

(NASA-CR-158631) MANTLE: A FINITE ELEMENT  
PROGRAM FOR THE THERMAL-MECHANICAL ANALYSIS  
OF MANTLE CONVECTION. A USER'S MANUAL WITH  
EXAMPLES Final Report (Colorado State  
Univ.) 115 p HC A06/MF A01

N79-24297

Unclas  
CSCL 20D G3/34 20903

## MANTLE

A Finite Element Program for the  
Thermal-Mechanical Analysis of  
Mantle Convection

A User's Manual with Examples

Final Report: NASA Grant NGR 06-002-191\*

by

Erik Thompson

Department of Civil Engineering  
Colorado State University  
Fort Collins, Colorado 80523



May 1979

\*The NASA Technical Officer for this Grant is: Dr. Lloyd Carpenter,  
NASA, Goddard Space Flight Center.

MANTLE

A Finite Element Program for the  
Thermal-Mechanical Analysis of  
Mantle Convection

A User's Manual with Examples

Final Report: NASA Grant NGR 06-002-191<sup>\*</sup>

by  
Erik Thompson

Department of Civil Engineering  
Colorado State University  
Fort Collins, Colorado 80523

May 1979

---

<sup>\*</sup>The NASA Technical Officer for this Grant is: Dr. Lloyd Carpenter,  
NASA, Goddard Space Flight Center.

#### ACKNOWLEDGMENT

Development of Program MANTLE was supported by NASA Grant NGR 06-002-191. Computer funds were also made available by the Department of Civil Engineering, Colorado State University. The author is particularly grateful to Drs. Atuo Sato and Paul R. Dawson and to Mr. Yoji Shimazaki, who have made significant contributions to this research.

## TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGMENT . . . . .	i
INTRODUCTION . . . . .	1
GOVERNING EQUATIONS . . . . .	3
VARIATIONAL PRINCIPLES AND FINITE ELEMENT EQUATIONS . . . . .	9
PROGRAM MANTLE . . . . .	15
Introduction . . . . .	15
Flow Charts . . . . .	17
Tape Usage . . . . .	24
Easy Input Reference . . . . .	25
Input Description . . . . .	26
PROGRAM DRIVE--Input . . . . .	26
PROGRAM MESH1--Input . . . . .	27
PROGRAM MESH2--Input . . . . .	33
PROGRAM WAVE--Input . . . . .	39
PROGRAM COUPLE--Input . . . . .	42
Output Description . . . . .	51
PROGRAM MESH1 or MESH2--Output . . . . .	51
PROGRAM WAVE--Output . . . . .	54
PROGRAM COUPLE, CREEP, TEMP--Output . . . . .	57
Error Analysis . . . . .	62
User Subroutines and Functions . . . . .	66
FUNCTIONS GAMX AND GAMY . . . . .	66
FUNCTION G . . . . .	67
SUBROUTINE VISC . . . . .	67
FUNCTIONS RHO, CP, RKX, RKY . . . . .	68
SUBROUTINE MSHADJ . . . . .	69
SUBROUTINES BNDRY, BNDRYC, BNDRYT . . . . .	69
SUBROUTINE STIFF . . . . .	70
ILLUSTRATIVE EXAMPLES . . . . .	71
Pure Conduction . . . . .	71
Thick-Walled Cylinder . . . . .	73
Simple Convection . . . . .	75
Low Viscosity Zone . . . . .	80
Axisymmetric Convection . . . . .	82
Crustal Plates . . . . .	84
Upper Mantle Convection . . . . .	93
CONCLUSION . . . . .	97

	<u>Page</u>
APPENDIX A--MESH2 GENERATOR AND IORDER ARRAY . . . . .	99
APPENDIX B--QUADRATURE POINTS, NODAL POINTS, AND STRESS POINTS IN THE $\xi$ - $\eta$ PLANE . . . . .	107
APPENDIX C--USEFUL REFERENCES . . . . .	109
APPENDIX D--INDEX OF FORTRAN VARIABLES . . . . .	110
APPENDIX E--PROGRAM MANTLE LISTING (under separate cover)	

## INTRODUCTION

This manual comprises the final report on NASA Grant NGR 06-002-191. It describes a finite element computer code for the analysis of mantle convection. The coupled equations for creeping viscous flow and heat transfer can be solved for either a transient analysis or steady-state analysis. For transient analyses, either a control volume or a control mass approach can be used. Non-Newtonian fluids with viscosities which have thermal and spacial dependencies can be easily incorporated. All material parameters may be written as function statements by the user or simply specified as constants. A wide range of boundary conditions, both for the thermal analysis and the viscous flow analysis can be specified. For steady-state analyses, elastic strain rates can be included. Although this manual was specifically written for users interested in mantle convection, the code is equally well suited for analyses in a number of other areas including metal forming, glacial flows, and creep of rock and soil.

## GOVERNING EQUATIONS

In this section the governing equations for the coupled thermal mechanical behavior of creeping viscous flows are presented. They are provided primarily for definition, therefore only a minimum of discussion is given.

Creeping Elasto Viscoplastic Flow

Equilibrium:

$$\sigma_{ij,j} + \rho X_i = 0 \quad (1)$$

Incompressibility:

$$\epsilon_{ii} = 0 \quad (2)$$

Velocity, strain-rate and vorticity:

$$\epsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}) \quad (3)$$

$$\omega_{ij} = \frac{1}{2} (u_{j,i} - u_{i,j}) \quad (4)$$

Constitutive:

$$\epsilon_{ij} = \frac{1}{2\mu} \sigma'_{ij} + \frac{1}{2G} \sigma'_{ij}{}^v \quad (5)$$

$$\sigma'_{ij}{}^v = \frac{\partial \sigma_{ij}}{\partial t} + u_k \frac{\partial \sigma'_{ij}}{\partial x_k} - \sigma'_{ip} \omega_{pj} - \sigma'_{jp} \omega_{pi} \quad (6)$$

$$\sigma'_{ij} = \sigma_{ij} + p \delta_{ij} \quad (7)$$

$$p = -\frac{1}{3} \sigma_{ii} \quad (8)$$

$$\epsilon_{ij} = \epsilon_{ij}^e + \epsilon_{ij}^p \quad (9)$$

Boundary Conditions:

$$\sigma_{ij} v_j = T_i = C_{ij} (\bar{u}_j - u_j) \quad \text{on } S_u \quad (10)$$

$$\sigma_{ij} v_j = T_i = \bar{T}_i \quad \text{on } S_\sigma \quad (11)$$

All terms are defined in the table presented at the end of this section. It should be noticed that the equations for pure viscous flow are obtained by letting  $G \rightarrow \infty$ .

In the above equations, the viscosity can be a function of temperature and the invariants of the stress tensor. Often it is assumed that the second invariant of the stress deviator is the only stress invariant of importance. The following equations define this invariant (the effective stress) and its relationship to the effective strain rate,

$$\sigma_{\text{eff}} = \sqrt{\frac{3}{2} \sigma'_{ij} \sigma'_{ij}} \quad (12)$$

$$\epsilon_{\text{eff}} = \sqrt{\frac{2}{3} \epsilon_{ij} \epsilon_{ij}} \quad (13)$$

The relationship between the viscosity and the effective stress and the effective plastic strain rate can be found as follows.

$$\sigma'_{ij} = 2\mu \epsilon_{ij}^p \quad (14)$$

$$\sigma'_{ij} \sigma'_{ij} = 2\mu \epsilon_{ij}^p 2\mu \epsilon_{ij}^p \quad (15)$$

$$\frac{2}{3} \sigma_{\text{eff}}^2 = 4\mu^2 \frac{3}{2} \epsilon_{\text{eff}}^2 \quad (16)$$

$$\mu = \frac{\sigma_{\text{eff}}}{3\epsilon_{\text{eff}}^p} \quad (17)$$

It should be noted that for a state of uniaxial stress and incompressible plastic flow the effective stress and effective plastic strain rate reduce to the simplified form:

$$\sigma_{\text{eff}} = \sigma_{\text{axial}} \quad (18)$$

$$\epsilon_{\text{eff}}^p = \epsilon_{\text{axial}}^p \quad (19)$$



This convenient relationship allows any one dimensional constitutive theory to be interpreted as a relationship between the effective stress and effective plastic strain rate.

### Heat Transfer

Conductive, convective heat transfer:

$$\rho C_p \left[ \frac{\partial \theta}{\partial t} + u_i \frac{\partial \theta}{\partial x_i} \right] = \frac{\partial}{\partial x_i} \left[ k \frac{\partial \theta}{\partial x_i} \right] + Q \quad (20)$$

Boundary conditions:

$$\left[ \rho C_p u_i \theta - k \frac{\partial \theta}{\partial x_i} \right] v_i = q^* \quad (21)$$

or

$$- k \frac{\partial \theta}{\partial x_i} v_i = q \quad (22)$$

where

$$\left. \begin{array}{l} q^* = \bar{q}^* \\ \text{or} \\ q = \bar{q} \end{array} \right\} \quad \text{on } S_q \quad (23)$$

and

$$q = h(\theta - \theta_\infty) \text{ on } S_\theta \quad (24)$$

and

$$v_i = \text{unit outward normal vector to } S$$

Viscous heating:

$$Q = \sigma_{ij} \epsilon_{ij}^p = 2\mu \epsilon_{ij}^p \epsilon_{ij}^p \quad (25)$$

Table of Terms and SI Units

Symbol	Definition	SI Units
$C_{ij}$	Surface viscosity tensor	$N \cdot S/m^3$
$C_p$	Specific heat	$J/kg \cdot K$
$h$	Coefficient of convective surface heat transfer	$J/m^2 \cdot S \cdot K$
$k$	Coefficient of thermal conductivity	$J/m \cdot S \cdot K$
$p$	Pressure = negative mean normal stress	$N/m^2$
$q$	Heat flow out of volume through surface due to conduction	$J/m^2 \cdot S$
$\bar{q}$	Known $q$ on surface $S_q$	$J/m^2 \cdot S$
$q^*$	Heat flow out of volume through surface due to conduction and mass transport	$J/m^2 \cdot S$
$\bar{q}^*$	Known $q^*$ on $S_q$	
$Q$	Heat source per unit volume	$J/m^3 \cdot S$
$S_q$	Segment of surface where $q$ on $q^*$ is known	
$S_u$	Segment of surface where velocity is known	
$S_\theta$	Segment of surface where temperature is known	
$S_\sigma$	Segment of surface where surface traction is known	
$T_i$	Surface traction	$N/m^2$
$\bar{T}_i$	Known $T_i$ on $S_\sigma$	$N/m^2$
$u_i$	Velocity vector	$m/s$
$\bar{u}_i$	Known $u_i$ on $S_u$	$m/s$
$\epsilon_{ij}$	Total strain rate tensor	$1/s$
$\epsilon_{ij}^e$	Elastic strain rate tensor	$1/s$
$\epsilon_{ij}^p$	Plastic strain rate tensor	$1/s$

Symbol	Definition	SI Units
$\dot{\epsilon}_{\text{eff}}$	Effective strain rate	1/s
$\dot{\epsilon}_{\text{eff}}^p$	Effective plastic strain rate	1/s
$\theta$	Temperature	$^{\circ}\text{K}$ or $^{\circ}\text{C}$
$\theta_{\infty}$	Ambient temperature outside of volume	$^{\circ}\text{K}$ or $^{\circ}\text{C}$
$\mu$	Viscosity	$\text{N}\cdot\text{s}/\text{m}^2$
$\nu_i$	Outward unit normal vector to surface	1
$\rho$	Density	$\text{Kg}/\text{m}^3$
$\sigma_{ij}$	Stress tensor	$\text{N}/\text{m}^2$
$\sigma'_{ij}$	Deviatoric stress tensor	$\text{N}/\text{m}^2$
$\sigma_{\text{eff}}$	Effective stress	$\text{N}/\text{m}^2$
$\omega_{ij}$	Vorticity	1/s

PRECEDING PAGE BLANK NOT FILMED

# VARIATIONAL PRINCIPLES and FINITE ELEMENT EQUATIONS

The following section is included to provide a short summary of the finite element equations used in MANTLE. For a more complete development of these equations the reader should consult the references given in the Bibliography.

## Creeping Elasto Viscoplastic Flow

The finite element equations develop from the following variational statement:

$$\begin{aligned} \delta J = & \int_V \sigma_{ij} \delta \epsilon_{ij} dV - \int_V p \delta \epsilon_{ii} dV - \int_V \epsilon_{ii} \delta p dV \\ & - \int_S T_i \delta u_i dS - \int_V \rho X_i \delta u_i dV = 0 \end{aligned} \quad (26)$$

where

$$\begin{aligned} T_i &= \sigma_{ij} v_j = C_{ij} (\bar{u}_j - u_j) \quad \text{on } S_u \\ T_i &= \bar{T}_i \quad \text{on } S_\sigma \end{aligned} \quad (27)$$

The above expression can be derived from the governing equations via Galerkin's method or written directly as a statement of virtual work with the constraint of incompressibility incorporated through a Lagrange multiplier,  $p$ .

Substitution of the constitutive equation presented in the previous section gives us

$$\begin{aligned} \delta J = & \int_V 2\mu \epsilon_{ij} \delta \epsilon_{ij} dV - \int_V \frac{\mu}{G} \sigma_{ij}^{\nabla} \delta \epsilon_{ij} dV - \int_V p \delta \epsilon_{ii} dV \\ & - \int_V \epsilon_{ii} \delta p dV - \int_S T_i \delta u_i dS - \int_V \rho X_i \delta u_i dV = 0 \end{aligned} \quad (28)$$

We now introduce the following matrix notations and finite element approximations within an element  $e$ :

$$u_i = \{u(x,y)\} = [N_u]\{U\}_e \quad (29)$$

$$p = p(x,y) = [N_p]\{p\}_e \quad (30)$$

$$e_{ij} = \{e(x,y)\} = [N_u']\{U\}_e \quad (31)$$

$$\sigma'_{ij} = \{\sigma'(x,y)\}$$

$$\sigma'^{\nabla}_{ij} = \{\sigma'^{\nabla}(x,y)\}$$

$$\{\sigma'_{ij}(x,y)\} = [D]\{e(x,y)\} - [R]\{\sigma'^{\nabla}(x,y)\} \quad (32)$$

$[D]$  = viscosity matrix

$[R]$  = relaxation matrix

$$e_{ii} = [h]\{e\} \quad (33)$$

$$T_i = [C]\{\bar{u}\}_e - [C]\{u\}_e \quad \text{on } S_u \quad (34)$$

$$T_i = \{\bar{T}\} \quad \text{on } S_\sigma \quad (35)$$

$$X_i = \{X\}$$

Using this notation, the contribution to  $\delta J$  from element  $e$  can be written as

$$\begin{aligned} \delta J_e = & \{u\}^T [K]_e \{u\}_e - \{\delta u\}_e^T [G]_e^T \{p\}_e - \{\delta p\}_e^T [G]_e \{u\}_e \\ & - \{\delta u\}_e^T \{F\}_e - \{\delta u\}_e^T \{F^H\}_e + \{\delta u\}_e^T [K_C]_e \{u\}_e \end{aligned} \quad (36)$$

where

$$[K]_e = \int_{V(e)} [N_u']^T [D] [N_u'] dV \quad (37)$$

$$[G]_e = \int_{V(e)} [N_p]^T [h] [N_u'] dV \quad (38)$$

$$\begin{aligned} \{F\}_e = & \int_{S_\sigma(e)} [N_u]^T \{\bar{T}\} dS + \int_{S_u(e)} [N_u]^T [C] \{\bar{u}\}_e dV \\ & + \int_{V(e)} \rho [N_u]^T \{X\} dV \end{aligned} \quad (39)$$

$$\{F^E\}_e = \int_{V(e)} [N_u']^T [R] \{\sigma, v\} dV \quad (40)$$

$$[K_c] = \int_{S_u(e)} [N_u]^T [C] dv \quad (41)$$

Summation of the element matrices and enforcement of  $\delta J = 0$  for all  $\{\delta u\}$  and  $\{\delta p\}$  provides the governing finite element equation

$$\begin{bmatrix} K & -G^T \\ -G & 0 \end{bmatrix} \begin{Bmatrix} u \\ p \end{Bmatrix} = \begin{Bmatrix} F \\ 0 \end{Bmatrix} + \begin{Bmatrix} F^E \\ 0 \end{Bmatrix} \quad (42)$$

where  $[K] = \sum [K]_e + \sum [K_c]_e$ .

Note that as the coefficients in  $[C]$  become large, the above equation will force  $\{u\} = \{\bar{u}\}$  on  $S_u$ . When this is the desired boundary condition on  $S_u$ , it can be specified directly rather than specifying a large  $C_{ij}$ .

The code has also been written so that the penalty function approach can be used. This is accomplished using

$$\sigma_{ij} = 2\mu \epsilon_{ij} + \lambda \epsilon_{kk} \delta_{ij}$$

where  $\lambda$  is specified as  $10,000 * \mu$ . The final governing equation under this condition has the form

$$[K]\{u\} = \{F\} + \{F^E\}. \quad (43)$$

Heat Transfer

The finite element equations for heat transfer are derived from the following variational statement found by Galerkin's principle.

$$\delta J = \int_V \left\{ k \frac{\partial \phi}{\partial x_i} \delta \frac{\partial \phi}{\partial x_i} - Q \delta \phi + \rho C_p \frac{\partial \phi}{\partial t} \delta \phi + U_i \frac{\partial \phi}{\partial x_i} \delta \phi \right\} dV - \int_{S_q} q \delta \phi dS = 0 \quad (44)$$

Substituting the finite element approximation for the temperature field and following the usual matrix summations, element by element, one obtains the governing equation

$$[H]\{\phi\} + [B]\{\dot{\phi}\} = \{Q\} \quad (45)$$

where

$$[H] = \int_V [N']^T [R] [N'] dV + \int_V [N] \rho C_p \{U\}^T [N'] dV$$

$$[R] = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix}$$

$$[B] = \int_V [N]^T \rho C_p [N] dV$$

$$\{Q\} = \int_V [N]^T Q dV - \int_{S_q} [N]^T q dS$$

$$\phi = [N]\{\phi\}$$

$$\{\nabla \phi\} = [N']\{\phi\}$$

$$q = -k \frac{\partial \phi}{\partial x_i} v_i$$

The heat flux across  $S_q$  can be specified either as the total heat flux due both to convection and conduction, or the heat flux due to

conduction only. When  $S_q$  is coincident with a streamline, there is no difference between the two specifications. However, if a control volume approach is used where mass is being transported across  $S_q$ , there is a significant difference between the two approaches. Specification of heat transfer by conduction only is tantamount to specifying the normal component of the gradient of the temperature. Because this is the more likely boundary condition to be known, MANTLE is written to accept either the temperature or heat transfer by conduction as known conditions. Note also that the coefficient of thermal conductivity can be specified for both the x-direction and the y-direction.



## PROGRAM MANTLE

Introduction

The name given to the overall computer code is MANTLE. It is composed of four major subprograms (overlays on CDC computers) which are called by an executive program, DRIVE. The four programs are, MESH1, a mesh generating program for mantle geometry; MESH2, a more general mesh generator; WAVE, a program which plans the solution procedure and auxiliary storage of the matrix equations associated with the finite element analysis; and COUPLE, which is itself an executive program which calls programs TEMP and CREEP for thermal and plastic flow analysis. COUPLE also monitors the transient and nonlinear analysis and calls all output subroutines. Flow charts are provided in the following section.

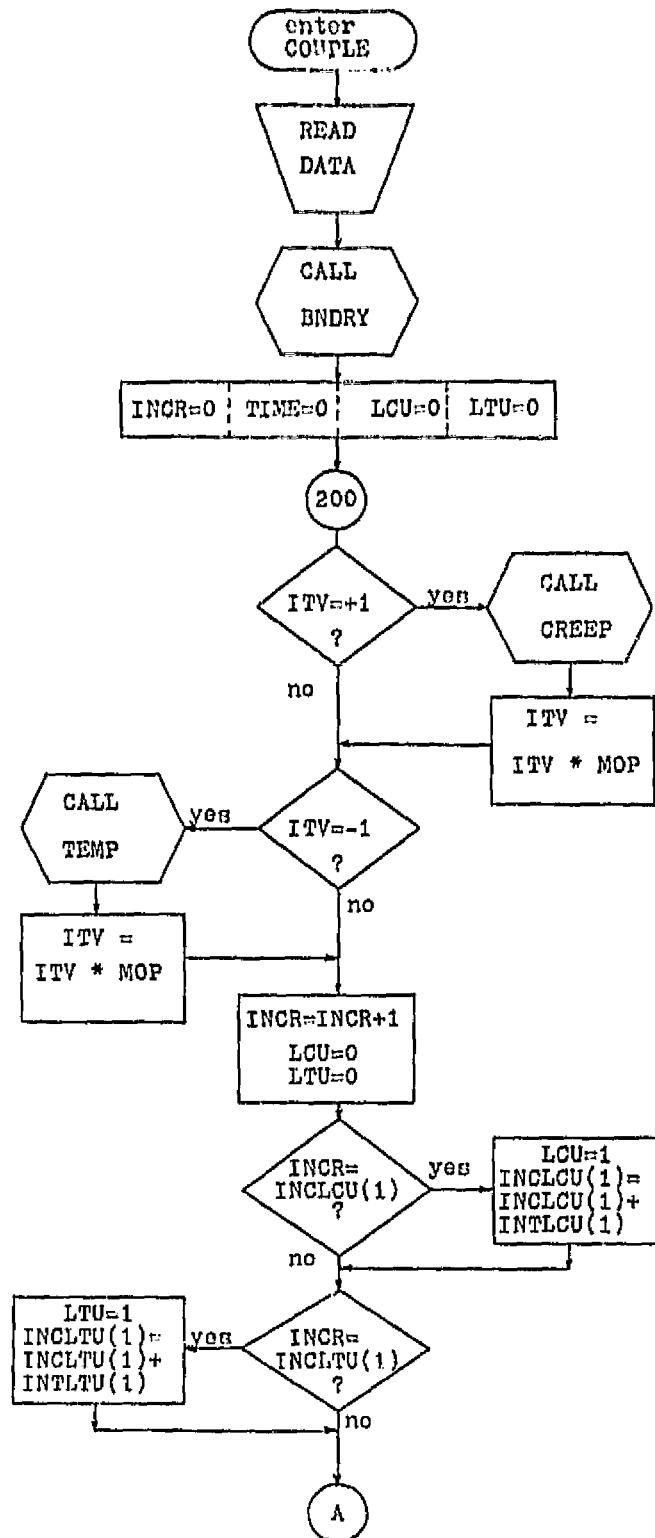
Each of the above programs will be discussed separately in terms of their input variables and whatever explanation is needed to define these variables. All input variables are echoed immediately after input with an output FORMAT identical to the input FORMAT.

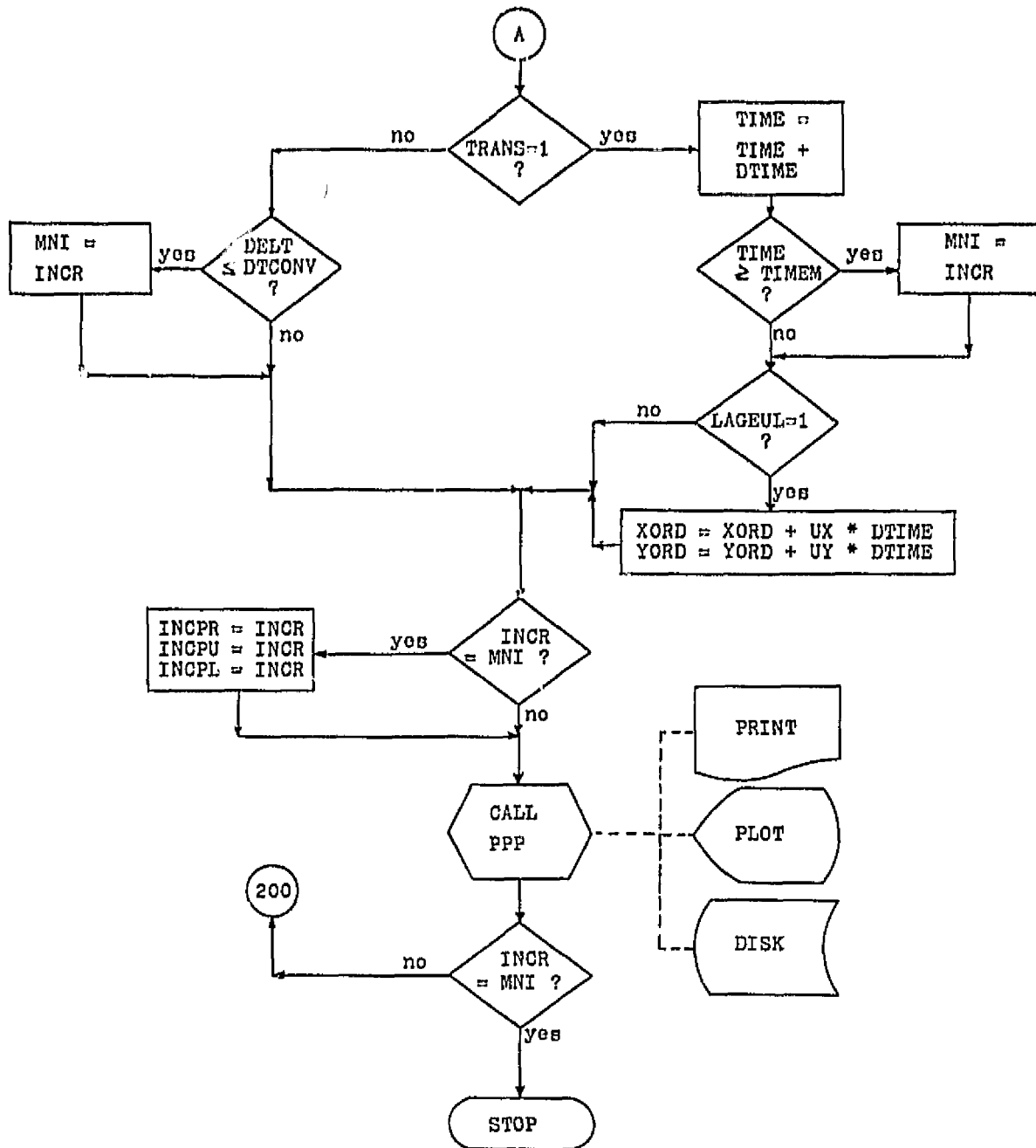
PRECEDING PAGE BLANK NOT FILMED

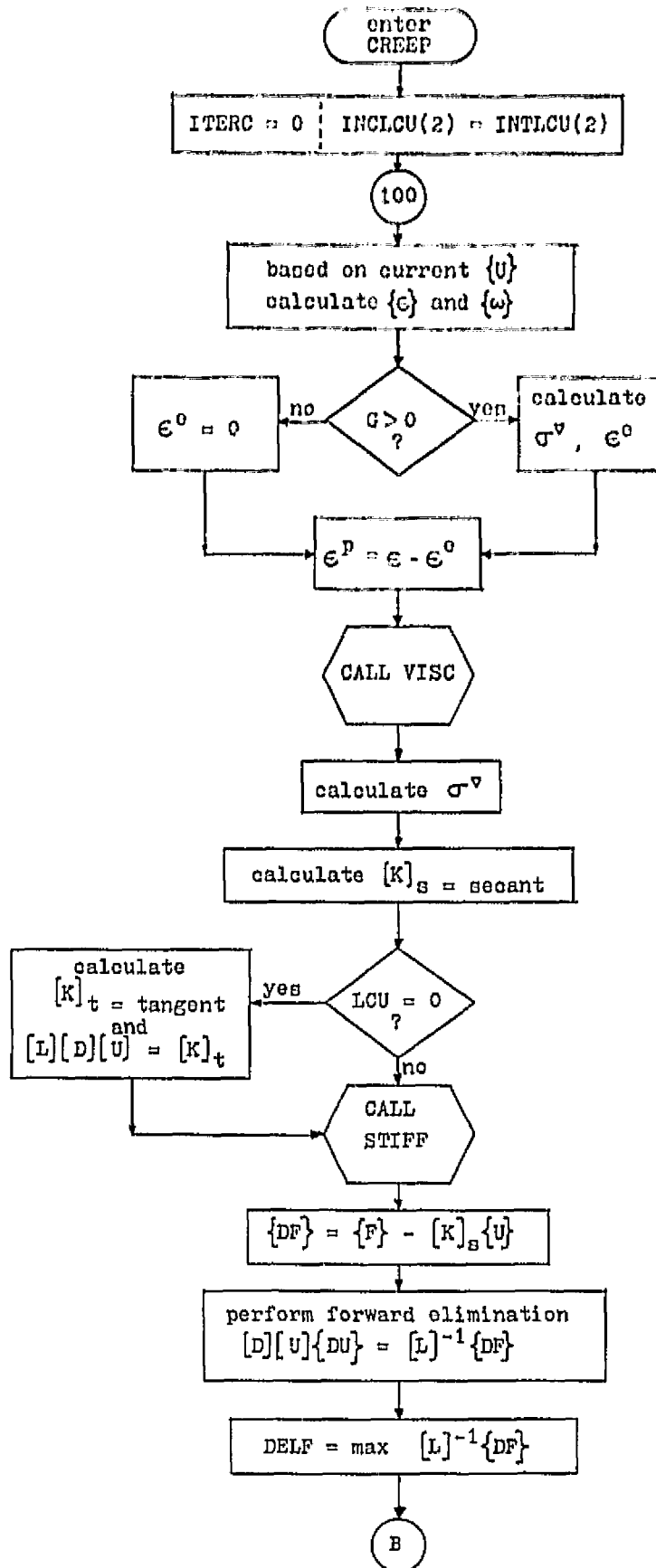
Flow Charts

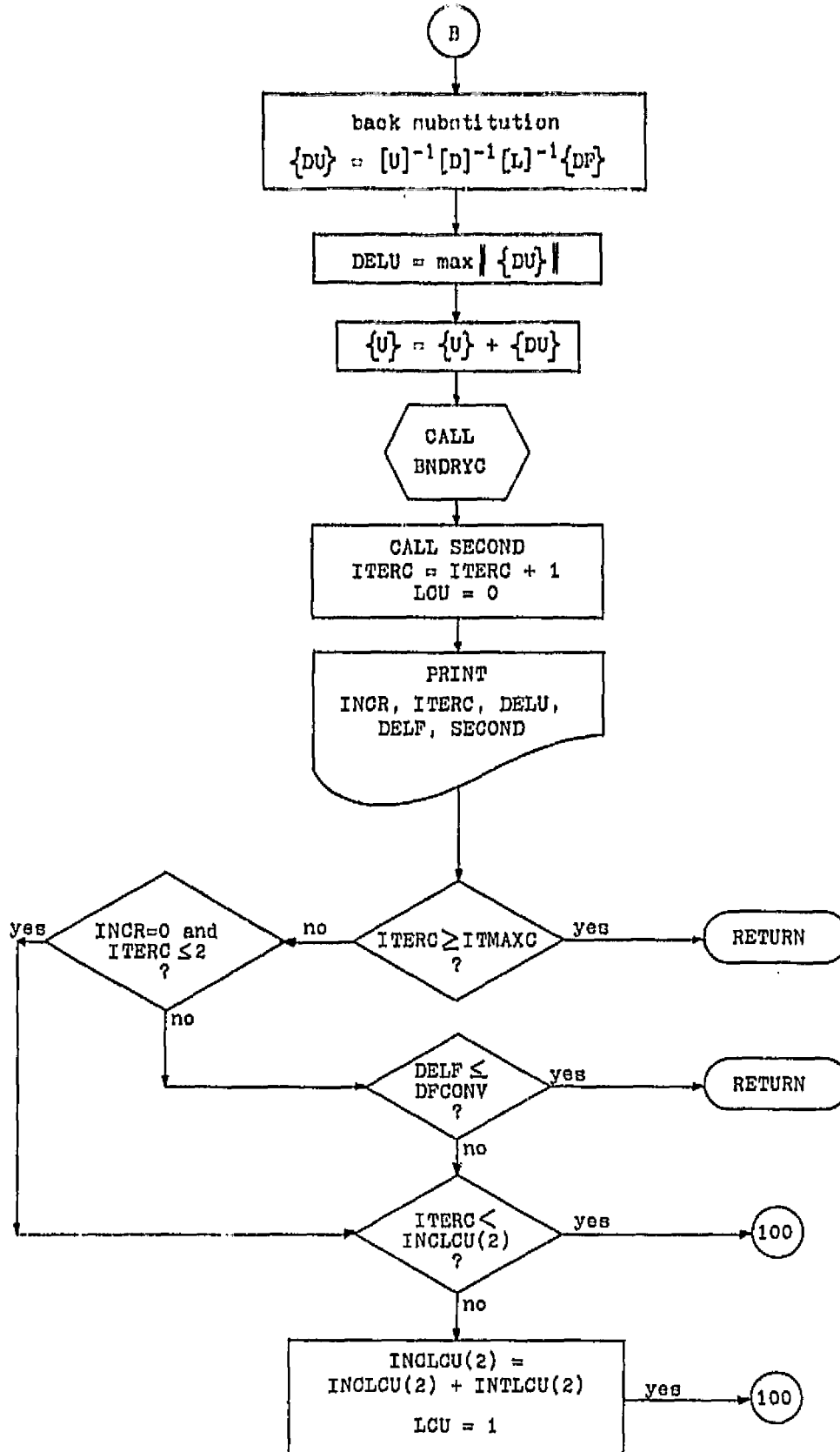
The following flow charts are presented to clarify the use of control variables and convergence limits. They also indicate where in the program certain calculations are made, e.g., CPU time, stresses, or mesh adjustment during a quasi-Lagrangian analysis. However, they are far from being a complete description of the finite element code and in some instances imply a program flow different from that used in the actual code. For instance, the flow charts indicate the stiffness matrices are completely formed and then solved, rather than the piece-wise procedure actually used with the frontal method. These discrepancies do not distract from the usefulness of the flow charts, and in fact were allowed so as to emphasize the basic logic of the analysis. However, the reader should proceed with reasonable caution if he uses them to help him decipher the FORTRAN Code.

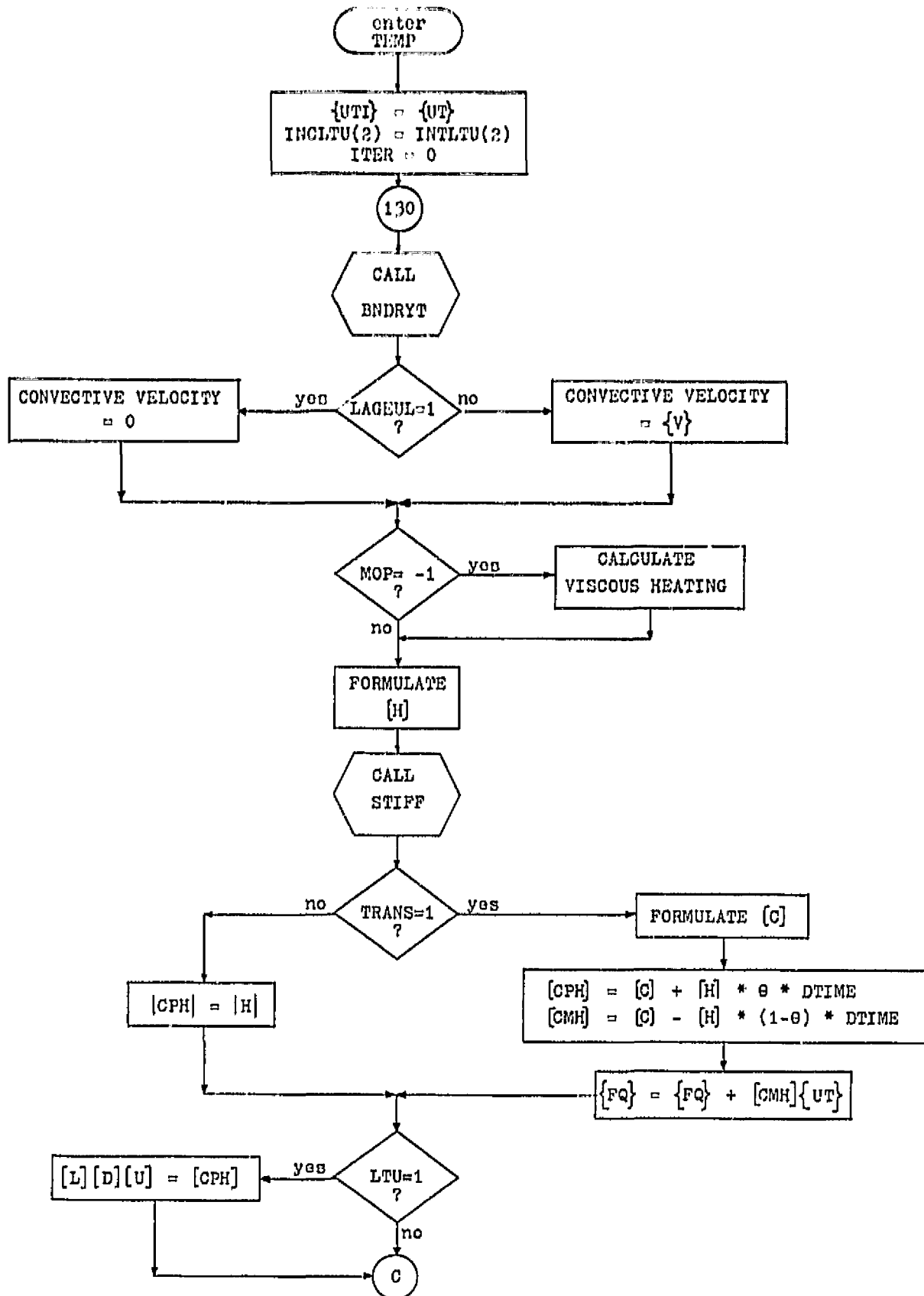
**PRECEDING PAGE BLANK NOT FILMED.**

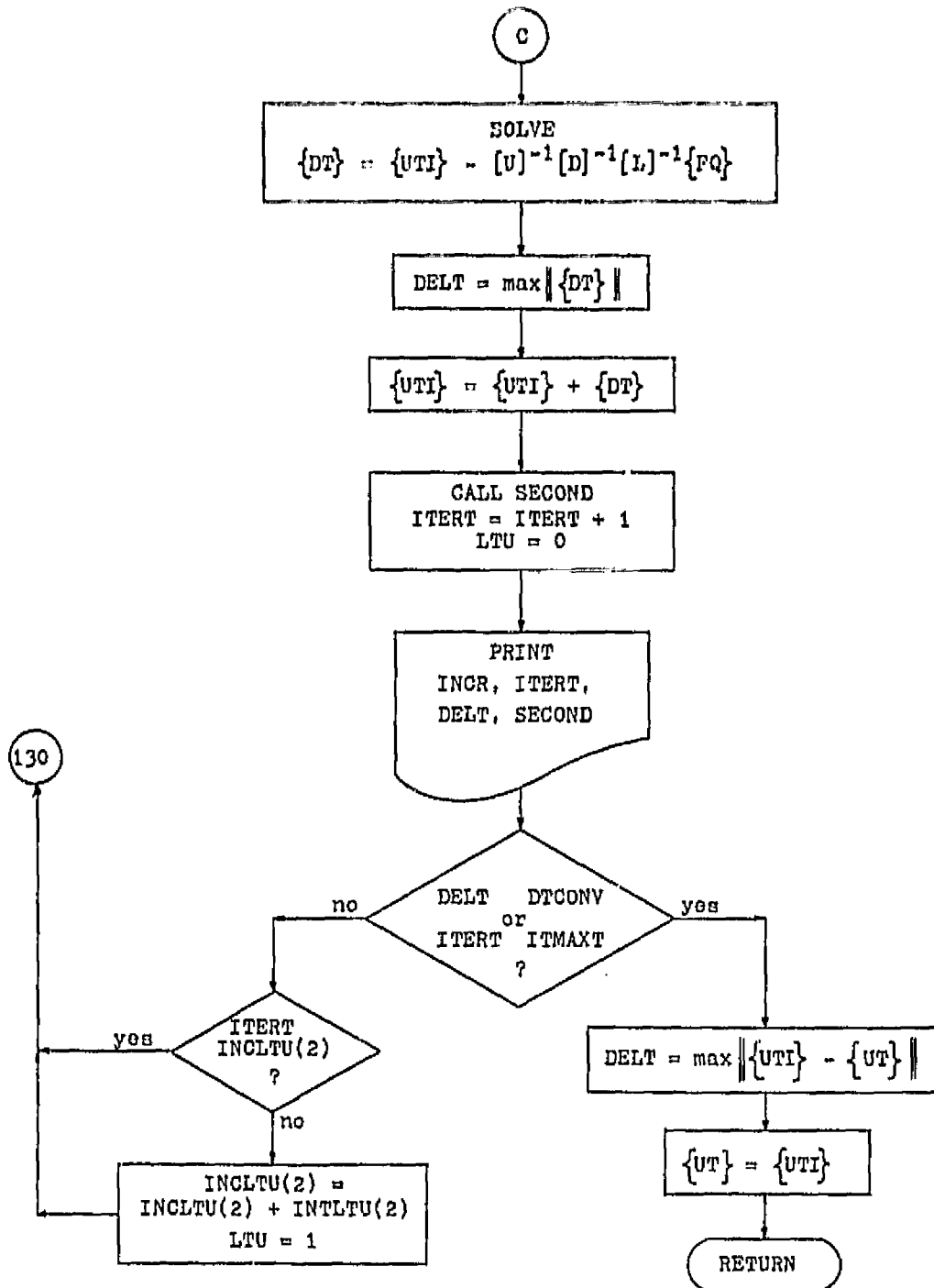














Tape Usage

A total of eight tapes are designated in MANTLE. If all three major subprograms (MESH1 or MESH2, WAVE, and COUPLE) are run as a unit, all eight tapes can be scratch files. However, if several different runs are to be made with the same mesh, a significant savings in computer time will result if only COUPLE is called. This requires equating the tapes to permanent files. The following is an explanation of the usage of each tape.

- TAPE1: This tape has the output from WAVE related to the tape segments for forming the stiffness matrix in CREEP. It must be a permanent file if this information is to be used for later runs without calling WAVE.
- TAPE2: This tape contains the segmented stiffness matrix for CREEP analysis. Because the matrix will be in its LDU form, it can be saved for later analysis to avoid a new decomposition of the stiffness matrix whenever it remains the same between runs.
- TAPE3: This tape is the counterpart of TAPE1 for TEMP.
- TAPE4: This tape is the counterpart of TAPE2 for TEMP.
- TAPE5: All READ statements in MANTLE are from this tape. It should be equated to the INPUT file when data is read from punched cards.
- TAPE6: Normal output from MANTLE is placed on this tape, that is, most WRITE statements are to this tape. It is normally equated to the OUTPUT file.
- TAPE7: Output which is useful for additional plotting and/or initialization for later runs in either time dependent problems or nonlinear analysis is read to this tape. If punch cards are used, it should be equated to the PUNCH file. Output from either MESH1 or MESH2 is read onto this file by specifying IPUNCH = 7.
- TAPE9: This tape contains all output from the MESH programs and WAVE not already on TAPE1 and TAPE3. It should be a permanent file if MESH and WAVE are not to be called for later runs.

Easy Input Reference

The following listing of the Input data is given for the user's quick reference. This listing is very abbreviated and primarily serves the purpose of reminding an already familiar user what variables are called for and in what order.

## PROGRAM DRIVE

IFLOW(I) 3I10

## PROGRAM MESH1

NPPE I10  
 RI,RO,RH,RPI 4E10,3  
 IPUNCH 4E10,3  
 NDIUTH,NDIVR 2I10  
 NPBC,XBC,YBC,TBC I10,3E10,3  
 CH,CX,CY,TX,TY,TQ 6E10,3  
 NPBC,XBC,YBC,TBC I10,3E10,3  
 CH,CX,CY,TX,TY,TQ I10,3E10,3  
 NUMBC I10  
 NP,NPBC,COSXXP,XBC,YBC,TBC 2I10,4E10,3  
 CH,CX,CY,TX,TY,TQ 6E10,3

## PROGRAM MESH2

NPPE I10  
 NUMLPS(1),NUMLPS(2) 2I10  
 XMIN,XMAX,YMIN,YMAX 4E10,3  
 IPUNCH I10  
 NDIV(I,1),NDIV(I,2) 2I10  
 JOIN(I,J,K) 4(I7,I3)  
 XCOR(I,J),YCOR(I,J) 2E10,3  
 NUMBC I10  
 J1,LPBC,COSLBC,XLBC,YLBC,TQLBC 2I10,4E10,3  
 CHBC,CXLBC,CYLBC,TXLBC,TYLBC,TQLBC 6E10,3  
 NUMBC I10  
 I1,NPBC,COSXXP,XBC,YBC,TBC 2I10,3E10,3  
 CH,CX,CY,TX,TY,TQ 6E10,3

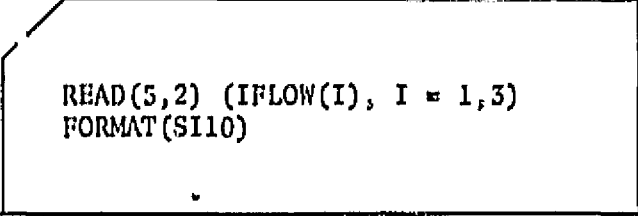
## PROGRAM COUPLE

NUMAT I10  
 MAT(I) 10I5  
 INCFR,INCPU,INCPL 3I10  
 INTPR,INTPU,INTPL 3I10  
 ITV,MOP,TRANS,THETA,INTMP,LAGEUL,IRZ 2I10,2E10,3,3I10  
 TIMEH,MNI,DMAX,ETHAX E10,3,I10,2E10,3  
 ITMAXC,ITMAXT 2I10  
 VECTL,CTEMP 2E10,3  
 INTLCU(1),INTLCU(2),INTLTU(1),INTLTU(2),LCU,LTU 4I10  
 DFCONV,DQCONV,DUCCONV,DTCONV 4E10,3  
 NPTS,NSEC 2I10  
 I1,XORD(I),YORD(I),UX(I),UY(I),UT(I) 15,2E10,3,3E18,10  
 JBGN,JEND,INCR,TEMPO,UXO,UYO 3I10,3E10,3

Input DescriptionPROGRAM DRIVE--Input

This is the main executive program whose only purpose is to call the other components of MANTLE and define the COMMON blocks which transfer data between these components.

INPUT:



```
READ(5,2) (IFLOW(I), I = 1,3)
FORMAT(SI10)
```

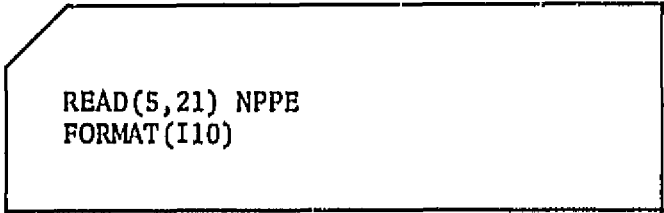
These three variables determine which of the four programs will be executed. The following code is employed:

IFLOW(1)	= 1	calls	MESH1
	= 2	calls	MESH2
IFLOW(2)	= 1	calls	WAVE
IFLOW(3)	= 1	calls	COUPLE

PROGRAM MESH1--Input

Program MESH1 is a specialized mesh generator for mantle geometry. It is also through this program that the boundary conditions are specified, although they can be altered later in the analysis. Figure 1 illustrates some typical meshes. The coordinate system used is rectangular rather than polar to accommodate the finite element approximation for pressure and its relationship to the derivatives of the velocity components. However, MESH1 accepts boundary conditions in terms of radial and tangential components. The mesh generated applies to both the thermal analysis and the creep analysis.

INPUT:



```

READ(5,21) NPPE
FORMAT(I10)
  
```

NPPE = number of pressure points per element

This variable can be either 0, 1 or 3. If zero, the penalty function approach will automatically be used during the analysis. When 1, the pressure will be constant throughout the element and compressibility will be satisfied only in an average sense in each element. When 3, the pressure is approximated by a linear function of  $x$  or  $y$  within each element. This last approach will insure satisfaction of the constraint of incompressibility everywhere within an element provided the mid-side nodes are truly mid-side nodes of a triangular element (i.e., curved sides will prevent the constraint from being satisfied everywhere within the element--although NPPE = 3 can still be used for such elements and will more nearly satisfy this constraint than NPPE = 1).

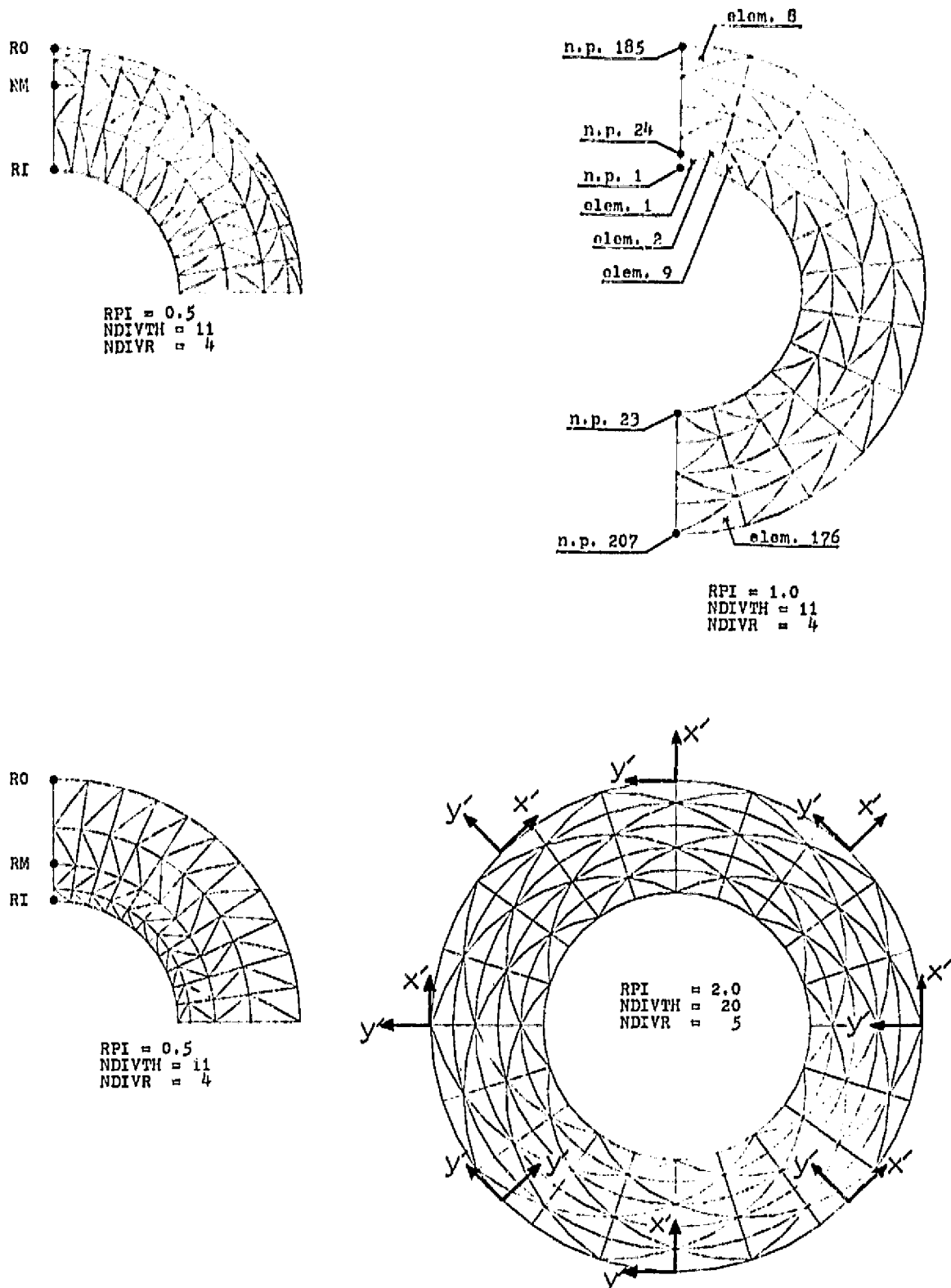


Figure 1. Meshes generated by MESH1.

INPUT:

```

READ(5,3) RI, RO, RM, RPI
FORMAT(4E10.3)

```

RI = inside radius

RO = outside radius

RM = mid-radius

RPI = multiple of  $\pi$  radians to be covered by the mesh.

The use of these variable is illustrated in Figure 1. Note that RM is used to grade the element size in the radial direction. RPI = 2.0 will create a completely circular mesh.

INPUT:

```

READ(5,19) IPUNCH
FORMAT (I10)

```

IPUNCH = 7 results in all mesh output data being read to TAPE7 for permanent storage

INPUT:

```

READ(5,1) NDIVTH, NDIVR
FORMAT(2I10)

```

NDIVTH = number of mesh divisions in the theta direction

NDIVR = number of mesh divisions in the radial direction

INPUT:

```

READ(5,10) NPBCI, XBCI, YBCI, TBCI
READ(5,11) CHI, CXI, CYI, TXI, TYI, TQI
FORMAT(I10, 3E10.3)
FORMAT(6E10.3)

```

INPUT:

```

READ(5,10) NPBCO, XBCO, YBCO, TBCO
READ(5,11) CHO, CXO, CYO, TXO, TYO, TQO
FORMAT(I10, 3E10.3)
FORMAT(6E10.3)

```

These next four cards define boundary conditions along the inside radius and the outside radius. The vector components are referred to the nodal point coordinates defined in Fig. 1.

NPBC defines the type of boundary condition and the meaning of XBC, YBC, and TBC

NPBC =  $\pm 1$  both  $x'$  and  $y'$  components of velocity are unknown

NPBC =  $\pm 2$   $x'$  component of velocity known to be XBC

NPBC =  $\pm 3$   $y'$  component of velocity known to be YBC

NPBC =  $\pm 4$   $x'$  and  $y'$  components of velocity known to be XBC and YBC respectively

NPBC < 0 temperature known to be TBC

If XBC and YBC are not designated by NPBC to be known components of velocity, they can have one of two different meanings depending on the value of CX and CY.

If CX = 0.0, then XBC equals known component of force in the  $x'$  direction

If CX  $\neq$  0.0, then XBC equals  $x'$  component of velocity external to boundary, to be used in viscous force boundary condition (see Eq. 10)

If  $CY = 0.0$ , then YBC equals known component of force in the  $y'$  direction

If  $CY \neq 0.0$ , then YBC equals  $y'$  component of velocity external to boundary, to be used in viscous force boundary condition (see Eq. 10)

CX = coefficient of boundary viscosity in  $x'$  direction. It is the ratio of the boundary surface traction in the  $x'$  direction to the difference between XBC and the actual boundary velocity in the  $x'$  direction

CY = coefficient of boundary viscosity in  $y'$  direction. It is the ratio of the boundary surface traction in the  $y'$  direction to the difference between YBC and the actual boundary velocity in the  $y'$  direction

TX = surface traction (force/unit area) in the  $x'$  direction

TY = surface traction in the  $y'$  direction

When NPBC > 0, then TBC can have one of two different meanings depending on the value of CH.

If  $CH = 0.0$ , then TBC is a point heat source

If  $CH \neq 0.0$ , then TBC equals the ambient temperature used for a convective heat transfer boundary condition (see Eq. 24)

CH = coefficient of convective heat transfer

TQ = heat source per unit area

COSXXP = cosine between the local  $x'$  axis and the global  $x$  axis. The  $x'$  axis should always be defined so that this describes an angle in either the first or second quadrants, i.e., the sine will always be assumed positive

INPUT:

```

READ(5,1) NUMBC
FORMAT(I10)

```

NUMBC = number of nodal points which are to have individual boundary conditions specified.



INPUT:

```

READ(5,24) I1, NPBC(I1), COSXXP(I1), XBC(I,1),
           YBC(I1), TBC(I1)
READ(5,27) CH(I1), CX(I1), CY(I1), TX(I1),
           TY(I1), TQ(I1)
FORMAT(2I10, F10.4, 3E10.3)
FORMAT(6E10.3)

```

As many pairs of the above cards will be read as specified by NUMBC.

I1            = nodal point number

NPBC(I1)     = type of boundary conditions (see page 30)

COSXXP(I1)   = cosine between  $x$  and  $x'$  axes. See Fig. 1  
for example. The  $x'$  axis must always be  
defined so that the  $\sin(x, x')$  will be  
positive

The remaining terms are defined on pages 31 and 31.

PROGRAM MESH2--Input

Program MESH2 is an automatic mesh generator. It is also through this program that the boundary conditions are specified. However, there are other opportunities to alter both the mesh and the boundary conditions. To understand the following input variables, one should be aware that two separate meshes are generated. The first mesh will always be contained within the second mesh and will be used in CREEP. The second mesh will at least coincide with the first but can be extended beyond the first. It will be used in TEMP.

INPUT:

```
READ(5,21) NPPE  
FORMAT(I10)
```

NPPE = number of pressure points per element

(See discussion under MESH1.)

INPUT:

```
READ(5,1) NUMLPS(1), NUMLPS (2)  
FORMAT(2I10)
```

NUMLPS(1) = number of loops in the finite element mesh for  
the plastic flow analysis

NUMLPS(2) = number of loops in the finite element mesh for  
the thermal analysis.

In all cases,  $\text{NUMLPS}(2) \geq \text{NUMLPS}(1)$ , whether a thermal analysis is to be conducted or not.  $\text{NUMLPS}(1)$  should be specified as zero if only a thermal analysis is to be conducted.

INPUT:

```
READ(5,3) XMIN, XMAX, YMIN, YMAX  
FORMAT(4E10.3)
```

These variables specify the minimum and maximum coordinates for plotting the mesh layout in an x-y Cartesian reference frame. These same variables are used for specifying coordinates for plotting results from program COUPLE. The exact use for these variables depends on the computer graphics available.

INPUT:

```
READ(5,19) IPUNCH  
FORMAT(I10)
```

IPUNCH = 7 results in the nodal point coordinates and the nodal points associated with each element being read onto TAPE7. This output is not necessary for an analysis but is included in case it is desirable to have this data on punched cards or permanent file

THE FOLLOWING DATA WILL BE NEEDED FOR EACH LOOP IN THE ORDER PRESENTED\*\*\*

INPUT:

```
READ(5,6) NDIV(I,1), NDIV(I,2)
FORMAT(2I10)
```

NDIV(I,1) = number of divisions on sides 1 and 3 of loop I

NDIV(I,2) = number of divisions on sides 2 and 4 of loop I

(See Appendix A for description of loops.)

INPUT:

```
READ(5,9) ((JOIN(I,J,K), K = 1,2), J = 1,4)
FORMAT(4(17,13))
```

This card indicates which sides of loop I are "joined" to loops already formed. It is best explained by example. In the following figure, there are four groupings of numbers, each group representing

2	3	3	1	0	0	4	2
Side 1	Side 2	Side 3	Side 4				

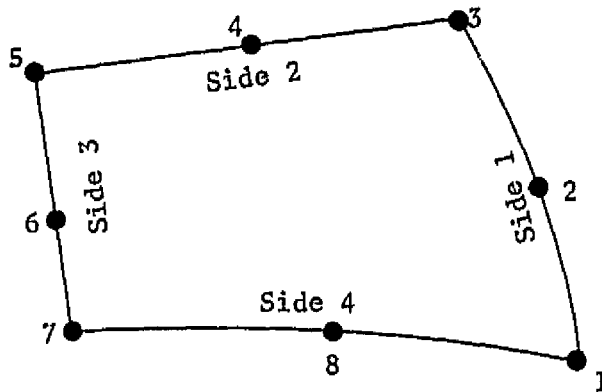
the 4 sides of the current loop I. The card indicates that side 1 of loop I joins loop 2, side 3 (3rd side of loop 2); side 2 of loop I joins loop 3, side 1; side 3 of loop I does not join any loop already formed; and side 4 of loop I joins loop 4, side 2. The card only "looks back" at loops already formed. It should not specify what loops not yet formed, i.e., loops > I, will connect to loop I. The

array has the following meaning: Loop I, side K, is joined to side JOIN(I,J,2) of loop JOIN(I,J,1)

INPUT:

```
READ(5,11) (XCOR(I,J), YCOR(I,J), J = 1,8)
FORMAT(2E10.3)
```

Each loop is a quadrilateral with parabolic sides. The geometry of the loop is defined by the eight points shown below. The eight data cards read in above define the x and y coordinates of these eight points. The first point read must always be a corner point. It, and the next two points, will define side 1 of the loop. The coordinates must be read counterclockwise, as shown:



INPUT:

```
READ(5,1) NUMBC
FORMAT(I10)
```

NUMBC = number of sides of loop I which have boundary conditions to be read. If a loop has zero sides which need boundary condition specified, NUMBC = 0 will cause the next input to be skipped.

INPUT:

```

READ(5,24) JI, LPBC(I,J), COSLBC(I,J)
      XLBC(I,J), YLBC(I,J1), TLBC(I,J1)
READ(5,27), CHBC(I,J), CXLBC(I,J1), CYLBC(I,J1),
      TXLBC(I,J1), TYLBC(I,J1), TQLBC(I,J1)
FORMAT(I10, 5E10.3)
FORMAT(6E10.3)

```

These two input cards are listed together since they are read in pairs for as many sides of loop I as specified by NUMBC.

JI = side of loop for which the boundary conditions apply

The remaining variables will be used to specify their corresponding boundary condition on all nodal points on side J1. They correspond to NPBC, COSXXP, XBC, YBC, TBC, CH, CX, CY, TX, TY, and TQ which are defined on page 30. It should be noted that the corner nodal points will have the boundary conditions corresponding to the highest numbered side to which they are contiguous. These boundary points as well as other special points may be specified individually with the set of data cards which follow.

*THIS CONCLUDES THE DATA CARDS READ FOR EACH LOOP SPECIFIED BY NUMLPS(2).\*\*\*\**

\*\*\*

INPUT:

```

READ(5,1) NUMBC
FORMAT(I10)

```

NUMBC = number of singular nodal points which have boundary conditions to be read. NUMBC = 0 will cause the next input to be skipped.

INPUT:

```
READ(5,24), I1, NPBC(I1), COSXXP(I1),  
      XBC(I1), YBC(I1), TBC(I1)  
READ(5,27), CH(I1), CX(I1), CY(I1),  
      TX(I1), TY(I1), TQ(I1)  
FORMAT(2I10, 4E10.3)  
FORMAT(6E10.3)
```

As many of these pairs of cards will be read as specified by the previous value of NUMBC. I1 is the nodal point number which the variables, NPBC, COSXXP, XBC, YBC, TBC, CH, CX, CY, TX, TY, and TQ will be assigned. Explanation of these variables is given on page 30.

PROGRAM WAVE--Input

Program WAVE takes the data from MESH and plans the tape sections which will be necessary to store the large finite element matrices in programs CREEP and TEMP.

INPUT:

```

READ(5,4) MSHCD(1), MSHCD(2)
FORMAT(2I10)

```

MSHCD(1) = 1 identifies that program CREEP will be called and hence tape segments should be planned

MSHCD(2) = 1 identifies that program TEMP will be called and tape segments should be planned for it

Zero should be specified when these programs are not to be called.

The remaining READ statements in WAVE will be read once for CREEP if MSHCD(1) = 1, and once for TEMP if MSHCD(2) = 1.

INPUT:

```

READ(5,4) MAXVOL, IBMIN
FORMAT(2I10)

```

MAXVOL = the storage specified for the finite element stiffness matrices. For program CREEP it is the dimension of SKXX, SKXY, SKYX and SKYY which must all be equal. For TEMP, it should be one-half the sum of these dimensions. Actually, TEMP will use the storage of all four arrays for its stiffness matrix, however WAVE calculates the size of each tape segment based on half the storage available for this nonsymmetrical matrix



IBMIN = the minimum bandwidth to be reserved in forming the K matrix. This normally should be zero but if it happens that a mesh is so constructed that it will allow a very small bandwidth to begin with and then suddenly require a large bandwidth, this variable should be set equal to the larger number.

INPUT:

```

READ(5,4) IREAD
FORMAT(110)

```

The finite element matrices are formed element by element with each row, or equation, corresponding to a particular nodal point. In order to keep the bandwidth for these matrices as small as possible, it is necessary to choose the order in which the elements are taken so that a minimum of rows, or equations, separate any two rows representing nodal points from the same element. The element numbering specified by MESH1 provides the best order for this to occur. Therefore IREAD should be read as zero which indicates the elements should be assembled in numerical order, and there is no need to read in a different order. However, when MESH2 is used, the element numbers may not indicate the best order to assemble the elements. In these cases, IREAD should be specified as 5. This designation will activate the next READ statement, otherwise it is skipped.

INPUT:

```

READ(5,11) (IORDER(IO,I), I = 1, NUMEL)
FORMAT(1015)

```

This is the order which the elements will be taken if other than their numerical order. See previous discussion on IREAD. Because two different meshes for CREEP and TEMP are used, IORDER must be specified for each. A discussion on how to select a suitable order is treated in Appendix A.

PROGRAM COUPLE--Input

Program COUPLE is the executive program which "couples" programs CREEP and TEMP. It also monitors the iterations between these programs to determine when steady-state conditions have been obtained as well as controlling the increments in a time dependent problem. All output is controlled by this same program.

INPUT:

```
READ(5,12) NUMAT
FORMAT(I10)
```

NUMAT = number of different materials to be identified in the analysis. Each element has a "material number."

INPUT:

```
READ(5,22) (MAT(I), I = 1, NUMEL)
FORMAT(20I4)
```

MAT(I) = material number for element I. If NUMAT = 1, this READ statement is by-passed and the array is initialized to 1.

INPUT:

```
READ(5,16) INCPR, INCPU, INCPL
FORMAT(3I10)
```

These variables stand for INCrement to PRint, Punch, and PLot respectively. They represent the increments in the analysis that Subroutine PPP will print, punch and/or plot the output. The values read in at this time will be the first increment that these outputs will be executed. Thereafter, they will be executed at regular intervals (see next input). For a steady-state analysis, increments represent iterations between CREEP and TEMP. In a transient problem, increments represent steps in time. If it is not desirable to execute one of these routines before completion of the analysis, the corresponding INC-value can be read in as zero (or less), or may be specified extremely large.

INPUT:

```

READ(5,18) INTPR, INTPU, INTPL
FORMAT(3E10)

```

These variables represent the interval, or number of increments, between each printing, punching, and plotting of the output. For example, if printing is called for at increment equal to INCPR, it will not be called for again until  $INCPR = INCPR + INTPR$ . This output is always called after the last increment of the analysis regardless of the values of INCPR, INCPU, and INCPL. If it is desired that one of these outputs not be used, even at that time, the corresponding Interval number should be read in as a negative number. For instance, if it is not desired to have any punched card (or TAPE7) output, INTPU should be specified as -1.

## INPUT:

```

READ(5,24) ITV, MOP TRANS, THETA, INTEMP, LAGEUL, IRZ
FORMAT(2I10, 2E10.3, 3I10)

```

These five variables control the type of solution desired, be it a pure creep solution, pure thermal analysis, coupled steady-state, transient, etc.

- ITV = +1 will cause program CREEP to be called from COUPLE
- = -1 will cause program TEMP to be called. During a coupled problem, this variable oscillates back and forth between plus and minus 1. The value read in here will determine which of these two programs will be called first. In the case of a pure thermal analysis the value must be -1 whereas for a pure creep analysis the value must be +1.
- MOP stands for minus or plus. After either CREEP or TEMP is called by COUPLE, ITV is multiplied by MOP. Therefore, MOP = -1 will cause ITV to oscillate and produce a coupled analysis, whereas MOP = +1 will let ITV remain the value initially read in and hence will dictate either a pure thermal analysis or a pure creep analysis.
- TRANS is a variable specifying if a transient analysis is desired. TRANS = 0.0 produces a steady-state analysis whereas TRANS = 1.0 produces a transient analysis.
- THETA is a variable that determines the type of numerical integration with respect to time that is used for the heat equation.  
 THETA = 0.0 specifies a pure explicit integration  
 THETA = 1.0 specifies a pure implicit integration  
 THETA = 0.5 specifies a Crank-Nicolson integration  
 It is recommended that THETA be taken between 0.5 and 1.0. For a complete discussion of this variable see Crandall, 1956.
- INTEMP is a flag to indicate whether the temperatures in a coupled problem should be initialized at their steady-state pure-conduction values. If INTEMP = 1 such an initialization will take place (it will involve a finite element solution in itself). If INTEMP = 0 no such initialization will take place.

LAGUEL stands for Lagrangian-Eulerian analysis. This type of analysis allows the mesh to follow the plastic flow in a transient analysis. It therefore corresponds to a control mass approach and is particularly convenient when the geometry of the problem is changing with the flow. LAGEUL = 1 implements such a solution procedure. LAGEUL = 0 will cause a pure Eulerian analysis to be made in which the elements are taken as divisions of space rather than divisions of material. Control volume analyses are made using this latter designations.

IRZ is a variable which specifies whether the coordinate system is axisymmetric or not. IRZ = 1 indicates it is. IRZ = 0 will result in a plane flow analysis being made. In the case of an axisymmetric analysis, x corresponds to the r direction. The axis of symmetry is taken as the y(=z) axis.

#### INPUT:

```

READ(5,26) TIMEM, MNI, DUMAX, DTMAX
FORMAT(E10.3, I10, 2E10.3)

```

TIMEM = maximum time to be simulated during a transient analysis. For a steady-state analysis this variable can be set equal to zero.

MNI = maximum number of increments. This variable limits the number of iterations during a coupled steady-state solution or the number of time steps in a transient analysis.

DUMAX = the maximum displacement any nodal point is to be given during an increment of a LAGEUL analysis.

DTMAX = the maximum value that is to be used for an increment of time during a transient analysis. The current version of the code allows only a constant value to be used for the increment of time.

INPUT:

```
READ(5,1) VECTL, CTEMP  
FORMAT(2E10.3)
```

These variables are used for plotting the output, and may be changed to fit the specific computer graphic equipment of the user.

VECTL = the length of the arrow representing the maximum velocity of any nodal point. It has the same units as are used for the nodal point coordinates. A good value to use is the shortest distance between any two nodal points.

CTEMP = the contour interval to be used for isotherm plots of the temperature field. All isotherms will be multiples of this variable.

INPUT:

```
READ(5,4) ITMAXC, ITMAXT  
FORMAT(2I10)
```

The code allows for the analysis of nonlinear creep behavior and nonlinear thermal behavior in addition to the nonlinearity resulting from the coupled problem. In these cases it is often desirable to iterate several times on either the creep equations or the thermal equations before returning to the other. ITMAXC specifies the maximum number of iterations to be made on the creep equations in such cases and ITMAXT represents the maximum number of iterations to be made on the temperature equations.

INPUT:

```

READ(5,29) INTLCU(1), INTLCU(2), INTLTU(1), INTLTU(2),
          LCU  LTU
FORMAT(6I10)

```

These six variables control how often an LDU decomposition of the finite element matrices should be made.

- INTLCU(1) = interval for LDU decompositions of creep matrix. This interval specifies the number of increments between CREEP and TEMP before a new decomposition is performed.
- INTLCU(2) = interval for LDU decompositions of creep matrix. This interval specifies the number of iterations within CREEP before a new decomposition is performed.
- INTLTU(1) = interval for LDU decompositions of temperature matrix. This interval specifies the number of increments between CREEP and TEMP before a new decomposition is performed.
- INTLTU(2) = interval of LDU decompositions of temperature matrix. This interval specifies the number of iterations within TEMP before a new decomposition is performed.
- LCU       = 1 should be specified when the stiffness matrix for CREEP has not been saved from a previous run.  
          = 0 should be specified when the LDU decomposition of the stiffness matrix in CREEP has been saved on a permanent file equated to TAPE2.
- LTU       is the same as LCU but applies to the stiffness matrix in TEMP. If it has been saved on permanent file, it should be equated to TAPE4.



INPUT:

```
READ(5,31) DFCONV, DQCONV, DUCONV, DTCONV  
FORMAT(4E10.3)
```

These four numbers are convergence limits. They represent the minimum values for changes in 1) the right-hand side of the creep equation (DF = change in force), 2) the right-hand side of the temperature equation (DQ = change in heat source), 3) the left-hand side of the creep equation (DU = change in velocity), and 4) the left-hand side of the temperature equation (DT = change in temperature). In the current version of the program not all four limits are used. DFCONV specifies convergence for iterations within program CREEP such as needed for nonlinear flow laws. It is used in conjunction with ITMAXC to control the iterations within CREEP. DQCONV is currently not used. It would normally be used to determine convergence within program TEMP by comparing it to the total change in the calculated heat sources between each iteration. It is provided for future use whenever the need might arise; for the present, the variable may be read in as zero. DUCONV is also inoperative but allows the possibility of adding a convergence check by comparing the change in velocity between iterations of CREEP or between iterations of CREEP-TEMP when a steady-state analysis is being conducted. DTCONV is currently used to test convergence within TEMP and also convergence for steady-state analyses involving coupled behavior between CREEP and TEMP. This latter use is the most common use for this variable. It might be noted here that there are few needs for a nonlinear temperature analysis, therefore ITMAXT most

likely will be specified as zero which will override any control DTCONV has within program TEMP. Therefore, DTCONV can be used exclusively for convergence tests on coupled thermal-mechanical analyses.

INPUT:

```

READ(5,4) NPTS, NSEC
FORMAT(2I10)

```

The remaining INPUT cards allow final adjustment of either the mesh or values of the independent variables before execution of an analysis.

NPTS = number of nodal point adjustments to be made

NSEC = number of sections or groups of nodal points which are to be initialized to the same values

INPUT:

```

READ(5,35) I1, XORD(I1), YORD(I1), UX(I1), UY(I1), UT(I1)
FORMAT(I5, 2E10.3, 3E18.10)

```

As many of these cards will be read as specified by NPTS. The FORMAT is the same as that used for TAPE7 output, hence it is at this point that output from a previous run may be read in as initial conditions for continued analysis, either transient or nonlinear. If punch cards are not to be used, it is best to write and read onto and off of TAPE7 directly without format. These cards and/or the next input can also be used to make changes in only a select number of nodal

point values, for instance to insure convection will begin in a particular pattern. I1 equals the nodal point to which the values will be given.

INPUT:

```
READ(5,9) JBGN, JEND, INCR, TEMPO, UXO, UYO  
FORMAT(3I10, 3E10.3)
```

As many of these cards will be read as specified by NSEC. Each card will activate the following DO-loop.

```
DO 185 J = JBGN, JEND, INCR  
  UT(J) = TEMPO  
  UX(J) = UXO  
  UY(J) = UYO  
185 CONTINUE
```

Output Description

All variables read into MANTLE are immediately "echoed" as output. Slash marks (/) appearing within this echoed output indicate separation of input data cards. Each new line of echoed data always represents a new input data card. Because these variables have already been defined in the previous section they will not be defined here.

PROGRAM MESH1 or MESH2--Output

OUTPUT:

NUMNP	NUMTP	NUMVP	NUMPP	NELMC	NELMT
520	520	520	0	240	240

NUMNP = number of nodal points. This number is equal to NUMVP + NUMPP or NUMTP, whichever is greater

NUMTP = number of temperature points, i.e., nodal points which specify temperatures in a thermal analysis

NUMVP = number of velocity points, i.e., nodal points which specify velocity in an analysis of creeping flow

NUMPP = number of pressure points, i.e., nodal points which specify pressure in a creeping flow analysis

NELMC = number of elements specified for the creep analysis

NELMT = number of elements specified for the temperature analysis

**OUTPUT:**

NP	XORD	YORD	NPBC	COBXXP	XBC	YBC
1	-.712E-03	.347E+07	-3	0.0000	0.	0.
2	.543E+04	.343E+07	-1	.1564	0.	0.
3	.107E+07	.330E+07	-1	.3090	0.	0.
4	.158E+07	.309E+07	-1	.4540	0.	0.
5	.204E+07	.201E+07	-1	.5078	0.	0.
6	.245E+07	.245E+07	-1	.7071	0.	0.
7	.201E+07	.204E+07	-1	.8090	0.	0.
8	.309E+07	.158E+07	-1	.8910	0.	0.
9	.330E+07	.107E+07	-1	.9511	0.	0.
10	.343E+07	.543E+04	-1	.9077	0.	0.
11	.347E+07	-.709E-07	-3	-1.0000	0.	0.
12	.343E+07	-.543E+04	-1	-.9877	0.	0.
13	.330E+07	-.107E+07	-1	-.9511	0.	0.
14	.309E+07	-.158E+07	-1	-.8910	0.	0.
15	.201E+07	-.204E+07	-1	-.8090	0.	0.
16	.245E+07	-.245E+07	-1	-.7071	0.	0.
17	.204E+07	-.201E+07	-1	-.5078	0.	0.
18	.158E+07	-.309E+07	-1	-.4540	0.	0.
19	.107E+07	-.330E+07	-1	-.3090	0.	0.
20	.543E+04	-.343E+07	-1	-.1564	0.	0.
21	-.712E-03	-.347E+07	-3	0.0000	0.	0.
22	-.543E+04	-.343E+07	-1	.1564	0.	0.
23	-.107E+07	-.330E+07	-1	.3090	0.	0.
	-.158E+07	-.309E+07	-1	.4540	0.	0.
	-.204E+07	-.201E+07	-1	.5078	0.	0.

TBC	TX	TY	TQ	CH	CX	CY
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900E+03 0.	0.	0.	0.	0.	0.	
.900F'						
.						

This is a complete listing of all nodal point data and parameters generated in program MESH. The only variables not already defined are the XORD and YORD arrays which are the x and y coordinates of the nodal point.

## OUTPUT:

ELEM	NP								
1	1	2	3	42	01	41	521	542	541
2	3	43	03	02	01	42	542	543	501
3	01	02	03	123	163	122	801	603	602
4	01	122	163	162	161	121	602	641	601
5	161	162	163	202	241	201	641	662	661
6	163	203	243	242	241	202	662	663	701
7	241	242	243	203	323	202	701	623	622
8	241	202	323	322	321	201	622	661	621
9	321	322	323	362	401	361	661	602	601
10	323	363	403	402	401	362	602	4'3	72'
11	401	402	403	443	403	442	721	767	
12	401	442	403	402	401	441	762		
13	3	4	5	44	03	43			
14	0	45	05	04	03	44			
15	03	04	05	125	165				
16	03	124	165	164	167				
17	163	164	165	204					
18	165	205	245	244					
19	243	244	245						
20	243	284	32*						
	323	324							

NPPE = 3

or

ELEM	NP								
1	1	2	3	42	01	41			
2	3	43	03	02	01	42			
3	01	02	03	123	163	122			
4	01	122	163	162	161	121			
5	161	162	163	202	241	201			
6	163	203	243	242	241	202			
7	241	242	243	203	323	202			
8	241	202	323	322	321	201			
9	321	322	323	362	401	361			
10	323	363	403	402	401	362			
11	401	402	403	443	403	442			
12	401	442	403	402	401	441			
13	3	4	5	44	03	43			
14	5	45	05	04	03	44			
15	03	04	05	125	165				
16	03	124	165	164	167				
17	163	164	165	204					
18	165	205	245	244					
19	243	244	245						
20	243	284	32*						
	323	324							

NPPE = 0

This is a listing of the NP-array which gives the nodal point numbers associated with each element. The first six numbers are the nodal points associated with the velocities and/or temperatures and they are listed in a counterclockwise sense. The next one or three numbers appear only for elements used for a creep analysis for which  $NPPE \neq 0$ . They represent nodal points for pressure.

PROGRAM WAVE--Output

Much of the output from WAVE is not used by the analyst and is provided more for the benefit of those who wish to know how the tape segments (blocks) are organized. This output is also helpful to determine storage requirements to within critical values. The following output is printed once for the creep analysis and once for the thermal analysis if both are called for.

## OUTPUT:

NSBG ID LISTX ICOMP IELEX MOVEX IEMPT IELE(I) LIST(I) NFR(I) MOVE(I) INTO(I)																
TAPE	SEGMENT	1	KVOL EQUALS				1365									
1	35	56	26	19	0	0										
1	2	3	4	5	6	7	0	9	10	11	12	13	14	15	16	17
2	42	82	122	162	202	242	282	322	362	402	442	482	522	562	602	642
124	163	203	164	204	244	1	81	41	161	121	241	201	243	283	323	321
363	403	443	483	481	441	5	45	85	125	165	205	245	285	325	284	
0																
0																
TAPE	SEGMENT	2	KVOL EQUALS				1220									
2	40	50	24	10	30	26										
20	21	22	23	24	25	26	27	28	29							
243	283	284	323	363	324	364	404	403	443	483	444	484	5	45		
126	165	205	166	1	81	41	161	121	241	201	321	281	401			
365	405	445	485	7	47	87	127	167	206							
34	35	56	36	41	42	43	44	47	48	49	50					
33	37	38	39	40	45	46	53	54	55							
1	2	3	4	5	9	10	11	14	15	19						
31	32	33	34	35	36	37	38	39	40							
TAPE	SEGMENT	3	KVOL EQUALS				1260									
3	40	51	23	11	26	24										
30	31	32	33	34	35	36	37	38								
246	246	245	285	286	325	365	326									
127	128	1	81	41	161	121										
467	407	447	487	9	49											
39	40	41														

This output shows the organization of each tape segment. The first two lines of headings describe the variables which are listed for each segment. Each listing begins with the tape segment number and KVOL which is the actual volume needed to store the matrix for that particular segment. For efficient use of storage this value should be "approximately" the same for all segments with the exception of the last segment.

The first row of six variables are defined by the first heading line mentioned above. They are:

- NSEG = segment number which has already been printed on the line with KVOL.
- IB = the maximum bandwidth for that particular segment.
- LISTX = the size of the array LIST. This is the number of equations in this tape segment.
- ICOMP = the number of completed equations in the current tape segment which are ready for the LDU decomposition.
- IELEX = the size of the array IELE. This represents the number of elements which will be added during this segment in the formation of the matrix.
- MOVEX = the size of the array MOVE. This is also the size of the array INTO and represents the number of rows and columns of the old segment of the stiffness matrix which must be moved to be compatible with the new segment.
- IEMPT = the number of empty rows of the current segment due to the completed rows of the last segment having been read onto disc.

The next several lines of output are the arrays indicated by the second heading line at the beginning of this output.

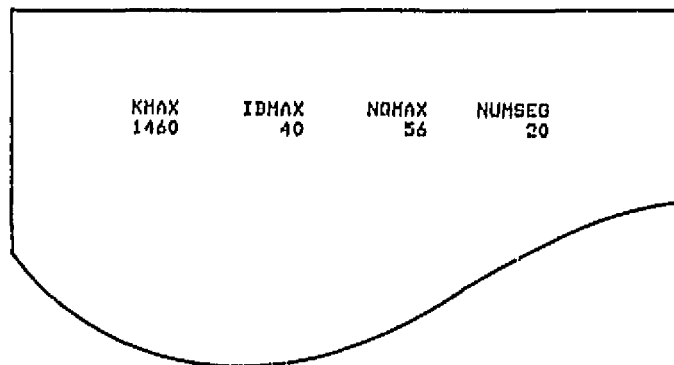
- IELE(I) = the elements added during this tape segment. The number will equal IELEX.
- LIST(I) = the nodal point numbers in this tape segment listed in the order they appear in the matrix. The number of entries equals LISTX.
- NPR(I) The printing of this array has been suppressed in the current version of the program. Its size is equal to the number of nodal points. Each entry has the following meaning:
  - NPR(I) = 0 indicates nodal point I has not yet been brought into the stiffness matrix, i.e., the front has not reached it.
  - NPR(I) = -1 indicates that nodal point I has been brought into the stiffness matrix and the decomposition has been completed and read onto low speed storage, i.e., the front has passed it.



NPR(I) = some positive integer value indicates that nodal point I is in the current segment (or front) and is located in the LIST array at the value indicated. This corresponds to the row of the matrix which corresponds to nodal point I.

MOVE(I) and INTO(I) indicate that row and column MOVE(I) should be "moved" "into" row and column INTO(I). The size of each array equals MOVEX.

#### OUTPUT:



This output summarizes the previous output from WAVE and probably represents all that is of interest to the analyst. It is printed both for the creep mesh data and for the thermal mesh data.

KMAX = maximum storage required for the stiffness matrix. For program CREEP this must be equal to or less than the dimensions specified for SKXX, SKXY, SKYX, and SKYY. For program TEMP this must be equal to or less than one-half the sum of these four dimensions.

IBMAX = the maximum bandwidth.

NQMAX = the maximum number of equations that appear any segment.

NUMSEG = number of tape segments.

PROGRAM COUPLE, CREEP, TEMP--Output

The remaining Output is the production data and the order in which it appears depends partly on the type of run being conducted. In the following discussion it is assumed that a coupled thermal-mechanical analysis has been made with program CREEP called first.

## OUTPUT:

INCR	ITERC	DELU	DELF	CP TIME
0	1	.101E-09	.121E+17	.276E+02
INCR	ITERC	DELT	CP TIME	
0	1	.226E+04	.451E+02	
INCR	ITERC	DELU	DELF	CP TIME
1	1	.111E-09	.133E+17	.650E+02
INCR	ITERC	DELT	CP TIME	
1	1	.223E+04	.824E+02	
INCR	ITERC	DELU		
2	1	.110E		

This information is printed every increment of the analysis.

INCR = the number of times control has been transferred back to program COUPLE from program CREEP and/or program TEMP. For a transient problem, this number will represent the number of increments of time taken since the beginning of the analysis. For a steady-state coupled analysis this number will represent the number of times both CREEP and TEMP have been called.

ITERC - ITERC = the number of iterations performed in CREEP and TEMP respectively within a given INCR (without returning control to COUPLE).

DELU = the absolute maximum change in the value of any component of the velocity matrix as a result of the current iteration.

DELF = the absolute value of the maximum change in the value found for any component of the force matrix as a result of the current iteration.

DELT = the absolute value of the maximum change in temperature found at any nodal point between iterations in TEMP (printed only when there are iterations in TEMP).

CP TIME = the time on the central processor "clock." By subtracting from the current value, the previous value, one obtains the length of time taken for the current iteration.

## OUTPUT:

```

TIME EQUALS 0.
NUMBER OF INCREMENTS EQUALS 2
      DELU      DELF      DELT      DELQ      ITERC      ITERI
      .151E+11 .341E+16 .324E+03 0.          1          1
DTIME EQUALS .100E+01
MAXIMUM DISPLACEMENT EQUALS 0.

```

This is the general heading at the beginning of a complete listing of the production output variables. It will be printed as often as specified by INCPR (see Input discussion).

TIME EQUALS refers to current simulated time during a transient analysis. For steady-state analyses this will always be zero.

NUMBER OF INCREMENTS EQUAL is self-explanatory.

DELU and DELF summarize the final values of these two variables printed in the previous output.

DELT = the absolute value of the maximum change found for any component of the temperature matrix during the last increment.

DTIME = the magnitude of the increment of time between this current time and the previous increment. It has meaning only for a transient analysis.

MAXIMUM DISPLACEMENT = maximum displacement given to any nodal point during the previous DTIME in a quasi Lagrangian analysis.

## OUTPUT:

N,P	XORD	YORD	UX	UY	UT	FTX	FTY
1	-.712E-03	.347E+07	-.11489E-34	.23892E-34	.30000E+04	-.42313E+15	.19444E+14
2	.543E+04	.347E+07	-.21404E-11	.34220E-12	.30000E+04	-.41155E+15	-.25452E+14
3	.107E+07	.330E+07	-.29774E-11	.94013E-12	.30000E+04	-.31477E+15	.91650E+13
4	.150E+07	.309E+07	-.25920E-11	.13711E-11	.30000E+04	-.05725E+15	.33356E+14
5	.204E+07	.201E+07	-.13106E-11	.95799E-12	.30000E+04	-.21691E+15	-.43117E+14
6	.245E+07	.245E+07	.33755E-13	.33755E-13	.30000E+04	-.69704E+15	.12743E+14
7	.201E+07	.204E+07	.96003E-12	-.13373E-11	.30000E+04	-.76792E+15	.15769E+13
8	.309E+07	.150E+07	.13067E-11	-.25635E-11	.30000E+04	-.76791E+15	.01054E+13
9	.310E+07	.107E+07	.95170E-12	-.29793E-11	.30000E+04	-.35530E+15	-.98711E+13
10	.347E+07	.543E+04	.34511E-12	-.21792E-11	.30000E+04	-.01563E+15	-.84936E+13
11	.347E+07	-.209E-07	.37200E-34	-.64006E-14	.30000E+04	.45313E+15	-.19341E+14
12	.347E+07	-.144E+04	.34130E-12	.21854E-11	.30000E+04	.01153E+15	.25452E+14
13	.330E+07	-.107E+07	.96727E-12	.29770E-11	.30000E+04	.31401E+15	-.91650E+13
14	.309E+07	-.150E+07	.13704E-11	.25914E-11	.30000E+04	.05724E+15	-.33356E+14
15	.204E+07	-.204E+07	.95799E-12	.13170E-11	.30000E+04	.21691E+15	.43117E+14
16	.245E+07	-.245E+07	-.33910E-13	-.33910E-13	.30000E+04	.69704E+15	-.12743E+14
17	.201E+07	-.201E+07	-.13324E-11	-.96002E-12	.30000E+04	.76792E+15	-.15769E+13
18	.150E+07	-.309E+07	-.25634E-11	-.13061E-11	.30000E+04	.76790E+15	-.01054E+13
19	.107E+07	-.330E+07	-.29793E-11	-.95174E-12	.30000E+04	.35530E+15	-.98711E+13
20	.543E+04	-.347E+07	-.21791E-11	-.34511E-12	.30000E+04	.01563E+15	-.84936E+13
21	-.712E-03	-.347E+07	-.23892E-34	.13001E-23	.30000E+04	.42313E+15	-.19444E+14
22	-.543E+04	-.347E+07	.21855E-11	-.34139E-12	.30000E+04	.01153E+15	.25452E+14
23	-.107E+07	-.330E+07	.29770E-11	-.96729E-12	.30000E+04	.31401E+15	-.91650E+13
24	-.150E+07	-.309E+07	.25914E-11	-.13704E-11	.30000E+04	.05724E+15	-.33356E+14
25	-.204E+07	-.201E+07	.13160E-11	-.95740E-12	.30000E+04	.21691E+15	.43117E+14
26	-.245E+07	-.245E+07	-.33910E-13	.33910E-13	.30000E+04	.69704E+15	-.12743E+14
27	-.201E+07	-.204E+07	-.96803E-12	.13324E-11	.30000E+04	.76792E+15	-.15769E+13
28	-.309E+07	-.150E+07	-.13061E-11	.25634E-11	.30000E+04	.76791E+15	-.01054E+13
29	-.310E+07	-.107E+07	-.95177E-12	.29793E-11	.30000E+04	.35530E+15	-.98711E+13
30	-.347E+07	-.543E+04	-.34511E-12	.21791E-11	.30000E+04	.01563E+15	-.84936E+13
31	.712E+07	.142E-02	-.31232E-34	.13001E-23	.30000E+04	.42313E+15	-.19444E+14
32	.543E+04	.347E+07	.23892E-34	.13001E-23	.30000E+04	.42313E+15	-.19444E+14
33	.107E+07	.330E+07	.29770E-11	.96729E-12	.30000E+04	.31401E+15	-.91650E+13
34	.150E+07	.309E+07	.25914E-11	.13704E-11	.30000E+04	.05724E+15	-.33356E+14
35	.204E+07	.201E+07	.13160E-11	.95740E-12	.30000E+04	.21691E+15	.43117E+14
36	.245E+07	.245E+07	-.33910E-13	.33910E-13	.30000E+04	.69704E+15	-.12743E+14
37	.201E+07	.204E+07	-.96803E-12	.13324E-11	.30000E+04	.76792E+15	-.15769E+13
38	.309E+07	.150E+07	-.13061E-11	.25634E-11	.30000E+04	.76791E+15	-.01054E+13
39	.310E+07	.107E+07	-.95177E-12	.29793E-11	.30000E+04	.35530E+15	-.98711E+13
40	.347E+07	.543E+04	-.34511E-12	.21791E-11	.30000E+04	.01563E+15	-.84936E+13
41	.712E+07	.142E-02	-.31232E-34	.13001E-23	.30000E+04	.42313E+15	-.19444E+14
42	.543E+04	.347E+07	.23892E-34	.13001E-23	.30000E+04	.42313E+15	-.19444E+14
43	.107E+07	.330E+07	.29770E-11	.96729E-12	.30000E+04	.31401E+15	-.91650E+13
44	.150E+07	.309E+07	.25914E-11	.13704E-11	.30000E+04	.05724E+15	-.33356E+14
45	.204E+07	.201E+07	.13160E-11	.95740E-12	.30000E+04	.21691E+15	.43117E+14
46	.245E+07	.245E+07	-.33910E-13	.33910E-13	.30000E+04	.69704E+15	-.12743E+14
47	.201E+07	.204E+07	-.96803E-12	.13324E-11	.30000E+04	.76792E+15	-.15769E+13
48	.309E+07	.150E+07	-.13061E-11	.25634E-11	.30000E+04	.76791E+15	-.01054E+13
49	.310E+07	.107E+07	-.95177E-12	.29793E-11	.30000E+04	.35530E+15	-.98711E+13
50	.347E+07	.543E+04	-.34511E-12	.21791E-11	.30000E+04	.01563E+15	-.84936E+13

This is the listing of all nodal point production variables.

Variables not yet defined are:

UX = the x-component of the nodal point velocity in global coordinates.

UY = the y-component of the nodal point velocity in global coordinates.

UT = nodal point temperature.

FTX, FTY are the  $x'$  and  $y'$  components of the reaction force acting on the nodal point. They are given in terms of the local coordinate axis defined by the direction cosine COSXXP(I). They are obtained by solving:

$$\{FT\} = [K]\{U\} - \{F_g\} - \{F_e\} - \{F_a\}$$

where  $F_g$  = the gravitational force

$F_e$  = the elastic force, and

$F_a$  = the applied forces.

Hence the FT forces arise only from reaction forces and should be zero except at constrained points. Because these values depend on the velocity vector, at least two iterations or increments must have been

made for them to have any meaning. For a nonlinear problem, these forces are in error until convergence has been reached. For this reason they serve as a convenient measure of convergence and indicate where the error is greatest.

OUTPUT:

NODAL POINT	PRS	QRS	IPQ	
28	.250000E+03	.250000E+03	1	2
29	.250000E+03	.250000E+03	3	4
30	.250000E+03	.250000E+03	5	6
31	.250000E+03	.250000E+03	7	8
32	.250000E+03	.250000E+03	9	10

These remaining nodal points (which have the same numbering as some nodal points for temperature when the temperature mesh is larger than the creep mesh) are used for specifying pressure. For ease in programming, each node has two pressures associated with it in the same manner as the components of velocity are specified. Both PRS and QRS are pressures associated with two different elements, designated by IPQ. Hence, in the example above, nodal point 30 is used to specify the pressure in elements 5 and 6. PRS is the pressure in element 5 and QRS is the pressure in element 6. For average incompressibility (NPPE = 1) there is only one pressure specified per element and therefore it is assumed constant throughout. For complete incompressibility (NPPE = 3), there are three pressures per element. Their values should be considered as located at the midside nodal points of the side connecting the elements designated by IPQ.

## OUTPUT:

ELM	SIGII	SIGXX1	SIGYY1	SIGXY1	SIGXX2	SIGYY2	SIGXY2	SIGXX3	SIGYY3	SIGXY3
1	.11970E+09	.3707E+08	-.121E+08	.733E+07	-.103E+09	.106E+09	-.922E+07	-.127E+09	.137E+09	.133E+08
2	.30530E+08	.434E+08	-.432E+08	-.141E+08	-.401E+08	.477E+08	.125E+08	.711E+08	-.476E+08	-.106E+07
3	.75360E+08	.499E+08	-.406E+08	.370E+08	.146E+08	-.130E+08	.260E+07	-.496E+08	.490E+08	.319E+08
4	.74360E+08	.159E+08	-.123E+08	.470E+08	-.201E+08	.297E+08	.217E+08	-.150E+08	.114E+08	.273E+08
5	.30490E+08	.209E+08	-.174E+08	.403E+08	.103E+07	-.114E+07	.217E+07	-.206E+08	.217E+08	.196E+08
6	.50344E+08	-.104E+08	.185E+08	.397E+08	.104E+08	-.220E+08	.203E+08	-.943E+07	.912E+07	.318E+08
7	.40017E+08	-.319E+08	.304E+08	.233E+08	.399E+08	.369E+08	.226E+08	.130E+08	-.170E+08	-.409E+07
8	.91660E+08	-.201E+08	.206E+08	.979E+07	.845E+08	-.065E+08	-.474E+08	.629E+08	-.441E+08	-.630E+08
9	.17650E+08	-.541E+08	.565E+08	-.620E+08	.973E+08	-.924E+08	-.103E+09	.855E+08	-.077E+08	-.105E+09
10	.1111E+09	-.670E+08	.525E+08	-.635E+08	.290E+08	-.859E+08	-.855E+08	-.313E+08	.377E+08	-.231E+08
11	.1327E+09	.397E+08	-.340E+08	-.436E+08	.447E+08	-.416E+08	-.372E+08	.846E+08	-.783E+08	-.555E+08
12	.1195E+09	.537E+08	-.517E+08	-.192E+08	.711E+08	-.689E+08	-.471E+07	.807E+08	-.740E+08	-.785E+08
13	.1323E+09	.654E+08	-.650E+08	-.910E+07	.701E+08	-.777E+08	.337E+08	.808E+08	-.785E+08	-.785E+08
14	.6773E+08	.200E+08	-.200E+08	.740E+07	.304E+08	-.304E+08	.274E+08	.413E+08	-.413E+08	-.413E+08
15	.5564E+08	.276E+08	-.273E+08	.106E+08	.224E+08	-.230E+08	.190E+08	.307E+08	-.307E+08	-.307E+08
16	.1201E+09	.321E+08	-.353E+08	.482E+08	.907E+07	-.100E+08	.701E+08	.701E+08	-.701E+08	-.701E+08
17	.7253E+08	.157E+08	-.174E+08	.429E+08	-.200E+08	.194E+08	.510E+08	.510E+08	-.510E+08	-.510E+08
18	.7255E+08	.219E+07	-.308E+07	.324E+08	-.304E+08	.202E+08	.460E+08	.460E+08	-.460E+08	-.460E+08
19	.6399E+08	.402E+07	-.474E+07	.318E+08	.110E+08	-.128E+08	.780E+08	.780E+08	-.780E+08	-.780E+08
20	.4327E+08	-.161E+08	.147E+08	-.256E+07	-.491E+08	.611E+08	.611E+08	-.611E+08	-.611E+08	-.611E+08
21	.5516E+08	.320E+08	-.291E+08	.663E+08	.474E+08	-.443E+08	.443E+08	-.443E+08	-.443E+08	-.443E+08
22	.9186E+08	.521E+08	-.409E+08	.371E+08	.377E+08	-.377E+08	.377E+08	-.377E+08	-.377E+08	-.377E+08
23	.1048E+09	.651E+08	-.618E+08	.270E+08	.679E+08	-.679E+08	.679E+08	-.679E+08	-.679E+08	-.679E+08
24	.7751E+08	.342E+08	-.349E+08	-.139E+08	.437E+08	-.437E+08	.437E+08	-.437E+08	-.437E+08	-.437E+08
25	.21E+08	.290E+08	-.304E+08	-.121E+08	.437E+08	-.437E+08	.437E+08	-.437E+08	-.437E+08	-.437E+08
26	.260E+08	-.276E+08	-.276E+08	-.276E+08	-.276E+08	-.276E+08	-.276E+08	-.276E+08	-.276E+08	-.276E+08

This figure shows the production output for each element.

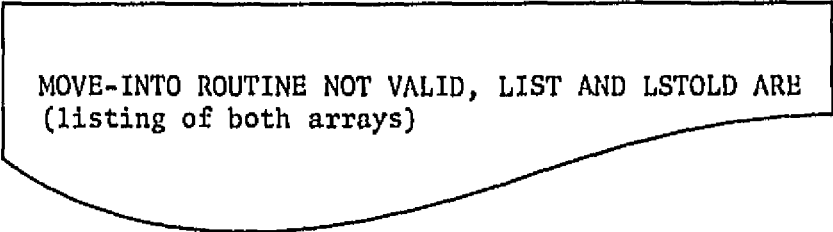
SIGII = the effective stress as calculated at the center quadrature point

SIGXX, SIGYY, and SIGXY 1,2, and 3 are the three components of the stress tensor at their separate quadrature points. (See Appendix for designation of all quadrature points.)

Error Messages

Most of the error messages in MANTLE are self-explanatory. Those which are not self-explanatory or which require special action to correct, will now be discussed.

## OUTPUT:



```
MOVE-INTO ROUTINE NOT VALID, LIST AND LSTOLD ARE
(listing of both arrays)
```

This message results from there not being sufficient room to rearrange the stiffness matrix from its orientation specified by the LIST array of the previous segment to the orientation needed for the segment under consideration. This causes an automatic STOP of MANTLE. The two LIST arrays printed are difficult to interpret and will not be discussed here. However, three pursuits to alleviate this problem can be made.

1. Increase KVOL (and correspondingly the dimension of COMMON/C7/).

When this is possible it is usually the easiest and best approach. If this is not possible, decreasing KVOL can sometimes remedy the situation. This occurs simply because any change in KVOL results in an entirely new arrangement of the tape segments. It often happens that one value of KVOL will work when a KVOL either higher or lower will not. KVOL can be made as low as zero and still produce legitimate tape segments.

2. Check to see if the first tape segment has a bandwidth (IB) significantly lower than the bandwidth of the last segment printed.

If so, increase IBMIN to equal the largest IB so far printed.

3. Specify a new order in which the elements are taken, to produce a smaller "front" (see Appendix for more explanation).

If all three methods have been exhausted the finite element matrix is too large for the computer. The only alternative, in this case, is to reduce the size of the matrix. This can be done by reducing the number of elements, or for a creep analysis, reducing NPPE to zero (penalty method).

#### OUTPUT:

NOTE, KMAX EQUALS, , WHICH EXCEEDS THE MAXVOL OF, ,

This is a warning message and will not cause the run to stop. It occurs when KVOL is greater than the specified MAXVOL. It is important to note that this is not a check on the actual dimensions used in COMMON/C7/. It can, in fact, be desirable to specify MAXVOL less than the available storage in order that other arrays associated with the size of tape segments remain within their limits. If KMAX does exceed the actual dimensions of COMMON/C7/, simply reduce the specified MAXVOL.



OUTPUT:

ERROR IN IORDER ARRAY, IORDER IS ARRAY PRINTED

This message indicates that in listing the order of the elements as data, the user has made an INPUT error. Automatic stop.

OUTPUT:

NEGATIVE DETJ, TAPE SEGMENT,

This occurs when the determinate of the Jacobian matrix is found to be negative during the Gaussian quadrature. This will be the case anytime the nodal point numbers in the NP array are not listed in a counterclockwise order as well as when an element has a reentrant side of significant magnitude. These conditions are most likely to occur in a quasi-Lagrangian analysis when the elements have become extremely distorted due to the flow. Automatic stop.

OUTPUT:

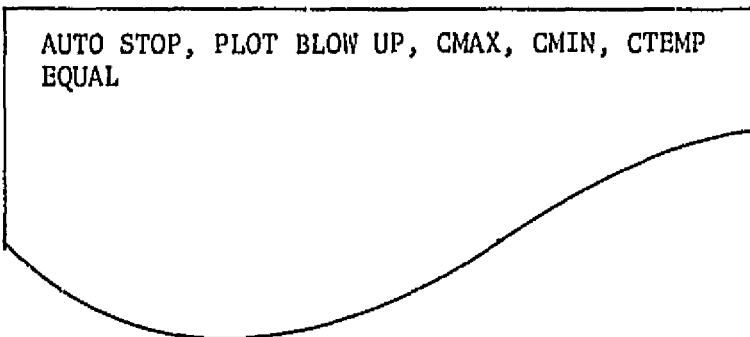
SINGULARITY DURING LDU



This occurs during the Gauss elimination whenever a diagonal term is found to be zero at the time that zeros are being placed beneath it. Strictly speaking this should never occur and most likely means a major error in the compiled program. It can possibly arise if a given element is overly constrained, i.e., most nodal points given known velocities. Automatic stop.

OUTPUT:

AUTO STOP, PLOT BLOW UP, CMAX, CMIN, CTEMP  
EQUAL



This occurs anytime that 200 or more contours are to be plotted in one element. This would normally represent an error in the analysis or a poor choice of contour interval. In either case the computer time to perform the plotting is often large and therefore an automatic stop is called for.

### User Subroutines and Functions

There is always a balance between selecting a very general program where a wide variety of problems may be solved by entering sufficient information as data, and selecting a very dedicated program designed for a select group of problems. The advantages of the general program are rather obvious but it has the disadvantage of requiring a large amount of input data and it is usually much larger than a dedicated program. On the other hand, the dedicated program has the obvious disadvantage of being limited in its scope but has the advantage of being operable with a minimum amount of input, usually shorter run times, and less total storage. One method which incorporates the advantages of both methods is to make available to users the ability to write their own dedicated subroutines. This approach was used in the design of MANTLE and these subroutines will be discussed here. The user will be able to bring most of the pertinent information into these subprograms through the common blocks. That information not contained in common will be brought in through arguments.

FUNCTION GAMX(TEMPK, XK, YK, MJ)

FUNCTION GAMY(TEMPK, XK, YK, MJ)

These two functions represent body forces (a force per unit volume) in the x and y directions. They are called by CREEP during the matrix assembly at each quadrature point so that Gaussain quadrature can be used to integrate the total body force. This takes place in a J-DO-loop over each element and a K-DO-loop over the quadrature points.

TEMPK = temperature at quadrature point K

XK, YK = x and y coordinates of quadrature point K

MJ = material number of element J

FUNCTION G (TEMPK, MJ)

G is the shear modulus of material MJ at temperature TEMPK. This value is needed for a steady-state elasto-visco-plastic analysis. There is much research still to be done with regard to these analyses and the user should proceed with caution. To by-pass this analysis in CREEP (i.e., produce a purely viscous analysis) G should be set equal to a negative number, e.g.,

$$G = -1.0$$

This has the same effect as using a very large G but with much less computer cost. The use of G is limited to steady-state analyses.

SUBROUTINE VISC (VS, VT, PENLTY, NPPE, EPSII, TEMPK, XK, YK, MJ)

Rather than a FUNCTION statement for viscosity, a subroutine was chosen so that more than one value could be returned to CREEP. These values are VS, VT and PENLTY and are defined below:

VS = the viscosity based on a secant modulus. That is, VS is the value of  $\mu$  at quadrature point K to be used in

$$\tau_{ij} = 2\mu\epsilon_{ij}$$

and is, in general, a function of the material, MJ; the effective strain rate, EPSII; the coordinates XK, YK, and the temperature, TEMPK. When  $\mu$  is a function of the strain rate it is possible to use a viscosity based on the tangent of the stress-strain rate curve. That is, the tangent viscosity, VT, defined as

$$d\tau_{ij} = 2\mu_T d\epsilon_{ij}.$$

The third variable, PENLTY, is the penalty function. Its value is zero unless the number of pressure points per element, NPPE, is zero. In this case it is set equal to  $1000.0 * VS$ , i.e., a large number in comparison to the viscosity. PENLTY appears as  $\lambda$  in the equations

$$\sigma_{ij} = 2\mu\epsilon_{ij} + \lambda\epsilon_{kk}\delta_{ij}$$

$$d\sigma_{ij} = 2\mu_T d\epsilon_{ij} + \lambda(d\epsilon_{kk})\delta_{ij} \quad .$$

It is seen therefore, that by equating  $\mu$  and  $\lambda$  to the elastic constants  $G$  and  $\lambda$  it is possible to use CREEP for the analysis of elastic solids. Because six nodal point elements are used, full integration is used for the penalty function rather than a reduced integration.

Two words of caution are necessary. In a viscous flow analysis, a small  $\lambda$  does not represent a compressible flow analysis. When a small  $\lambda$  is used in an elastic analysis, the stresses calculated in MANTLE represent neither the total stress nor the stress deviator.

FUNCTION RHO(MJ, IELEJ)

FUNCTION CP(MJ, IELEJ)

FUNCTION RKX(MJ, IELEJ)

FUNCTION RKY(MJ, IELEJ)

These four functions represent the density, specific heat, and thermal conductivity in the  $x$  and  $y$  direction respectively for material MJ. These parameters are assumed constant within each element and hence are called only once for each element (in contrast to VISC for example, which is called at each quadrature point). The element number is also made an argument in case it is desirable to know location, temperature or other information which can be obtained by knowing the element rather than the material. In these cases it is only necessary to include the appropriate COMMON blocks in the function programs.

SUBROUTINE MSHADJ

This subroutine is called near the end of both MESH programs and is provided in order that the analyst can make adjustments in the finite element mesh. These adjustments can take a number of forms, the most common being slight adjustments of boundary coordinates to fit unusual shapes. Other adjustments could be specification of boundary conditions along curved boundaries which vary according to some specified function, changing a direction of one or more diagonals in the mesh, and adding additional elements.

SUBROUTINE BNDRYSUBROUTINE BNDRYCSUBROUTINE BNDRYT

These three routines are called during COUPLE, CREEP and TEMP respectively. BNDRY is called only once and can be used to make any adjustments necessary for a specific run. The next two are called every increment and are intended for the purpose of altering boundary conditions during an iterative analysis, but can be used for other purposes as well. As an example, boundary conditions could be altered due to material coming into contact with other boundaries during the flow analysis. Free surface adjustments for steady-state analysis also are made within these sub-routines. BNDRYC is called at the end of each iteration within CREEP. BNDRYT is called at the beginning of each iteration in TEMP.

SUBROUTINE STIFF(IELEJ, ITV)

This subroutine is called immediately after the stiffness matrix has been formed and before it has been assembled into the global stiffness matrix. The element number is an argument as well as ITV which indicates whether a creep analysis or a temperature analysis is in progress.

The purpose of this subroutine is to provide the user the opportunity to make unusual changes in the matrix before assembly. As examples, a penalty function could be used to force two or more components of velocity to be the same. Also, alterations in the matrix could be made to simulate a failure plane across the element. The use of this routine is limited to those users who are fairly familiar with the finite element method of approximation.

## ILLUSTRATIVE EXAMPLES

In this section several problems are presented to help the user establish some valid test runs. The examples are limited to mantle convection, however the user may consult the reference to obtain additional examples.

Example 1. Pure Conduction

This problem can be used to check SUBROUTINE TEMP. Figure 2 illustrates the problem which is one of steady-state, pure conduction in an infinite cylinder. The dimensions used coincide with the earth's mantle. Both the inside and outside temperatures are constant and there is no convection. Table I compares the exact solution with that obtained by MANTLE. Any row of nodal points in the radial direction, such as emphasized in Figure 2, can be used for comparison.

Inside:

$$r = 0.347 \times 10^7 \text{ m}$$

$$T = 900.0 \text{ }^\circ\text{C}$$

Outside:

$$r = 0.637 \times 10^7 \text{ m}$$

$$T = 300.0 \text{ }^\circ\text{C}$$

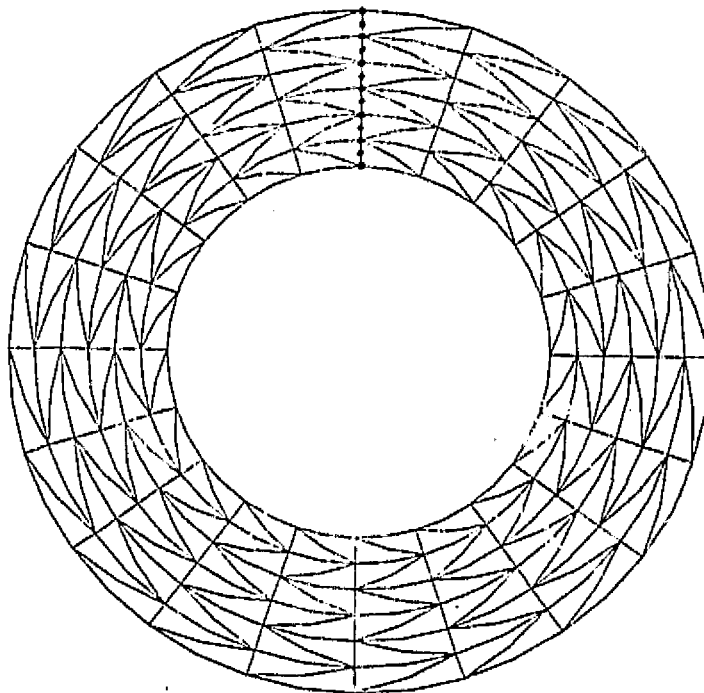


Figure 2



TABLE I. Solution to Example Problem I

Nodal Point	Radial Distance $\times 10^{-7}$	TEMPERATURE	
		MANTLE Solution	Exact Solution
1	0.347	900.00	900.00
41	0.371	833.54	833.94
81	0.395	771.26	772.02
121	0.419	712.67	713.76
161	0.444	657.36	656.52
201	0.468	604.97	604.52
241	0.492	555.22	555.12
281	0.516	507.85	508.08
321	0.540	462.64	463.18
361	0.565	419.41	418.47
401	0.589	377.98	377.38
441	0.613	338.23	337.93
481	0.637	300.00	300.00

Exact Solution:

$$\text{Temperature} = -977.744 \, r + 15775.093$$

Example 2. Thick-Walled Cylinder

This problem can be used to check SUBROUTINE CREEP. Figure 3 illustrates the problem which is a thick-walled cylinder under an external traction. Again, the dimensions are those of the earth's mantle and the element layout is the same as that used in the previous example. Table II compares the exact solution with that obtained by MANTLE, where the velocity is in the radial direction.

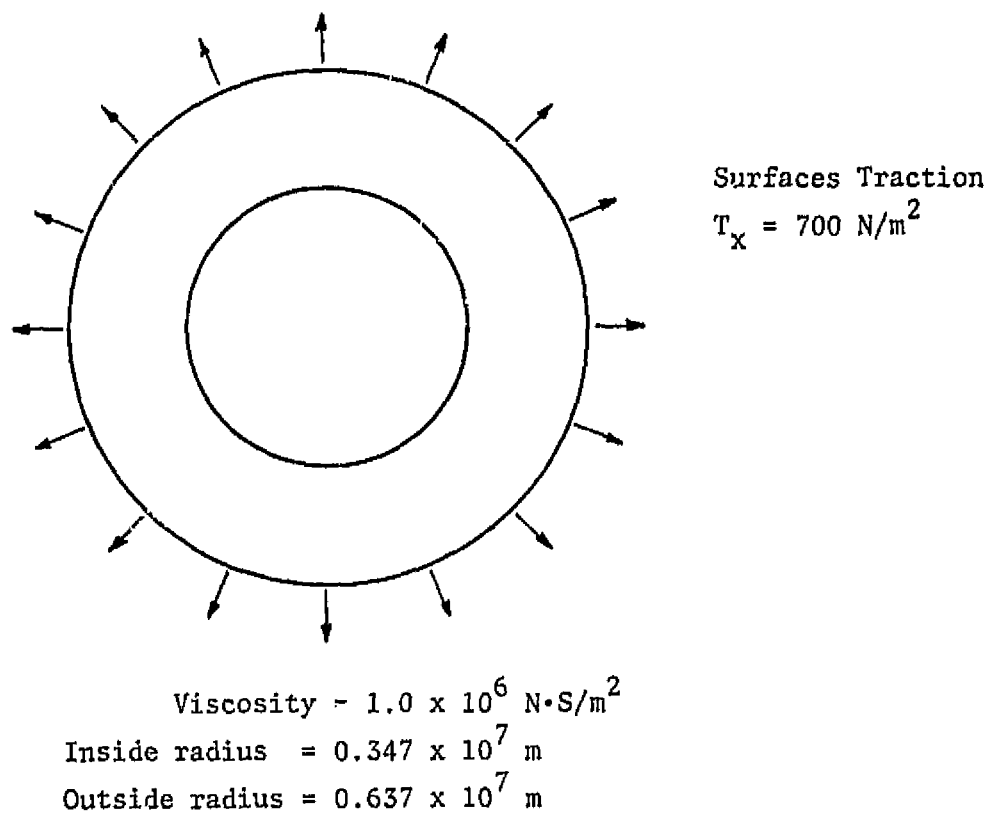


Figure 3

TABLE II. Solution to Example Problem 2

Nodal Point	Radial Distance $\times 10^{-7}$	RADIAL VELOCITY	
		MANTLE Solution $\times 10^{-4}$	Exact Solution $\times 10^{-4}$
1	0.347	0.17255	0.17270
41	0.371	0.16139	0.16152
81	0.395	0.15157	0.15171
121	0.419	0.14288	0.14302
161	0.444	0.13513	0.13497
201	0.468	0.12819	0.12805
241	0.492	0.12193	0.12180
281	0.516	0.11625	0.11613
321	0.540	0.11108	0.11097
361	0.565	0.10636	0.10603
401	0.589	0.10202	0.10174
441	0.615	0.09803	0.09776
481	0.637	0.09433	0.09407

Exact Solution:

$$\text{Velocity} = \frac{5.992565 \times 10^9}{r}$$

General Solution:

$$\text{Velocity} = \frac{1}{2\mu} \left[ \frac{r_i r_o (T_o - T_i)}{r_o^2 - r_i^2} \right] \frac{1}{r}$$

where

 $r_i$  = inside radius $r_o$  = outside radius $T_i$  = inside surface traction $T_o$  = outside surface traction

### Example 3. Simple Convection

This example as well as those that follow, deal with convection within the mantle. With the exception of the final example, it has been assumed that convection extends the entire depth of the mantle. Although there is some evidence that this might be the case, there is no attempt in this report to try to prove or disprove that theory. The choice was made simply to provide some interesting examples with a minimum of computer cost. If the upper mantle is modeled, and an entire cross section of the earth is taken, many more elements are needed to provide a reasonable element aspect ratio than are needed for modeling the full mantle.

The parameters used for these analyses (see Table III) represent typical values as reported in the literature, the exception to this being the viscosity. In order for the iterative process used in the analysis to converge, the number of elements must increase with the Rayleigh number. For this reason, the Rayleigh number was kept near its critical value, which for full mantle convection requires a viscosity larger than that normally assumed for the mantle. However, the examples do indicate the versatility of the code, and are of interest in themselves.

This example assumes a constant viscosity within the mantle equal to  $10^{25}$  N·S/m<sup>2</sup>. The fluid is allowed to slip at both the inside and outside boundaries. In order to decrease computer cost, a slightly coarser mesh was used than in the previous examples and is shown in Figure 4. In all examples, NPPE = 0, which results in the penalty function approach. There are, therefore, a total of 440 nodal points, giving 880 degrees of freedom for the velocity field.

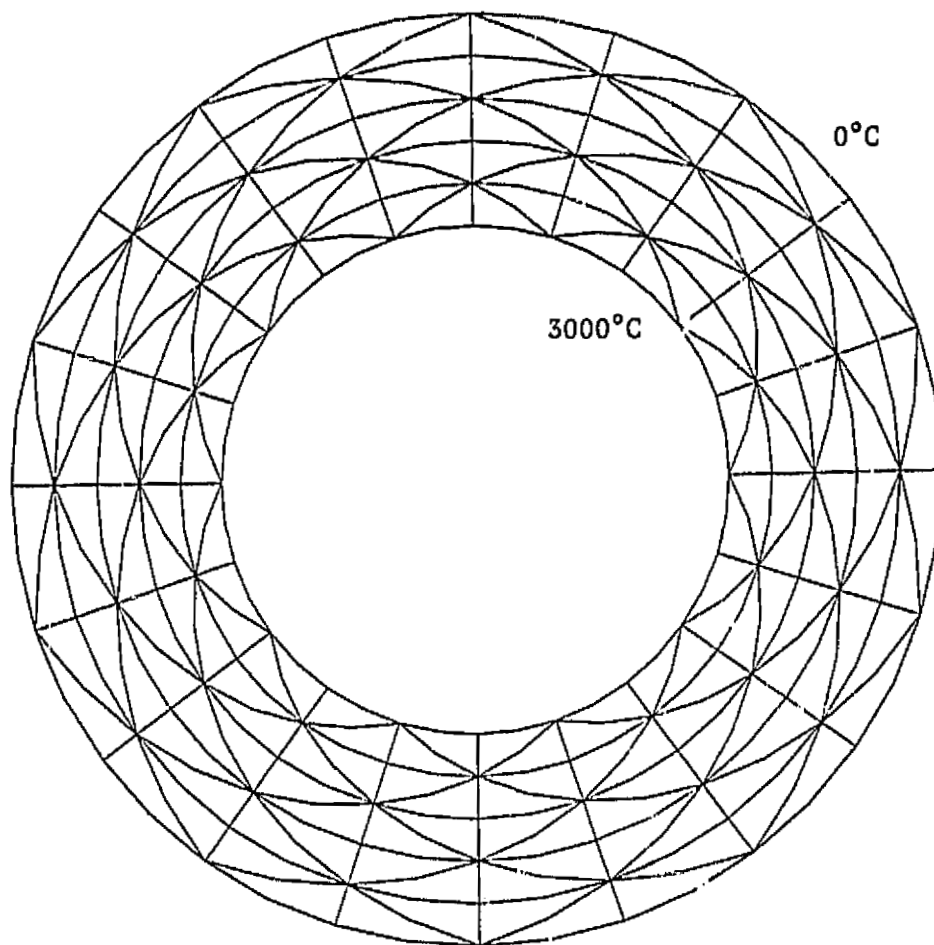
	FUNCTION GAMX(TEMPK,XK,YK,MJ) F=-0.74*TEMPK R=SQRT(XK**2+YK**2) GAMX=-F*XK/R RETURN END	Dimensions $\rho g \alpha = [\text{kg/m}^2 \cdot \text{s}^2 \cdot ^\circ\text{C}]$ R, XK, YK [m] TEMPK [ $^\circ\text{C}$ ]
C	FUNCTION GAMY(TEMPK,XK,YK,MJ) F=-0.74*TEMPK R=SQRT(XK**2+YK**2) GAMY=-F*YK/R RETURN END	GAMX, GAMY [ $\text{kg/m}^2 \cdot \text{s}^2$ = $\text{N/m}^3$ ]
C	FUNCTION G(TEMPK,MI) G=-1.0 RETURN END	G [ $\text{N/m}^2$ ]
C	FUNCTION RHO(MJ,IELEJ) RHO=.370E+04 RETURN END	RHO [ $\text{kg/m}^3$ ]
C	FUNCTION CP(MJ,IELEJ) CP=1.2E+03 RETURN END	CP [ $\text{J/kg} \cdot ^\circ\text{C}$ ]
C	FUNCTION RKX(MJ,IELEJ) RKX=6.66 RETURN END	RKX, RKY [ $\text{J/m} \cdot \text{s} \cdot ^\circ\text{C}$ ]
C	FUNCTION RKY(MJ,IELEJ) RKY=6.66 RETURN END	
C	SUBROUTINE VISC(VS,VT,PENLTY,NPPE,EPSII,TEMPK,XK,YK,MJ) VS=5.0E+24 VT=5.0E+24 PENLTY=0.0 IF(NPPE,EQ,0) PENLTY=1000.0*VS RETURN END	VS, VT [ $\text{N} \cdot \text{s/m}^2$ ]
C		

TABLE III. Parameters for Example Problems 3-7  
Excluding Viscosity

The steady state isotherms are shown in Figure 5. Iterations were begun using a constant temperature field with four temperature "spikes" evenly spaced. After five iterations, the largest change in any nodal point temperature from the previous iterations was 10.5°C. The largest surface velocity was found to be  $1.793 \times 10^{-11}$  m/s or 0.05655 cm/year. If the Rayleigh number is defined as

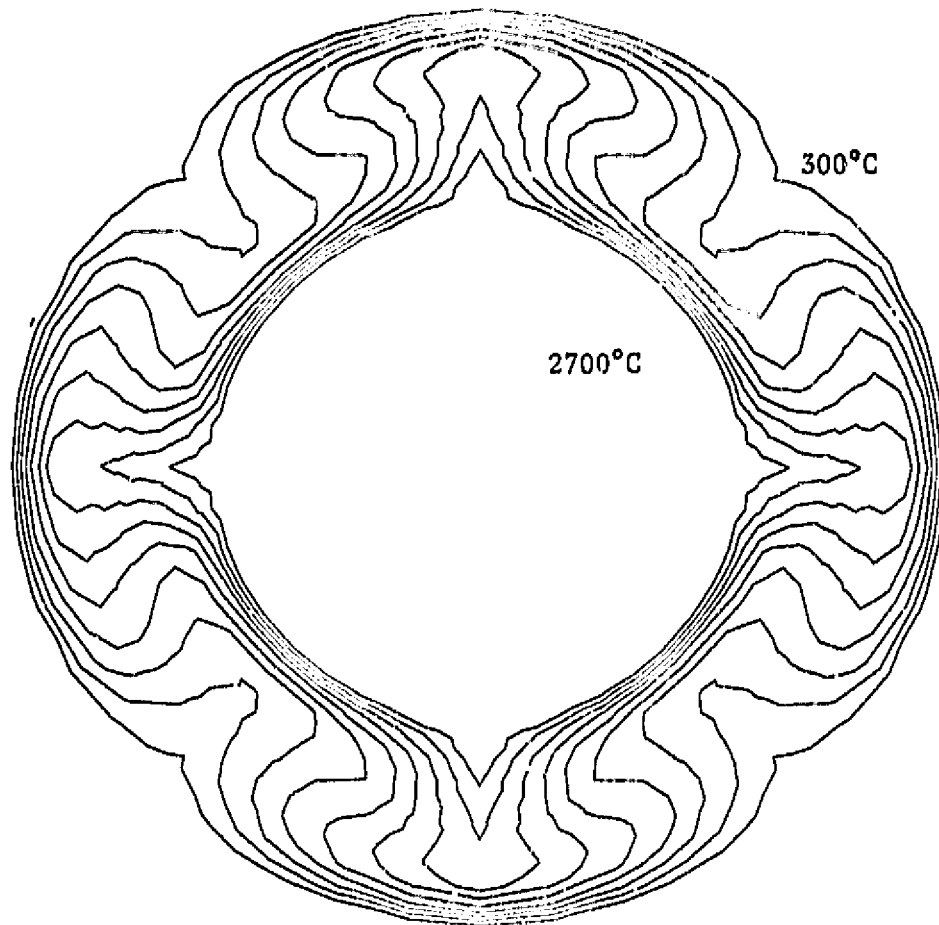
$$R = \frac{(g\alpha\rho)(\Delta T)(\rho C_p)d^3}{k\mu}$$

where  $d$  = depth of mantle, then, for this example  $R = 3610$  or about 5.5 the critical value.



Inside radius =  $0.347 \times 10^7$  m  
Outside radius =  $0.637 \times 10^7$  m

Figure 4. Mesh for Examples 3-6



Contour interval = 300°C

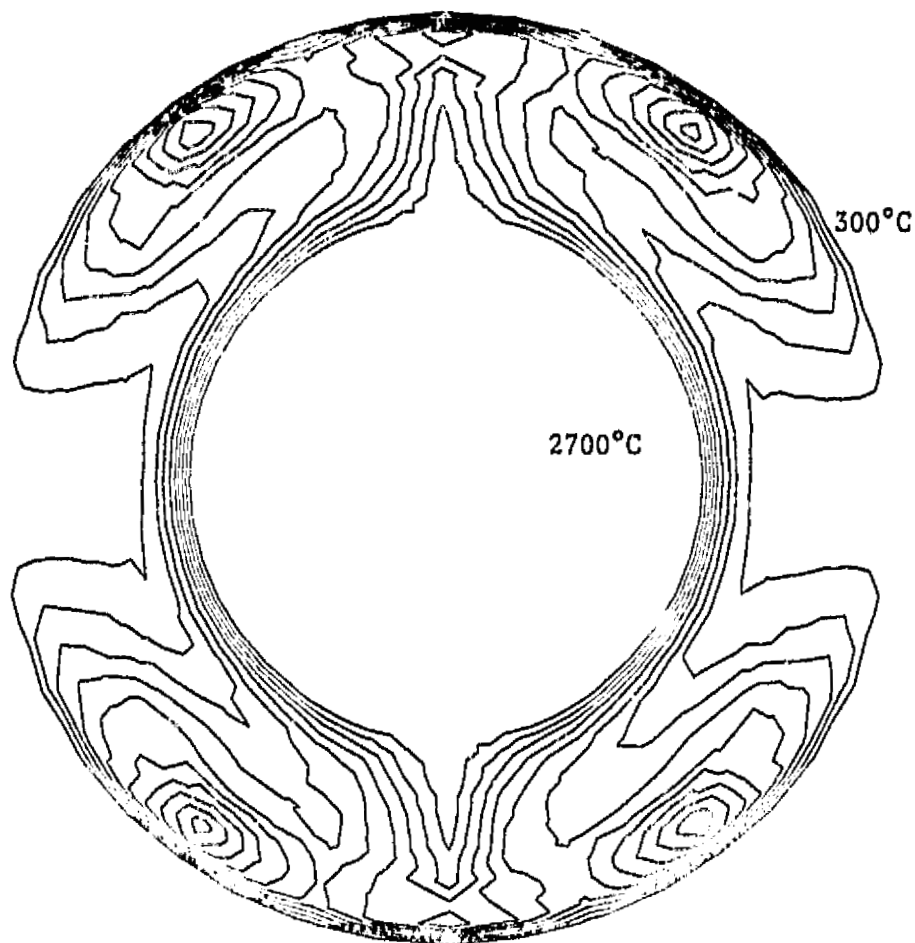
Figure 5. Isotherms--  
Example Problem 3



Example 4. Low Viscosity Zone

This example was chosen to illustrate the use of a variable viscosity. Although viscosity can be specified as a function of temperature, position, and/or strain-rate, only position (in this case, depth) was used for this example. The viscosity of the mantle at a depth greater than 700 km was taken as  $10^{25} \text{ N}\cdot\text{S}/\text{m}^2$  as in the previous example. In the upper mantle, depths less than 700 km, the viscosity was taken as  $10^{23} \text{ N}\cdot\text{S}/\text{m}^2$ .

The problem was initialized with the temperature field from the previous example. As the iterations continued, the eight-cell pattern broke into a four-cell pattern. For this particular case, steady-state was not reached. Figure 6 illustrates the results found on the 13th iteration. During the next iteration, the temperature became unstable and the solutions begin to diverge. However, it is believed that the overall flow pattern is reasonably correct. It is interesting to note that reducing the viscosity in the upper mantle resulted in larger convection cells. It is not unlikely that a finer mesh in the upper mantle would have resulted in smaller convection cells in that region, giving rise to a two-scale convection pattern.



Contour interval = 300°C

Figure 6. Isotherms--  
Example Problem 4

Example 5. Axisymmetric Convection

The previous two examples have assumed plane flow. This example shows what effect the constraint of plane flow can have on the analysis. Only half of the mantle cross section, as shown in Figure 7, was used. The viscosity was assumed to be a constant  $10^{25}$  N·S/m<sup>2</sup>. The problem was first analyzed as one of plane flow. The isotherms for this are shown in Figure 7a. Next, the flow was analyzed as axisymmetric. These results are shown in Figure 7b. Again, the four-cell (eight-cell full cross section) convection pattern broke into a two-cell (four-cell full cross section) pattern. Both results were obtained to a convergence limit of less than 50°C between iterations. The maximum surface velocity found for the two cases was essentially the same at 0.06 cm/year.

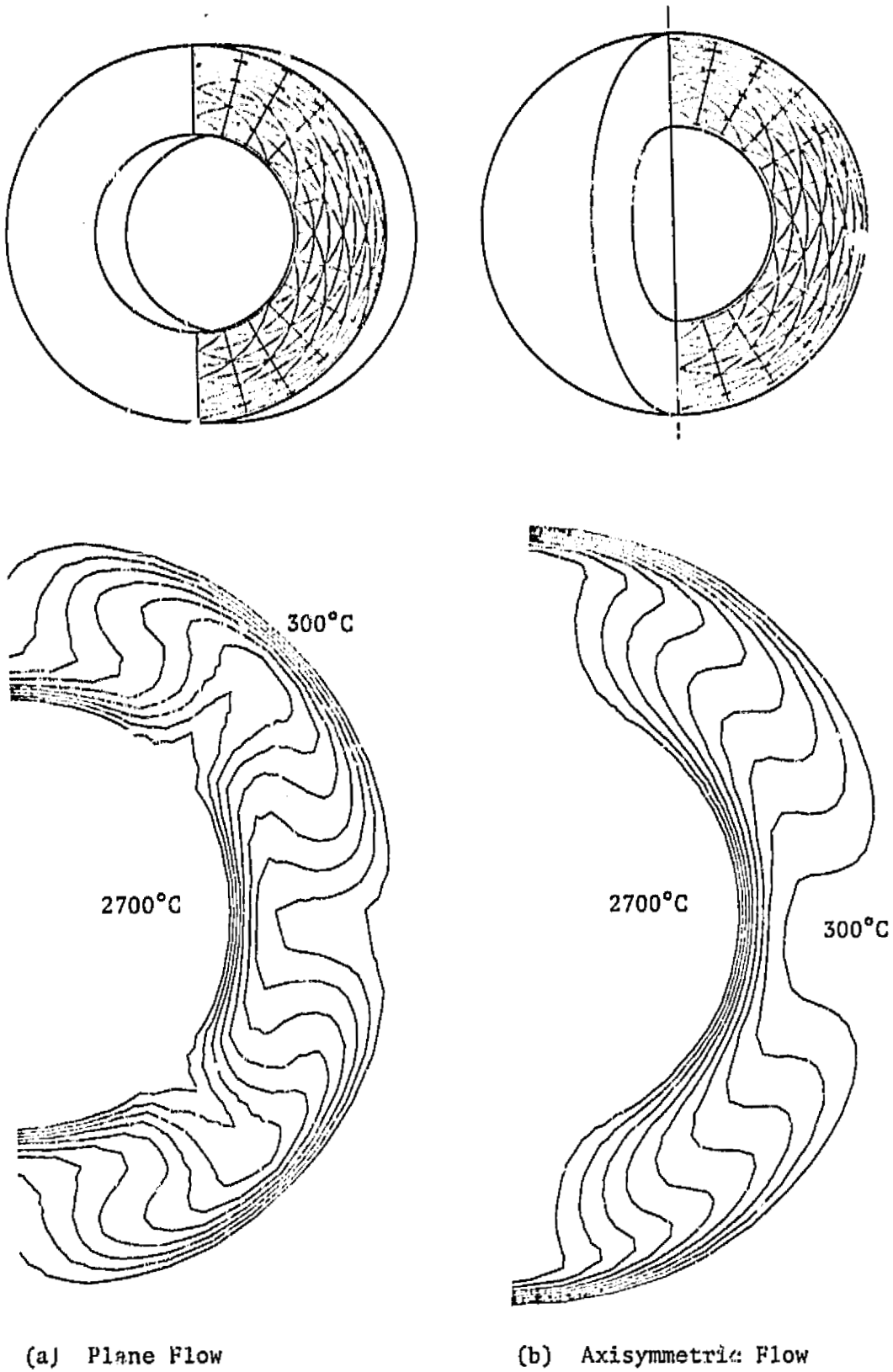
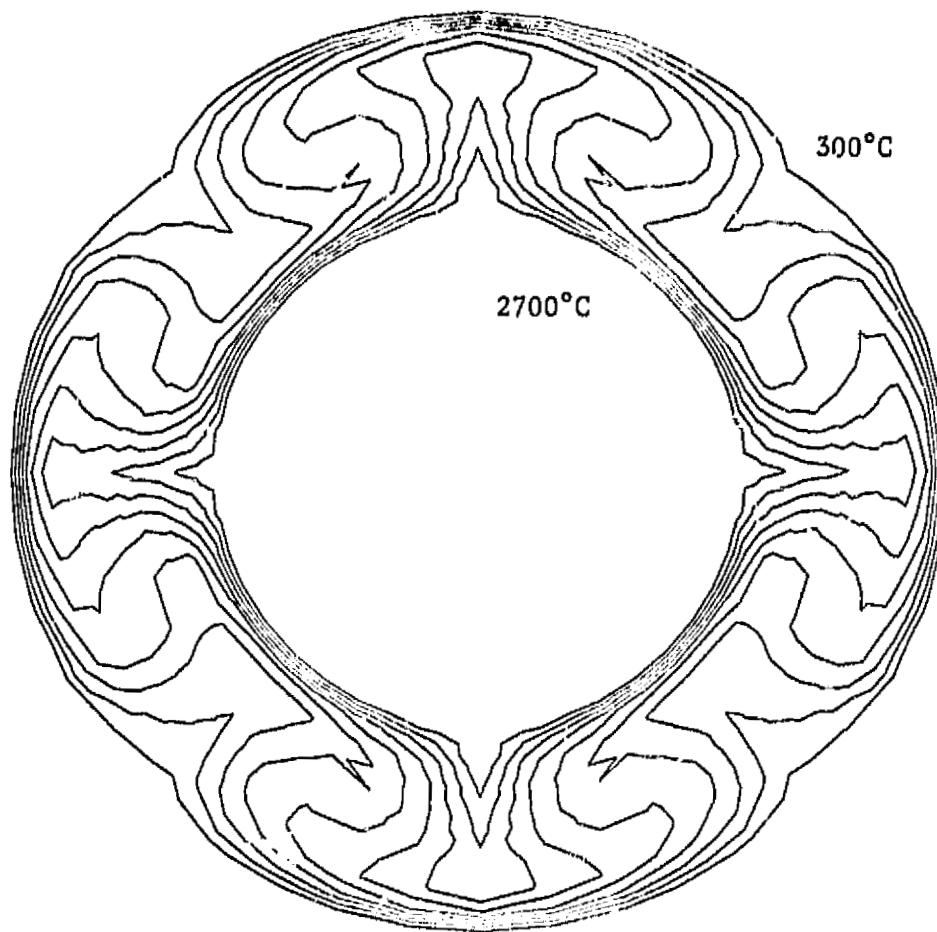


Figure 7. Comparison of Axisymmetric and Plane Flow--  
Example Problem 5

### Example 6. Crustal Plates

This example was conducted to illustrate one use for SUBROUTINE STIFF, as well as to examine possible effects on mantle convection due to rigid crustal plates. The plates were simulated by requiring that groups of nodal points on the surface have the same tangential velocity. Once again, plane flow was assumed, hence the "plates" are on the outside of an infinite cylinder. The viscosity was taken as  $0.5 \times 10^{25} \text{ N}\cdot\text{S}/\text{m}^2$ . The steady-state solution with no plates (similar to Example Problem 3) is shown in Figure 8. Figure 9 shows the plates that were first assumed and the isotherms and streamlines resulting from the analysis. Clearly, the additional constraint imposed by the plates results in an entirely different flow pattern. For this four-plate example, the flow was counterclockwise at all surface points. In an effort to eliminate this characteristic and thus create a more interesting flow pattern, one plate was removed. Figure 10 illustrates the new plate configuration with the resulting isotherms and streamlines.

For both the four-plate analysis and the three-plate analysis, convergence was not obtained. Although the iterative process did not diverge, large changes in temperature (and thus flow patterns) resulted between each iteration. The figures shown are simply typical of those that were found. It is not fully understood why convergence was not obtained but it is assumed that additional elements would help to alleviate the oscillations. One could study these oscillations by terminating the iterative procedure and beginning a transient analysis. This should eventually lead to a steady state condition.



Contour interval =  $300^{\circ}\text{C}$

Figure 8. Steady State Isotherms for  
Example 6, No Plates

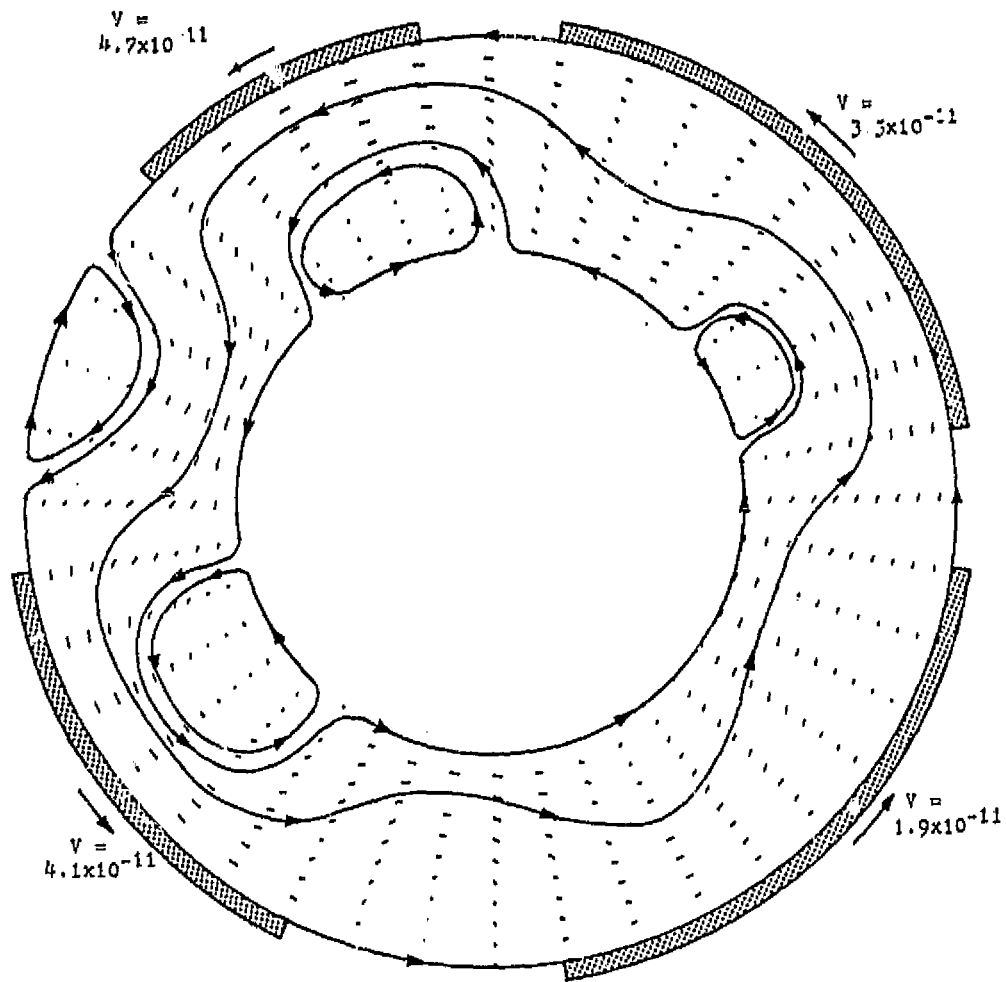


Plate velocities in m/s

Figure 9a. Streamlines for Four-Plate Analysis--  
Example Problem 6

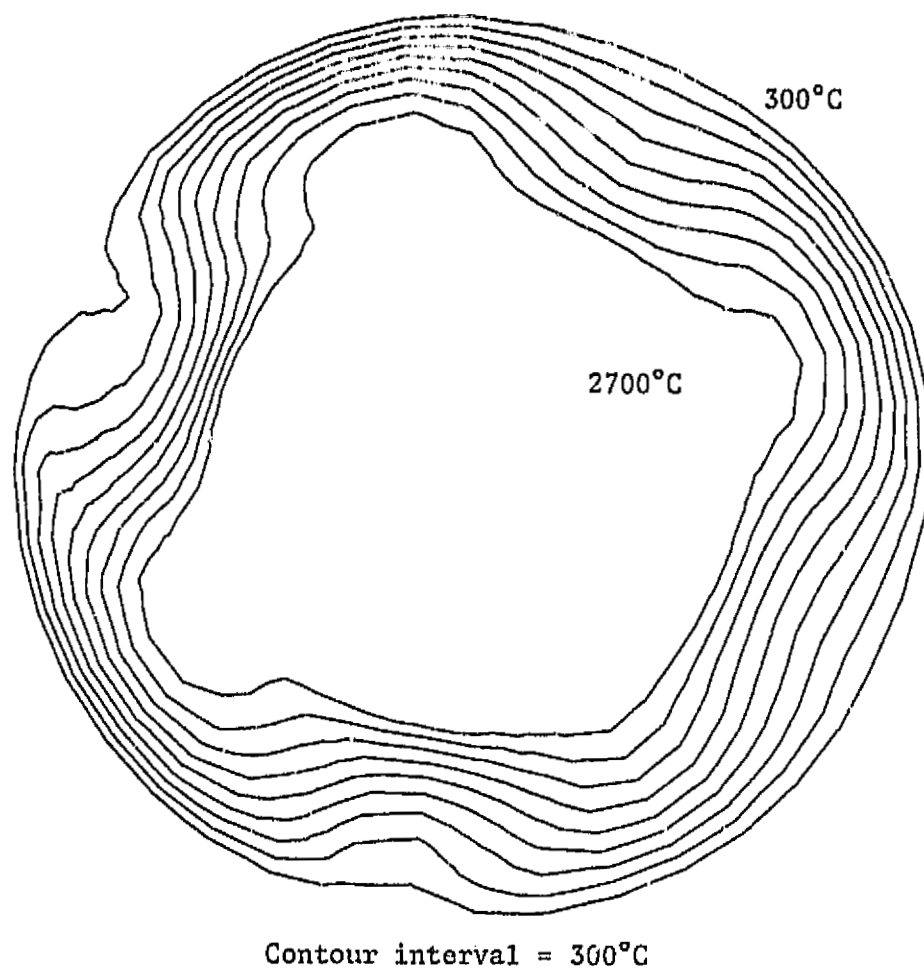


Figure 9b. Isotherms for Four-Plate Analysis--  
Example Problem 6



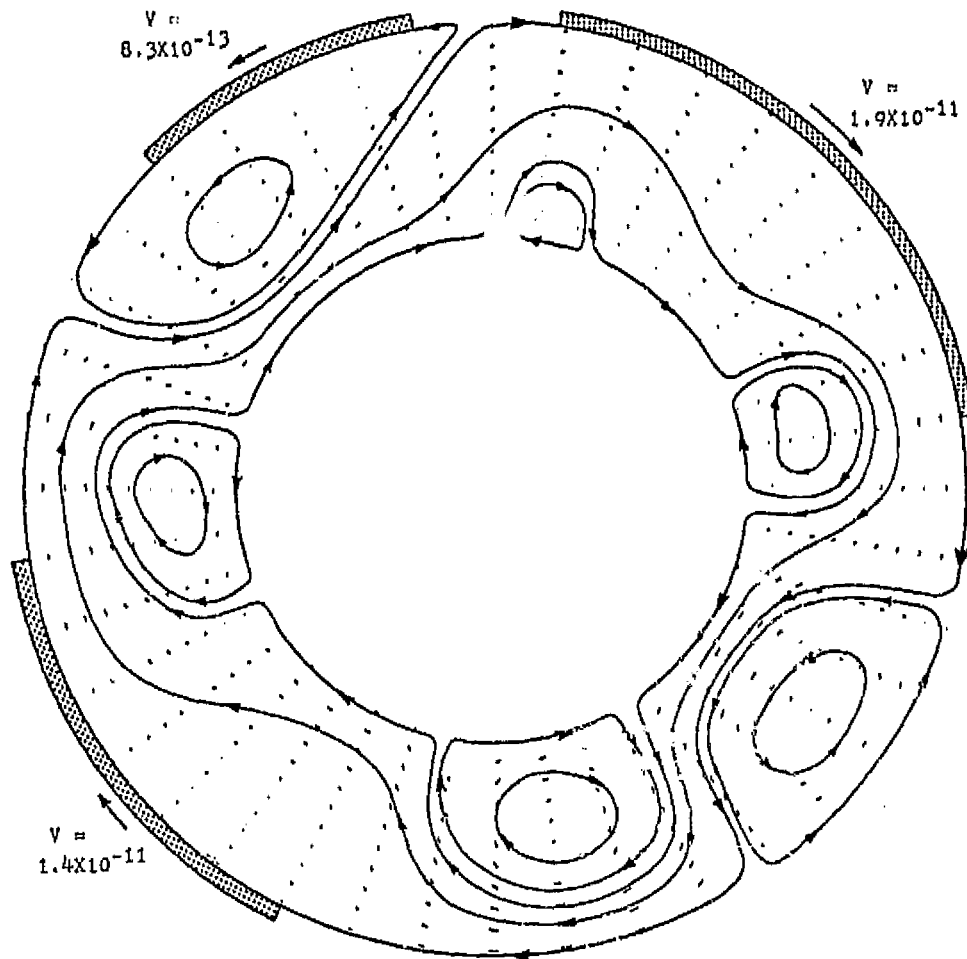


Plate velocities in m/s

Figure 10a. Streamline for Three-Plate Analysis--  
Example Problem 6

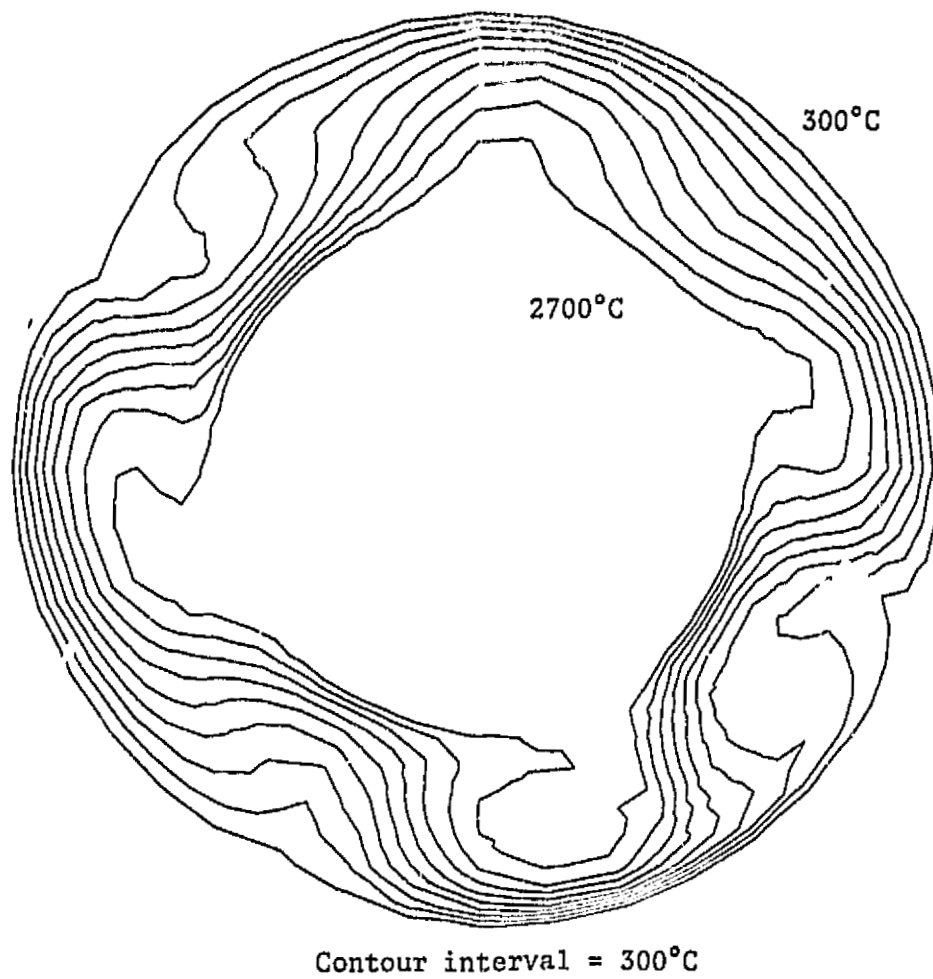


Figure 10b. Isotherms for Three-Plate Analysis--  
Example Problem 6

The surface constraints were imposed using the penalty function method. The tangential components of two points  $n$  and  $m$  are the same if

$$UY(m) - UY(n) = 0.$$

This constraint can be incorporated into the variational principles by adding to the original functional term

$$\lambda (UY(m) - UY(n))^2$$

where  $\lambda$  is assigned a large value. Perhaps an easier to understand explanation of the method is to consider an additional one-dimensional element which connects the two nodal points being considered. Figure 11 illustrates such an element as a connecting link between the two points

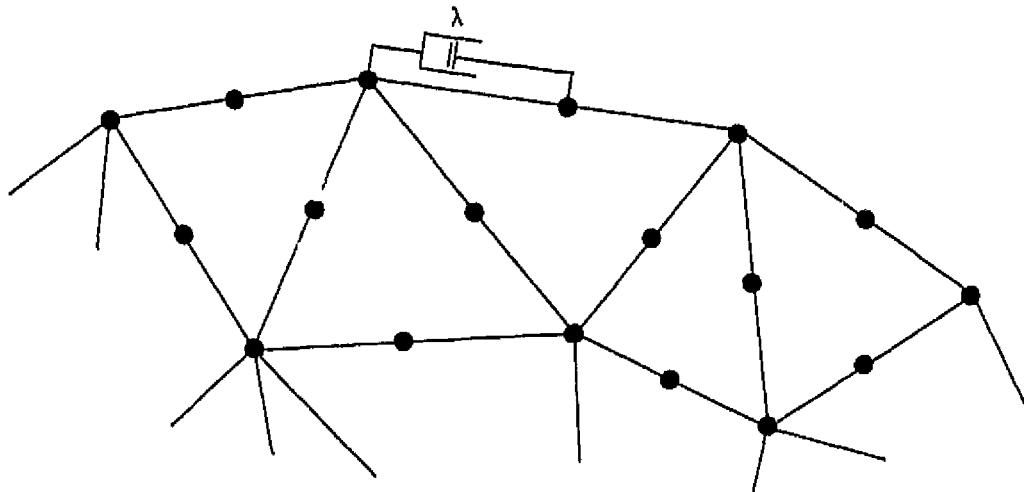


Figure 11

with a viscosity  $\lambda$ . The stiffness for this one-dimensional element is expressed by

$$\begin{bmatrix} \lambda & -\lambda \\ -\lambda & \lambda \end{bmatrix} \begin{Bmatrix} UY(m) \\ UY(n) \end{Bmatrix} = \begin{Bmatrix} FY(m) \\ FY(n) \end{Bmatrix}.$$

If  $\lambda$  becomes large and there are no corresponding large terms added to the right-hand side, it is clear that the desired constraint will be realized. The small  $2 \times 2$  stiffness matrix is simply added to the element stiffness in SUBROUTINE STIFF.

One other aspect of the code is illustrated in this example. The nodal point boundary conditions specified in the array NPBC(I) will be unchanged by even additions of  $\pm 10$ , provided the sign is not changed. That is, the following specifications will accomplish the same nodal point boundary conditions:

NPBC = 4 and NPBC = 14 and NPBC = 24  
 or  
 NPBC = -2 and NPBC = -12 and NPBC = -32  
 or  
 NPBC = 3 and NPBC = 43 and NPBC = 123, etc.

This allows the user to specify special boundary conditions to be used in any of the subroutines provided. In the present case, any two adjacent nodal points in the same element with an absolute value of NPBC greater than 10 were taken to have the same  $y'$ -component of velocity.

```

SUBROUTINE STIFF(IELEJ,ITV)
COMMON/C1/
1 XORD( 550),YORD( 550),XBC( 550),YBC( 550),TBC( 550),
2 CX( 550),CY( 550),CH( 550),TX( 550),TY( 550),TQ( 550),
3 COSXP( 550),NPBC( 550),NP( 300, 6)
COMMON
1 TXX(9,9),TXY(9,9),TYX(9,9),TTY(9,9),
1 SXX(9,9),SXY(9,9),SYX(9,9),SYY(9,9),
1 SPX(6,3),SPY(6,3),
1 SIGXXJ(3),SIGYYJ(3),SIGXYJ(3),SIGTHJ(3),
1 DNQDX(3),DNQDY(3),
1 RJAC(2,2),RJACI(2,2),DNDX(6),DNDY(6)
IF(ITV.EQ.-1) RETURN
DO 500 I=1,6
  NPI=NP(IELEJ,I)
  NBCI=IABS(NPBC(NPI))
  IF(NBCI.LT.10) GO TO 500
  BIG=SYI(I,I)*10000.0
  DO 400 J=I,6
    IF(I.EQ.J) GO TO 400
    NPJ=NP(IELEJ,J)
    NBCJ=IABS(NPBC(NPJ))
    IF(NBCJ.LT.10) GO TO 400
    TYY(I,I)=TYY(I,I)+BIG
    TYY(I,J)=TYY(I,J)-BIG
    TYY(J,I)=TYY(J,I)-BIG
    TYY(J,J)=TYY(J,J)+BIG
    SYY(I,I)=SYY(I,I)+BIG
    SYY(I,J)=SYY(I,J)-BIG
    SYY(J,I)=SYY(J,I)-BIG
    SYY(J,J)=SYY(J,J)+BIG
  GO TO 500
400 CONTINUE
500 CONTINUE
RETURN
END

```

SUBROUTINE STIFF for Example Problem 6

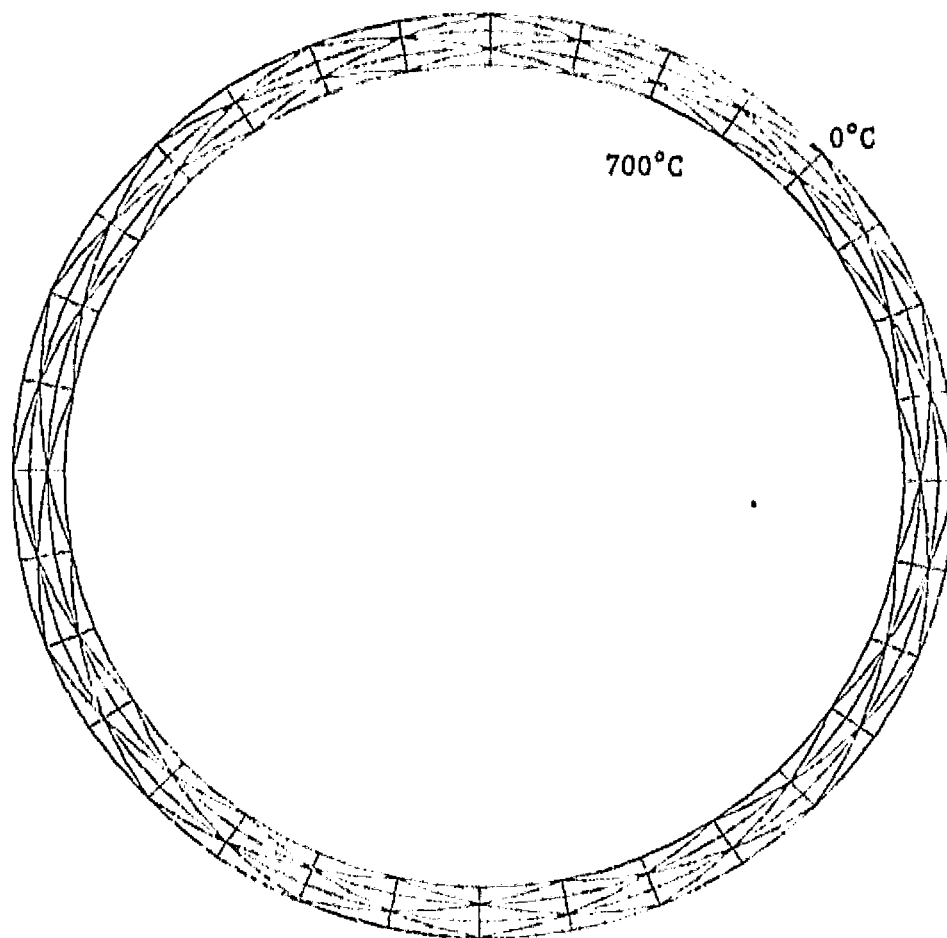
Example 7. Upper Mantle Convection

The final example illustrates convection in the upper mantle. Figure 12 shows the element layout and dimensions for the analysis. Because of the reduced depth of convection, the viscosity was reduced to  $0.3 \times 10^{23} \text{ N}\cdot\text{S}/\text{m}^2$  which gave a Rayleigh number equal to

$$R = 3948.$$

The analysis was not completed to convergence and after five iterations showed signs it might diverge. This could easily be corrected by the addition of more elements, particularly in the radial direction.

Figure 12b, which illustrates the isotherms obtained on the fifth iteration, does indicate, however, what one would expect, that is, convection cells with aspect ratios approximately equal to one. It is interesting to note that the maximum surface velocity for this case was found to be 0.34 cm/year, which compares favorably with actual plate velocities.



viscosity =  $0.3 \times 10^{23} \text{ N}\cdot\text{S}/\text{m}^2$   
inside radius =  $0.567 \times 10^7 \text{ m}$   
outside radius =  $0.637 \times 10^7$

Figure 12a. Mesh for Upper Mantle--  
Example Problem 7

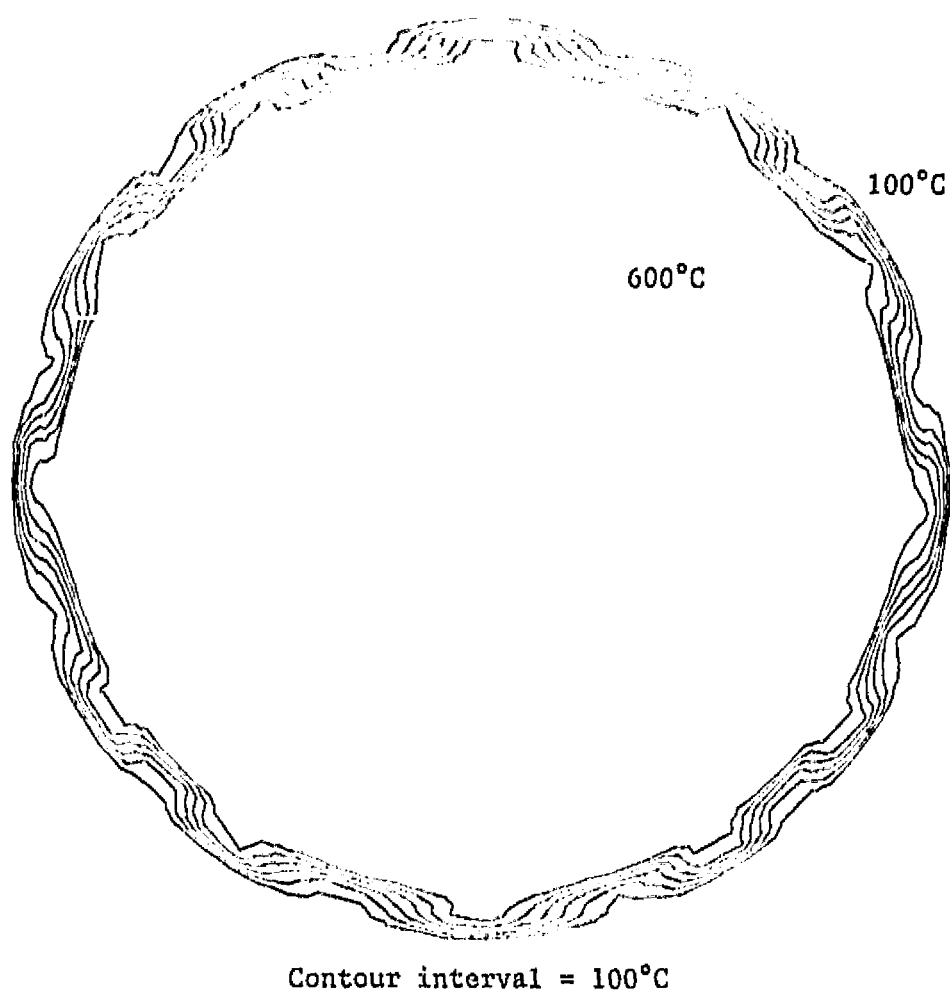


Figure 12b. Isotherms in Upper Mantle--  
Example Problem 7



## CONCLUSION

The previous examples demonstrate the use of MANTLE to solve many of the complex questions related to mantle convection. In these examples, the coarseness of the mesh adds a significant constraint on the flow. Actual investigations into particular problems should be made with a much finer mesh. Also, the examples did not demonstrate many of the features of MANTLE, particularly transient problems including the use of quasi-Lagrangian analysis. The user is referred to the literature cited in Appendix D for many more examples covering a wider scope of application.

As with all computer codes, improvements are continually being incorporated into MANTLE. At the time of this report, a new version of the code which includes inertia terms in the equations of motion, the ability to model compressible elastic solids in the elasto-viscoplastic analysis, an improved iterative technique for elasto-viscoplastic analysis, and improved solution algorithms for the matrix equations are being incorporated. These additions are not likely to significantly change the input/output variables as described in this report. New versions of the code will be obtainable from the author when ready.

**PRECEDING PAGE BLANK NOT FILLED**

## APPENDIX A

## MESH2 GENERATOR AND IORDER ARRAY

MESH2

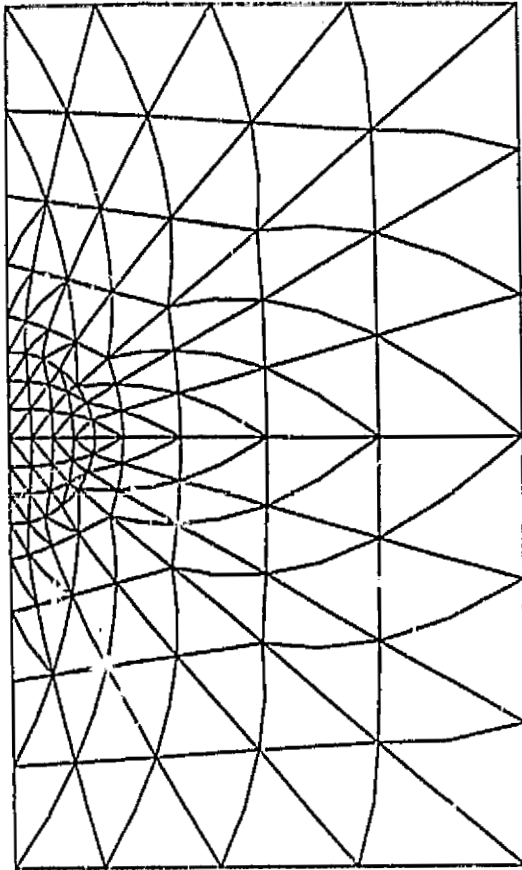
PRECEDING PAGE BLANK NOT FILLED

The basic building block of MESH2 is a quadrilateral region referred to as a loop. Any number of loops may be joined together to form a completed mesh. Figure 1 illustrates how four such loops were joined together to form the completed mesh shown. The four sides of a loop are parabolas specified by two corner points (dark circles in Figure 2) and side point (open circles in Figure 2).

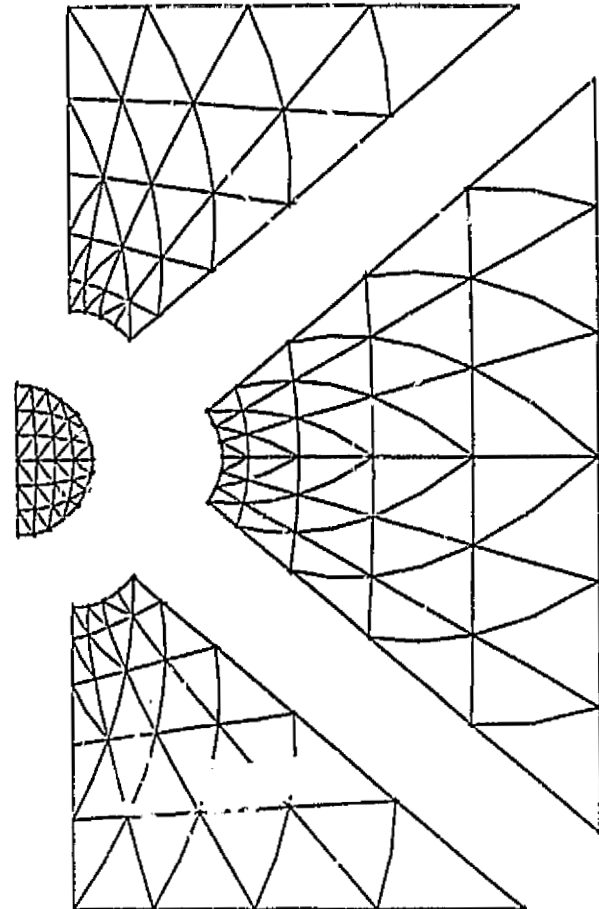
The eight points specifying the shape of a given loop must begin with a corner point and be read in counterclockwise. The first three points then define side 1 of that loop with the other sides numbered consecutively counterclockwise. The user specifies how many divisions are desired on side 1 and side 2 of each loop (NDIV-array). By use of the coordinates of the side points, the divisions (or elements) may then be "pushed" to one side or the other, e.g., compare the elements of loop 3 and the location of the side points for that loop.

Loops are joined together through the use of array-JOIN. The data cards for JOIN are listed below for the four loops shown in Figure 1 and Figure 2. Note, that each card only specifies how the present loop is joined to already existing loops.

loop	side							
	1		2		3		4	
1	0	0	0	0	0	0	0	0
2	0	0	0	0	1	1	0	0
3	2	2	0	0	0	0	1	2
4	1	3	3	3	0	0	0	0



Example of a Completed Mesh



Exploded View Showing the 4  
Loops Used in MESH

Figure 1

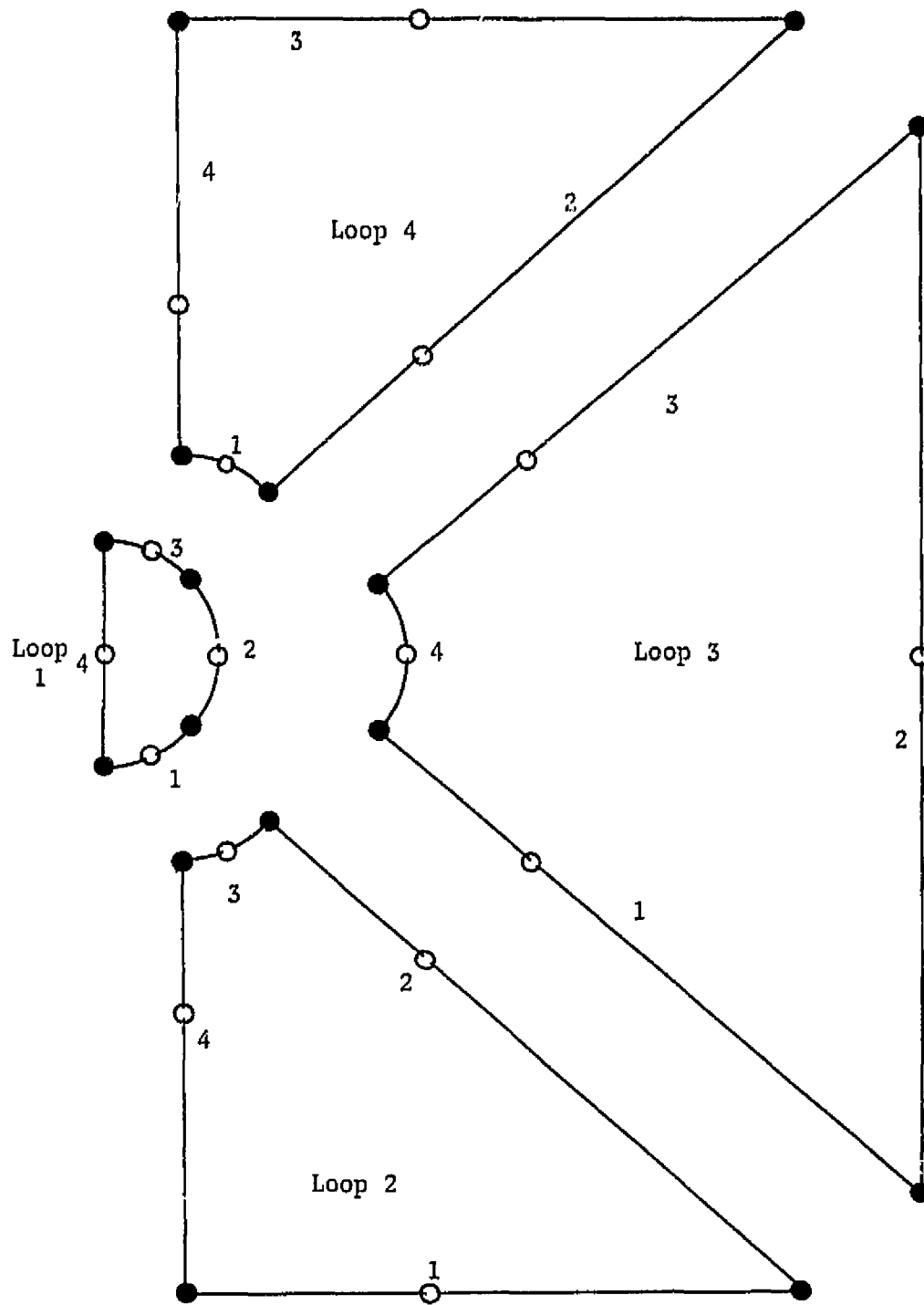


Figure 2

The two numbers listed for each side represent the loop and side respectively to which that side is joined. Hence, the above cards state for loop 1, there are no previous loops to which it is joined (this is always the case); for loop 2, its side 3 is joined to loop 1, side 1; for loop 3, its side 1 is joined to loop 2, side 2 and its side 4 is joined to loop 1, side 2; and for loop 4, its side 1 is joined to loop 1, side 3 and its side 2 is joined to loop 3, side 3.

Figure 3 indicates the nodal point numbering and element numbering assigned by MESH for a given loop. For loops greater than the first, these numbers simply take up where the last previous loop left off. Where a side is joined to a previous loop, the previous loop numbers are used and the numbers of the current loop are adjusted so that they remain in consecutive order.

The diagonals within each "square" are taken as the shortest distances. When those distances are equal, the direction of the diagonal is alternated as shown.

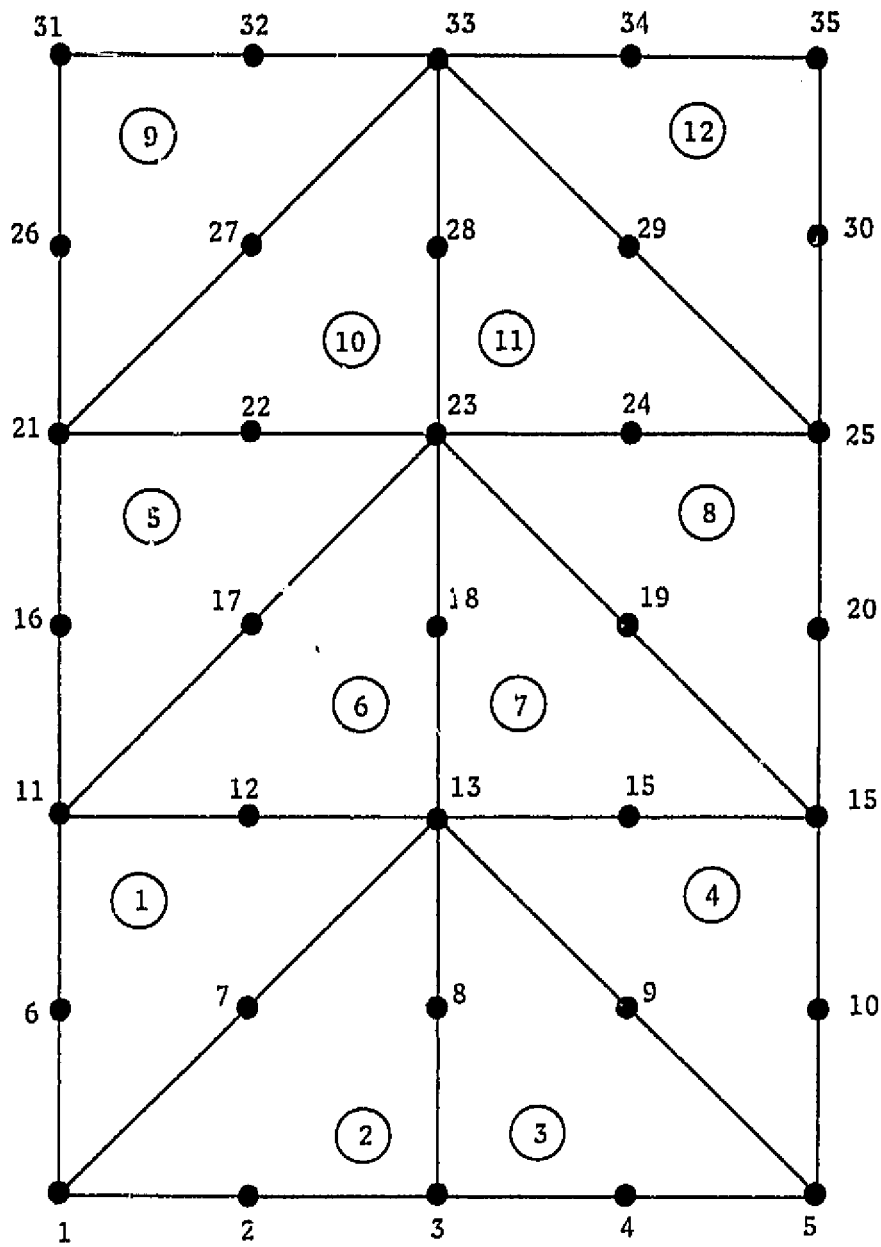


Figure 3

### Nodal Point Numbering and the IORDER Array

In order to keep the bandwidth of the stiffness matrix as small as possible it is possible to specify the order in which the elements will be selected. Some general guidelines are now given which will help the user select an efficient ordering.

At any stage of a frontal analysis it is always possible to draw one or more lines across the mesh which separates the elements not yet assembled from those already assembled. It is the number of nodal points along these lines which generally determine the bandwidth necessary for the analysis. It is therefore desirable to select an element order so that the maximum number of nodes at any stage in the frontal process is a minimum. Figure 4 shows a sequence of such lines, a-a through m-m, for the sample mesh. Although none of the lines shown happen to be the one where the maximum number of points will occur, they do give a good visual sensation of the "front" passing through the mesh. Lines d-d through lines j-j all have 19 nodal points along them. As the front moves from the line g-g, so that it picks up the next element on the left between line g-g and h-h, four new nodal points will be added without a single loss. Hence, in this case the maximum number of points appears to be 23 rather than 19. The easiest way to select the path of the front is to locate what would appear to be the line passing through the mesh which would contain the largest number of points and yet be an obvious front for the analysis. In the example this was assumed to be line g-g. From this line, it is a rather relatively easy matter to work outward from there so as to cover the entire mesh with a sequence of such lines. In the above example the IORDER proceeded from the element at the beginning of row A,

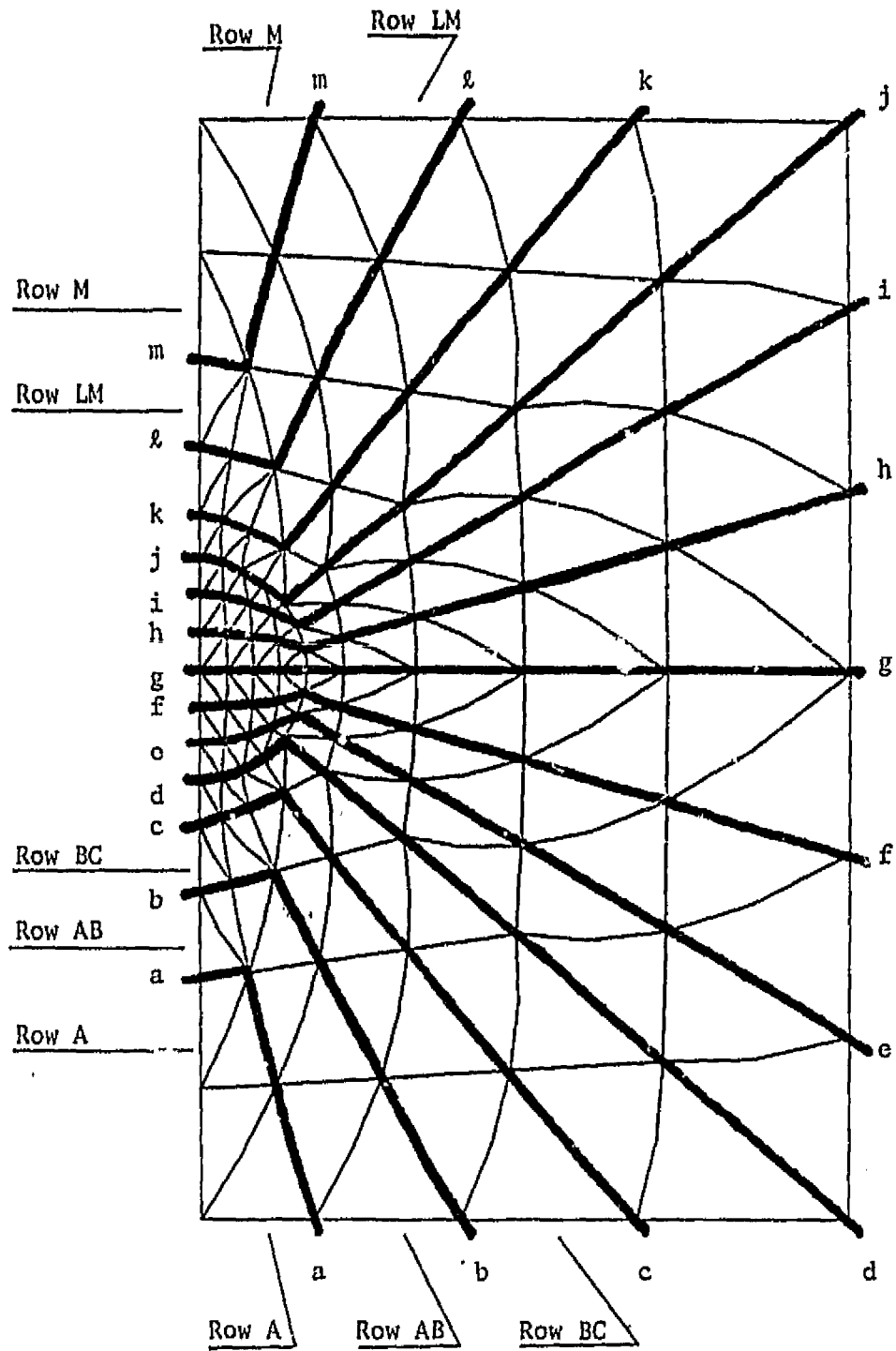


Figure 4

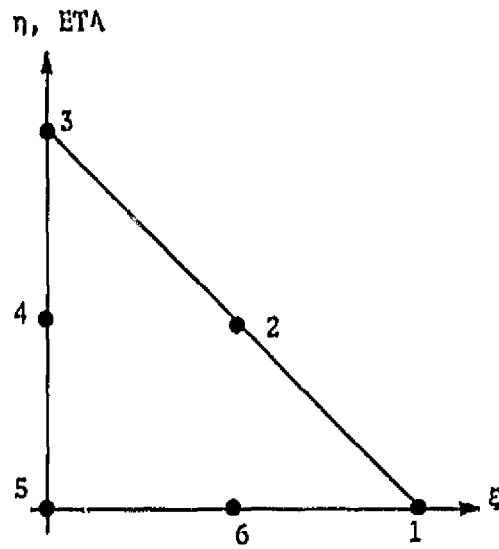


through the four elements of that row and then on to the eight elements in row B beginning at the left and so forth through the mesh. It is important to emphasize that this order does not influence the nodal point or element numbering generated by MESH2.

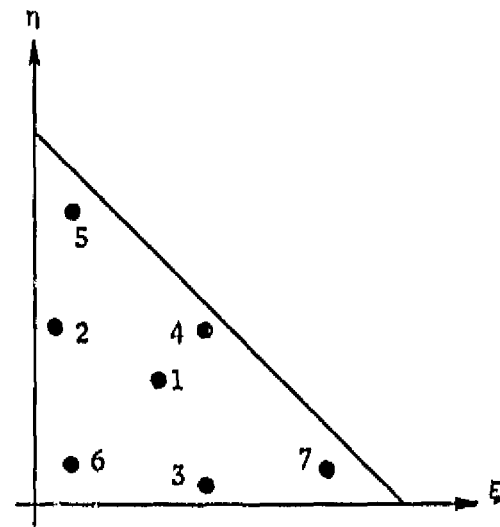
## APPENDIX B

QUADRATURE POINTS, NODAL POINTS, AND STRESS POINTS  
IN THE  $\xi$ - $\eta$  PLANE

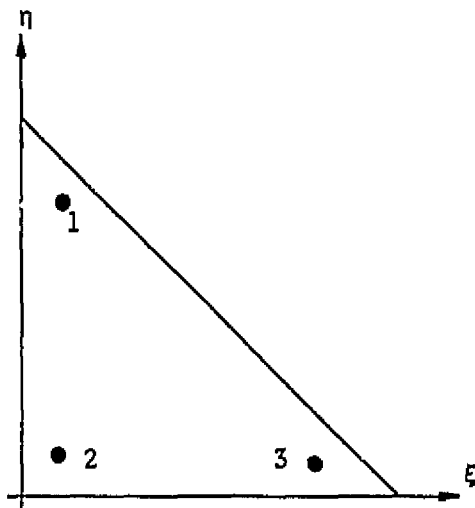
The following figure and table locate the quadrature points, nodal points, and stress points in the  $\xi$ - $\eta$  plane (here, the  $\xi$  and  $\eta$  coordinates also correspond to two of the three area coordinates commonly referred to with Gaussian quadrature). In order to locate a particular quadrature point or stress point on an element in the  $x$ - $y$  plane, it is only necessary to orient the nodal point numbers in the  $\xi$ - $\eta$  plane with those in the  $x$ - $y$  plane and assume the same relative orientation for the quadrature points and the stress points.



(a) Nodal Points



(b) Quadrature Points



(c) Stress Points

Quadrature Point	$\xi$	$\eta$	Stress Point
1	0.33333	0.33333	
2	0.05972	0.47014	
3	0.47014	0.05972	
4	0.47014	0.47014	
5	0.10129	0.79743	1
6	0.10129	0.10129	2
7	0.79743	0.10129	3

Figure 1. Nodal Points, Quadrature Points, and Stress Points

## APPENDIX C

## USEFUL REFERENCES

The following references pertain directly to the material presented in this report.

- Thompson, E. G. and M. I. Hague, "A High Order Finite Element for Completely Incompressible Creeping Flow," Int. Journal Num. Meth. Engr., Vol. 6, 315-321, 1973.
- Sato, A. and E. G. Thompson, "Finite Element Models for Creeping Convection," Journal Computational Physics.
- Thompson, E. G., "Average and Complete Incompressibility in the Finite Element Method," Int. Journal Num. Meth. Engr., Vol. 9, 925-932, 1975.
- Dawson, P. R., Finite Element Thermomechanical Models for Metal Forming, Ph.D. Dissertation, Colorado State University, 1976.
- Grandall, S. H., Engineering Analysis, McGraw-Hill, 1956.
- Dawson, P. R. and E. G. Thompson, "Steady State Thermomechanical Finite Element Analysis of Elastoviscoplastic Metal Forming Processes," Numerical Modeling of Manufacturing Processes, ASME, PVP-PB-02S, 1977.
- Dawson, P. R. and E. G. Thompson, "Finite Element Analysis of Steady-State Elasto-Visco-Plastic Flow by the Initial Stress-Rate Method," Int. Journal Num. Meth. Engr., Vol. 12, 1978.
- Dawson, P. R. and E. G. Thompson, "Addendum to the Paper Finite Element Analysis of Steady-State Elasto-Visco-Plastic Flow by the Initial Stress-Rate Method," Int. Journal Num. Meth. Engr., Vol. 12, pp. 382-383, 1978.

## APPENDIX D

## INDEX OF FORTRAN VARIABLES

<u>Variable</u>	<u>Page</u>	<u>Variable</u>	<u>Page</u>
BNDRY	69	IEMPT	54
BNDRYC	69	IFLOW	26
BNDRYT	69	INCPL	42
CH	32, 37, 52	INCPR	42
CHBC	37	INCPU	42
CHI	30	INCR	50
CHO	30	INTEMP	44
COSLBC	37	INTLCU	47
COSXXP	32, 37, 52	INTLTU	47
CP	68, 76	INTO	54
CP TIME	57	INTPL	43
CTEMP	46	INTPR	43
CX	32, 37, 52	INTPU	43
CXI	30	IPORDER	40
CXLBC	37	IPUNCH	29, 34
CXO	30	IPQ	60
CY	32, 37, 52	IREAD	40
CYI	30	IFZ	44
CYLBC	37	ITERC	57, 58
CYO	30	ITERT	57, 58
DELF	57, 58	ITMAXC	46
DELQ	58	ITMAXT	46
DELT	57, 58	ITV	44, 70
DELU	57, 58	JBGN	50
DFCONV	48	JEND	50
DTCONV	48	JOIN	35
DTIME	58	KMAX	56
DTMAX	45	KVOL	54
DQCONV	48	LAGEUL	44
DUCONV	48	LCU	47
DUMAX	45	LIST	54
EPSII	67	LISTX	54
FTX	59	LPBC	37
FTY	59	LTU	47
G	67, 76	MAT	42
GAMX	66, 76	MAXVOL	39
GAMY	66, 76	MJ	66, 67, 68
IB	54	MOP	44
IBMAX	56	MOVE	54
IBMIN	39	MOVEX	54
ICOMP	54	MSHADJ	69
IELE	54	MSHCD	39
IELEJ	68, 70	MNI	45
IELEX	54	NDIV	35
		NDIVR	29

<u>Variable</u>	<u>Page</u>	<u>Variable</u>	<u>Page</u>
NDIVTH	29	TX	32,37,52
NELMC	51	TXI	30
NELMT	51	TXO	30
NP	53	TXLBC	37
NPBC	32,37,52	TY	32,37,52
NPBCI	30	TLBC	37
NPBCO	30	UT	49,59
NPPE	27,33,67	UX	49,59
NPR	54	UXO	50
NPTS	49	UY	49,59
NQMAX	56	UYO	50
NSEC	49	VECTL	46
NSEG	54	VISC	67,76
NUMAT	42	VS	67,76
NUMBC	31,36,37	VT	67,76
NUMLPS	33	XBC	32,37,52
NUMNP	51	XBCI	30
NUMPP	51	XBCO	30
NUMSEG	56	XCOR	36
NUMTP	51	XK	66,67
NUMVP	51	XLBC	37
PENLTY	67	XMAX	34
PRS	60	XMIN	34
QRS	60	XORD	49,52,59
RHO	68	YBC	32,37,52
RI	29	YBCI	30
RKX	68,76	YBCO	30
RKY	68,76	YCOR	36
RM	29	YK	67,76
RPI	29	YLBC	37
RO	20	YMAX	34
SIGII	61	YMIN	34
SIGXX	61	YORD	49,52,59
SIGXY	61		
SIGYY	61		
STIFF	70,92		
TBC	32,37,52		
TBCI	30		
TBCD	30		
TEMPK	66,67		
TEMPO	50		
THETA	44		
TIMEM	45		
TLBC	37		
TRANS	44		
TQ	32,37,52		
TQI	30		
TQO	30		
TQLBC	37		