAPS - - NOT NEEDED FOR ITH EQN
            BEGIN
              M := 4 * NCOLM;
              MJ := 2*ABS(PT(I,NCOLM)-1); USR(I);
            END;
            IF PAR(I) = 1
            THEN COMMENT: ON BOUNDARY NAPS--LOOK FOR IMAGE COEFF.;
              FORC(NCOLM,NFINSTINDANT-I-1,1,I)=1;
            BEGIN
              FORC(J,1,1,RZV(NCOLM)-1) IF -PT(NCOLM,J) = 1+1
            THEN
              M := 4*XJ; MJ:=2*NCOLM;
              USR(NCOLM);
            END;
          END;
        END;

```

```

END) SOLVE*
FOR(NCOLM+J,I=1)
IF (PT(NCOLM,COLMAT(NCOLM))) = 1+1
THEN
BEGIN
M:=2*COLMAT(NCOLM)
VSL:
DO COLMAT(NCOLM) = COLMAT(NCOLM) + 1 UNTIL
PT(NCOLM,COLMAT(NCOLM)) GT 0;
COMMENT: * * * ONCE A COEFFICIENT HAS BEEN FOUND, IT WILL NOT BE
NEEDED AGAIN, SO THE COLUMNS POINTED (MATRIX COLMAT) IS INCRE-
MENTED TO POINT TO THE NEXT COLUMN. * * * * *
IF PT(NCOLM,COLMAT(NCOLM)) = 1023 AND NCOLM = NJ
THEN NJ := NJ + 1;
COMMENT: * * * IF THE END OF A ROW OF (PT) HAS BEEN REACHED AND THE
END OF ALL ROWS ABOVE THAT ROW HAVE BEEN REACHED, THEN THERE
IS NO NEED TO CHECK THAT ROW AGAIN, SO NJ = THE ROW OF (PT)
AT WHICH TO BEGIN SEARCHING, IS INCREMENTED. * * * * *
END)
COMMENT: * * SEARCH (PT) FOR LOCATION OF COEFF. K(I,K),K=1,1,...,N,* * *
FOR(NCOLM, 1,92V(1)-1)
BEGIN
IF PT(1,NCOLM) GT 0 THEN
M := 1 V NCOLM
MM := 2*PT(1,NCOLM)-1 VSR(1)
END)
IF RAR(1) = 1
THEN COMMENT: ON BOUNDARY NOOP--LOOK FOR IMAG COEFF.)
FOR(NCOLM,FIRSTIMPART-1,1,1,1)
BEGIN FOR(J,I,HZ(NCOLM)-1) IF PT(NCOLM,J) = 1+1
THEN
BEGIN M := 4+1; MM:=2*NCOLM
VSR(NCOLM)
END)
END)
SOLVE*
END)
WRITE(LINE,ITERNO,COUNT);
IF (COUNT) MOD 10 = 0 THEN
WRITE(ITERNO,MM,M10.6,M V =,R10.6" EPSILON =,
R10.6,R10.6,R10.6+1), EPSILON)
END UNTIL EPSILON < EPS OR COUNT = ITER;
END)
END OF SOLVE*

```

```

USL=
BEGIN
MMI := 2*NCOLM;
XUSUMLEFT := XUSUMLEFT + (PT(NCOLM,M) * Q(MMI));
XVSUMLEFT := XVSUMLEFT + (PT(NCOLM,M+2) * Q(MMI+1));
END)
VSL=
BEGIN
MMI := 2*NCOLM;
YUSUMLEFT := YUSUMLEFT + (PT(NCOLM,M+1) * Q(MMI));
YVSUMLEFT := YVSUMLEFT + (PT(NCOLM,M+3) * Q(MMI+1));
END)
USR=
BEGIN (USR1) COMMENTS: U-UNKNOWN SUM RIGHT USR1 IS ROW OFPT)
XUSUMRIGHT := XUSUMRIGHT + (PT(USR1,M) * Q(MMI));
XVSUMRIGHT := XVSUMRIGHT + (PT(USR1,M+1) * Q(MMI+1));
END)
VSR=
BEGIN (VSR1) COMMENTS: V-UNKNOWN SUM RIGHT USR1 IS ROW OFPT)
YUSUMRIGHT := YUSUMRIGHT + (PT(VSR1,M+2) * Q(MMI+1));
YVSUMRIGHT := YVSUMRIGHT + (PT(VSR1,M+3) * Q(MMI+1));
END)

```



```

SOLVEX =
  BEGIN
    XUSUMLEFT := XUSUMLEFT + YVSUMLEFT;
    XUSUMRIGHT := (XUSUMRIGHT + XVSUMRIGHT + (RT(1,1)*
    Q(R+1))) ;
    DELTAQ := (F(R+MM)-YVSUMLEFT-XUSUMRIGHT)/RT(1,01);
    DELTAQ := DELTAQ-Q(R);
    Q(R) := Q(R) + (RFAC*DELTAQ);
    EPSILON := MAX(EPSILON,ABS(DELTAQ));
  END;

SOLVEY =
  BEGIN
    YVSUMLEFT := YVSUMLEFT + XUSUMLEFT + (BT(1,2)*Q(R));
    YVSUMRIGHT := YVSUMRIGHT + XUSUMRIGHT;
    DELTAQ := (F(R+1,MM)-YVSUMLEFT-YVSUMRIGHT)/RT(1,3);
    DELTAQ := DELTAQ - Q(R+1);
    Q(R+1) := Q(R+1) + (RFAC*DELTAQ);
    EPSILON := MAX(EPSILON,ABS(DELTAQ));
  END;

```

```

INCREMENT1 = MM*MM*MM*IT. DISPLACE NODAL POINTS;
WRITE(UNIT,DELTA,INFORM,LOGSTRAIN*100*MMUL,
  YEARS:= YEARS* DELTA/(STRAINRATE*2.81*7));
FOR(I:=0,MM*MM) FOR(J:=0,1,NCOL*MM)

COMMENT: COMPUTE DISPLACEMENTS DUE TO ALL FORCE VECTORS(MM), N=VT;

  S(I,MM) := 4(I) *DELTA;
  FOR(I:=0,MM*MM) FOR(J:=0,1,NCOL*MM)
    BEGIN
      X(I,0) := X(I,0) + (S(I,2*1) *MM);
      X(I,1) := X(I,1) + (S(I,2*1)+1*MM);
    END;

  WRITE(UNIT,FILE(0),'*LOGSTRAIN*STRAINRATE*DELTA')
  ENDS;

```

```

COMPSR (COMPSTR1,COMPSTR2)=COMMENT; COMPUTE STRESSES;
  BEGIN
    WRITE(LINE,STRITILECOMPSTR1);
  COMMENT: ** READ STRESS MATRICES FROM DISK FILE. ** * * * * *
  BEGIN
    READ(SRSTRSTRN(COMPSTR1));
    READ(SRSTRSTRN(COMPSTR1),JJ:=540,COMPSTR2(1:3));
    FM := 0;
    M := 3; N := 6; H := 4; S := 1;
    FOR(CLK,1,1*NELEMM1)
      IF FM = 540 THEN
        COMMENT: (SAV) IS EXHAUSTED, READ MORE STRESS MATRICES FROM DISK FILE.
        BEGIN
          READ(SRSTRSTRN(COMPSTR1),JJ:=540,COMPSTR2(1:3));
          FM := 0;
        END
      ** CONSTRUCT (SFI. ** * * * * *
      END;
    FOR(CI,0,1*2)
      FOR(CJ,0,1*5)
        FM := FM + 1;
        SELECT,JJ := COMPSTR2(FM);
      END;
    STRESS(ST,P,ELEMMSTR); FOR(CI,0,1*2)
      DT(1)=FIFMSTR(LK,1); PRINC(DT,FILINDEX);
      DELSTR(1) := ABS(DT(1)-DT(1));
    WRITE(LINE,PRINCOPALK+1,DT(1),DT(1),DT(1)*57.3);
    DELSTR(1)=FOR(CI,0,1*2) FLEMMSTR(LK,1);
    WRITE(PLATEFILE,COMPSTR1,*,FOR(CI,0,1*2) DT(1));
  REPLACE PINTERCELEMMSTR(LK,*) BY PINTERCOT(1);
  FOR I WDOENST % PRINCIPAL STRAINS READY FOR VICOMP
  END;
  END;

```

```

COMMENT: ** UTILITY PACKAGE IF DEFINES. ** * * * * *
  FOR(CI,0,2*ZRO3,FOR(CJ,0,1*2))
    FOR(CI,0,1) := FOR(CJ,0,1) UNTIL FOR(CI,0,1)
    FOR(CI,0,1) := FOR(CJ,0,1) UNTIL FOR(CI,0,1)
    FOR(CI,0,1) := FOR(CJ,0,1) UNTIL FOR(CI,0,1)
    RITAPE
      RITAPE1:=RITAPE2,RITAPE2:=RITAPE3,RITAPE3:=RITAPE2;
      RITAPE4:=RITAPE5,RITAPE5:=RITAPE6,RITAPE6:=RITAPE5;
      RITAPE7:=RITAPE8,RITAPE8:=RITAPE9,RITAPE9:=RITAPE8;
      RITAPE10:=RITAPE11,RITAPE11:=RITAPE12,RITAPE12:=RITAPE11;
      RITAPE13:=RITAPE14,RITAPE14:=RITAPE15,RITAPE15:=RITAPE14;
      RITAPE16:=RITAPE17,RITAPE17:=RITAPE18,RITAPE18:=RITAPE16;
      RITAPE19:=RITAPE20,RITAPE20:=RITAPE21,RITAPE21:=RITAPE19;
      RITAPE22:=RITAPE23,RITAPE23:=RITAPE24,RITAPE24:=RITAPE22;
      RITAPE25:=RITAPE26,RITAPE26:=RITAPE27,RITAPE27:=RITAPE25;
      RITAPE28:=RITAPE29,RITAPE29:=RITAPE30,RITAPE30:=RITAPE28;
      RITAPE31:=RITAPE32,RITAPE32:=RITAPE33,RITAPE33:=RITAPE31;
      RITAPE34:=RITAPE35,RITAPE35:=RITAPE36,RITAPE36:=RITAPE34;
      RITAPE37:=RITAPE38,RITAPE38:=RITAPE39,RITAPE39:=RITAPE37;
      RITAPE40:=RITAPE41,RITAPE41:=RITAPE42,RITAPE42:=RITAPE40;
      RITAPE43:=RITAPE44,RITAPE44:=RITAPE45,RITAPE45:=RITAPE43;
      RITAPE46:=RITAPE47,RITAPE47:=RITAPE48,RITAPE48:=RITAPE46;
      RITAPE49:=RITAPE50,RITAPE50:=RITAPE51,RITAPE51:=RITAPE49;
      RITAPE52:=RITAPE53,RITAPE53:=RITAPE54,RITAPE54:=RITAPE52;
      RITAPE55:=RITAPE56,RITAPE56:=RITAPE57,RITAPE57:=RITAPE55;
      RITAPE58:=RITAPE59,RITAPE59:=RITAPE60,RITAPE60:=RITAPE58;
      RITAPE61:=RITAPE62,RITAPE62:=RITAPE63,RITAPE63:=RITAPE61;
      RITAPE64:=RITAPE65,RITAPE65:=RITAPE66,RITAPE66:=RITAPE64;
      RITAPE67:=RITAPE68,RITAPE68:=RITAPE69,RITAPE69:=RITAPE67;
      RITAPE70:=RITAPE71,RITAPE71:=RITAPE72,RITAPE72:=RITAPE70;
      RITAPE73:=RITAPE74,RITAPE74:=RITAPE75,RITAPE75:=RITAPE73;
      RITAPE76:=RITAPE77,RITAPE77:=RITAPE78,RITAPE78:=RITAPE76;
      RITAPE79:=RITAPE80,RITAPE80:=RITAPE81,RITAPE81:=RITAPE79;
      RITAPE82:=RITAPE83,RITAPE83:=RITAPE84,RITAPE84:=RITAPE82;
      RITAPE85:=RITAPE86,RITAPE86:=RITAPE87,RITAPE87:=RITAPE85;
      RITAPE88:=RITAPE89,RITAPE89:=RITAPE90,RITAPE90:=RITAPE88;
      RITAPE91:=RITAPE92,RITAPE92:=RITAPE93,RITAPE93:=RITAPE91;
      RITAPE94:=RITAPE95,RITAPE95:=RITAPE96,RITAPE96:=RITAPE94;
      RITAPE97:=RITAPE98,RITAPE98:=RITAPE99,RITAPE99:=RITAPE97;
      RITAPE100:=RITAPE101,RITAPE101:=RITAPE102,RITAPE102:=RITAPE100;
      RITAPE103:=RITAPE104,RITAPE104:=RITAPE105,RITAPE105:=RITAPE103;
      RITAPE106:=RITAPE107,RITAPE107:=RITAPE108,RITAPE108:=RITAPE106;
      RITAPE109:=RITAPE110,RITAPE110:=RITAPE111,RITAPE111:=RITAPE109;
      RITAPE112:=RITAPE113,RITAPE113:=RITAPE114,RITAPE114:=RITAPE112;
      RITAPE115:=RITAPE116,RITAPE116:=RITAPE117,RITAPE117:=RITAPE115;
      RITAPE118:=RITAPE119,RITAPE119:=RITAPE120,RITAPE120:=RITAPE118;
      RITAPE121:=RITAPE122,RITAPE122:=RITAPE123,RITAPE123:=RITAPE121;
      RITAPE124:=RITAPE125,RITAPE125:=RITAPE126,RITAPE126:=RITAPE124;
      RITAPE127:=RITAPE128,RITAPE128:=RITAPE129,RITAPE129:=RITAPE127;
      RITAPE130:=RITAPE131,RITAPE131:=RITAPE132,RITAPE132:=RITAPE130;
      RITAPE133:=RITAPE134,RITAPE134:=RITAPE135,RITAPE135:=RITAPE133;
      RITAPE136:=RITAPE137,RITAPE137:=RITAPE138,RITAPE138:=RITAPE136;
      RITAPE139:=RITAPE140,RITAPE140:=RITAPE141,RITAPE141:=RITAPE139;
      RITAPE142:=RITAPE143,RITAPE143:=RITAPE144,RITAPE144:=RITAPE142;
      RITAPE145:=RITAPE146,RITAPE146:=RITAPE147,RITAPE147:=RITAPE145;
      RITAPE148:=RITAPE149,RITAPE149:=RITAPE150,RITAPE150:=RITAPE148;
      RITAPE151:=RITAPE152,RITAPE152:=RITAPE153,RITAPE153:=RITAPE151;
      RITAPE154:=RITAPE155,RITAPE155:=RITAPE156,RITAPE156:=RITAPE154;
      RITAPE157:=RITAPE158,RITAPE158:=RITAPE159,RITAPE159:=RITAPE157;
      RITAPE160:=RITAPE161,RITAPE161:=RITAPE162,RITAPE162:=RITAPE160;
      RITAPE163:=RITAPE164,RITAPE164:=RITAPE165,RITAPE165:=RITAPE163;
      RITAPE166:=RITAPE167,RITAPE167:=RITAPE168,RITAPE168:=RITAPE166;
      RITAPE169:=RITAPE170,RITAPE170:=RITAPE171,RITAPE171:=RITAPE169;
      RITAPE172:=RITAPE173,RITAPE173:=RITAPE174,RITAPE174:=RITAPE172;
      RITAPE175:=RITAPE176,RITAPE176:=RITAPE177,RITAPE177:=RITAPE175;
      RITAPE178:=RITAPE179,RITAPE179:=RITAPE180,RITAPE180:=RITAPE178;
      RITAPE181:=RITAPE182,RITAPE182:=RITAPE183,RITAPE183:=RITAPE181;
      RITAPE184:=RITAPE185,RITAPE185:=RITAPE186,RITAPE186:=RITAPE184;
      RITAPE187:=RITAPE188,RITAPE188:=RITAPE189,RITAPE189:=RITAPE187;
      RITAPE190:=RITAPE191,RITAPE191:=RITAPE192,RITAPE192:=RITAPE190;
      RITAPE193:=RITAPE194,RITAPE194:=RITAPE195,RITAPE195:=RITAPE193;
      RITAPE196:=RITAPE197,RITAPE197:=RITAPE198,RITAPE198:=RITAPE196;
      RITAPE199:=RITAPE200,RITAPE200:=RITAPE201,RITAPE201:=RITAPE199;
      RITAPE202:=RITAPE203,RITAPE203:=RITAPE204,RITAPE204:=RITAPE202;
      RITAPE205:=RITAPE206,RITAPE206:=RITAPE207,RITAPE207:=RITAPE205;
      RITAPE208:=RITAPE209,RITAPE209:=RITAPE210,RITAPE210:=RITAPE208;
      RITAPE211:=RITAPE212,RITAPE212:=RITAPE213,RITAPE213:=RITAPE211;
      RITAPE214:=RITAPE215,RITAPE215:=RITAPE216,RITAPE216:=RITAPE214;
      RITAPE217:=RITAPE218,RITAPE218:=RITAPE219,RITAPE219:=RITAPE217;
      RITAPE220:=RITAPE221,RITAPE221:=RITAPE222,RITAPE222:=RITAPE220;
      RITAPE223:=RITAPE224,RITAPE224:=RITAPE225,RITAPE225:=RITAPE223;
      RITAPE226:=RITAPE227,RITAPE227:=RITAPE228,RITAPE228:=RITAPE226;
      RITAPE229:=RITAPE230,RITAPE230:=RITAPE231,RITAPE231:=RITAPE229;
      RITAPE232:=RITAPE233,RITAPE233:=RITAPE234,RITAPE234:=RITAPE232;
      RITAPE235:=RITAPE236,RITAPE236:=RITAPE237,RITAPE237:=RITAPE235;
      RITAPE238:=RITAPE239,RITAPE239:=RITAPE240,RITAPE240:=RITAPE238;
      RITAPE241:=RITAPE242,RITAPE242:=RITAPE243,RITAPE243:=RITAPE241;
      RITAPE244:=RITAPE245,RITAPE245:=RITAPE246,RITAPE246:=RITAPE244;
      RITAPE247:=RITAPE248,RITAPE248:=RITAPE249,RITAPE249:=RITAPE247;
      RITAPE250:=RITAPE251,RITAPE251:=RITAPE252,RITAPE252:=RITAPE250;
      RITAPE253:=RITAPE254,RITAPE254:=RITAPE255,RITAPE255:=RITAPE253;
      RITAPE256:=RITAPE257,RITAPE257:=RITAPE258,RITAPE258:=RITAPE256;
      RITAPE259:=RITAPE260,RITAPE260:=RITAPE261,RITAPE261:=RITAPE259;
      RITAPE262:=RITAPE263,RITAPE263:=RITAPE264,RITAPE264:=RITAPE262;
      RITAPE265:=RITAPE266,RITAPE266:=RITAPE267,RITAPE267:=RITAPE265;
      RITAPE268:=RITAPE269,RITAPE269:=RITAPE270,RITAPE270:=RITAPE268;
      RITAPE271:=RITAPE272,RITAPE272:=RITAPE273,RITAPE273:=RITAPE271;
      RITAPE274:=RITAPE275,RITAPE275:=RITAPE276,RITAPE276:=RITAPE274;
      RITAPE277:=RITAPE278,RITAPE278:=RITAPE279,RITAPE279:=RITAPE277;
      RITAPE280:=RITAPE281,RITAPE281:=RITAPE282,RITAPE282:=RITAPE280;
      RITAPE283:=RITAPE284,RITAPE284:=RITAPE285,RITAPE285:=RITAPE283;
      RITAPE286:=RITAPE287,RITAPE287:=RITAPE288,RITAPE288:=RITAPE286;
      RITAPE289:=RITAPE290,RITAPE290:=RITAPE291,RITAPE291:=RITAPE289;
      RITAPE292:=RITAPE293,RITAPE293:=RITAPE294,RITAPE294:=RITAPE292;
      RITAPE295:=RITAPE296,RITAPE296:=RITAPE297,RITAPE297:=RITAPE295;
      RITAPE298:=RITAPE299,RITAPE299:=RITAPE300,RITAPE300:=RITAPE298;
      RITAPE301:=RITAPE302,RITAPE302:=RITAPE303,RITAPE303:=RITAPE301;
      RITAPE304:=RITAPE305,RITAPE305:=RITAPE306,RITAPE306:=RITAPE304;
      RITAPE307:=RITAPE308,RITAPE308:=RITAPE309,RITAPE309:=RITAPE307;
      RITAPE310:=RITAPE311,RITAPE311:=RITAPE312,RITAPE312:=RITAPE310;
      RITAPE313:=RITAPE314,RITAPE314:=RITAPE315,RITAPE315:=RITAPE313;
      RITAPE316:=RITAPE317,RITAPE317:=RITAPE318,RITAPE318:=RITAPE316;
      RITAPE319:=RITAPE320,RITAPE320:=RITAPE321,RITAPE321:=RITAPE319;
      RITAPE322:=RITAPE323,RITAPE323:=RITAPE324,RITAPE324:=RITAPE322;
      RITAPE325:=RITAPE326,RITAPE326:=RITAPE327,RITAPE327:=RITAPE325;
      RITAPE328:=RITAPE329,RITAPE329:=RITAPE330,RITAPE330:=RITAPE328;
      RITAPE331:=RITAPE332,RITAPE332:=RITAPE333,RITAPE333:=RITAPE331;
      RITAPE334:=RITAPE335,RITAPE335:=RITAPE336,RITAPE336:=RITAPE334;
      RITAPE337:=RITAPE338,RITAPE338:=RITAPE339,RITAPE339:=RITAPE337;
      RITAPE340:=RITAPE341,RITAPE341:=RITAPE342,RITAPE342:=RITAPE340;
      RITAPE343:=RITAPE344,RITAPE344:=RITAPE345,RITAPE345:=RITAPE343;
      RITAPE346:=RITAPE347,RITAPE347:=RITAPE348,RITAPE348:=RITAPE346;
      RITAPE349:=RITAPE350,RITAPE350:=RITAPE351,RITAPE351:=RITAPE349;
      RITAPE352:=RITAPE353,RITAPE353:=RITAPE354,RITAPE354:=RITAPE352;
      RITAPE355:=RITAPE356,RITAPE356:=RITAPE357,RITAPE357:=RITAPE355;
      RITAPE358:=RITAPE359,RITAPE359:=RITAPE360,RITAPE360:=RITAPE358;
      RITAPE361:=RITAPE362,RITAPE362:=RITAPE363,RITAPE363:=RITAPE361;
      RITAPE364:=RITAPE365,RITAPE365:=RITAPE366,RITAPE366:=RITAPE364;
      RITAPE367:=RITAPE368,RITAPE368:=RITAPE369,RITAPE369:=RITAPE367;
      RITAPE370:=RITAPE371,RITAPE371:=RITAPE372,RITAPE372:=RITAPE370;
      RITAPE373:=RITAPE374,RITAPE374:=RITAPE375,RITAPE375:=RITAPE373;
      RITAPE376:=RITAPE377,RITAPE377:=RITAPE378,RITAPE378:=RITAPE376;
      RITAPE379:=RITAPE380,RITAPE380:=RITAPE381,RITAPE381:=RITAPE379;
      RITAPE382:=RITAPE383,RITAPE383:=RITAPE384,RITAPE384:=RITAPE382;
      RITAPE385:=RITAPE386,RITAPE386:=RITAPE387,RITAPE387:=RITAPE385;
      RITAPE388:=RITAPE389,RITAPE389:=RITAPE390,RITAPE390:=RITAPE388;
      RITAPE391:=RITAPE392,RITAPE392:=RITAPE393,RITAPE393:=RITAPE391;
      RITAPE394:=RITAPE395,RITAPE395:=RITAPE396,RITAPE396:=RITAPE394;
      RITAPE397:=RITAPE398,RITAPE398:=RITAPE399,RITAPE399:=RITAPE397;
      RITAPE400:=RITAPE401,RITAPE401:=RITAPE402,RITAPE402:=RITAPE400;
      RITAPE403:=RITAPE404,RITAPE404:=RITAPE405,RITAPE405:=RITAPE403;
      RITAPE406:=RITAPE407,RITAPE407:=RITAPE408,RITAPE408:=RITAPE406;
      RITAPE409:=RITAPE410,RITAPE410:=RITAPE411,RITAPE411:=RITAPE409;
      RITAPE412:=RITAPE413,RITAPE413:=RITAPE414,RITAPE414:=RITAPE412;
      RITAPE415:=RITAPE416,RITAPE416:=RITAPE417,RITAPE417:=RITAPE415;
      RITAPE418:=RITAPE419,RITAPE419:=RITAPE420,RITAPE420:=RITAPE418;
      RITAPE421:=RITAPE422,RITAPE422:=RITAPE423,RITAPE423:=RITAPE421;
      RITAPE424:=RITAPE425,RITAPE425:=RITAPE426,RITAPE426:=RITAPE424;
      RITAPE427:=RITAPE428,RITAPE428:=RITAPE429,RITAPE429:=RITAPE427;
      RITAPE430:=RITAPE431,RITAPE431:=RITAPE432,RITAPE432:=RITAPE430;
      RITAPE433:=RITAPE434,RITAPE434:=RITAPE435,RITAPE435:=RITAPE433;
      RITAPE436:=RITAPE437,RITAPE437:=RITAPE438,RITAPE438:=RITAPE436;
      RITAPE439:=RITAPE440,RITAPE440:=RITAPE441,RITAPE441:=RITAPE439;
      RITAPE442:=RITAPE443,RITAPE443:=RITAPE444,RITAPE444:=RITAPE442;
      RITAPE445:=RITAPE446,RITAPE446:=RITAPE447,RITAPE447:=RITAPE445;
      RITAPE448:=RITAPE449,RITAPE449:=RITAPE450,RITAPE450:=RITAPE448;
      RITAPE451:=RITAPE452,RITAPE452:=RITAPE453,RITAPE453:=RITAPE451;
      RITAPE454:=RITAPE455,RITAPE455:=RITAPE456,RITAPE456:=RITAPE454;
      RITAPE457:=RITAPE458,RITAPE458:=RITAPE459,RITAPE459:=RITAPE457;
      RITAPE460:=RITAPE461,RITAPE461:=RITAPE462,RITAPE462:=RITAPE460;
      RITAPE463:=RITAPE464,RITAPE464:=RITAPE465,RITAPE465:=RITAPE463;
      RITAPE466:=RITAPE467,RITAPE467:=RITAPE468,RITAPE468:=RITAPE466;
      RITAPE469:=RITAPE470,RITAPE470:=RITAPE471,RITAPE471:=RITAPE469;
      RITAPE472:=RITAPE473,RITAPE473:=RITAPE474,RITAPE474:=RITAPE472;
      RITAPE475:=RITAPE476,RITAPE476:=RITAPE477,RITAPE477:=RITAPE475;
      RITAPE478:=RITAPE479,RITAPE479:=RITAPE480,RITAPE480:=RITAPE478;
      RITAPE481:=RITAPE482,RITAPE482:=RITAPE483,RITAPE483:=RITAPE481;
      RITAPE484:=RITAPE485,RITAPE485:=RITAPE486,RITAPE486:=RITAPE484;
      RITAPE487:=RITAPE488,RITAPE488:=RITAPE489,RITAPE489:=RITAPE487;
      RITAPE490:=RITAPE491,RITAPE491:=RITAPE492,RITAPE492:=RITAPE490;
      RITAPE493:=RITAPE494,RITAPE494:=RITAPE495,RITAPE495:=RITAPE493;
      RITAPE496:=RITAPE497,RITAPE497:=RITAPE498,RITAPE498:=RITAPE496;
      RITAPE499:=RITAPE500,RITAPE500:=RITAPE501,RITAPE501:=RITAPE499;
      RITAPE502:=RITAPE503,RITAPE503:=RITAPE504,RITAPE504:=RITAPE502;
      RITAPE505:=RITAPE506,RITAPE506:=RITAPE507,RITAPE507:=RITAPE505;
      RITAPE508:=RITAPE509,RITAPE509:=RITAPE510,RITAPE510:=RITAPE508;
      RITAPE511:=RITAPE512,RITAPE512:=RITAPE513,RITAPE513:=RITAPE511;
      RITAPE514:=RITAPE515,RITAPE515:=RITAPE516,RITAPE516:=RITAPE514;
      RITAPE517:=RITAPE518,RITAPE518:=RITAPE519,RITAPE519:=RITAPE517;
      RITAPE520:=RITAPE521,RITAPE521:=RITAPE522,RITAPE522:=RITAPE520;
      RITAPE523:=RITAPE524,RITAPE524:=RITAPE525,RITAPE525:=RITAPE523;
      RITAPE526:=RITAPE527,RITAPE527:=RITAPE528,RITAPE528:=RITAPE526;
      RITAPE529:=RITAPE530,RITAPE530:=RITAPE531,RITAPE531:=RITAPE529;
      RITAPE532:=RITAPE533,RITAPE533:=RITAPE534,RITAPE534:=RITAPE532;
      RITAPE535:=RITAPE536,RITAPE536:=RITAPE537,RITAPE537:=RITAPE535;
      RITAPE538:=RITAPE539,RITAPE539:=RITAPE540,RITAPE540:=RITAPE538;
      RITAPE541:=RITAPE542,RITAPE542:=RITAPE543,RITAPE543:=RITAPE541;
      RITAPE544:=RITAPE545,RITAPE545:=RITAPE546,RITAPE546:=RITAPE544;
      RITAPE547:=RITAPE548,RITAPE548:=RITAPE549,RITAPE549:=RITAPE547;
      RITAPE550:=RITAPE551,RITAPE551:=RITAPE552,RITAPE552:=RITAPE550;
      RITAPE553:=RITAPE554,RITAPE554:=RITAPE555,RITAPE555:=RITAPE553;
      RITAPE556:=RITAPE557,RITAPE557:=RITAPE558,RITAPE558:=RITAPE556;
      RITAPE559:=RITAPE560,RITAPE560:=RITAPE561,RITAPE561:=RITAPE559;
      RITAPE562:=RITAPE563,RITAPE563:=RITAPE564,RITAPE564:=RITAPE562;
      RITAPE565:=RITAPE566,RITAPE566:=RITAPE567,RITAPE567:=RITAPE565;
      RITAPE568:=RITAPE569,RITAPE569:=RITAPE570,RITAPE570:=RITAPE568;
      RITAPE571:=RITAPE572,RITAPE572:=RITAPE573,RITAPE573:=RITAPE571;
      RITAPE574:=RITAPE575,RITAPE575:=RITAPE576,RITAPE576:=RITAPE574;
      RITAPE577:=RITAPE578,RITAPE578:=RITAPE579,RITAPE579:=RITAPE577;
      RITAPE580:=RITAPE581,RITAPE581:=RITAPE582,RITAPE582:=RITAPE580;
      RITAPE583:=RITAPE584,RITAPE584:=RITAPE585,RITAPE585:=RITAPE583;
      RITAPE586:=RITAPE587,RITAPE587:=RITAPE588,RITAPE588:=RITAPE586;
      RITAPE589:=RITAPE590,RITAPE590:=RITAPE591,RITAPE591:=RITAPE589;
      RITAPE592:=RITAPE593,RITAPE593:=RITAPE594,RITAPE594:=RITAPE592;
      RITAPE595:=RITAPE596,RITAPE596:=RITAPE597,RITAPE597:=RITAPE595;
      RITAPE598:=RITAPE599,RITAPE599:=RITAPE600,RITAPE600:=RITAPE598;
      RITAPE601:=RITAPE602,RITAPE602:=RITAPE603,RITAPE603:=RITAPE601;
      RITAPE604:=RITAPE605,RITAPE605:=RITAPE606,RITAPE606:=RITAPE604;
      RITAPE607:=RITAPE608,RITAPE608:=RITAPE609,RITAPE609:=RITAPE607;
      RITAPE610:=RITAPE611,RITAPE611:=RITAPE612,RITAPE612:=RITAPE610;
      RITAPE613:=RITAPE614,RITAPE614:=RITAPE615,RITAPE615:=RITAPE613;
      RITAPE616:=RITAPE617,RITAPE617:=RITAPE618,RITAPE618:=RITAPE616;
      RITAPE619:=RITAPE620,RITAPE620:=RITAPE621,RITAPE621:=RITAPE619;
      RITAPE622:=RITAPE623,RITAPE623:=RITAPE624,RITAPE624:=RITAPE622;
      RITAPE625:=RITAPE626,RITAPE626:=RITAPE627,RITAPE627:=RITAPE625;
      RITAPE628:=RITAPE629,RITAPE629:=RITAPE630,RITAPE630:=RITAPE628;
      RITAPE631:=RITAPE632,RITAPE632:=RITAPE633,RITAPE633:=RITAPE631;
      RITAPE634:=RITAPE635,RITAPE635:=RITAPE636,RITAPE636:=RITAPE634;
      RITAPE637:=RITAPE638,RITAPE638:=RITAPE639,RITAPE639:=RITAPE637;
      RITAPE640:=RITAPE641,RITAPE641:=RITAPE642,RITAPE642:=RITAPE640;
      RITAPE643:=RITAPE644,RITAPE644:=RITAPE645,RITAPE645:=RITAPE643;
      RITAPE646:=RITAPE647,RITAPE647:=RITAPE648,RITAPE648:=RITAPE646;
      RITAPE649:=RITAPE650,RITAPE650:=RITAPE651,RITAPE651:=RITAPE649;
      RITAPE652:=RITAPE653,RITAPE653:=RITAPE654,RITAPE654:=RITAPE652;
      RITAPE655:=RITAPE656,RITAPE656:=RITAPE657,RITAPE657:=RITAPE655;
      RITAPE658:=RITAPE659,RITAPE659:=RITAPE660,RITAPE660:=RITAPE658;
      RITAPE661:=RITAPE662,RITAPE662:=RITAPE663,RITAPE663:=RITAPE661;
      RITAPE664:=RITAPE665,RITAPE665:=RITAPE666,RITAPE666:=RITAPE664;
      RITAPE667:=RITAPE668,RITAPE668:=RITAPE669,RITAPE669:=RITAPE667;
      RITAPE670:=RITAPE671,RITAPE671:=RITAPE672,RITAPE672:=RITAPE670;
      RITAPE673:=RITAPE674,RITAPE674:=RITAPE675,RITAPE675:=RITAPE673;
      RITAPE676:=RITAPE677,RITAPE677:=RITAPE678,RITAPE678:=RITAPE676;
      RITAPE679:=RITAPE680,RITAPE680:=RITAPE681,RITAPE681:=RITAPE679;
      RITAPE682:=RITAPE683,RITAPE683:=RITAPE684,RITAPE684:=RITAPE682;
      RITAPE685:=RITAPE686,RITAPE686:=RITAPE687,RITAPE687:=RITAPE685;
      RITAPE688:=RITAPE689,RITAPE689:=RITAPE690,RITAPE690:=RITAPE688;
      RITAPE691:=RITAPE692,RITAPE692:=RITAPE693,RITAPE693:=RITAPE691;
      RITAPE694:=RITAPE695,RITAPE695:=RITAPE696,RITAPE696:=RITAPE694;
      RITAPE697:=RITAPE698,RITAPE698:=RITAPE699,RITAPE699:=RITAPE697;
      RITAPE700:=RITAPE701,RITAPE701:=RITAPE702,RITAPE702:=RITAPE700;
      RITAPE703:=RITAPE704,RITAPE704:=RITAPE705,RITAPE705:=RITAPE703;
      RITAPE706:=RITAPE707,RITAPE707:=RITAPE708,RITAPE708:=RITAPE706;
      RITAPE709:=RITAPE710,RITAPE710:=RITAPE711,RITAPE711:=RITAPE709;
      RITAPE712:=RITAPE713,RITAPE713:=RITAPE714,RITAPE714:=RITAPE712;
      RITAPE715:=RITAPE716,RITAPE716:=RITAPE717,RITAPE717:=RITAPE715;
      RITAPE718:=RITAPE719,RITAPE719:=RITAPE720,RITAPE720:=RITAPE718;
      RITAPE721:=RITAPE722,RITAPE722:=RITAPE723,RITAPE723:=RITAPE721;
      RITAPE724:=RITAPE725,RITAPE725:=RITAPE726,RITAPE726:=RITAPE724;
      RITAPE727:=RITAPE728,RITAPE728:=RITAPE729,RITAPE729:=RITAPE727;
      RITAPE730:=RITAPE731,RITAPE731:=RITAPE732,RITAPE732:=RITAPE730;
      RITAPE733:=RITAPE734,RITAPE734:=RITAPE735,RITAPE735:=RITAPE733;
      RITAPE736:=RITAPE737,RITAPE737:=RITAPE738,RITAPE738:=RITAPE736;
      RITAPE739:=RITAPE740,RITAPE740:=RITAPE741,RITAPE741:=RITAPE739;
      RITAPE742:=RITAPE743,RITAPE743:=RITAPE744,RITAPE744:=RITAPE742;
      RITAPE745:=RITAPE746,RITAPE746:=RITAPE747,RITAPE747:=RITAPE745;
      RITAPE748:=RITAPE749,RITAPE749:=RITAPE750,RITAPE750:=RITAPE748;
      RITAPE751:=RITAPE752,RITAPE752:=RITAPE753,RITAPE753:=RITAPE751;
      RITAPE754:=RITAPE755,RITAPE755:=RITAPE756,RITAPE756:=RITAPE754;
      RITAPE757:=RITAPE758,RITAPE758:=RITAPE759,RITAPE759:=RITAPE757;
      RITAPE760:=RITAPE761,RITAPE761:=RITAPE762,RITAPE762:=RITAPE760;
      RITAPE763:=RITAPE764,RITAPE764:=RITAPE765,RITAPE765:=RITAPE763;
      RITAPE766:=RITAPE767,RITAPE767:=RITAPE768,RITAPE768:=RITAPE766;
      RITAPE769:=RITAPE770,RITAPE770:=RITAPE771,RITAPE771:=RITAPE769;
      RITAPE772:=RITAPE773,RITAPE773:=RITAPE774,RITAPE774:=RITAPE772;
      RITAPE775:=RITAPE776,RITAPE776:=RITAPE777,RITAPE777:=RITAPE775;
      RITAPE778:=RITAPE779,RITAPE779:=RITAPE780,RITAPE780:=RITAPE778;
      RITAPE781:=RITAPE782,RITAPE782:=RITAPE783,RITAPE783:=RITAPE781;
      RITAPE784:=RITAPE785,RITAPE785:=RITAPE786,RITAPE786:=RITAPE784;
      RITAPE787:=RITAPE788,RITAPE788:=RITAPE789,RITAPE789:=RITAPE787;
      RITAPE790:=RITAPE791,RITAPE791:=RITAPE792,RITAPE792:=RITAPE790;
      RITAPE793:=RITAPE794,RITAPE794:=RITAPE795,RITAPE795:=RITAPE793;
      RITAPE796:=RITAPE797,RITAPE797:=RITAPE798,RITAPE798:=RITAPE796;
      RITAPE799:=RITAPE800,RITAPE800:=RITAPE801,RITAPE801:=RITAPE799;
      RITAPE802:=RITAPE803,RITAPE803:=RITAPE804,RITAPE804:=RITAPE802;
      RITAPE805:=RITAPE806,RITAPE806:=RITAPE807,RITAPE807:=RITAPE805;
      RITAPE808:=RITAPE809,RITAPE809:=RITAPE810,RITAPE810:=RITAPE808;
      RITAPE811:=RITAPE812,RITAPE812:=RITAPE813,RITAPE813:=RITAPE811;
      RITAPE814:=RITAPE815,RITAPE815:=RITAPE816,RITAPE816:=RITAPE814;
      RITAPE817:=RITAPE818,RITAPE818:=RITAPE819,RITAPE819:=RITAPE817;
      RITAPE820:=RITAPE821,RITAPE821:=RITAPE822,RITAPE822:=RITAPE820;
      RITAPE823:=RITAPE824,RITAPE824:=RITAPE825,RITAPE825:=RITAPE823;
      RITAPE826:=RITAPE827,RITAPE827:=RITAPE828,RITAPE828:=RITAPE826;
      RITAPE829:=RITAPE830,RITAPE830:=RITAPE831,RITAPE831:=RITAPE829;
      RITAPE832:=RITAPE833,RITAPE833:=RITAPE834,RITAPE834:=RITAPE832;
      RITAPE835:=RITAPE836,RITAPE836:=RITAPE837,RITAPE837:=RITAPE835;
      RITAPE838:=RITAPE839,RITAPE839:=RITAPE840,RITAPE840:=RITAPE838;
      RITAPE841:=RITAPE842,RITAPE842:=RITAPE843,RITAPE843:=RITAPE841;
      RITAPE844:=RITAPE845,RITAPE845:=RITAPE846,RITAPE846:=RITAPE844;
      RITAPE847:=RITAPE848,RITAPE848:=RITAPE849,RITAPE849:=RITAPE847;
      RITAPE850:=RITAPE851,RITAPE851:=RITAPE852,RITAPE8
```

```

FORMAT OUT F4, 'MATRICES NOT COMPATIBLE')
IF MATM2*MATM3 THEN GO TO GOOD
ZROCMATM7,MATM1,MATM4)
BEGIN
  FOR I=0 STEP 1 UNTIL MATM1-1 DO
    FOR JI=0 STEP 1 UNTIL MATM1-1 DO
      FOR KI=0 STEP 1 UNTIL MATM2-1 DO
        MATM7 := MATM7 + (MATM5 * MATM6)
      END
    GO TO GOOD
  END
  GOOD: WRITE(UNIT,F4)
  GOOD: END
  LASTOFLINE = 1

```

 COMMENT: *****

```

PROCEDURE LGAP;
COMMENT: COMPUTE CONCENTRATED NODAL FORCES EQUIVALENT TO
        BODY FORCES ON STRUCTURE. * * * * *
BEGIN

```

```

        FORUC(1,1,0,FRM1)
        F(1,NCOLN1) := 0.;
        FORUC(1,0,1,NELEMM1)
        BEGIN
            REAL VOL,DELTA,VT,UT;
            FORUC(1,0,1,2)
            BEGIN
                XE2(2*1) := X(NOD(LK,1))-1.0;
                XE2((2*1)+1) := X(NOD(LK,1))-1.1;
            END
            VOL := 0.166667 * THICK(1) * DET(XE2,DELTA);
            VT := VOL * FVUY;
            UT := VOL * FVUX;
            FOR J := 0 STEP 1 UNTIL 2 DO
                BEGIN
                    NJ := 2 * (NOD(1,2))-1;
                    F( NJ ,NCOLN) := F( NJ ,NCOLN)
                        + UT;
                    F(NJ+1 ,NCOLN) := F((NJ+1) ,NCOLN) + VT;
                END;
            END;
        END LGAP;

```

```

PROCEDURE PRINC(D,FLGG);
COMMENT: COMPUTE PRINCIPAL STRESSES OR STRAINS FROM X,Y, AND SHAR
        STRESSES OR STRAINS. FLGG=0.,STRESSES, FLGG=1.,STRAINS.
        MPAL ARRAY D(0); INTEGER FLAG;
        BEGIN
            REAL RTE,MEEN,THETA;
            MEEN := (D(0) + D(1))*0.5;
            RTE := SQRT(((D(0) - D(1))*0.5)*2)+(D(2)*2);
            IF RTE = 0
            THEN D(2) := 0.5 * ARCTAN(D(2)/(D(0)-D(1)))
            ELSE D(2) := 0.5*ARCTAN(D(2)/(D(0)-D(1)));
            D(0) := MEEN + RTE;
            D(1) := MEEN - RTE;
            END OF PRINC;

```

```

COMMENT: COMPUTE STRESSES (C(0,0),ELENSTR)
      (SEJ = STRESS OR STRAIN MATRIX.
      (Q) = MODAL VELOCITY VECTOR
      (ELENSTR) = MATRIX FOR RESULTANT STRESSES OR STRAINS. * * * *)
REAL ARRAY C(0,0),SE,FILE,STRM(0,0)
      BEGIN
      INTEGER I,J,K
      FORU(1,0,1,2)
      BEGIN
      Q(I(2,1)) = Q(2*(MOD(LK,1))-1)
      Q(I(2,1+1)) = Q(2*(MOD(LK,1))-1)
      END
      MATM(N,N,0,0,SE,FILE,STRM(0,0),J(TK),ELENSTR(LK,1))
      END

```

```

PROCEDURE FEM COMMENT: COMPUTE ELEMENT STIFFNESS MATRIX.
      BEGIN
      REAL EC,DEVC,CS,SS,OMX,OMY,DELTA,VOL
      INTEGER N1,M1,M2,N2
      LABEL
      LOCAL NORDI
      COMMENT: CONVERT GLOBAL DATION COORDINATES TO LOCAL COORDINATES
      WITH ORIGIN AT CENTROID OF ELEMENT. * * * *)
      FORM(1,0,1,2) FORU(0,1,5) R(1,J)=0.
      COMMENT FORMATION OF (P)
      DEN = 1./((DET(XE2,DELTA)))
      R(0,0) = R(2,1) * (XE2(3) - XE2(5)) * DEN
      R(0,2) = R(2,3) * (XE2(5) - XE2(1)) * DEN
      R(0,4) = R(2,5) * (XE2(1) - XE2(3)) * DEN
      R(2,0) = R(1,1) * (XE2(4) - XE2(2)) * DEN
      R(2,2) = R(1,3) * (XE2(0) - XE2(4)) * DEN
      R(2,4) = R(1,5) * (XE2(2) - XE2(0)) * DEN
      BEGIN
      LABEL
      PSTRAIN,ROTAT
      FORU(1,0,1,2) D(1,1) = 0.
      IF NP THEN GO TO PSTRAIN
      COMMENT FORMATION OF (P) = PLASTICITY MATRIX PLANE STRESS CASE
      DEN = 1. / (1. - (PHI*2))
      D(1,1) = VM1 * DEN
      D(1,2) = (VM1/2. * (1. + PHI))
      GO TO ROTAT
      COMMENT FORMATION OF (P) = PLASTICITY MATRIX PLANE STRAIN CASE
      PSTRAIN
      DEN = 1. / ((1. + PHI) * (1. - (2. * PHI)))
      D(1,1) = VM1 * (1. - PHI) * DEN
      D(1,2) = VM1 * PHI * DEN
      D(1,3) = (VM1/2. * (1. + PHI))
      ROTAT
      END
      COMMENT FORMATION OF ROTATION MATRIX TAKES PLACE HERE IF DEFSTROD
      N1 = 3
      N2 = 3
      N1 = 3
      N2 = 6
      FORU(1,0,2,4) SE(0,1) = D(1,1) * R(0,1)
      FORU(1,2,5) SE(0,1) = D(1,1) * R(1,1)
      FORU(1,2,4) SE(1,1) = D(1,1) * R(0,1)
      FORU(1,1,2,5) SE(1,1) = D(1,1) * R(1,1)
      FORU(1,0,1,5) SE(2,1) = D(1,2) * R(2,1)
      COMMENT (SE) = (D(1,1) * SE) = STRESS MATRIX
      IF WANTSTRAIN = 1 THEN
      STOP(SAV,STRMSTRM(0,0),FM)
      COMMENT: STORE STRAIN MATRIX ON DISK FILE FOR LATER
      COMMENT: COMPUTATION OF STRAINS. * * * * *
      IF WANTSTRESS = 1 THEN
      STOP(SAV,STRMSTRM(0,0),FM)
      COMMENT: STORE STRESS MATRIX ON DISK FILE FOR LATER
      COMMENT: COMPUTATION OF STRESSES. * * * * *
      N1 = 6
      MATM(N,N,0,0,M1,M2,1,1,SE,K,J,C(1,0))
      VOL = (0.5 * 1H * DELTA)
      FORU(1,0,1,5) FORU(0,1,5)
      C(1,1) = C(1,1) * VOL
      COMMENT (C) = (C(1,1) * C) * VOL = STIFF MATRIX
      END OF FEM

```



```

      WRITE(LINE,PARFORMC2)
      FOR(I=0,1,NBOUND1)
      BEGIN
        READ (CARD,PARFORM1,NF(I),NB(I,0),NF(I,1),RV(I,0),
             BV(I,1),BOUNDARY)(FINISH)
        WRITE(LINE,PARFORM1,NF(I),NB(I,0),NF(I,1),RV(I,0),
             BV(I,1),BOUNDARY)
        IF BOUNDARY = 1 THEN CNDF := NF(I);
      END;
      IF NCARD ≠ NBOUND THEN GO TO GUF3;

      COMMENT: *****
      PARTITIONING DATA: *****
      NSTART = FIRST ELEMENT IN EACH PARTITION.
      NEND = LAST ELEMENT IN EACH PARTITION
      NFIRST = FIRST NODAL POINT IN EACH PARTITION.
      NLAST = LAST NODAL POINT IN EACH PARTITION.
      *****

      READ (CARD,PARFORM1,NSTART(I),NEND(I),NFIRST(I),
           NLAST(I))(FINISH)
      WRITE(LINE,PARFORM1,NSTART(I),NEND(I),NFIRST(I),
           NLAST(I))
      PARTITION := NSTART(I)-1,NLAST(I)+1,PARFORM1)
      END;

      IF NCARD ≠ NPART THEN GO TO GUF4;

      COMMENT: *****
      MATERIAL PARAMETERS: E1,E2 = VISCOSITY IN X,Y DIRECTIONS
                        P1,P2 = POISSON RATIO IN X,Y DIRECTIONS
      *****

      READ (CARD,PARFORM1,NCARD)(FINISH)
      WRITE(LINE,PARFORM1)
      FOR(I=0,1,NYMM1)
      BEGIN
        READ (CARD,PARFORM1,E1(I),E2(I),P1(I),P2(I),RF(I))
        (FINISH)
        WRITE(LINE,PARFORM1,I+1,E1(I),E2(I),P1(I),P2(I),
             RF(I))
      END
      IF NOT LINEAR THEN
        COMMENT: CONVERT VISCOSITY IN POISES TO VISCOSITY IN KILOGRAM-POISES.
      BEGIN
        E1(I) := E1(I) × 9.8; E2(I) := E2(I) × 9.8;
      END;
      END;
      IF NCARD ≠ NYM THEN GO TO GUF5;

      COMMENT: *****
      CONCENTRATED FORCES:
      (F) = ARRAY OF FORCE VECTORS TO BE APPLIED TO STRUCTURE.
      *****
      FOR(I=0,1,NCOLNM1)
      BEGIN

```

```

      L BEL
      BRANCH:
      READ (CARD,PARFORM1,NCARD)(FINISH)
      WRITE(LINE,PARFORM1)
      IF NCARD ≠ 0 THEN
      BEGIN
        FOR(I=0,1,NDFRM1)
        FOR(J=0,1,N)
        FOR(K=0,1,NCONCM1)
        READ (CARD,PARFORM1,K,F1(X(K-1),J),F2(X(K-1),J))
        (FINISH)
        WRITE(LINE,PARFORM1,K,F1(X(K-1),J),F2(X(K-1),J))
      END;
      IF NCARD ≠ NCONC THEN GO TO GUF6;
      END;
      END;
      BEGIN
        LABEL
        NOLPAD:
        IF FPUV = 0 AND FPUX = 0 THEN GO TO NOLPAD;
        LOAD:
        NCOLN := NCOLN + 1;
        PTHX := PTHX + 1;
      END;
      COMMENT: STRUCTURE BLOCK ANALYSES INITIAL STRUCTURE, PARTITION BY
      PARTITION;
      BEGIN LABEL STRUCT;
      INTEGER ARRAY
      REAL ARRAY
      PTHX(NPDM1,0,PTMAX),COLMATION(NPDM1)
      RTION(NPDM1,0,(4*PTMAX)-1))
      COMMENT:
      * * * * * (PT) = INDEX MATRIX REFERENCEING NON-ZERO STIFFNESS COEFF-
      ICIENTS STORED IN (PT).
      (COLMATION) = WHAT COLUMN OF EACH ROW OF (PT) IS TO BE REFERENCE-
      ED NEXT, USED IN SOLVER.
      (RT) = UPPER TRIANGULAR HALF OF NON-ZERO STIFFNESS COEFF.
      * * * * *
      CYCLE := 1;
      NUL TWO:
      MAKEPT := MAKE INDEX MATRIX.
      ZROK(X(J),N,NPDM1)
      IF PROMPTIE NEW 0
      THEN
      BEGIN
        READ RESTART DATA FROM DISK FILE.
        IF PROMPTIE = 1 THEN BEGIN REWIND(COORU(1));
        FOR(I=0,1,NPDM1)
        READ(COORU(1),X(I,0),X(I,1),GART(I)) ENR;
        REWIND(COORU(1));
        REWIND(PN(1));
        REWIND(VIC(1));
        READ(COORU(1),CYCLE,CYCLE,CYCLE);
        FOR(LX=0,1,NLEMM1)
        READ(COORU(1),FLEMM1,FLEMM1,FLEMM1);
        READ(COORU(1),REFFQ(1));
        END
      END
      END OF RESTART DATA BLOCK;
      IF NCARD NEQ J THEN NDESONROUND;

```


PROBLEM NUMBER 1 L=94, I=30, H=1.67, AMP=0.1H =0.167
 MODAL
 PARTITIONS 4 POINTS 125 ELEMENTS 256
 BOUNDARY CONDITIONS 42
 FORCE VECTORS 1
 U.D.F. OFR 2
 MODAL POINT FORCES 0
 FORCE PLR 0.000000
 FORCE PER VOLUME IN X 0.000000
 FORCE PER VOLUME IN Y 0.000000

NON-LINEAR PROBLEM

PLANE STRAIN PROBLEM

RESTART DATA WILL BE SAVED.

PROFILE IS 0
 ALL DATA READ FROM CAPS.

STRESSES WILL BE COMPUTED.

STRAINS WILL BE COMPUTED.

DELTA TIME NTIMES
 1.00E+00 1

RELAXATION FACTOR EPSILON ITERATIONS
 1.750 0.00010 200

CONSTANT A ACTIV. ENERGY TEMPERATURE POWER
 1.94900E+12 6.20000E+04 5.00000E+02 8
 6.49000E-02 4.50000E+04 5.00000E+02 5

STRAIN RATE LOG STRAIN
 1.00000E-14 0.050000

DATUM COORDINATES		BAR	
MODE	X	Y	
1	-7.500000	23.000000	0
2	0.000000	23.000000	0
3	7.500000	23.000000	0
4	-7.500000	18.000000	0
5	-3.500000	18.000000	0
6	3.500000	18.000000	0
7	7.500000	19.000000	0
8	-7.500000	12.800000	0
9	-4.500000	12.800000	0
10	0.000000	12.800000	0
11	4.500000	12.800000	0
12	7.500000	12.800000	0
13	-7.500000	9.000000	0
14	-5.000000	9.000000	0
15	-1.750000	9.000000	0
16	1.750000	9.000000	0
17	5.000000	9.000000	0
18	7.500000	9.000000	0
19	-7.500000	5.500000	0
20	-6.000000	5.500000	0
21	-3.900000	5.500000	0
22	0.000000	5.500000	0
23	3.900000	5.500000	0
24	6.000000	5.500000	0
25	7.500000	5.500000	0

SHFAR MODULUS
0.00000
0.00000

Y
0.40000
0.40000

X

Y
1.00000
02.00000

NEP
1
2

1.00000
02.00000

CONCENTRATED NODAL FORCES
NODE
FX
FY

EQUIVALENT VISCOSITY FOR MATERIAL 1 = 2.9777510+12
EQUIVALENT VISCOSITY FOR MATERIAL 2 = 3.0391400+13

ENERGY(1) = 5.6270200-04
ENERGY(1) = 1.2077050-10

PROCESSOR TIME = 58.5167 SECONDS.

ITERATION = 1	U = -12.975000	V = -10.632734	EPSTILN = 12.133164
ITERATION = 2	U = -3.503250	V = 12.604183	EPSTILN = 30.475233
ITERATION = 3	U = -10.417027	V = -11.169321	EPSTILN = 30.851961
ITERATION = 4	U = -5.370132	V = 4.229619	EPSTILN = 23.991145
ITERATION = 5	U = -9.054804	V = -4.224408	EPSTILN = 13.304373
ITERATION = 6	U = -6.364993	V = 2.029214	EPSTILN = 8.876111
ITERATION = 7	U = -8.328555	V = -3.783712	EPSTILN = 7.123934
ITERATION = 8	U = -6.895155	V = 0.083294	EPSTILN = 5.570400
ITERATION = 9	U = -7.941537	V = -2.503050	EPSTILN = 4.422064
ITERATION = 10	U = -7.177674	V = -0.288239	EPSTILN = 3.409551
ITERATION = 11	U = -7.735295	V = -2.010948	EPSTILN = 2.816882
ITERATION = 12	U = -7.328235	V = -0.688337	EPSTILN = 2.109586
ITERATION = 13	U = -7.625389	V = -1.080225	EPSTILN = 1.958482
ITERATION = 14	U = -7.008464	V = -2.005326	EPSTILN = 1.718822
ITERATION = 15	U = -7.566820	V = -1.022634	EPSTILN = 1.629049
ITERATION = 16	U = -7.451222	V = -0.992423	EPSTILN = 1.336499
ITERATION = 17	U = -7.535608	V = -1.694567	EPSTILN = 0.902274
ITERATION = 18	U = -7.474006	V = -1.134033	EPSTILN = 0.613794
ITERATION = 19	U = -7.510976	V = -1.504361	EPSTILN = 0.402773
ITERATION = 20	U = -7.486144	V = -1.234574	EPSTILN = 0.513868
ITERATION = 21	U = -7.510112	V = -1.064103	EPSTILN = 0.440272
ITERATION = 22	U = -7.492619	V = -1.327461	EPSTILN = 0.260174
ITERATION = 23	U = -7.502389	V = -1.107739	EPSTILN = 0.233220
ITERATION = 24	U = -7.490066	V = -1.163595	EPSTILN = 0.161884
ITERATION = 25			

```

ITERATION = 176
U = -7.500000 V = -0.922033 EPSILON = 0.000752
ITERATION = 179
U = -7.500000 V = -0.920757 EPSILON = 0.000745
ITERATION = 180
U = -7.500000 V = -0.919495 EPSILON = 0.000737
ITERATION = 181
U = -7.500000 V = -0.918245 EPSILON = 0.000730
ITERATION = 182
U = -7.500000 V = -0.917008 EPSILON = 0.000722
ITERATION = 183
U = -7.500000 V = -0.915784 EPSILON = 0.000715
ITERATION = 184
U = -7.500000 V = -0.914572 EPSILON = 0.000708
ITERATION = 185
U = -7.500000 V = -0.913372 EPSILON = 0.000700
ITERATION = 186
U = -7.500000 V = -0.912184 EPSILON = 0.000693
ITERATION = 187
U = -7.500000 V = -0.911009 EPSILON = 0.000686
ITERATION = 188
U = -7.500000 V = -0.909845 EPSILON = 0.000679
ITERATION = 189
U = -7.500000 V = -0.908693 EPSILON = 0.000672
ITERATION = 190
U = -7.500000 V = -0.907553 EPSILON = 0.000666
ITERATION = 191
U = -7.500000 V = -0.906424 EPSILON = 0.000659
ITERATION = 192
U = -7.500000 V = -0.905307 EPSILON = 0.000652
ITERATION = 193
U = -7.500000 V = -0.904201 EPSILON = 0.000646
ITERATION = 194
U = -7.500000 V = -0.903106 EPSILON = 0.000639
ITERATION = 195
U = -7.500000 V = -0.902022 EPSILON = 0.000633
ITERATION = 196
U = -7.500000 V = -0.900950 EPSILON = 0.000626
ITERATION = 197
U = -7.500000 V = -0.899888 EPSILON = 0.000620
ITERATION = 198
U = -7.500000 V = -0.898837 EPSILON = 0.000614
ITERATION = 199
U = -7.500000 V = -0.897796 EPSILON = 0.000607
ITERATION = 200
U = -7.500000 V = -0.896767 EPSILON = 0.000601
PROCESSOR TIME = 518.4343 SECONDS.

```

```

NODE VELOCITIES
U
1 7.500000E-14 1.537977E-13
2 4.793214E-14 1.533751E-13
3 -7.500000E-14 1.529531E-13
4 7.500000E-14 1.207037E-13
5 3.535567E-14 1.205384E-13
6 -3.464350E-14 1.105452E-13
7 -7.500000E-14 1.153107E-13
8 7.500000E-14 8.691046E-14
9 4.557840E-14 9.455900E-14
10 0.562054E-14 8.537643E-14
V

```

ELF	SIGMA 1	SIGMA 2	THETA	DFLSIG	SIGMA-X	SIGMA-Y	SIGMA-Z
72	-0.819121e-14	3.762402e-15					
73	-5.460216e-14	3.198718e-15					
74	-6.102076e-14	2.830003e-15					
75	-6.706415e-14	2.572335e-15					
76	-7.500000e-14	2.500766e-15					
77	7.500000e-14	1.732890e-14					
78	7.126401e-14	1.735212e-14					
79	6.566068e-14	1.716276e-14					
80	6.000873e-14	1.688402e-14					
81	5.438803e-14	1.660691e-14					
82	4.865650e-14	1.583656e-14					
83	-4.552949e-14	1.423384e-15					
84	-5.181273e-14	8.336776e-16					
85	-5.811641e-14	3.900182e-16					
86	-6.448102e-14	7.358073e-17					
87	-7.077024e-14	-1.083482e-16					
88	-7.500000e-14	-1.250513e-17					
89	7.500000e-14	1.315608e-14					
90	6.823944e-14	1.305672e-14					
91	6.243728e-14	1.281128e-14					
92	5.682753e-14	1.241570e-14					
93	5.080671e-14	1.189765e-14					
94	4.056753e-14	1.295150e-14					
95	2.936215e-14	1.132430e-14					
96	1.951948e-14	9.708633e-15					
97	7.602431e-15	7.586931e-15					
98	-3.489003e-15	5.568203e-15					
99	-1.563175e-14	3.839180e-15					
100	-2.584060e-14	1.831642e-15					
101	-3.765265e-14	1.864100e-16					
102	-4.939327e-14	-3.887508e-15					
103	-5.553720e-14	-4.015518e-15					
104	-6.168355e-14	-4.408415e-15					
105	-6.782770e-14	-4.660718e-15					
106	-7.500000e-14	-4.758637e-15					
107	7.500000e-14	8.970254e-15					
108	7.101878e-14	9.002427e-15					
109	6.504818e-14	8.888206e-15					
110	5.907395e-14	8.888206e-15					
111	5.309267e-14	8.017025e-15					
112	4.311103e-14	6.862836e-15					
113	3.310834e-14	5.656435e-15					
114	2.208856e-14	3.939408e-15					
115	1.105389e-14	2.042302e-15					
116	-4.105403e-14	1.226706e-15					
117	-1.105395e-14	-2.042066e-15					
118	-2.208863e-14	-3.939188e-15					
119	-3.310830e-14	-5.655640e-15					
120	-4.311103e-14	-6.862021e-15					
121	-5.309265e-14	-8.017623e-15					
122	-5.907389e-14	-8.888152e-15					
123	-6.504810e-14	-8.888223e-15					
124	-7.101872e-14	-9.003909e-15					
125	-7.500000e-14	-9.867666e-15					

TIME FOR 5.00 PERCENT DEFORMATION =17/935.947 YEARS

1	-9.20130e-04	-0.09508	0.06758	0.09507	-0.09508	-0.09501	-0.00016
2	-0.00030	-0.09551	0.13221	0.09521	-0.09551	-0.00022	-0.00022
3	1.96845e-06	-0.09597	0.20156	0.09521	-0.09592	7.81507e-07	-0.00034
4	0.00030	-0.09634	0.13041	0.09664	-0.09634	0.00030	-0.00022
5	0.00001	-0.09677	0.09608	0.09674	-0.09677	0.00001	-0.00016
6	-0.00050	-0.09681	0.20219	0.09391	-0.09681	-0.00050	-0.00034
7	-0.00107	-0.09579	0.20228	0.09472	-0.09579	-0.00107	-0.00033
8	-0.00043	-0.09542	0.55854	0.09490	-0.09542	-0.00043	-0.00093
9	2.83104e-06	-0.09593	0.42333	0.09593	-0.09593	-1.28502e-06	-0.00074
10	0.00046	-0.09644	0.54010	0.09690	-0.09644	0.00045	-0.00093
11	0.00107	-0.09607	0.19756	0.09714	-0.09607	0.00107	-0.00033
12	-0.00051	-0.09744	0.20147	0.09795	-0.09744	0.00051	-0.00034
13	-0.00171	-0.09384	0.33150	0.09213	-0.09384	-0.00172	-0.00054
14	-0.00261	-0.09582	0.37834	0.09321	-0.09582	-0.00261	-0.00062
15	-0.00117	-0.09435	0.97550	0.09318	-0.09432	-0.00120	-0.00159
16	-0.00277	-0.09567	0.84390	0.09490	-0.09565	-0.00279	-0.00140
17	0.00003	-0.09597	1.14464	0.09600	-0.09593	-8.41788e-06	-0.00192
18	0.00080	-0.09623	0.82442	0.09703	-0.09621	0.00078	-0.00140
19	0.00123	-0.09755	0.92131	0.09878	-0.09753	0.00120	-0.00159
20	0.00262	-0.09604	0.35759	0.09866	-0.09603	0.00262	-0.00062
21	0.00174	-0.09800	0.30009	0.09974	-0.09800	0.00173	-0.00054
22	-0.00285	-0.09287	0.40417	0.09003	-0.09287	-0.00285	-0.00064
23	-0.00465	-0.09581	0.47460	0.09115	-0.09580	-0.00466	-0.00076
24	-0.00171	-0.09279	1.12204	0.09108	-0.09276	-0.00175	-0.00178
25	-0.00369	-0.09604	1.11962	0.09235	-0.09601	-0.00373	-0.00179
26	-0.00050	-0.09455	1.65705	0.09404	-0.09447	-0.00058	-0.00272
27	0.00005	-0.09597	1.23184	0.09602	-0.09592	1.94851e-06	-0.00206
28	0.00068	-0.09750	1.58441	0.09814	-0.09743	0.00056	-0.00272
29	-0.00376	-0.09587	1.02745	0.09684	-0.09584	0.00373	-0.00179
30	0.00179	-0.09910	1.01118	0.10089	-0.09907	6.00176	-0.00178
31	0.00468	-0.09604	0.42757	0.10772	-0.09604	0.00467	-0.00075
32	0.00288	-0.09890	0.36474	0.1178	-0.09889	0.00288	-0.00062
33	-0.00451	-0.09286	0.42492	0.08835	-0.09286	-0.00452	-0.00046
34	-0.00581	-0.09485	0.49499	0.09202	-0.09485	-0.00582	-0.00077
35	-0.00386	-0.09274	0.75456	0.09688	-0.09273	-0.00386	-0.00124
36	-0.00621	-0.09580	1.15491	0.09591	-0.09576	-0.00625	-0.00181
37	-0.00315	-0.09320	1.51786	0.09004	-0.09313	-0.00322	-0.00238
38	-0.00396	-0.09458	2.22416	0.09063	-0.09445	-0.00399	-0.00351
39	-0.00261	-0.09604	2.18716	0.09344	-0.09591	-0.00274	-0.00356
40	-0.00096	-0.09502	2.38712	0.09598	-0.09485	-0.00080	-0.00393
41	-0.00066	-0.09745	2.34155	0.09679	-0.09728	-0.00082	-0.00395
42	0.00301	-0.09600	2.08455	0.09701	-0.09588	0.00289	-0.00347
43	0.00415	-0.09770	1.94123	0.10185	-0.09758	0.00403	-0.00345
44	0.00341	-0.09965	1.31481	0.1206	-0.09860	0.00336	-0.00234
45	0.00630	-0.09609	0.97779	0.10239	-0.09606	0.00627	-0.00175
46	0.00411	-0.09889	0.66394	0.10300	-0.09887	0.00410	-0.00119
47	0.00592	-0.09687	0.40229	0.10280	-0.09687	0.00592	-0.00072
48	0.00463	-0.09879	0.33432	0.11342	-0.09879	0.00462	-0.00040
49	-0.00554	-0.09424	0.43428	0.08764	-0.09424	-0.00559	-0.00046
50	-0.00536	-0.09298	0.59499	0.08762	-0.09297	-0.00537	-0.00091
51	-0.00768	-0.09530	0.89765	0.08762	-0.09527	-0.00770	-0.00118
52	-0.00431	-0.09267	1.28423	0.08434	-0.09262	-0.00436	-0.00198
53	-0.00812	-0.09655	1.78161	0.08442	-0.09646	-0.00821	-0.00275
54	-0.00186	-0.09250	2.09713	0.09064	-0.09238	-0.00198	-0.00331
55	-0.00580	-0.09580	2.57095	0.09020	-0.09562	-0.00578	-0.00404
56	-0.00430	-0.09430	2.35360	0.09415	-0.09415	-0.00436	-0.00379
57	-0.00784	-0.09784	2.78438	0.09481	-0.09761	-0.00785	-0.00460
58	-0.00435	-0.09435	2.74010	0.09465	-0.09413	-0.00461	-0.00441
59	-0.00866	-0.09866	2.61788	0.09824	-0.09866	-0.00866	-0.00448
60	-0.00762	-0.09762	2.61788	0.10350	-0.09768	-0.00768	-0.00372
61	-0.00962	-0.00962	2.15830	0.10261	-0.009647	0.00367	-0.00385

ELF	STRAIN-1	STRAIN-2	THETA	STRAIN DIFF.	STRAIN-Y	STRAIN-X	STRAIN-Y	STRAIN-X
186	-0.01141	-1.15058	-0.76131	1.15917	-1.15641	-0.01158	0.01394	0.01394
187	0.00917	-1.11592	-0.81269	1.1509	-1.11570	0.00894	0.01596	0.01596
188	-0.00729	-1.14784	-0.78957	1.18055	-1.14762	-0.00750	0.01571	0.01571
189	0.00324	-1.12876	-0.86065	1.13200	-1.12850	0.00299	0.01700	0.01700
190	-0.00229	-1.14559	-0.81740	1.14331	-1.14536	-0.00252	0.01421	0.01421
191	0.00051	-1.14351	-0.85168	1.14402	-1.14325	0.00026	0.01700	0.01700
192	-0.00262	-1.14233	-0.81736	1.14494	-1.14209	-0.00238	0.01633	0.01633
193	-0.00379	-1.15897	-0.83719	1.15518	-1.15872	-0.00403	0.01688	0.01688
194	-0.00599	-1.13902	-0.78753	1.14898	-1.13878	-0.00577	0.01573	0.01573
195	-0.01011	-1.17097	-0.80780	1.16087	-1.17074	-0.01034	0.01636	0.01636
196	0.01142	-1.13537	-0.69463	1.14679	-1.13520	0.01125	0.01390	0.01390
197	-0.00450	-1.17804	-0.67935	1.15954	-1.17788	-0.00866	0.01375	0.01375
198	0.01449	-1.13299	-0.58185	1.16788	-1.13287	0.01437	0.01165	0.01165
199	-0.00866	-1.17821	-0.62919	1.16884	-1.17807	-0.008750	0.01263	0.01263
200	0.00565	-1.13866	-0.32023	1.14352	-1.13863	0.00482	0.00639	0.00639
201	-0.01311	-1.13278	-0.37743	1.16686	-1.13245	-0.01370	0.00769	0.00769
202	-0.01486	-1.13345	-0.36239	1.14590	-1.13275	-0.01308	0.00605	0.00605
203	-0.01352	-1.13178	-0.25566	1.1859	-1.13142	-0.01488	0.00521	0.00521
204	-0.01614	-1.13178	-0.21763	1.16530	-1.13177	-0.01650	0.00435	0.00435
205	-0.01497	-1.18327	-0.15294	1.16813	-1.18326	-0.01497	0.00312	0.00312
206	-0.01024	-1.13054	-0.08492	1.14551	-1.13054	-0.01024	0.00138	0.00138
207	0.00201	-1.18139	-0.10112	1.17115	-1.18138	0.00201	0.00207	0.00207
208	6.66699e-15	-1.13969	-0.15440	1.14170	-1.13968	0.00200	0.00312	0.00312
209	6.66709e-15	-0.91127e-15	0.09758	1.45183e-14	-0.91108e-15	6.66699e-15	STRAIN-Y	STRAIN-X
210	6.66747e-15	-0.93648e-15	0.11321	1.45335e-14	-0.93609e-15	6.66747e-15	6.66699e-15	6.66699e-15
211	6.66769e-15	-1.00007e-15	0.20156	1.66682e-14	-0.99986e-15	6.66769e-15	6.66699e-15	6.66699e-15
212	6.66788e-15	-1.00644e-15	0.37441	1.67912e-14	-1.00639e-15	6.66788e-15	6.66699e-15	6.66699e-15
213	6.66807e-15	-1.00644e-15	0.09408	1.67162e-14	-1.00639e-15	6.66807e-15	6.66699e-15	6.66699e-15
214	6.66826e-15	-1.00644e-15	0.20919	1.68180e-14	-1.00639e-15	6.66826e-15	6.66699e-15	6.66699e-15
215	6.66845e-15	-1.00644e-15	0.20228	1.68180e-14	-1.00639e-15	6.66845e-15	6.66699e-15	6.66699e-15
216	6.66864e-15	-1.00644e-15	0.54454	1.65137e-14	-0.91604e-15	6.66864e-15	6.66699e-15	6.66699e-15
217	6.66883e-15	-1.00644e-15	0.42733	1.66742e-14	-0.99986e-15	6.66883e-15	6.66699e-15	6.66699e-15
218	6.66902e-15	-1.00644e-15	0.50110	1.66445e-14	-1.00639e-15	6.66902e-15	6.66699e-15	6.66699e-15
219	6.66921e-15	-1.00644e-15	0.19754	1.67802e-14	-1.00639e-15	6.66921e-15	6.66699e-15	6.66699e-15
220	6.66940e-15	-1.00644e-15	0.20147	1.67019e-14	-1.00639e-15	6.66940e-15	6.66699e-15	6.66699e-15
221	6.66959e-15	-1.00644e-15	0.33150	1.60103e-14	-0.66322e-15	6.66959e-15	6.66699e-15	6.66699e-15
222	6.66978e-15	-1.00644e-15	0.37434	1.61998e-14	-0.66322e-15	6.66978e-15	6.66699e-15	6.66699e-15
223	6.66997e-15	-1.00644e-15	0.97550	1.42174e-14	-0.74951e-15	6.66997e-15	6.66699e-15	6.66699e-15
224	6.67016e-15	-1.00644e-15	0.64390	1.45094e-14	-0.91604e-15	6.67016e-15	6.66699e-15	6.66699e-15
225	6.67035e-15	-1.00644e-15	1.44664	1.67201e-14	-1.00000e-15	6.67035e-15	6.66699e-15	6.66699e-15
226	6.67054e-15	-1.00644e-15	0.8242	1.67058e-14	-1.00000e-15	6.67054e-15	6.66699e-15	6.66699e-15
227	6.67073e-15	-1.00644e-15	0.92131	1.71898e-14	-1.00000e-15	6.67073e-15	6.66699e-15	6.66699e-15
228	6.67092e-15	-1.00644e-15	0.35759	1.71455e-14	-1.00000e-15	6.67092e-15	6.66699e-15	6.66699e-15
229	6.67111e-15	-1.00644e-15	1.58401	1.71303e-14	-1.00000e-15	6.67111e-15	6.66699e-15	6.66699e-15
230	6.67130e-15	-1.00644e-15	1.02745	1.73445e-14	-1.00000e-15	6.67130e-15	6.66699e-15	6.66699e-15
231	6.67149e-15	-1.00644e-15	1.01118	1.75624e-14	-1.00000e-15	6.67149e-15	6.66699e-15	6.66699e-15
232	6.67168e-15	-1.00644e-15	0.42757	1.75057e-14	-1.00000e-15	6.67168e-15	6.66699e-15	6.66699e-15
233	6.67187e-15	-1.00644e-15	0.34478	1.76879e-14	-1.00000e-15	6.67187e-15	6.66699e-15	6.66699e-15
234	6.67206e-15	-1.00644e-15	0.42492	1.75555e-14	-1.00000e-15	6.67206e-15	6.66699e-15	6.66699e-15
235	6.67225e-15	-1.00644e-15	0.40499	1.74777e-14	-1.00000e-15	6.67225e-15	6.66699e-15	6.66699e-15
236	6.67244e-15	-1.00644e-15	0.79456	1.74611e-14	-1.00000e-15	6.67244e-15	6.66699e-15	6.66699e-15
237	6.67263e-15	-1.00644e-15	1.15491	1.76027e-14	-1.00000e-15	6.67263e-15	6.66699e-15	6.66699e-15
238	6.67282e-15	-1.00644e-15	1.51786	1.77104e-14	-1.00000e-15	6.67282e-15	6.66699e-15	6.66699e-15

73	5.326949	1.529994	0
74	5.874874	1.521150	0
75	6.452644	1.514862	0
76	7.125000	1.512574	0
77	-7.125000	1.557644	0
78	-6.743640	1.551761	0
79	-6.171796	1.544814	0
80	-5.599556	1.530225	0
81	-5.028260	1.517035	0
82	-4.456717	1.517183	0
83	0.472353	1.144117	0
84	5.040936	1.150169	0
85	5.609416	1.147950	0
86	6.177795	1.137368	0
87	6.746149	1.131478	0
88	7.125000	1.125597	0
89	-7.125000	0.845740	0
90	-6.456803	0.843284	0
91	-5.887814	0.854036	0
92	-5.316862	0.860078	0
93	-4.745944	0.874488	0
94	-4.169712	1.174758	0
95	-3.603189	1.134622	0
96	-3.026403	1.097519	0
97	-2.451194	1.045935	0
98	0.532755	0.907841	0
99	1.671841	0.947191	0
100	2.620797	0.809150	0
101	3.711737	0.867932	0
102	4.753034	0.471562	0
103	5.322314	0.458922	0
104	5.691542	0.450978	0
105	5.460841	0.444696	0
106	7.125000	0.443227	0
107	-7.125000	0.211851	1
108	-6.744906	0.212012	1
109	-6.174759	0.204031	1
110	-5.604630	0.190421	1
111	-5.034537	0.187055	1
112	-4.084445	0.164814	1
113	-3.134459	0.131242	1
114	-2.089557	0.091697	1
115	-1.044731	0.044212	1
116	-2.052701	6.133571	1
117	1.044730	-0.044210	1
118	2.039557	-0.031696	1
119	3.134458	-0.131242	1
120	4.084445	-0.164810	1
121	5.034537	-0.187058	1
122	5.604631	-0.190421	1
123	6.174760	-0.207031	1
124	6.744906	-0.212020	1
125	7.125000	-0.211838	1
I AM A STUDIO MACHINE.			
I AM A STUDIO MACHINE.			
PROFESSOR TIME #			
559.0167 SECONDS.			
I AM A STUDIO MACHINE.			