

GENERAL MOTORS CORPORATION

TEDDY—A TWO-DIMENSIONAL LAGRANGIAN  
CODE FOR DETERMINING THE WAVE  
PROPAGATION IN ELASTIC-PLASTIC MEDIA

A.H. Jones

C.E. Kriech

Prepared For

National Aeronautics and Space Administration

Manned Spacecraft Center, Houston, Texas

Under Contract No. NAS 9-3081

AC ELECTRONICS - DEFENSE RESEARCH LABORATORIES

SANTA BARBARA, CALIFORNIA

AEROSPACE OPERATIONS DEPARTMENT

PRECEDING PAGE BLANK NOT FILMED.

TR66-85

## CONTENTS

|   |     |
|---|-----|
| I. INTRODUCTION                                 | 1   |
| II. EQUATIONS OF MOTION                         | 3   |
| A. Conservation of Mass                         | 3   |
| B. Conservation of Momentum                     | 4   |
| C. Conservation of Energy                       | 4   |
| III. FINITE-DIFFERENCE EQUATIONS                | 7   |
| A. Conservation of Momentum                     | 7   |
| B. Continuity Equation                          | 11  |
| C. Energy Equation                              | 13  |
| D. Artificial Viscosity and Stability Condition | 17  |
| E. Boundary Conditions                          | 19  |
| F. Sliding Interface                            | 21  |
| G. Equation of State                            | 29  |
| IV. PROGRAM DESCRIPTION                         | 37  |
| A. Computer Configuration                       | 37  |
| B. Mesh Numbering Scheme                        | 37  |
| C. Program Structure                            | 39  |
| D. Program Features and Limitations             | 40  |
| E. Input Data Format                            | 42  |
| F. Program Output                               | 50  |
| G. Example Problems                             | 50  |
| H. Program Flow Charts                          | 56  |
| I. Fortran Source Decks                         | 56  |
| REFERENCES                                      | 59  |
| APPENDIXES                                      |     |
| A MATERIAL ROTATION                             | A-1 |
| B PROGRAM FLOW CHARTS                           | B-1 |
| C LISTINGS OF FORTRAN SOURCE DECKS              | C-1 |

TR66-85

## ILLUSTRATIONS

| <u>Figure</u> | <u>Title</u>   | <u>Page</u> |
|---------------|--|-------------|
| 1             | Mesh Numbering Scheme  | 38          |
| 2             | Type 1 Load Sheet  | 46          |
| 3             | Type 2 Load Sheet  | 47          |
| 4             | Type 3 Load Sheet  | 48          |
| 5             | Type 4 Load Sheet  | 49          |
| 6             | Example of Printed Input Data  | 51          |
| 7             | Example of Printed Output at Each Mesh   | 52          |
| 8             | Example of Position Plot   | 53          |
| 9             | Example of Stress Plot   | 54          |
| 10            | Example 1: Shock Tube Analysis   | 55          |
| 11            | Example 2: Stress Wave Profiles Due to the<br>Impact of an Aluminum Plate on a Rigid Wall  | 57          |
| 12            | Example 3: Wave Propagation in an Elastic Medium Surrounding<br>a Spherical Cavity Subjected to a Time-Varying Internal Pressure |             |
|               | a. Velocity and Stress vs Radius   |             |
|               | b. Velocity and Displacement vs Time   | 58          |

## I. INTRODUCTION

The problem set up here is the solution, by finite difference methods, of wave propagation in compressible media in two space dimensions, with either rectangular or cylindrical geometry. Lagrangian grid representation is used. For this system the motion of the medium is described with reference to a mesh attached to the material. This results in a limitation of the method, and serious difficulties arise when physical situations involve severe distortions of the original mesh.

The program can be used with either a fluid or elastic-perfectly plastic solid. Either the polynomial or the Tillotson<sup>(1)</sup> equation of state can be used for describing the hydrodynamic component of the stress. Other equations of state may be incorporated, as well as equations of state to describe the behavior of other media (for example, the explosives).

Several two-dimensional time-dependent Lagrangian codes have been developed, such as TENSOR,<sup>(2)</sup> HEMP,<sup>(3)</sup> PIPE,<sup>(4)</sup> and RAVE I.<sup>(5)</sup> The method outlined here follows that formulated earlier by Wilkins (HEMP)<sup>(3)</sup> and Herrmann (RAVE I).<sup>(5)</sup> In order to make the present report self-contained the equations of motion are given in the following section, Section II, and the finite difference derivations are given fully in Section III. (Additional equations for material rotation are given in Appendix A.) The program itself is described in Section IV, and example problems with printed and plotted outputs are also given there. Finally, flow charts for the program are given in Appendix B and complete listings of the Fortran source decks are presented in Appendix C.

## II. EQUATIONS OF MOTION

The governing equations of motion in their Lagrangian form are as follows.

### A. CONSERVATION OF MASS

$$\dot{\rho} + \rho (\dot{e}^{xx} + \dot{e}^{yy} + \dot{e}^{zz}) = 0$$

where  $\rho$  is the medium density,  $e^{ij}$  the strain, and the dot indicates a time derivative. With the present nomenclature  $x$ ,  $y$ ,  $z$  can either be rectangular coordinates or cylindrical coordinate systems. For the axial symmetric problem the cylindrical coordinate system is used, in which case  $x$  denotes the radial coordinate, and  $z$  the axial coordinate.

The strain rates are defined as

$$\dot{e}^{ij} = \frac{1}{2} \left( \frac{\partial u^i}{\partial x^j} + \frac{\partial u^j}{\partial x^i} \right)$$

where  $u^i$  indicates particle velocity in direction  $i$ , e. g.

$$u^x = \frac{\partial x}{\partial t}$$

and

$$\left[ \frac{\partial u^i}{\partial x^j} \right] = \begin{bmatrix} \frac{\partial u^x}{\partial x} & 0 & \frac{\partial u^x}{\partial z} \\ 0 & (\alpha-1) \frac{u^x}{x} & 0 \\ \frac{\partial u^z}{\partial x} & 0 & \frac{\partial u^z}{\partial x} \end{bmatrix}$$

Here  $\alpha$  is a coefficient which takes on a value unity for motion in the  $x$ - $z$  plane and 2 for axially symmetric motion.

TR66-85

**B. CONSERVATION OF MOMENTUM**

$$\rho a^x = \frac{\partial t^{xx}}{\partial x} + \frac{\partial t^{xz}}{\partial z} + (\alpha-1) \frac{t^{xx}-t^{yy}}{x}$$

$$\rho a^z = \frac{\partial t^{xz}}{\partial x} + \frac{\partial t^{zz}}{\partial z} + (\alpha-1) \frac{t^{xz}}{x}$$

where  $a^i$  is the acceleration in direction  $i$ , and  $t^{ij}$  are the stress components.

**C. CONSERVATION OF ENERGY**

The rate of change of internal energy/unit mass  $\dot{e}$ , is given by

$$\rho \dot{e} = p \frac{\dot{\rho}}{\rho} + \rho d_e \dot{e}$$

where

$$p = -\frac{1}{3} (t^{xx} + t^{yy} + t^{zz})$$

$$\rho d_e \dot{e} = 2 t^{rr} d_e^{rr} + 2 t^{rz} d_e^{rz} + 2 t^{zz} d_e^{zz} + t^{rr} d_e^{zz} + t^{zz} d_e^{rr}$$

$$d_t^{ij} = t^{ij} + p$$

$$d_e^{ij} = \dot{e}^{ij} + \frac{1}{3} \frac{\dot{\rho}}{\rho} \delta^{ij}$$

$$\delta^{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

so that

$$d_t^{xx} + d_t^{yy} + d_t^{zz} = 0$$

$$d_e^{xx} + d_e^{yy} + d_e^{zz} = 0$$

Since during a given time step an element of the body rotates it is necessary to correct the stress so as to refer to the fixed coordinate system (see Appendix A).

The objective stress rates are defined by

TR66-85

$$\begin{aligned} \dot{t}^{XX} &= \dot{t}^{XX} - 2w_{XZ} t^{XZ} \\ \dot{t}^{XZ} &= \dot{t}^{XZ} + w_{XZ} (t^{XX} - t^{ZZ}) \\ \dot{t}^{ZZ} &= \dot{t}^{ZZ} + 2w_{XZ} t^{XZ} \end{aligned}$$

where

$$w_{XZ} = \frac{1}{2} \left( \frac{\partial u^X}{\partial Z} - \frac{\partial u^Z}{\partial X} \right)$$

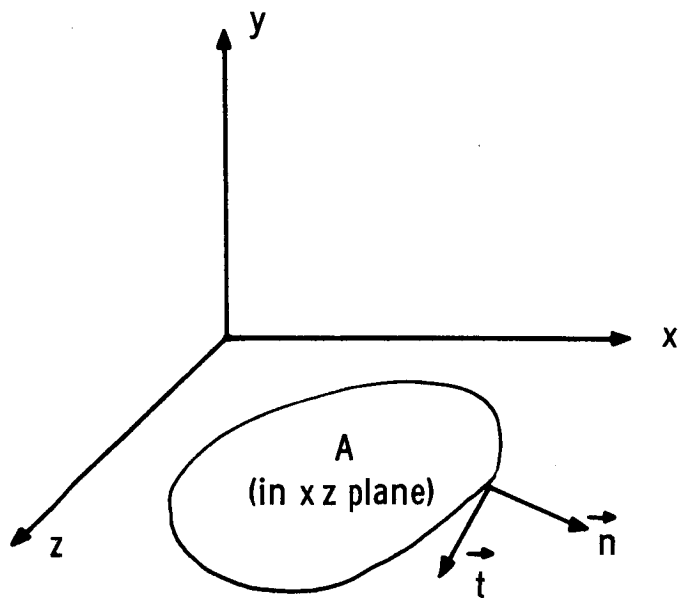
The above together with an equation of state are the required equations for solution of wave propagation problems. With given initial and boundary conditions specific problems can be solved within the limit of the Lagrangian code described.

### III. FINITE-DIFFERENCE EQUATIONS

#### A. CONSERVATION OF MOMENTUM

The partial derivatives are set into finite difference form by the following method.

Integral definition  
of the partial derivatives



$$\frac{\partial \psi}{\partial x} = \lim_{A \rightarrow 0} \frac{\oint_S F(\vec{n}, \vec{i}) d\ell}{A}$$

$$\frac{\partial \psi}{\partial z} = \lim_{A \rightarrow 0} \frac{\oint_S F(\vec{n}, \vec{k}) d\ell}{A}$$

where  $s$  is the boundary of area  $A$ ,  $\ell$  an arc length,  $\vec{n}$  the normal vector and  $\vec{t}$  the tangent vector to the arc all lying in  $xz$  plane. These are related to the unit vectors  $\vec{i}$ ,  $\vec{j}$ ,  $\vec{k}$  along the axes  $x$ ,  $y$ ,  $z$  respectively by the relations

$$\vec{t} = \vec{i} \frac{\partial x}{\partial s} + \vec{k} \frac{\partial z}{\partial s}$$



TR66-85

$$\begin{aligned}
 \vec{n} &= \vec{i} \frac{\partial x}{\partial n} + \vec{k} \frac{\partial z}{\partial n} \\
 &= \vec{j} \times \vec{t} \\
 &= \vec{j} \times \left( \vec{i} \frac{\partial x}{\partial s} + \vec{k} \frac{\partial z}{\partial s} \right) \\
 &= \vec{i} \frac{\partial z}{\partial s} - \vec{k} \frac{\partial x}{\partial s}
 \end{aligned}$$

On substitution we find that

$$\frac{\partial \psi}{\partial x} = \lim_{A \rightarrow 0} \frac{\oint_s \psi \frac{\partial z}{\partial s} ds}{A}$$

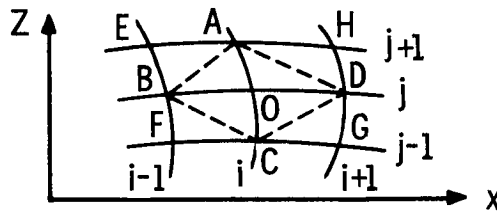
and

$$\frac{\partial \psi}{\partial z} = \lim_{A \rightarrow 0} \frac{-\oint_s \psi \frac{\partial x}{\partial s} ds}{A}$$

Returning to the equations of motion

$$\frac{1}{\rho} \frac{\partial t^{xx}}{\partial x} = \lim_{A \rightarrow 0} \frac{\oint t^{xx} \frac{\partial z}{\partial s} ds}{\rho A}$$

etc.



Applying these to quadrilateral ABCD,

$$\left( \frac{1}{\rho} \frac{\partial t^{xx}}{\partial x} \right)_0 = \frac{-1}{(\rho A)_0} \left\{ (t^{xx})_1 (z_D - z_A) + (t^{zz})_2 (z_A - z_B) + (t^{xx})_3 (z_B - z_C) + (t^{xx})_4 (z_C - z_D) \right\}$$

$$\left( \frac{1}{\rho} \frac{\partial t^{xz}}{\partial z} \right)_0 = \frac{1}{(\rho A)_0} \left\{ (t^{xz})_1 (x_D - x_A) + (t^{xz})_2 (x_A - x_B) + (t^{xz})_3 (x_B - x_C) + (t^{xz})_4 (x_C - x_D) \right\}$$

The term  $(\rho A)$  appearing in these equations is required at 0, where  $\rho$  and  $A$  are quantities averaged over the meshes 1, 2, 3 and 4. This term is well represented by the expression

$$(\rho A)_0 = A_{I1}\rho_1 + A_{I2}\rho_2 + A_{I3}\rho_3 + A_{I4}\rho_4$$

where  $A_{I1}$  is the area of triangle AOD,  $A_{I2}$  of AOB,  $A_{I3}$  of BOC, and  $A_{I4}$  of COD.

The last term on the right of the momentum equation is written

$$\begin{aligned} \left( \frac{t^{xx} - t^{yy}}{\rho x} \right)_0 &= \frac{1}{4} \left\{ \frac{[(t^{xx})_1 - (t^{yy})_1]}{\rho_1 \bar{x}_{I1}} + \frac{[(t^{xx})_2 - (t^{yy})_2]}{\rho_2 \bar{x}_{I2}} \right. \\ &\quad \left. + \frac{[(t^{xx})_3 - (t^{yy})_3]}{\rho_3 \bar{x}_{I3}} + \frac{[(t^{xx})_4 - (t^{yy})_4]}{\rho_4 \bar{x}_{I4}} \right\} \end{aligned}$$

where

$$\bar{x}_{I1} = \frac{1}{3} (x_O + x_D + x_A)$$

$$\bar{x}_{I2} = \frac{1}{3} (x_O + x_A + x_B)$$

$$\bar{x}_{I3} = \frac{1}{3} (x_O + x_B + x_C)$$

$$\bar{x}_{I4} = \frac{1}{3} (x_O + x_C + x_D)$$

Similarly for the second momentum equation

$$\begin{aligned} \left( \frac{1}{\rho} \frac{\partial t^{xz}}{\partial x} \right)_0 &= \frac{-1}{(\rho A)_0} \left\{ (t^{xz})_1 (z_D - z_A) + (t^{xz})_2 (z_A - z_B) \right. \\ &\quad \left. + (t^{xz})_3 (z_B - z_C) + (t^{xz})_4 (z_C - z_D) \right\} \end{aligned}$$

$$\begin{aligned} \left( \frac{1}{\rho} \frac{\partial t^{zz}}{\partial z} \right)_0 &= \frac{1}{(\rho A)_0} \left\{ (t^{zz})_1 (x_D - x_A) + (t^{zz})_2 (x_A - x_B) \right. \\ &\quad \left. + (t^{zz})_3 (x_B - x_C) + (t^{zz})_4 (x_C - x_D) \right\} \end{aligned}$$

TR66-85

and

$$\left( \frac{t^{zz}}{\rho x} \right)_0 = \frac{1}{4} \left\{ \frac{(t^{xz})_1}{\rho_1 \bar{x}_{11}} + \frac{(t^{xz})_2}{\rho_2 \bar{x}_{12}} + \frac{(t^{xz})_3}{\rho_3 \bar{x}_{13}} + \frac{(t^{xz})_4}{\rho_4 \bar{x}_{14}} \right\}$$

Substituting these finite difference quantities in the momentum equation enables us to evaluate the accelerations  $a^x$  and  $a^z$ :

$$\begin{aligned} (a^x)_0^n &= - \frac{(t^{xx})_1^n (z_D^n - z_A^n) + (t^{xx})_2^n (z_A^n - z_B^n) + (t^{xx})_3^n (z_B^n - z_C^n) + (t^{xx})_4^n (z_C^n - z_D^n)}{A_{11}^n \rho_1^n + A_{12}^n \rho_2^n + A_{13}^n \rho_3^n + A_{14}^n \rho_4^n} \\ &+ \frac{(t^{xz})_1^n (x_D^n - x_A^n) + (t^{xz})_2^n (x_A^n - x_B^n) + (t^{xz})_3^n (x_B^n - x_C^n) + (t^{xz})_4^n (x_C^n - x_D^n)}{A_{11}^n \rho_1^n + A_{12}^n \rho_2^n + A_{13}^n \rho_3^n + A_{14}^n \rho_4^n} \\ &+ \frac{(\alpha-1)}{4} \left\{ \frac{[(t^{xx})_1^n - (t^{yy})_1^n]}{\rho_1^n \bar{x}_{11}^n} + \frac{[(t^{xx})_2^n - (t^{yy})_2^n]}{\rho_2^n \bar{x}_{12}^n} + \frac{[(t^{xx})_3^n - (t^{yy})_3^n]}{\rho_3^n \bar{x}_{13}^n} \right. \\ &\left. + \frac{[(t^{xx})_4^n - (t^{yy})_4^n]}{\rho_4^n \bar{x}_{14}^n} \right\} \\ (a^z)_0^n &= - \frac{(t^{xz})_1^n (z_D^n - z_A^n) + (t^{xz})_2^n (z_A^n - z_B^n) + (t^{xz})_3^n (z_B^n - z_C^n) + (t^{xz})_4^n (z_C^n - z_D^n)}{A_{11}^n \rho_1^n + A_{12}^n \rho_2^n + A_{13}^n \rho_3^n + A_{14}^n \rho_4^n} \\ &+ \frac{(t^{zz})_1^n (x_D^n - x_A^n) + (t^{zz})_2^n (x_A^n - x_B^n) + (t^{zz})_3^n (x_B^n - x_C^n) + (t^{zz})_4^n (x_C^n - x_D^n)}{A_{11}^n \rho_1^n + A_{12}^n \rho_2^n + A_{13}^n \rho_3^n + A_{14}^n \rho_4^n} \\ &+ \frac{(\alpha-1)}{4} \left\{ \frac{(t^{xz})_1^n}{\rho_1^n \bar{x}_{11}^n} + \frac{(t^{xz})_2^n}{\rho_2^n \bar{x}_{12}^n} + \frac{(t^{xz})_3^n}{\rho_3^n \bar{x}_{13}^n} + \frac{(t^{xz})_4^n}{\rho_4^n \bar{x}_{14}^n} \right\} \end{aligned}$$

The areas in these equations are given by

$$A_{I1}^n = \frac{1}{2} \left\{ (x_D^n - x_O^n)(z_A^n - z_O^n) - (z_D^n - z_O^n)(x_A^n - x_O^n) \right\}$$

$$A_{I2}^n = \frac{1}{2} \left\{ (x_A^n - x_O^n)(z_B^n - z_O^n) - (z_A^n - z_O^n)(x_B^n - x_O^n) \right\}$$

$$A_{I3}^n = \frac{1}{2} \left\{ (x_B^n - x_O^n)(z_C^n - z_O^n) - (z_B^n - z_O^n)(x_C^n - x_O^n) \right\}$$

$$A_{I4}^n = \frac{1}{2} \left\{ (x_C^n - x_O^n)(z_D^n - z_O^n) - (z_C^n - z_O^n)(x_D^n - x_O^n) \right\}$$

Knowing the accelerations, the velocities and the coordinates of the vertices at time (n+1) may be evaluated

$$(u^x)_O^{n+1/2} = (u^x)_O^{n-1/2} + 1/2 (\Delta t^{n+1/2} + \Delta t^{n-1/2})(a^x)_O^n$$

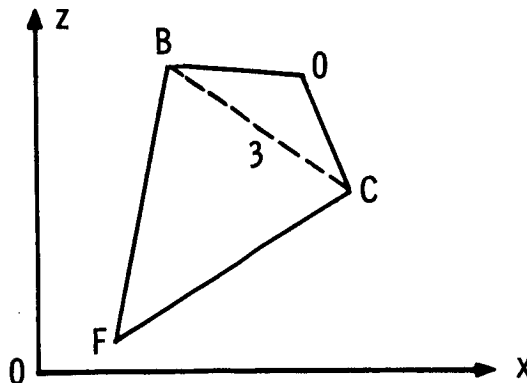
$$(u^z)_O^{n+1/2} = (u^z)_O^{n-1/2} + 1/2 (\Delta t^{n+1/2} + \Delta t^{n-1/2})(a^z)_O^n$$

and

$$x_O^{n+1} = x_O^n + \Delta t^{n+1/2} (u^x)_O^{n+1/2}$$

$$z_O^{n+1} = z_O^n + \Delta t^{n+1/2} (u^z)_O^{n+1/2}$$

### B. CONTINUITY EQUATION



TR66-85

Knowing the positions of the vertices for the quadrilaterals at time (n+1) the densities may be calculated, e. g.,

$$\rho_3^{n+1} = \frac{M_3}{A_{I3}^{n+1} (\bar{x}_{I3}^{n+1})^{\alpha-1} + A_{\Phi 3}^{n+1} (\bar{x}_{\Phi 3}^{n+1})^{\alpha-1}}$$

where

$$A_{I3}^{n+1} = \frac{1}{2} \left\{ (x_B^{n+1} - x_O^{n+1}) (z_C^{n+1} - z_O^{n+1}) - (z_B^{n+1} - z_O^{n+1}) (x_C^{n+1} - x_O^{n+1}) \right\}$$

$$A_{\Phi 3}^{n+1} = \frac{1}{2} \left\{ (x_C^{n+1} - x_F^{n+1}) (z_B^{n+1} - z_F^{n+1}) - (z_C^{n+1} - z_F^{n+1}) (x_B^{n+1} - x_F^{n+1}) \right\}$$

$$\bar{x}_{I3}^{n+1} = \frac{1}{3} (x_B^{n+1} + x_C^{n+1} + x_O^{n+1})$$

$$\bar{x}_{\Phi 3}^{n+1} = \frac{1}{3} (x_C^{n+1} + x_B^{n+1} + x_F^{n+1})$$

$M_3$  is given by

$$M_3 = \rho_3^0 \left\{ A_{I3}^0 (\bar{x}_{I3}^0)^{\alpha-1} + A_{\Phi 3}^0 (\bar{x}_{\Phi 3}^0)^{\alpha-1} \right\}$$

since mass is conserved within each zone.

Using the evaluated density

$$\left( \frac{\dot{\rho}}{\rho} \right)_3^{n+1/2} = \frac{2}{\Delta t^{n+1/2}} \frac{\rho_3^{n+1} - \rho_3^n}{\rho_3^{n+1} + \rho_3^n}$$

$$\left( \frac{\Delta \rho}{\rho} \right)_3^{n+1/2} = 4 \frac{\rho_3^{n+1} - \rho_3^n}{(\rho_3^{n+1} + \rho_3^n)^2}$$

and

$$\eta_3^{n+1} = \frac{\rho_3^{n+1}}{\rho_3^0}$$

## C. ENERGY EQUATION

This energy equation in finite difference form becomes

$$e_3^{n+1} - e_3^n = \frac{1}{2} \left( p_3^{n+1} + p_3^n \right) \left( \frac{\Delta \rho}{\rho} \right)_3 + \Delta_q e_3 + \Delta^d e_3$$

$$\Delta_q e_3 = \frac{1}{2} \left( q_3^{n+1/2} + q_3^{n-1/2} \right) \left( \frac{\Delta \rho}{\rho} \right)_3$$

$$\Delta^d e_3 = \Delta_e^d e_3 + \Delta_q^d e_3$$

$$\begin{aligned} \Delta_e^d e_3 = & \frac{\Delta t^{n+1/2}}{\rho_3^o (\eta_3^{n+1} + \eta_3^n)} \left\{ \left[ \left( d_{t,xx} \right)_3^{n+1} + \left( d_{t,xx} \right)_3^n \right] \left[ 2 \left( d_{e,xx} \right)_3^{n+1/2} \right. \right. \\ & + \left. \left. \left( d_{e,zz} \right)_3^{n+1/2} \right] + \left[ \left( d_{t,zz} \right)_3^{n+1} + \left( d_{t,zz} \right)_3^n \right] \left[ 2 \left( d_{e,zz} \right)_3^{n+1/2} \right. \right. \\ & \left. \left. + \left( d_{e,xx} \right)_3^{n+1/2} \right] + \left[ \left( d_{t,xz} \right)_3^{n+1} + \left( d_{t,xz} \right)_3^n \right] \left[ 2 \left( d_{e,xz} \right)_3^{n+1/2} \right] \right\} \end{aligned}$$

and

$$\begin{aligned} \Delta_q^d e_3 = & \frac{\Delta t^{n+1/2}}{\rho_3^o (\eta_3^{n+1} + \eta_3^n)} \left\{ \left[ \left( d_{q,xx} \right)_3^{n+1/2} + \left( d_{q,xx} \right)_3^{n-1/2} \right] \left[ 2 \left( d_{e,xx} \right)_3^{n+1/2} \right. \right. \\ & + \left. \left. \left( d_{e,zz} \right)_3^{n+1/2} \right] + \left[ \left( d_{q,zz} \right)_3^{n+1/2} + \left( d_{q,zz} \right)_3^{n-1/2} \right] \left[ 2 \left( d_{e,zz} \right)_3^{n+1/2} \right. \right. \\ & \left. \left. + \left( d_{e,xx} \right)_3^{n+1/2} \right] + \left[ \left( d_{q,xz} \right)_3^{n+1/2} + \left( d_{q,xz} \right)_3^{n-1/2} \right] \left[ 2 \left( d_{e,xz} \right)_3^{n+1/2} \right] \right\} \end{aligned}$$

TR66-85

where

$$(t^{xx})^{n+1} = -(p^{n+1} + q^{n+1/2}) + (d t^{xx})^{n+1} + (d_q^{xx})^{n+1/2}$$

$$(t^{yy})^{n+1} = -3(p^{n+1} + q^{n+1/2}) - (t^{xx})^{n+1} - (t^{zz})^{n+1}$$

$$(t^{zz})^{n+1} = -(p^{n+1} + q^{n+1/2}) + (d t^{zz})^{n+1} + (d_q^{zz})^{n+1/2}$$

$$(t^{xz})^{n+1} = (d t^{xz})^{n+1} + (d_q^{xz})^{n+1/2}$$

q being an artificial viscosity to be introduced later.

Using the definition derived earlier for partial derivative, namely

$$\frac{\partial u^x}{\partial x} = \lim_{A \rightarrow 0} \frac{\oint_s u^x \frac{\partial z}{\partial s} ds}{A}$$

the deviatoric strain rates can be evaluated. The required partial derivatives for quadrilateral OBFC are

$$\begin{aligned} \left(\frac{\partial u^x}{\partial x}\right)_3^{n+1/2} = & \frac{-2}{A_3^{n+1} + A_3^n} \left\{ \frac{\left[ \left(u^x\right)_C^{n+1/2} + \left(u^x\right)_O^{n+1/2} \right]}{2} \left[ \frac{z_C^{n+1} + z_C^n - z_O^{n+1} - z_O^n}{2} \right] \right. \\ & + \frac{\left[ \left(u^x\right)_O^{n+1/2} + \left(u^x\right)_B^{n+1/2} \right]}{2} \left[ \frac{z_O^{n+1} + z_O^n - z_B^{n+1} - z_B^n}{2} \right] \\ & + \frac{\left[ \left(u^x\right)_B^{n+1/2} + \left(u^x\right)_F^{n+1/2} \right]}{2} \left[ \frac{z_B^{n+1} + z_B^n - z_F^{n+1} - z_F^n}{2} \right] \\ & \left. + \frac{\left[ \left(u^x\right)_F^{n+1/2} + \left(u^x\right)_C^{n+1/2} \right]}{2} \left[ \frac{z_F^{n+1} + z_F^n - z_C^{n+1} - z_C^n}{2} \right] \right\} \end{aligned}$$

$$= \frac{-1}{2(A_3^{n+1} + A_3^n)} \left\{ \left[ \left( u^x \right)_O^{n+1/2} - \left( u^x \right)_F^{n+1/2} \right] \left[ z_C^{n+1} + z_C^n - z_B^{n+1} - z_B^n \right] \right. \\ \left. - \left[ \left( u^x \right)_C^{n+1/2} - \left( u^x \right)_B^{n+1/2} \right] \left[ z_O^{n+1} + z_O^n - z_F^{n+1} - z_F^n \right] \right\}$$

$$\left( \frac{\partial u^z}{\partial z} \right)_3^{n+1/2} = \frac{1}{2(A_3^{n+1} + A_3^n)} \left[ \left( u^z \right)_O^{n+1/2} - \left( u^z \right)_F^{n+1/2} \right] \left[ x_C^{n+1} + x_C^n - x_B^{n+1} - x_B^n \right]$$

$$- \left[ \left( u^z \right)_C^{n+1/2} - \left( u^z \right)_B^{n+1/2} \right] \left[ x_O^{n+1} + x_O^n - x_F^{n+1} - x_F^n \right]$$

$$\left( \frac{\partial u^z}{\partial x} \right)_3^{n+1/2} = \frac{-1}{2(A_3^{n+1} + A_3^n)} \left\{ \left[ \left( u^z \right)_O^{n+1/2} - \left( u^z \right)_F^{n+1/2} \right] \left[ z_C^{n+1} + z_C^n - z_B^{n+1} - z_B^n \right] \right. \\ \left. - \left[ \left( u^z \right)_C^{n+1/2} - \left( u^z \right)_B^{n+1/2} \right] \left[ z_O^{n+1} + z_O^n - z_F^{n+1} - z_F^n \right] \right\}$$

and

$$\left( \frac{\partial u^x}{\partial z} \right)_3^{n+1/2} = \frac{1}{2(A_3^{n+1} + A_3^n)} \left\{ \left[ \left( u^x \right)_O^{n+1/2} - \left( u^x \right)_F^{n+1/2} \right] \left[ x_C^{n+1} + x_C^n - x_B^{n+1} - x_B^n \right] \right. \\ \left. - \left[ \left( u^x \right)_C^{n+1/2} - \left( u^x \right)_B^{n+1/2} \right] \left[ x_O^{n+1} + x_O^n - x_F^{n+1} - x_F^n \right] \right\}$$

Substituting in the expression for the deviatoric strain rate and spin

$$\left( d_e^{xx} \right)_3^{n+1/2} = \frac{-1}{2(A_3^{n+1} + A_3^n)} \left\{ \left[ \left( u^x \right)_O^{n+1/2} - \left( u^x \right)_F^{n+1/2} \right] \left[ z_C^{n+1} + z_C^n - z_B^{n+1} - z_B^n \right] \right. \\ \left. - \left[ \left( u^x \right)_C^{n+1/2} - \left( u^x \right)_B^{n+1/2} \right] \left[ z_O^{n+1} + z_O^n - z_F^{n+1} - z_F^n \right] \right\} + \frac{1}{3} \left( \frac{\dot{\rho}}{\rho} \right)_3^{n+1/2}$$



TR66-85

$$(d_e \cdot yy)_3^{n+1/2} = - \left\{ (d_e \cdot xx)_3^{n+1/2} + (d_e \cdot zz)_3^{n+1/2} \right\}$$

$$(d_e \cdot zz)_3^{n+1/2} = \frac{1}{2(A_3^{n+1} + A_3^n)} \left\{ (u^z)_O^{n+1/2} - (u^z)_F^{n+1/2} \right\} \left[ x_C^{n+1} + x_C^n - x_B^{n+1} - x_B^n \right] \\ - \left[ (u^z)_C^{n+1/2} - (u^z)_B^{n+1/2} \right] \left[ x_O^{n+1} + x_O^n - x_F^{n+1} - x_F^n \right] + \frac{1}{3} \left( \frac{\dot{\rho}}{\rho} \right)_3^{n+1/2}$$

$$(d_e \cdot xz)_3^{n+1/2} = \frac{1}{4(A_3^{n+1} + A_3^n)} \left\{ (u^x)_O^{n+1/2} - (u^x)_F^{n+1/2} \right\} \left[ x_C^{n+1} + x_C^n - x_B^{n+1} - x_B^n \right] \\ - \left[ (u^x)_C^{n+1/2} - (u^x)_B^{n+1/2} \right] \left[ x_O^{n+1} + x_O^n - x_F^{n+1} - x_F^n \right] \\ - \left[ (u^z)_O^{n+1/2} - (u^z)_F^{n+1/2} \right] \left[ z_C^{n+1} + z_C^n - z_B^{n+1} - z_B^n \right] \\ + \left[ (u^z)_C^{n+1/2} - (u^z)_B^{n+1/2} \right] \left[ z_O^{n+1} + z_O^n - z_F^{n+1} - z_F^n \right]$$

and

$$(w^{xz})_3^{n+1/2} = \frac{1}{4(A_3^{n+1} + A_3^n)} \left\{ (u^x)_O^{n+1/2} - (u^x)_F^{n+1/2} \right\} \left[ x_C^{n+1} + x_C^n - x_B^{n+1} - x_B^n \right] \\ - \left[ (u^x)_C^{n+1/2} - (u^x)_B^{n+1/2} \right] \left[ x_O^{n+1} + x_O^n - x_F^{n+1} - x_F^n \right] \\ + \left[ (u^z)_O^{n+1/2} - (u^z)_F^{n+1/2} \right] \left[ z_C^{n+1} + z_C^n - z_B^{n+1} - z_B^n \right] \\ - \left[ (u^z)_C^{n+1/2} - (u^z)_B^{n+1/2} \right] \left[ z_O^{n+1} + z_O^n - z_F^{n+1} - z_F^n \right]$$

## D. ARTIFICIAL VISCOSITY AND STABILITY CONDITION

Numerical calculations which include the effect of a moving boundary, such as a shock wave, become complicated because shocks move relative to the grid system. Furthermore, the motion of the discontinuity is not known in advance but is governed by the differential equations and the Rankine-Hugoniot equations. As a result, the treatment of shocks requires lengthy calculations at each step in time.

A scheme which overcomes this deficiency calls for the inclusion of an artificial viscosity in the calculations.<sup>(6)</sup> This smears out the shock so that the discontinuity varies rapidly but continuously (typically about three grids). Shocks are readily recognized in the numerical results as near discontinuities which move through the media at nearly the correct speed and across which the pressure, density, etc. have nearly the correct values, provided the thickness of the shock layer is small in comparison with other physically relevant dimensions of the system. The inclusion of the artificial viscosity term does require a shorter time step for stability; however, this is not serious provided the artificial viscosity is only sufficient to produce a shock thickness comparable with the grid size.

The artificial viscosity used in the present calculations are composed of two terms. Following von Neumann and Richtmyer<sup>(6)</sup> the first term is a bulk viscosity coefficient dependent on the dilatation so that stresses are negligible in areas of moderate gradient. The second term is to provide damping in the sonic range, giving a composite term

$$q_3^{n+1/2} = \frac{1}{2} \rho_3^0 \left( \eta_3^{n+1} + \eta_3^n \right) \left\{ C_1^2 \left( \frac{A_3^{n+1} + A_3^n}{2} \right) \left[ \left( \frac{\dot{p}}{\rho} \right)_3^{n+1/2} \right]^2 \right. \\ \left. + C_2 \left( \frac{A_3^{n+1} + A_3^n}{2} \right)^{1/2} C_3^n \left( \frac{\dot{p}}{\rho} \right)_3^{n+1/2} \right\}$$

if

$$\left( \frac{\dot{p}}{\rho} \right)_3^{n+1/2} > 0$$

TR66-85

otherwise

$$q_3^{n+1/2} = 0$$

 where  $C_1 \simeq 2.0$  and  $C_2 \simeq 0.1$ .

Viscous stress deviator, analogous to the viscous pressure, may also be introduced. The following are listed within the program.

$$\left( d_q^{xx} \right)_3^{n+1/2} = \frac{1}{2} C_3 \rho_3^o \left( \eta_3^{n+1} + \eta_3^n \right) C_3^n \left( \frac{A_3^{n+1} + A_3^n}{2} \right)^{1/2} \left( d_e^{xx} \right)_3^{n+1/2}$$

$$\left( d_q^{yy} \right)_3^{n+1/2} = - \left[ \left( d_q^{xx} \right)_3^{n+1/2} + \left( d_q^{zz} \right)_3^{n+1/2} \right]$$

$$\left( d_q^{zz} \right)_3^{n+1/2} = \frac{1}{2} C_3 \rho_3^o \left( \eta_3^{n+1} + \eta_3^n \right) C_3^n \left( \frac{A_3^{n+1} + A_3^n}{2} \right)^{1/2} \left( d_e^{zz} \right)_3^{n+1/2}$$

$$\left( d_q^{xz} \right)_3^{n+1/2} = \frac{1}{2} C_3 \rho_3^o \left( \eta_3^{n+1} + \eta_3^n \right) C_3^n \left( \frac{A_3^{n+1} + A_3^n}{2} \right)^{1/2} \left( d_e^{xz} \right)_3^{n+1/2}$$

In the above equations  $C_1$ ,  $C_2$ ,  $C_3$  are constants and the velocity of sound to be used,  $C_3^n$ , is approximated by

$$C_3^n = a_o + b_o \left( p_3^n \right)^\epsilon$$

$a_o$ ,  $b_o$ ,  $\epsilon$  being constants.

The stability criterion used for the two-dimensional case is

$$\Delta t^{n+3/2} = \frac{K\delta}{\sqrt{\left( C_3^n \right)^2 + 4C_1^2 \delta^2 \left| \frac{\rho}{\rho} \right|^2}} \quad \left| \text{minimum over } j \right.$$

for  $|\dot{\rho}/\rho| > 0$ , otherwise

$$\Delta t^{n+3/2} = \frac{K\delta}{C_3^n}$$

where  $\delta$  is the area of zone 3 divided by the longest diagonal,  $K$  an artificial stability criterion.

If the calculated time step

$$\Delta t^{n+3/2} > (1.1)\Delta t^{n+1/2}$$

then

$$\Delta t^{n+3/2} = (1.1)\Delta t^{n+1/2}$$

is used as the time step.

#### E. BOUNDARY CONDITIONS

At the boundaries the equations of motion must allow for either a free or fixed surface. Along fixed surfaces the material may slide, but it may not move normal to the wall. To calculate the accelerations along the wall, phantom zones are considered to exist on the opposite side of the fixed surface, for which the stresses  $t^{xx}$ ,  $t^{yy}$ ,  $t^{zz}$ , density  $\rho$ , radius  $\bar{x}$ , are equal, while the shear stress reflects antisymmetrically.

Consider the motion of point O along the fixed boundary on the z-axis. The coordinate position of point B is given by

$$x_B = 2x_O - x_D, \quad z_B = z_D$$

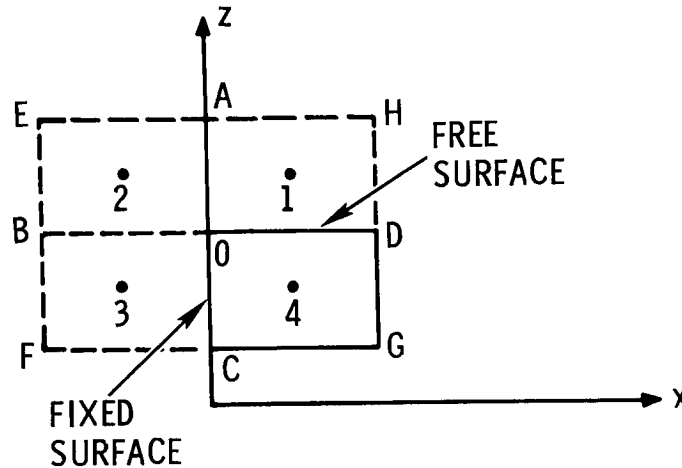
Similarly for a fixed boundary along the x-axis coordinates of C are

$$x_C = x_A, \quad z_C = 2z_O - z_A$$

Phantom zones on the opposing side of a free surface are given zero stresses and density.

TR66-85

Exactly the same scheme can be used for corner zones, with fixed-fixed, free-fixed, or free-free boundary without any ambiguity. As an example, consider the corner zone on the Z-axis.



$$(a^x)_0^n = 0$$

$$(u^x)_{0}^{n+1/2} = 0$$

$$(t^{zz})_1^n = (t^{zz})_2^n = 0$$

$$(t^{zz})_3^n = (t^{zz})_4^n$$

$$(t^{xz})_1^n = - (t^{xz})_2^n = 0$$

$$(t^{xz})_4^n = - (t^{xz})_3^n$$

$$\rho_3 = \rho_4$$

$$z_B = z_D$$

$$x_B = 2x_0 - x_D$$

$$A_{I1}^n = A_{I2}^n = 0$$

$$A_{I3}^n = A_{I4}^n$$

$$\rho_{1\bar{x}I1}^{n-} = \rho_{2\bar{x}I2}^{n-} = 0$$

$$\rho_{3\bar{x}I3}^{n-} = \rho_{4\bar{x}I4}^{n-}$$

For these conditions

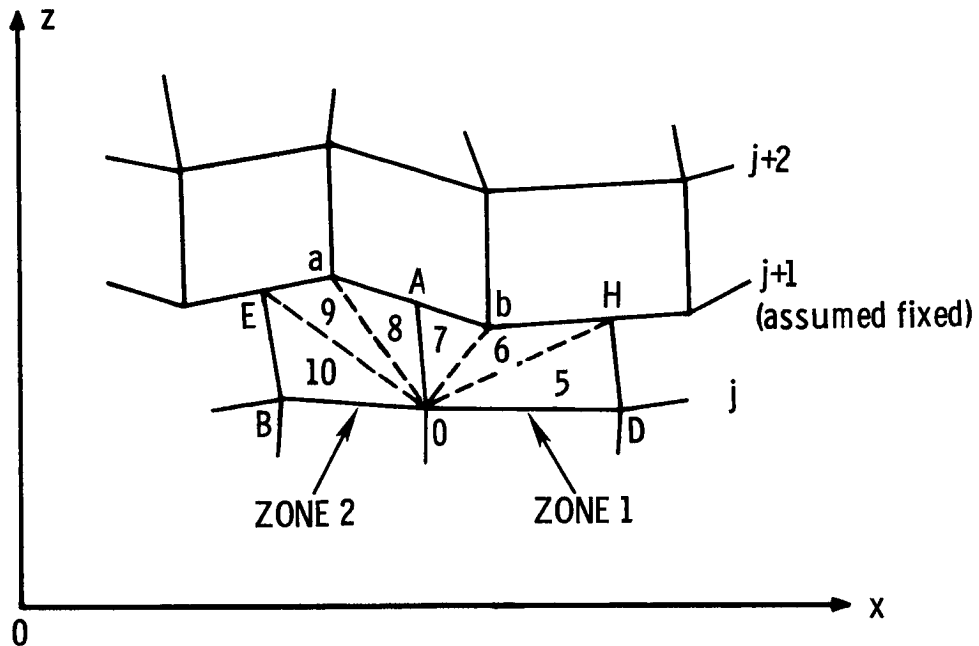
$$(a^z)_0^n = - \frac{(t^{xz})_4^n (z_C^n - z_D^n)}{A_{I4}^n \rho_4^n} + \frac{(t^{zz})_4^n (x_C^n - x_D^n)}{A_{I4}^n \rho_4^n}$$

Interfaces between materials require no special treatment, unless the materials are allowed to slide relative to each other. This case is treated below.

F. SLIDING INTERFACE

The problem of interest in the present study involves the pulse propagating in a plate on impact by a gaseous debris. It is important in this case to allow the two materials to slide relative to one another, otherwise the gaseous interface zones become excessively distorted in a very short time. We follow the approach initiated by Wilkins,<sup>(3, 7)</sup> which considers a simple decoupling of the grid points at the interface. The upper material interface is considered as a fixed boundary for a given time step  $\Delta t$ . During this time the equations of motion for the sliding material are the same as those given for motion along a fixed boundary. The fixed boundary is then advanced in time using the force field of the sliding boundary next to it. The new position of the fixed boundary provides a new boundary for the sliding material.

(1) Point A on the slide line (j+1)



TR66-85

In the diagram above, the acceleration of A along the fixed boundary ab is

$$(a^{ab})_A^n = \left[ \frac{(p_1^n + q_1^{n-1/2})(z_O^n - z_H^n) + (p_2^n + q_2^{n-1/2})(z_E^n - z_O^n)}{\rho_1^n (A_6^n + A_7^n) + \rho_2^n (A_8^n + A_9^n)} \right] \cos \theta_{ab}^n$$

$$- \left[ \frac{(p_1^n + q_1^{n-1/2})(x_O^n - x_H^n) + (p_2^n + q_2^{n-1/2})(x_E^n - x_O^n)}{\rho_1^n (A_6^n + A_7^n) + \rho_2^n (A_8^n + A_9^n)} \right] \sin \theta_{ab}^n$$

where

$$\sin \theta_{ab}^n = \frac{z_b^n - z_a^n}{\sqrt{(x_b^n - x_a^n)^2 + (z_b^n - z_a^n)^2}}$$

$$\cos \theta_{ab}^n = \frac{x_b^n - x_a^n}{\sqrt{(x_b^n - x_a^n)^2 + (z_b^n - z_a^n)^2}}$$

The velocities and position of A after a time increment (A moves to \*A) for the above restrictions are given by:

$$(u^x)_{*A}^{n+1/2} = (u^x)_O^{n-1/2} + \frac{1}{2} (\Delta t^{n+1/2} + \Delta t^{n-1/2}) (a^{ab} \cos \theta_{ab}^n)_A^n$$

$$(u^z)_{*A}^{n+1/2} = (u^z)_O^{n-1/2} + \frac{1}{2} (\Delta t^{n+1/2} + \Delta t^{n-1/2}) (a^{ab} \sin \theta_{ab}^n)_A^n$$

$$x_{*A}^{n+1} = x_A^n + \Delta t^{n+1/2} (u^x)_{*A}^{n+1/2}$$

$$z_{*A}^{n+1} = z_A^n + \Delta t^{n+1/2} (u^z)_{*A}^{n+1/2}$$

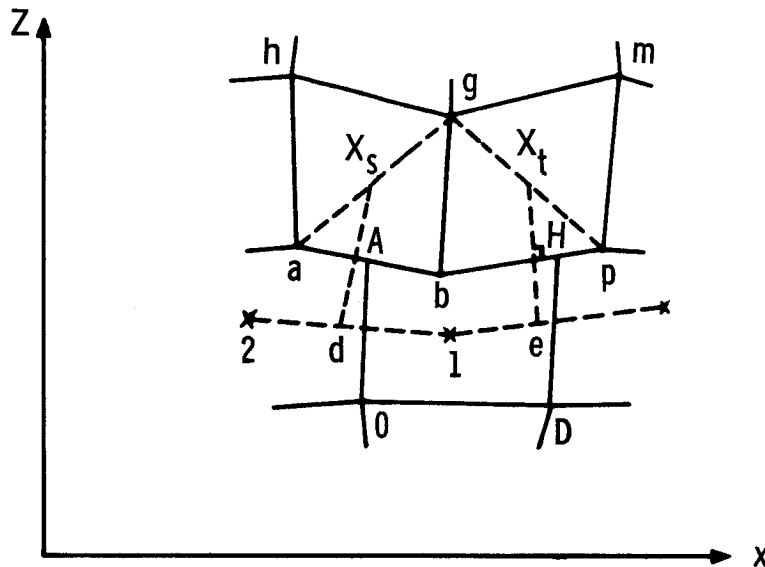
Here the velocities at point A at time (n-1/2) are assumed to be the same as the velocities of point O. Also the point \*A may pass to another line segment.

The slope of line O\*A is given by

$$\frac{z_{*A}^{n+1} - z_O^{n+1}}{x_{*A}^{n+1} - x_O^{n+1}} = (\tan \theta_{O^*A})^{n+1} = (\tan \theta_{OA})^{n+1}$$

To advance in time points on the slide line.

The motion of points a, b, etc. are calculated assuming that the grid points are accelerated by the forces from below as well as above the interface. Forces from below the interface are calculated by the scheme given below.



Given the points 1, 2, x<sub>S</sub>, etc. where

$$\bar{x}_1^n = \frac{1}{4} (x_O^n + x_D^n + x_H^n + x_A^n)$$

$$\bar{z}_1^n = \frac{1}{4} (z_O^n + z_D^n + z_H^n + z_A^n)$$

$$\bar{x}_2^n = \frac{1}{4} (x_O^n + x_A^n + x_E^n + x_B^n)$$

$$\bar{z}_2^n = \frac{1}{4} (z_O^n + z_A^n + z_E^n + z_B^n)$$



TR66-85

$$\bar{x}_S^n = \frac{1}{4} (x_a^n + x_b^n + x_g^n + x_h^n)$$

$$\bar{z}_S^n = \frac{1}{4} (z_a^n + z_b^n + z_g^n + z_h^n)$$

and the slope  $m_2$  of the line through  $X_S$  perpendicular to  $ab$ , i. e.

$$m_2 = -\cot \theta_{ab}$$

$$= -\frac{x_b^n - x_a^n}{z_b^n - z_a^n}$$

which bisects the line 12 at  $d$ . In order to calculate the coordinates of point  $d$ , it is necessary to find between which points  $d$  lies.

For each point, i. e. 1, 2, etc., calculate

$$x_2^1 - x_S^1 = (x_2 - x_O) \cos \theta_{ab} - (z_2 - z_S) \sin \theta_{ab}$$

Test consecutive values of  $(x_i^1 - x_S^1)$  until a change in sign is found. The points 1 and 2 between which  $d$  lies will then have been located. Coordinates of point  $d$  can now be calculated.

$$x_d^n = \frac{(m_{12} \bar{x}_2^n - m_2 \bar{x}_S^n + \bar{z}_S^n - \bar{z}_2^n)}{m_{12} - m_2}$$

$$z_d^n = \frac{m_2 \left[ \bar{z}_2^n - m_{12} (\bar{x}_2^n - \bar{x}_S^n) \right] - m_{12} \bar{z}_S^n}{m_2 - m_{12}}$$

$$m_{12} = \frac{\bar{z}_1^n - \bar{z}_2^n}{\bar{x}_1^n - \bar{x}_2^n}$$

If  $|m_{12}| < 10^{-4}$  and  $|m_2| > 10^4$ ,  $x_d = \bar{x}_S^n$  and  $z_d = \bar{z}_2^n$ .

The stresses at d are calculated as follows:

$$p_d^n + q_d^{n-1/2} = \frac{(p_1^n + q_1^{n-1/2})|d - X_2| + (p_2^n + q_2^{n-1/2})|d - X_1|}{|X_1 - X_2|}$$

where

$$|d - X_2| = \sqrt{(x_d^n - \bar{x}_2^n)^2 + (z_d^n - \bar{z}_2^n)^2}$$

$$|d - X_1| = \sqrt{(x_d^n - \bar{x}_1^n)^2 + (z_d^n - \bar{z}_2^n)^2}$$

$$|X_1 - X_2| = \sqrt{(\bar{x}_1^n - \bar{x}_2^n)^2 + (\bar{z}_1^n - \bar{z}_2^n)^2}$$

Similarly the stresses at e are found by following the same steps.

The accelerations at point b are calculated parallel and perpendicular to ap in the following manner:

$$(r)^n = - \frac{(t^{xx})_t^n (z_p^n - z_g^n) + (t^{xx})_s^n (z_g^n - z_a^n) - (p_d^n + q_d^{n-1/2})(z_a^n - z_b^n) - (p_e^n + q_e^{n-1/2})(z_b^n - z_p^n)}{A_{It}^n \rho_t^n + A_{Is}^n \rho_s^n}$$

$$+ \frac{(t^{xz})_t^n (x_p^n - x_g^n) + (t^{xz})_s^n (x_g^n - x_a^n)}{A_{It}^n \rho_t^n + A_{Is}^n \rho_s^n}$$

$$+ \frac{(\alpha-1)}{2} \left\{ \frac{(t^{xx})_t^n - (t^{yy})_t^n}{\rho_t^n \bar{x}_{It}^n} + \frac{(t^{xx})_s^n - (t^{yy})_s^n}{\rho_s^n \bar{x}_{Is}^n} \right\}$$

TR66-85

and

$$\begin{aligned}
 (g)^n = & - \frac{(t^{xz})_t^n (z_p^n - z_g^n) + (t^{xz})_s^n (z_g^n - z_a^n)}{A_{It}^n \rho_t^n + A_{Is}^n \rho_s^n} \\
 & + \frac{(t^{zz})_t^n (x_p^n - x_g^n) + (t^{zz})_s^n (x_g^n - x_a^n) - (p_d^n + q_d^{n-1/2}) (x_a^n - x_b^n) - (p_e^n + q_e^{n-1/2}) (x_b^n - x_p^n)}{A_{It}^n \rho_t^n + A_{Is}^n \rho_s^n} \\
 & + \frac{(\alpha-1)}{2} \left\{ \frac{(t^{xz})_t^n}{\rho_t^n \bar{x}_t^n} + \frac{(t^{xz})_s^n}{\rho_s^n \bar{x}_s^n} \right\}
 \end{aligned}$$

where

$$\bar{x}_t^n = \frac{1}{3} (x_b^n + x_p^n + x_g^n)$$

$$\bar{x}_s^n = \frac{1}{3} (x_a^n + x_b^n + x_g^n)$$

The accelerations of b parallel and perpendicular to ap are given by

$$(R)_b^n = (r)^n (\cos \theta_{ap})^n + (g)^n (\sin \theta_{ap})^n \quad \text{parallel to ap}$$

$$(G)_b^n = \frac{M_t}{M_t + M_1} \left\{ - (r)^n (\sin \theta_{ap})^n + (g)^n (\cos \theta_{ap})^n \right\} \quad \text{perpendicular to ap}$$

where

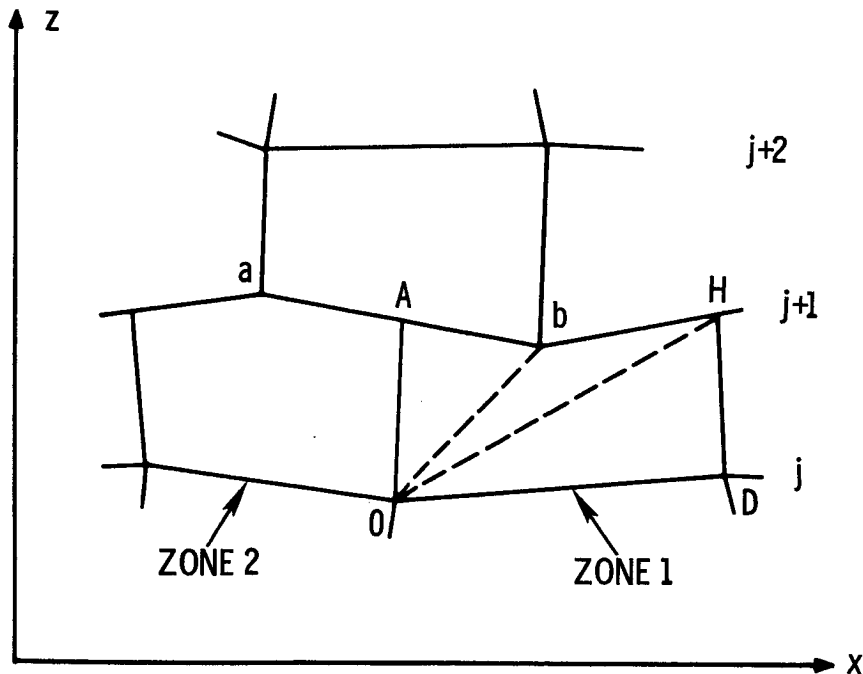
$$(\sin \theta_{ap})^n = \frac{z_p^n - z_a^n}{\sqrt{(z_p^n - z_a^n)^2 + (x_p^n - x_a^n)^2}}$$

$$(\cos \theta_{ap})^n = \frac{x_p^n - x_a^n}{\sqrt{(z_p^n - z_a^n)^2 + (x_p^n - x_a^n)^2}}$$

Using the values calculated above for the accelerations, the velocities and position of point b at a later time are:

$$\begin{aligned} (u^x)_b^{n+1/2} &= (u^x)_b^{n-1/2} + \frac{1}{2} (\Delta t^{n+1/2} + \Delta t^{n-1/2}) \left[ (R)_b^n (\cos \theta_{ap})^n - (G)_b^n (\sin \theta_{ap})^n \right] \\ (u^z)_b^{n+1/2} &= (u^z)_b^{n-1/2} + \frac{1}{2} (\Delta t^{n+1/2} + \Delta t^{n-1/2}) \left[ (R)_b^n (\sin \theta_{ap})^n + (G)_b^n (\cos \theta_{ap})^n \right] \\ x_b^{n+1} &= x_b^n + \Delta t^{n+1/2} (u^x)_b^{n+1/2} \\ z_b^{n+1} &= z_b^n + \Delta t^{n+1/2} (u^z)_b^{n+1/2} \end{aligned}$$

At time (n+1) it is necessary to search for points between which A lies and the coordinates of point A , as well as the densities and areas in zones 1, 2, etc., and 5, 6, 7, 8, 9, 10, etc. We begin with the knowledge of the coordinates of 0 and the slope of the line OA (see diagram below)



For each point a, b, p, etc. we calculate

$$(\tan \theta_{AOB})^{n+1} = \frac{(\tan \theta_{OA})^{n+1} - (\tan \theta_{Ob})^{n+1}}{1 + (\tan \theta_{OA})^{n+1} (\tan \theta_{Ob})^{n+1}}$$

TR66-85

where

$$(\tan \theta_{Ob})^{n+1} = \frac{z_b^{n+1} - z_O^{n+1}}{x_b^{n+1} - x_O^{n+1}}$$

with the following limitations.

$$(i) \quad \text{if } |\tan \theta_{OA}| > 10^4 \quad \tan \theta_{AOB} = \frac{1}{\tan \theta_{Ob}}$$

$$(ii) \quad \text{if } |\tan \theta_{Ob}| > 10^4 \quad \tan \theta_{AOB} = -\frac{1}{\tan \theta_{OA}}$$

$$(iii) \quad \text{if } |\tan \theta_{OA}| > 10^4 \text{ and } |\tan \theta_{Ob}| > 10^4, \quad \tan \theta_{AOB} = 0$$

Test for  $\tan \theta_{AOp}$  etc., i. e. consecutive points along the slide line, until a change of sign is found. The points where this will occur are the points between which A lies, i. e. a and b in the diagram.

The coordinates of A may now be found

$$x_A^{n+1} = \frac{\left[ z_a^{n+1} - z_O^{n+1} + (\tan \theta_{OA})^{n+1} x_O^{n+1} - (\tan \theta_{ab})^{n+1} x_a^{n+1} \right]}{\left[ (\tan \theta_{OA})^{n+1} - (\tan \theta_{ab})^{n+1} \right]}$$

$$z_A^{n+1} = \frac{(\tan \theta_{OA})^{n+1} \left[ z_a^{n+1} - (\tan \theta_{ab})^{n+1} (x_a^{n+1} - x_O^{n+1}) \right] - (\tan \theta_{ab})^{n+1} z_O^{n+1}}{\left[ (\tan \theta_{OA})^{n+1} - (\tan \theta_{ab})^{n+1} \right]}$$

$$(\tan \theta_{ab})^{n+1} = \frac{z_b^{n+1} - z_a^{n+1}}{x_b^{n+1} - x_a^{n+1}}$$

Similarly for point H.

We can now calculate the areas enclosed by AODHb

$$A_5^{n+1} = \frac{1}{2} \left\{ (x_D^{n+1} - x_O^{n+1})(z_H^{n+1} - z_O^{n+1}) - (z_D^{n+1} - z_O^{n+1})(x_H^{n+1} - x_O^{n+1}) \right\}$$

$$A_6^{n+1} = \frac{1}{2} \left\{ (x_H^{n+1} - x_O^{n+1})(z_b^{n+1} - z_O^{n+1}) - (z_H^{n+1} - z_O^{n+1})(x_b^{n+1} - x_O^{n+1}) \right\}$$

$$A_7^{n+1} = \frac{1}{2} \left\{ (x_b^{n+1} - x_O^{n+1})(z_A^{n+1} - z_O^{n+1}) - (z_b^{n+1} - z_O^{n+1})(x_A^{n+1} - x_O^{n+1}) \right\}$$

The density of zone 1 is given by

$$\rho_1^{n+1} = \frac{M_1}{A_5^{n+1} (\bar{x}_5^{n+1})^{\alpha-1} + A_b^{n+1} (\bar{x}_b^{n+1})^{\alpha-1} + A_7^{n+1} (\bar{x}_7^{n+1})^{\alpha-1}}$$

where

$$\bar{x}_5^{n+1} = \frac{1}{3} (x_O^{n+1} + x_D^{n+1} + x_H^{n+1})$$

$$\bar{x}_6^{n+1} = \frac{1}{3} (x_O^{n+1} + x_H^{n+1} + x_b^{n+1})$$

$$\bar{x}_7^{n+1} = \frac{1}{3} (x_O^{n+1} + x_b^{n+1} + x_A^{n+1})$$

Coordinates at the center of the zone are taken to be

$$\bar{x}_1^{n+1} = \frac{1}{4} (x_O^{n+1} + x_D^{n+1} + x_H^{n+1} + x_A^{n+1})$$

$$\bar{z}_1^{n+1} = \frac{1}{4} (z_O^{n+1} + z_D^{n+1} + z_H^{n+1} + z_A^{n+1})$$

(Although these are not, in general, the center of the zone they are adequate to the current approximation.)

Similarly, the areas and densities in zone OAaEB, etc. are obtained.

## G. EQUATION OF STATE

### (1) Polynomial

The hydrodynamic pressure is assumed to be given by the relation

$$p = f_1(\rho) + f_2(\rho) \varepsilon$$

TR66-85

which in finite difference form is

$$p^{n+1} = f_1^{n+1} + f_2^{n+1} e^{n+1}$$

Substituting in the energy equation

$$e^{n+1} - e^n = \frac{1}{2} (f_1^{n+1} + f_2^{n+1} e^{n+1} + p^n) \frac{\Delta \rho}{\rho} + \Delta_q e + \Delta^d e$$

or

$$e^{n+1} = \frac{e^n + 1/2 (f_1^{n+1} + p^n) \frac{\Delta \rho}{\rho} + \Delta_q e + \Delta^d e}{1 - 1/2 f_2^{n+1} \frac{\Delta \rho}{\rho}}$$

On evaluating  $e^{n+1}$  the pressure  $p^{n+1}$  can be calculated.

## (2) Tillotson

In this equation of state the hydrodynamic pressure is given by

$$p = a + \left\{ \frac{b}{1 + \epsilon/\epsilon_0 \eta^2} \right\} \rho \epsilon + A \mu + B \mu^2$$

for

$$\eta > 1 \quad \text{for all } \epsilon > 0 \quad (\text{Region II})$$

and

$$\eta > \eta_s \quad \text{for all } \epsilon < \epsilon'_s \quad (\text{Region III})$$

and in Region IV

$$p = a e_\rho + \left\{ \frac{b \epsilon \rho}{1 + \epsilon/\epsilon_0 \eta^2} + A \mu e^{\beta[1-1/\eta]} \right\} e^{-\alpha[1-1/\eta]^2}$$

for

$$\eta_s < \eta < 1 \quad \text{for } \epsilon < \epsilon'_s$$

and

$$\eta_s > \eta \quad \text{for all } \epsilon > 0$$

In the above equations

$$\mu = n - 1$$

Two methods of solving for the pressure and energy are outlined below.

In Regions II and III, i. e. for

$$\eta_3^{n+1} > 1 \quad \text{and} \quad \epsilon_3^{n+1} > 0$$

and

$$\eta_3^{n+1} > \eta_s \quad \text{for all} \quad \epsilon_3^{n+1} < \epsilon'_s$$

the equation of state can be written

$$p_3^{n+1} = \epsilon_3^{n+1} \left( b_1 + \frac{b_2}{b_3 \epsilon_3^{n+1} + 1} \right) + b_4$$

where

$$b_1 = a \eta_3^{n+1} \rho_3^0$$

$$b_2 = b \eta_3^{n+1} \rho_3^0$$

$$b_3 = \frac{1}{\epsilon_0 (\eta_3^{n+1})^2}$$

$$b_4 = A (\eta_3^{n+1} - 1) + B (\eta_3^{n+1} - 1)^2$$

Combining with the energy equation

$$p_3^{n+1} = a_1 \epsilon_3^{n+1} + a_2$$

where

$$a_1 = \frac{2}{(\Delta\rho/\rho^2)_3^{n+1}}$$



TR66-85

$$a_2 = \frac{-2}{(\Delta\rho/\rho^2)_3^{n+1}} \left( e_3^n + \Delta_q e_3 + \Delta^d e_3 \right) - p_3^n$$

we obtain

$$e_3^{n+1} = \frac{-K_2 + \sqrt{K_2^2 - 4K_1K_3}}{2K_1}$$

where

$$K_1 = b_3 (a_1 - b_1)$$

$$K_2 = (a_1 - b_1) + b_3 (a_2 - b_4) - b_2$$

$$K_3 = a_2 - b_4$$

In Region IV i. e. where

$$\eta_s < \eta_3^{n+1} < 1 \text{ for } e_3^{n+1} < e'_s$$

and

$$\eta_s > \eta_3^{n+1} \text{ for all } e_3^{n+1} > 0$$

the equation of state can be written

$$p_3^{n+1} = b_1 e_3^{n+1} + \frac{c_2 e_3^{n+1}}{b_3 e_3^{n+1} + 1} + c_3$$

where

$$c_1 = \frac{1}{\eta_3^{n+1}} - 1$$

$$c_2 = b \eta_3^{n+1} \rho_3^0 e^{-\alpha c_1^2}$$

$$c_3 = A (\eta_3^{n+1} - 1) e^{-\beta c_1} e^{-\alpha c_1^2}$$

Combining with the energy equation

$$p_3^{n+1} = a_1 e_3^{n+1} + a_2$$

we obtain

$$e_3^{n+1} = \frac{-K_5 + \sqrt{K_5^2 - 4K_4 K_6}}{2K_4}$$

where

$$K_4 = b_3 (a_1 - b_1)$$

$$K_5 = (a_1 - b_1) + b_3 (a_2 - c_3) - c_2$$

$$K_6 = a_2 - c_3$$

Alternatively an iterative procedure may be used

$$\tilde{e}^{n+1} = e^n + (p^n + q^{n-1/2}) \frac{\Delta\rho}{\rho}$$

$$\tilde{p}^{n+1} = p(\tilde{e}^{n+1}, \rho^{n+1})$$

where  $p(e, \rho)$  are the equations of state given previously, and  $\tilde{p}^{n+1}, \tilde{e}^{n+1}$  are the first approximations to the pressure and energy respectively at time  $(n+1)$ .

The values assumed at time  $(n+1)$  are taken to be

$$e^{n+1} = \tilde{e}^{n+1} + 1/2 (\tilde{p}^{n+1} + p^n) \frac{\Delta\rho}{\rho}$$

$$p^{n+1} = p(\tilde{e}^{n+1}, \rho^{n+1})$$

### (3) Elastic-Perfectly Plastic

The equations relating stress deviations to the deviatoric strain rate written in finite difference form leads to an implicit set of equations

$$\begin{aligned} \left( \frac{d}{e} \right)_{t,xx}^{n+1} - \left( \frac{d}{e} \right)_{t,xx}^n - \Delta t^{n+1/2} (w^{xz})^{n+1/2} \left\{ \left( \frac{d}{e} \right)_{t,xz}^{n+1} + \left( \frac{d}{e} \right)_{t,xz}^n \right\} \\ = 2 \Delta t^{n+1/2} G^{n+1/2} (d_e)_{xx}^{n+1/2} \end{aligned}$$

TR66-85

$$\left(\frac{d}{e}t^{yy}\right)^{n+1} - \left(\frac{d}{e}t^{yy}\right)^n = 2\Delta t^{n+1/2} G^{n+1/2} \left(\frac{d}{e}\dot{y}y\right)^{n+1/2}$$

$$\begin{aligned} \left(\frac{d}{e}t^{zz}\right)^{n+1} - \left(\frac{d}{e}t^{yy}\right)^n + \Delta t^{n+1/2} (w^{xz})^{n+1/2} \left\{ \left(\frac{d}{e}t^{xz}\right)^{n+1} + \left(\frac{d}{e}t^{xz}\right)^n \right\} \\ = 2\Delta t^{n+1/2} G^{n+1/2} \left(\frac{d}{e}\dot{z}z\right)^{n+1/2} \end{aligned}$$

$$\begin{aligned} \left(\frac{d}{e}t^{xz}\right)^{n+1} - \left(\frac{d}{e}t^{xz}\right)^n + 1/2\Delta t^{n+1/2} (w^{xz})^{n+1/2} \left\{ \left(\frac{d}{e}t^{xx}\right)^{n+1} + \left(\frac{d}{e}t^{xx}\right)^n \right. \\ \left. - \left(\frac{d}{e}t^{zz}\right)^{n+1} - \left(\frac{d}{e}t^{zz}\right)^n \right\} = 2\Delta t^{n+1/2} G^{n+1/2} \left(\frac{d}{e}\dot{x}z\right)^{n+1/2} \end{aligned}$$

If we choose the time increment  $\Delta t^{n+1/2}$  sufficiently small so that

$$\left[ \Delta t^{n+1/2} (w^{xz})^{n+1/2} \right]^2 \ll 1$$

i. e. the rotation from  $t^n$  to  $t^{n+1}$  is small, then these equations may be expressed in a relatively simple explicit form. The rotation in time  $\Delta t^{n+1/2}$  is

$$\Delta\theta = \Delta t^{n+1/2} w^{xz}{}^{n+1/2}$$

and the stress deviators at  $t^{n+1}$  given approximately by

$$\begin{aligned} \frac{d}{e}t^{xx} = \left(\frac{d}{e}t^{xx}\right)^n + 2\Delta t^{n+1/2} G^{n+1/2} \left(\frac{d}{e}\dot{x}x\right)^{n+1/2} + 2\Delta\theta \left\{ \left(\frac{d}{e}t^{xz}\right)^n \right. \\ \left. + \Delta t^{n+1/2} G^{n+1/2} \left(\frac{d}{e}\dot{x}z\right)^{n+1/2} \right\} \end{aligned}$$

$$\frac{d}{e}t^{yy} = \left(\frac{d}{e}t^{yy}\right)^n + 2\Delta t^{n+1/2} G^{n+1/2} \left(\frac{d}{e}\dot{y}y\right)^{n+1/2}$$

$$\begin{aligned} \frac{d}{e}t^{zz} = \left(\frac{d}{e}t^{zz}\right)^n + 2\Delta t^{n+1/2} G^{n+1/2} \left(\frac{d}{e}\dot{z}z\right)^{n+1/2} - 2\Delta\theta \left\{ \left(\frac{d}{e}t^{xz}\right)^n \right. \\ \left. + \Delta t^{n+1/2} G^{n+1/2} \left(\frac{d}{e}\dot{x}z\right)^{n+1/2} \right\} \end{aligned}$$

$$\begin{aligned} d_{e_t}^{xz} = & \left( d_{e_t}^{xz} \right)^n + 2\Delta t^{n+1/2} G^{n+1/2} \left( d_{e_t}^{xz} \right)^{n+1/2} - \Delta\theta \left\{ \left( d_{e_t}^{xx} \right)^n \right. \\ & \left. + \Delta t^{n+1/2} G^{n+1/2} \left( d_{e_t}^{xx} \right)^{n+1/2} - \left( d_{e_t}^{zz} \right)^n - \Delta t^{n+1/2} G^{n+1/2} \left( d_{e_t}^{zz} \right)^{n+1/2} \right\} \end{aligned}$$

These equations are valid only if the material remains elastic, and

$$\left( d_{e_t}^{xx} \right)^{n+1} = d_{e_t}^{xx}$$

$$\left( d_{e_t}^{yy} \right)^{n+1} = d_{e_t}^{yy}$$

$$\left( d_{e_t}^{zz} \right)^{n+1} = d_{e_t}^{zz}$$

$$\left( d_{e_t}^{xz} \right)^{n+1} = d_{e_t}^{xz}$$

If the material becomes plastic, i. e.

$$\bar{\Pi} = \left\{ \left( d_{e_t}^{xx} \right)^2 + \left( d_{e_t}^{yy} \right)^2 + \left( d_{e_t}^{zz} \right)^2 + 2 \left( d_{e_t}^{xz} \right)^2 \right\} > \frac{2}{3} y_o^2$$

or

$$\bar{\Pi} = 2 \left\{ \left( d_{e_t}^{xx} \right)^2 + \left( d_{e_t}^{xx} \right) \left( d_{e_t}^{zz} \right) + \left( d_{e_t}^{zz} \right)^2 + \left( d_{e_t}^{xz} \right)^2 \right\} > \frac{2}{3} y_o^2$$

then

$$\left( d_{e_t}^{xx} \right)^{n+1} = \sqrt{\frac{\frac{2}{3} y_o^2}{\bar{\Pi}}} \left( d_{e_t}^{xx} \right)$$

$$\left( d_{e_t}^{yy} \right)^{n+1} = \sqrt{\frac{\frac{2}{3} y_o^2}{\bar{\Pi}}} \left( d_{e_t}^{yy} \right)$$

$$\left( d_{e_t}^{zz} \right)^{n+1} = \sqrt{\frac{\frac{2}{3} y_o^2}{\bar{\Pi}}} \left( d_{e_t}^{zz} \right)$$

TR66-85

$$\left(d_{t,xz}\right)^{n+1} = \sqrt{\frac{2}{3} \frac{y_0^2}{\bar{\Pi}}} \left(d_{e,t,xz}\right)$$

The total stresses are given by

$$\left(t^{xx}\right)^{n+1} = -\left(p^{n+1} + q^{n+1/2}\right) + \left(d_{t,xx}\right)^{n+1} + \left(d_{q,xx}\right)^{n+1/2}$$

$$\left(t^{yy}\right)^{n+1} = -\left(p^{n+1} + q^{n+1/2}\right) + \left(d_{t,yy}\right)^{n+1} + \left(d_{q,yy}\right)^{n+1/2}$$

$$\left(t^{zz}\right)^{n+1} = -\left(p^{n+1} + q^{n+1/2}\right) + \left(d_{t,zz}\right)^{n+1} + \left(d_{q,zz}\right)^{n+1/2}$$

and

$$\left(t^{xz}\right)^{n+1} = \left(d_{t,xz}\right)^{n+1} + \left(d_{q,xz}\right)^{n+1/2}$$

where  $p$  is obtained from the hydrodynamic equation of state and the  $q$ 's by the expressions for artificial viscosity.

## IV. PROGRAM DESCRIPTION

### A. COMPUTER CONFIGURATION

The available computer configuration greatly influenced the form the program took. The program was set up to run on an IBM 7040/44 computer with 32 K of core storage and eight tape drives on a single channel. The system monitor program, input, output, plotter results, and the chain feature of the loader reduce the available scratch tapes to three. One of these tapes is used to give the program a re-start capability.

An IBM 1401 computer prepares the input tapes and processes both the printed and plotted results. A CalComp Model 565 plotter is driven on-line by the IBM 1401 computer.

The two remaining scratch tapes were not used to provide auxiliary storage, since the reading and writing of data from core to tape and back again would make computation time excessive with the single-channel computer configuration.

The fact that all data must remain in the core at all times puts a severe limitation on the number of zones into which a problem may be divided. At present the program will accept up to a total of 600 zones, which may be spread over five plates.

### B. MESH NUMBERING SCHEME

In order to promote a better understanding of the code itself, term definitions and the mesh numbering system will be explained. Figure 1 should be referred to during the following discussion.

#### TERM

plate

#### DEFINITION

an integral piece of material (for example, the target)

TR66-85

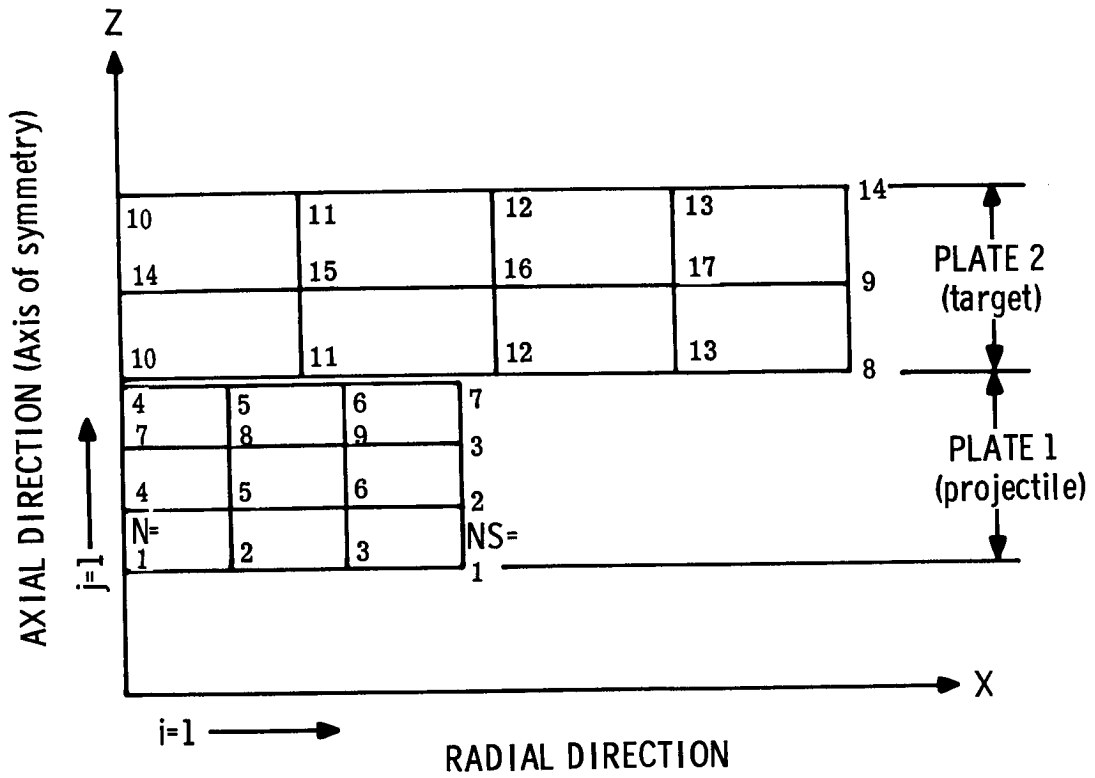


Figure 1 Mesh Numbering Scheme

|            |  |
|------------|--|
| zone       | a subdivision of a plate whose boundaries are formed by the grid lines (sometimes referred to as a mesh) |
| mesh point | the point of intersection of two grid lines (also referred to as a station)                              |

The zones themselves, with all their material properties, are numbered consecutively starting at the lower left corner, moving in the direction of increasing  $x$ , and then toward increasing  $z$ . The active zone indicator is  $N$ . The position and velocity of a zone are taken as the lower-left corner. All other zone properties are assumed to be at the center of the zone.

The right-hand and upper boundary mesh points are considered "special." Only position and velocity are stored. The active special station indicator is  $NS$ .

### C. PROGRAM STRUCTURE

The program is divided into a main program and three links, and uses the chain feature of the loader. The function of each part is as follows:

- a) Main program — directs sequence in which the links are called into core. Main program remains in core at all times.
- b) Link 1 — reads all input, and prints it for verification. Generates problem configuration. Initializes all variables prior to start of integration.
- c) Link 2 — integrates until time to output results or stop.
- d) Link 3 — prints and plots results, and updates restart tape on signal from Link 2.

The program has been divided into a large number of subroutines to facilitate program modifications and checkout (see Appendixes B and C).

In an attempt to make the code as machine-independent as possible, FORTRAN IV was used wherever possible. As a result, there are only two machine-language subroutines in the entire code.



TR66-85

One is a modified version of the IBM-supplied floating-point trap routine. It is used to set all underflows to zero, and to continue execution without an error message. However, on encountering the first overflow, a location-error message is printed, and the subroutine then transfers to a specified location in the main program. The code should run equally well with the standard IBM-supplied floating-point trap routine.

The other machine-language routine is a tape rewind subroutine to allow rewinding of a tape without first writing an end-of-file mark. This routine is used to initially rewind the plot tape (tape 2), and to rewind the chain tape (tape 0) to reduce the amount of time to load Link 2 after outputting results in Link 3.

All the plotting subroutines are made up of calls to the standard CALCOMP-supplied routines. However, CALCOMP does change its argument lists from time to time, so the calling sequences may be different from those at other installations.

#### D. PROGRAM FEATURES AND LIMITATIONS

Summarized below are some of the features incorporated into the code, along with some of its limitations and notes of caution.

1. Plane or cylindrically symmetric geometry may be specified.
2. Hydrodynamic or elastic-perfectly plastic media may be specified.
3. There is a choice of two equations of state:
  - a) Polynomial
  - b) Tillotson
4. Interface sliding is allowed between hydrodynamic and elastic-plastic plates and between two hydrodynamic plates, but not between two elastic-plastic plates.
5. If the interface between plates is not allowed to slide, then the zone width (radial component) in each plate must be the same.
6. To save computation time, if two complete rows of zones in the target plate are found to have no velocity, computation for this cycle is stopped, and computation for the next cycle is started.

TR66-85

7. The plates are not limited to rectangular shapes. By specifying the angle the boundaries make with respect to the X-axis, any quadrilateral shape may be used.
8. The "projectile" plate may be specified entirely as input. This allows the plate to take on any shape ("free-form").
9. Any individual plate must contain a minimum of four zones. That is, there must be two rows and two columns of zones in each plate.

At this writing not every combination of problem configurations has been tried, and some program errors may still exist.

TR66-85

## E. INPUT DATA FORMAT

The input data format is listed out below; Figures 2 through 5 show samples of the four types of load sheets used.

| DEFINITION OF INPUT QUANTITIES   | FORMAT   |
|--|----------|
| CARD 1.  | (12A6)   |
| TITLE - 72 CHARACTERS OF ALPHA-NUMERICS FOR PROBLEM IDENTIFICATION.<br>PRINTS OUT ON FIRST SHEET OF OUTPUT, AND ON EACH PLOT.      |          |
| CARD 2.  | (6E12.8) |
| STRTC - STARTING TIME CYCLE.<br>IF 0.0 - SUBROUTINE INTLIZ CALLED.<br>IF NOT 0.0 - READS TAPE 4 FOR VALUES OF COMMON A D PROCEEDS. |          |
| CARD 3.  | (6E12.8) |
| CZ - COEFFICIENT IN ARTIFICIAL VISCOSITY EQUATION  |          |
| CL - COEFFICIENT IN ARTIFICIAL VISCOSITY EQUATION  |          |
| C3 - COEFFICIENT IN DEVIATORIC ARTIFICIAL VISCOSITY EQUATION   |          |
| C4 - COEFFICIENT IN DEVIATORIC ARTIFICIAL VISCOSITY EQUATION   |          |
| FK - COEFFICIENT IN TIME INCREMENT EQUATION  |          |
| SBAR - UPPER LIMIT OF STRESS. COMPRESSION IS NEGATIVE.<br>IF STRESS EXCEEDS THIS VALUE, PROGRAM WILL STOP.                         | (D/SQCM) |
| CARD 4.  | (6E12.8) |
| TPRNT - TIME INCREMENT FOR PRINTOUT AND PLOT   | (SEC)    |
| TMAX - PROBLEM TERMINATE TIME  | (SEC)    |
| PRNTC - PRINT CYCLE INCREMENT.<br>ALLOWS PRINTOUT AND PLOT AFTER EACH PRNTC CYCLES IF DESIRED.                                     |          |
| WRTPC - WRITE TAPE INCREMENT<br>UPDATES THE RESTART TAPE AFTER EACH WRTPC CYCLES+  |          |
| STOPC - STOP CYCLE.<br>ALLOWS PROGRAM TO BE TERMINATED AFTER STOPC CYCLES.   |          |
| CARD 5 0.  | (12I6)   |
| M - NO. OF PLATES IN PROBLEM.  |          |
| NBL - BOUNDARY INDICATOR - LOWER X OF FIRST PLATE.<br>= 1 - FREE SURFACE<br>= 2 - FIXED SURFACE                                    |          |
| NBU - BOUNDARY INDICATOR - UPPER X OF LAST PLATE   |          |
| IALF - GEOMETRY INDICATOR<br>= 1 - PLANE<br>= 2 - CYLINDRICAL  |          |
| NOSLID - BOUNDARY INTERFACE INDICATOR<br>= 0 - THE INTERFACES ARE ALLOWED TO SLIDE. ONLY LAST PLATE                                |          |

TR86-85

C IS ALLOWED TO INCLUDE STRENGTH.  
 C = 1 - NO RELATIVE MOVEMENT ALONG INTERFACE BOUNDARIES.  
 C MESH SIZE IN X-DIRECTION MUST BE SAME FOR ALL PLATES.  
 C N1D - ONE DIMENSIONAL CODE SIMULATOR INDICATOR  
 C = 0 - NORMAL MODE - 2-D PROBLEM  
 C = 1 - 1-D PROBLEM  
 C NBO - ARTIFICIAL VISCOSITY INDICATOR  
 C = 0 - EVALUATE VISCOSITY ONLY IN COMPRESSION  
 C = 1 - ALWAYS EVALUATE VISCOSITY  
 C NPLOTT - PLOT INDICATOR  
 C = 0 OR BLANK - PLOT X VS. Z  
 C = 1 - PLOT TXX1 VS. X AND Z  
 C = 2 - PLOT TYY1 VS. X AND Z  
 C = 3 - PLOT TZZ1 VS. X AND Z  
 C = 4 - PLOT TXZ1 VS. X AND Z  
 C = 5 - DRAW ALL 5 PLOTS  
 C = 6 - SKIP PLOTS  
 C NCRVS - THE NUMBER OF COLUMNS (I VALUES) TO BE PLOTTED. MAX. OF 20.  
 C  
 C CARD 5 1. IF NCRVS = 0, OMIT THIS CARD. (1216)  
 C  
 C JCRVS(L),L=1,NCRVS  
 C - THE COLUMN VALUES TO BE PLOTTED.  
 C  
 C CARD 6 0. THERE ARE 8 CARDS TO DEFINE EACH PLATE. THERE (1216)  
 C WILL BE M SETS OF THE FOLLOWING 8 CARDS.  
 C  
 C IM - NO. OF ZONES IN RADIAL (X) DIRECTION  
 C JM - NO. OF ZONES IN AXIAL (Z) DIRECTION  
 C NBZL - INDICATOR FOR BOUNDARY CONDITION - LEFT Z OF EACH PLATE  
 C NBZR - INDICATOR FOR BOUNDARY CONDITION - RIGHT Z OF EACH PLATE  
 C MEQ - INDICATOR FOR EQUATION-OF-STATE  
 C = 1 - TILLOTSON  
 C = 2 - POLYNOMIAL  
 C  
 C  
 C CARD 6 1. (6E12.8)  
 C  
 C XI - INITIAL POSITION IN X-DIRECTION OF LOWER-LEFT CORNER (CM)  
 C XL - WIDTH OF PLATE IN X-DIRECTION (CM)  
 C ZI - INITIAL POSITION IN Z-DIRECTION OF UPPER-LEFT CORNER (CM)  
 C ZL - LENGTH OF PLATE IN Z-DIRECTION (CM)  
 C UXZ - INITIAL VELOCITY IN X-DIRECTION (CM/SEC)  
 C UZZ - INITIAL VELOCITY IN Z-DIRECTION (CM/SEC)  
 C  
 C  
 C CARD 6 2. (6E12.8)  
 C  
 C UXINT - INITIAL INTERFACE VELOCITY IN X-DIRECTION (CM/SEC)  
 C UZINT - INITIAL INTERFACE VELOCITY IN Z-DIRECTION (CM/SEC)  
 C RHOI - INITIAL VALUE OF MATERIAL DENSITY (G/CC)  
 C TXXZ - INITIAL VALUE OF STRESS IN X-DIRECTION (D/SQCM)  
 C TYYZ - INITIAL VALUE OF STRESS IN Y-DIRECTION (D/SQCM)  
 C TZZZ - INITIAL VALUE OF STRESS IN Z-DIRECTION (D/SQCM)

TR66-85

C  
C CARD 6 3. (6E12.8)  
C  
C TX7Z - INITIAL VALUE OF SHEAR STRESS (D/SQCM)  
C EZ - INITIAL VALUE OF ENERGY (ERG/GM)  
C PHIL - ANGLE OF INCLINATION OF LEFT Z-BOUNDARY (DEG)  
C PHIR - ANGLE OF INCLINATION OF RIGHT Z-BOUNDARY (DEG)  
C PSIL - ANGLE OF INCLINATION OF LOWER X-BOUNDARY (DEG)  
C PSIU - ANGLE OF INCLINATION OF UPPER X-BOUNDARY (DEG)

C  
C IF IM = 0, THE PLATE WILL BE --FREE-FORM-- IN SHAPE (EX. A  
C DEBRIS CLOUD). ALL QUANTITIES NEEDED TO DEFINE THE  
C PLATE ARE INPUTTED.

C ONLY ONE FREE-FORM PLATE PER PROBLEM.

C CARDS 6 1, 6 2, 6 3 ARE OMITTED AND REPLACED WITH  
C THE FOLLOWING CARDS.

C THERE WILL BE JM SETS OF CARDS. EACH SET CONSISTS  
C OF THREE TYPES OF CARDS, AND DEFINES A ROW.

C  
C CARD 1. (16)

C IM1(J) - J IS THE ROW INDICATOR. NO. OF ZONES IN ROW J. U7 TO 20 ROWS  
C ARE ALLOWED.

C  
C CARD 2. EACH PAIR OF CARDS (ONE CARD 2 AND ONE CARD 3) (5E15.8)  
C DEFINES A ZONE. THEREFORE, THERE WILL BE IM1 PAIRS  
C OF CARDS.

C UX(N) - N IS THE ZONE COUNTER. RADIAL VELOCITY (CM/SEC)  
C UZ(N) - AXIAL VELOCITY (CM/SEC)  
C P(N) - HYDRODYNAMIC PRESSURE IN THE ZONE (D/SQCM)  
C E(N) - ENERGY (ERG/GM)  
C RHO(N) - DENSITY (G/CC)

C  
C CARD 3. (5E15.8)

C X(N) - RADIAL POSITION (CM)  
C Z(N) - AXIAL POSITION (CM)

C  
C AFTER ALL ZONE-DEFINING CARDS HAVE BEEN READ IN,  
C CARDS DEFINING THE POSITION AND VELOCITIES OF THE  
C RIGHT-HAND AND INTERFACE BOUNDARY MESH POINTS ARE  
C READ.

C  
C CARD 4. EACH CARD DEFINES ONE BOUNDARY MESH POINT. (5E15.8)

C UXS(NS) - RADIAL VELOCITY (CM/SEC)  
C UZS(NS) - AXIAL VELOCITY (CM/SEC)  
C XS(NS) - RADIAL POSITION (CM)  
C ZS(NS) - AXIAL POSITION (CM)

TR66-85

```

C
C   THE FOLLOWING QUANTITIES DESCRIBE THE MATERIAL PROPERTIES OF EACH PLATE
C
C   FOR MEQ = 1
C   CARD 6 4.                                     (6E12.8)
C
C   RHOZ   - DENSITY                               (G/CC)
C   SA     - SMALL A IN E.O.S.
C   SB     - SMALL B IN E.O.S.
C   CAPA   - BIG A IN E.O.S.                       (D/SQCM)
C   CAPB   - BIG B IN E.O.S.                       (D/SQCM)
C   ALPHA  - ALPHA IN E.O.S.
C
C   FOR MEQ = 2
C   CARD 6 4.                                     (6E12.8)
C
C   RHOZ   - DENSITY                               (G/CC)
C   K0     - COEFFICIENT OF ARG**0 IN F1
C   K1     - COEFFICIENT OF ARG**1 IN F1
C   K2     - COEFFICIENT OF ARG**2 IN F1
C   K3     - COEFFICIENT OF ARG**3 IN F1
C   H0     - COEFFICIENT OF ARG**0 IN F2
C
C   FOR MEQ = 1
C   CARD 6 5.                                     (6E12.8)
C
C   BETA   - BETA IN E.O.S.
C   EZERO  - E SUB ZERO IN E.O.S.                 (ERG/GM)
C   RHOS   - RHO SUB S IN E.O.S.
C   FPRS   - E SUB S IN E.O.S.                   (ERG/GM)
C
C   FOR MEQ = 2
C   CARD 6 5.                                     (6E12.8)
C
C   H1     - COEFFICIENT OF ARG**1 IN F2
C   H2     - COEFFICIENT OF ARG**2 IN F2
C   H3     - COEFFICIENT OF ARG**3 IN F2
C
C   CARD 6 6.                                     (6E12.8)
C
C   CNO    - CONSTANT TERM IN SOUND SPEED EQUATION (CM/SEC)
C   CNA    - PRESSURE MULTIPLIER TERM IN SOUND SPEED EQUATION (CMPS/MB)
C   CNEXP  - EXPONENT OF PRESSURE TERM IN SOUND SPEED EQUATION
C
C   CARD 6 7.                                     (6E12.8)
C
C   G0     - COEFFICIENT IN MODULUS OF RIGIDITY FIT (D/SQCM)
C   G1     - COEFFICIENT IN MODULUS OF RIGIDITY FIT (D/SQCM)
C   G2     - COEFFICIENT IN MODULUS OF RIGIDITY FIT (D/SQCM)
C   G3     - COEFFICIENT IN MODULUS OF RIGIDITY FIT (D/SQCM)
C   CAPY   - YIELD STRENGTH                       (D/SQCM)
C   UTS    - ULTIMATE TENSILE STRENGTH OF MATERIAL (D/SQCM)
C
C   END OF INPUT QUANTITIES.

```











TR66-85

## F. PROGRAM OUTPUT

1. Printed Output

- a. At the start of a new problem, the input data read in from cards is printed out to serve as a check in case the results are not as expected. Figure 6 is an example.
- b. Figure 7 is an example of the printed output at each mesh, and is self-explanatory.

2. Plotted Output

There are two different types of plots which may be drawn:

- a. Mesh position (shape) at some instance in time
- b. Stress vs position at a given time.

The stress plots are drawn for each plate on a separate graph. Figures 8 and 9 are examples of the position and stress plots.

## G. EXAMPLE PROBLEMS

Three problems, to serve as examples, are solved and compared with analytical solutions here. They are as follows:

1. The first is for a one-dimensional shock-tube problem. The high-pressure and low-pressure initial conditions are as follows.

High Pressure

$$p = 48 \text{ Mb}$$

$$\rho = 0.8 \text{ gm/cm}^3$$

$$E = 90 \times 10^{12} \text{ ergs/gm}$$

Low Pressure

$$p = 0$$

$$\rho = 1.2 \text{ gm/cm}^3$$

$$E = 0$$

Both materials behave as perfect gases for which the ratio of specific heats is 5/3.

Comparison of the analytical and computer solution for the wave at 1.0 and 1.5  $\mu\text{sec}$  are given in Figure 10. It is seen that excellent agreement is obtained.

INPUT DATA

DEBRIS CLOUD TEST

| CO    | CL    | C3 | C4 | FK    | TPRNT-SEC      | TMAX-SEC       | SBAR-D/CSQ      | START C | PRINT C | WRTP C | STOP C |
|-------|-------|----|----|-------|----------------|----------------|-----------------|---------|---------|--------|--------|
| 2.000 | 0.100 | 0. | 0. | 0.100 | 0.20000000E-06 | 0.20000000E-06 | -0.99999998E 19 | 0.      | 0.      | 200.   | 0.     |

NO. PLATES X-LOW.B. X-UP.B. ALPHA SLDNG. IFCE VISC. IND.

|   |   |   |   |   |
|---|---|---|---|---|
| 2 | 1 | 1 | 2 | 0 |
|---|---|---|---|---|

PLATE X-STATIONS Z-STATIONS X-LFT.B. Z-RGMT.B. E,O,S.

|   |    |    |   |   |   |
|---|----|----|---|---|---|
| 1 | U  | 20 | 2 | 1 | 1 |
| 2 | 20 | 15 | 2 | 1 | 1 |

PLATE XI-CM XL-CM ZI-CM ZL-CM UXZ-CM/SEC UZZ-CM/SEC UXINT-CM/SEC UZINT-CM/SEC

|   |    |                |                |                |    |    |    |    |
|---|----|----------------|----------------|----------------|----|----|----|----|
| 2 | 0. | 0.10000000E 02 | 0.52264996E 01 | 0.30000000E 00 | 0. | 0. | 0. | 0. |
|---|----|----------------|----------------|----------------|----|----|----|----|

PLATE PHIL-DEG PMIR-DEG PSIL-DEG PSIU-DEG

|   |                |                |    |    |
|---|----------------|----------------|----|----|
| 2 | 0.90000000E 02 | 0.90000000E 02 | 0. | 0. |
|---|----------------|----------------|----|----|

PLATE RHOZ-G/CC TXXZ-D/CSQ TYYZ-D/CSQ TZZZ-D/CSQ TXZZ-D/CSQ EZ-ERG/GM

|   |                |    |    |    |    |    |
|---|----------------|----|----|----|----|----|
| 2 | 0.27000000E 01 | 0. | 0. | 0. | 0. | 0. |
|---|----------------|----|----|----|----|----|

PLATE 1 IS A DEBRIS CLOUD.

MESH UX-CM/SEC UZ-CM/SEC STRESS-D/CSQ E-ERG/G RHO-G/CC X-CM Z-CM

|    |                |                |    |                |                |                |                |
|----|----------------|----------------|----|----------------|----------------|----------------|----------------|
| 1  | 0.             | 0.16232680E 06 | 0. | 0.14626053E 12 | 0.36368664E-02 | 0.             | 0.41100000E 00 |
| 2  | 0.63592570E 05 | 0.11793859E 06 | 0. | 0.40497578E 11 | 0.70473591E-02 | 0.13900000E 00 | 0.41100000E 00 |
| 3  | 0.15490812E 06 | 0.92124053E 05 | 0. | 0.29810278E 11 | 0.11584522E-01 | 0.27800000E 00 | 0.41100000E 00 |
| 4  | 0.25979609E 06 | 0.59116723E 05 | 0. | 0.33381018E 11 | 0.25427544E-02 | 0.71499999E 00 | 0.41100000E 00 |
| 5  | 0.             | 0.23242365E 06 | 0. | 0.14457737E 12 | 0.36168440E-02 | 0.             | 0.48049999E 00 |
| 6  | 0.58597682E 05 | 0.17135757E 06 | 0. | 0.41166542E 11 | 0.79036359E-02 | 0.13900000E 00 | 0.48049999E 00 |
| 7  | 0.14597362E 06 | 0.12535069E 06 | 0. | 0.29982896E 11 | 0.89543827E-02 | 0.27800000E 00 | 0.48049999E 00 |
| 8  | 0.30838989E 06 | 0.10779782E 06 | 0. | 0.36400917E 11 | 0.24543573E-02 | 0.71499999E 00 | 0.48049999E 00 |
| 9  | 0.             | 0.32159433E 06 | 0. | 0.14241594E 12 | 0.32867227E-02 | 0.             | 0.54999998E 00 |
| 10 | 0.56538855E 05 | 0.24337544E 06 | 0. | 0.44008555E 11 | 0.66583531E-02 | 0.13900000E 00 | 0.54999998E 00 |
| 11 | 0.14721478E 06 | 0.18945691E 06 | 0. | 0.30779859E 11 | 0.71246836E-02 | 0.27800000E 00 | 0.54999998E 00 |
| 12 | 0.3423738E 07  | 0.18486100E 06 | 0. | 0.395681E 12   | 0.22308168E-02 | 0.13900000E 00 | 0.54999998E 00 |

POINT UXZ-CM/SEC UZS-CM/SEC XS-CM ZS-CM

|    |                |                |                |                |
|----|----------------|----------------|----------------|----------------|
| 1  | 0.53649670E 06 | 0.42494797E 05 | 0.12710000E 01 | 0.41100000E 00 |
| 2  | 0.57374787E 06 | 0.10104062E 06 | 0.12710000E 01 | 0.48049999E 00 |
| 3  | 0.60599460E 06 | 0.16805187E 06 | 0.12710000E 01 | 0.54999998E 00 |
| 4  | 0.94631103E 06 | 0.24827040E 06 | 0.18269999E 01 | 0.61949997E 00 |
| 5  | 0.10253670E 07 | 0.44339401E 06 | 0.23829999E 01 | 0.77849996E 00 |
| 6  | 0.11366269E 07 | 0.67577897E 06 | 0.23829999E 01 | 0.10565000E 01 |
| 7  | 0.15167652E 07 | 0.11083706E 07 | 0.23829999E 01 | 0.13345000E 01 |
| 8  | 0.1592223E 07  | 0.1292223E 07  | 0.23829999E 01 | 0.1512223E 01  |
| 9  | 0.16625416E 07 | 0.15092541E 07 | 0.29389999E 01 | 0.18904999E 01 |
| 10 | 0.18607950E 07 | 0.17616570E 07 | 0.29389999E 01 | 0.21584999E 01 |
| 11 | 0.18968692E 07 | 0.19380253E 07 | 0.29389999E 01 | 0.24444999E 01 |
| 12 | 0.19400254E 07 | 0.21224722E 07 | 0.29389999E 01 | 0.27244999E 01 |
| 13 | 0.19625958E 07 | 0.22940918E 07 | 0.29389999E 01 | 0.30024998E 01 |
| 14 | 0.19928162E 07 | 0.24096199E 07 | 0.29389999E 01 | 0.32804998E 01 |
| 15 | 0.1710578E 07  | 0.25081730E 07 | 0.29389999E 01 | 0.35584998E 01 |
| 16 | 0.17069826E 07 | 0.26368723E 07 | 0.29389999E 01 | 0.38364998E 01 |
| 17 | 0.17021509E 07 | 0.27475747E 07 | 0.29389999E 01 | 0.41144997E 01 |
| 18 | 0.14668726E 07 | 0.28153426E 07 | 0.29389999E 01 | 0.43924997E 01 |
| 19 | 0.14586341E 07 | 0.29236888E 07 | 0.29389999E 01 | 0.46704996E 01 |
| 20 | 0.14597778E 07 | 0.30090224E 07 | 0.29389999E 01 | 0.49484996E 01 |
| 21 | 0.             | 0.28810549E 07 | 0.             | 0.52264996E 01 |
| 22 | 0.16510378E 06 | 0.29044222E 07 | 0.13900000E 00 | 0.52264996E 01 |
| 23 | 0.3883575E 06  | 0.29245874E 07 | 0.27800000E 00 | 0.52264996E 01 |
| 24 | 0.74739373E 06 | 0.29681583E 07 | 0.71499999E 00 | 0.52264996E 01 |
| 25 | 0.11008820E 07 | 0.30384203E 07 | 0.12710000E 01 | 0.52264996E 01 |
| 26 | 0.11008820E 07 | 0.30384203E 07 | 0.18269999E 01 | 0.52264996E 01 |
| 27 | 0.11008820E 07 | 0.30384203E 07 | 0.23829999E 01 | 0.52264996E 01 |
| 28 | 0.11008820E 07 | 0.30384203E 07 | 0.29389999E 01 | 0.52264996E 01 |

MATERIAL CONSTANTS

PLATE RHO-G/CC SA SB A B ALPHA BETA EZERO

|   |                |                |                |                |                |                |                |                |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1 | 0.27000000E 01 | 0.50000000E 00 | 0.16300000E 01 | 0.75200000E 12 | 0.65000000E 12 | 0.50000000E 01 | 0.50000000E 01 | 0.50000000E 11 |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|

RHOS ES CNO CNA CNEXP

|                |                |                |                |                |
|----------------|----------------|----------------|----------------|----------------|
| 0.91000000E 00 | 0.30000000E 11 | 0.52800000E 06 | 0.44000000E 01 | 0.45000000E 00 |
|----------------|----------------|----------------|----------------|----------------|

GO G1 G2 G3 CAPY UTS

|    |    |    |    |    |                |
|----|----|----|----|----|----------------|
| 0. | 0. | 0. | 0. | 0. | 0.10000000E 20 |
|----|----|----|----|----|----------------|

PLATE RHO-G/CC SA SB A B ALPHA BETA EZERO

|   |                |                |                |                |                |                |                |                |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 2 | 0.27000000E 01 | 0.50000000E 00 | 0.16300000E 01 | 0.75200000E 12 | 0.65000000E 12 | 0.50000000E 01 | 0.50000000E 01 | 0.50000000E 11 |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|

RHOS ES CNO CNA CNEXP

|                |                |                |                |                |
|----------------|----------------|----------------|----------------|----------------|
| 0.91000000E 00 | 0.30000000E 11 | 0.52800000E 06 | 0.44000000E 01 | 0.45000000E 00 |
|----------------|----------------|----------------|----------------|----------------|

GO G1 G2 G3 CAPY UTS

|                |                |                |    |    |                |                |
|----------------|----------------|----------------|----|----|----------------|----------------|
| 0.28630000E 12 | 0.58610000E 01 | 0.42670000E 01 | 0. | 0. | 0.27600000E 10 | 0.12000000E 11 |
|----------------|----------------|----------------|----|----|----------------|----------------|

RESTART TAPE UPDATED AT CYCLE 98

Figure 6 Example of Printed Input Data

TR66-85

DEBRIS CLOUD TEST

CYCLE TIME-SEC DELTA T-SEC  
 98 0.20099201E-06 0.17736918E-08

PER CYCLE ENERGIES - INTERNAL-ERG KINETIC-ERG TOTAL-ERG RADIAL MOM.-G CM/SEC AXIAL MOM.-G CM/SEC  
 0.13242272E 11 0.18597794E 12 0.19922023E 12 0.65656913E 05 0.14474099E 06

ACCUMULATED ENERGIES - VISC.ST.-ERG PLASTIC-ERG  
 0.92204328E 10 0.53013678E 11

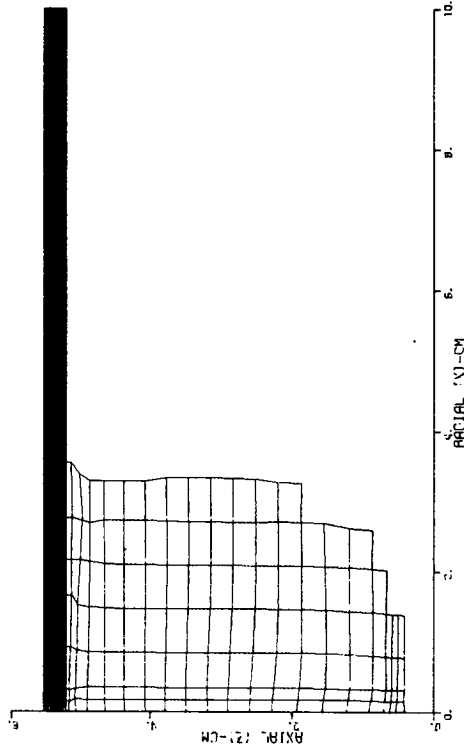
| PLATE Z | X   | SIGMA X-D/CSQ                      | SIGMA Y-D/CSQ                      | SIGMA Z-D/CSQ                      | SIGMA XZ-D/CSQ                    | Q-D/CSQ                           | RMO-G/CC             | ENERGY-ERG/GM                    |
|---------|-----|------------------------------------|------------------------------------|------------------------------------|-----------------------------------|-----------------------------------|----------------------|----------------------------------|
| ST.     | ST. | X VEL.-CM/SEC                      | Z VEL.-CM/SEC                      | X-CM                               | Z-CM                              | Q X-D/CSQ                         | Q Z-D/CSQ            | Q XZ-D/CSQ                       |
| 1       | 1   | 1                                  | -0.10616871E 09<br>0.              | -0.10616871E 09<br>-0.13754007E 06 | -0.10616871E 09<br>0.             | 0.<br>0.40930738E 00              | 0.<br>0.             | 0.19709845E-02<br>0.10773091E 12 |
| 1       | 2   | -0.65543364E 08<br>0.88902756E 05  | -0.65543364E 08<br>-0.49057465E 05 | -0.65543364E 08<br>0.15455030E 00  | 0.<br>0.41605540E 00              | 0.<br>0.                          | 0.41934414E-02<br>0. | 0.31259687E 11<br>0.             |
| 1       | 3   | -0.90973737E 08<br>0.15224580E 06  | -0.90973737E 08<br>-0.19856104E 05 | -0.90973737E 08<br>0.30888072E 00  | 0.<br>0.41362525E 00              | 0.<br>0.                          | 0.75541121E-02<br>0. | 0.24085669E 11<br>0.             |
| 1       | 4   | -0.21970023E 08<br>0.26674557E 06  | -0.21970023E 08<br>-0.31743041E 05 | -0.21970023E 08<br>0.76795061E 00  | 0.<br>0.41117065E 00              | 0.<br>0.                          | 0.16387303E-02<br>0. | 0.26813244E 11<br>0.             |
| 1       | 4   | 0.54682711E 06                     | -0.35619604E 05                    | 0.13799244E 01                     | 0.41108781E 00                    |                                   |                      |                                  |
| 2       | 1   | -0.13887767E 09<br>0.              | -0.13887767E 09<br>0.22125749E 06  | -0.13887767E 09<br>0.              | 0.<br>0.52672721E 00              | 0.<br>0.                          | 0.23712154E-02<br>0. | 0.11713900E 12<br>0.             |
| 2       | 2   | -0.89730697E 08<br>0.81543677E 05  | -0.89730697E 08<br>0.16153804E 06  | -0.89730697E 08<br>0.19331455E 00  | 0.<br>0.51421955E 00              | 0.<br>0.                          | 0.53133050E-02<br>0. | 0.33775391E 11<br>0.             |
| 2       | 3   | -0.77964517E 08<br>0.14548466E 06  | -0.77964517E 08<br>0.13003949E 06  | -0.77964517E 08<br>0.30730387E 00  | 0.<br>0.50627167E 00              | 0.<br>0.                          | 0.62307004E-02<br>0. | 0.29025659E 11<br>0.             |
| 20      | 5   | -0.12738908E 10<br>0.24105352E 07  | -0.12738908E 10<br>0.45359393E 06  | -0.12738908E 10<br>0.16511711E 01  | 0.<br>0.51543037E 01              | 0.39605458E 09<br>0.              | 0.12059008E-03<br>0. | 0.14559071E 14<br>0.             |
| 20      | 6   | -0.73985686E 09<br>0.21017092E 07  | -0.73985686E 09<br>0.29467254E 06  | -0.73985686E 09<br>0.21642155E 01  | 0.<br>0.51539557E 01              | 0.14187053E 09<br>0.              | 0.97899499E-04<br>0. | 0.12216468E 14<br>0.             |
| 20      | 7   | -0.30911336E 09<br>0.22228069E 07  | 0.30911336E 09<br>0.30457271E 06   | -0.30911336E 09<br>0.27550267E 01  | 0.<br>0.51366759E 01              | 0.34664475E 08<br>0.              | 0.74359832E-04<br>0. | 0.73818173E 13<br>0.             |
| 20      | 7   | 0.38474041E 07                     | 0.36195002E 06                     | 0.35537726E 01                     | 0.51584222E 01                    |                                   |                      |                                  |
| 21      | 1   | 0.                                 | 0.26895355E 04                     | 0.                                 | 0.52268645E 01                    |                                   |                      |                                  |
| 1       | 21  | 2                                  | 0.32582105E 06                     | 0.25875733E 04                     | 0.15247205E 00                    | 0.52268541E 01                    |                      |                                  |
| 21      | 3   | 0.41525090E 06                     | 0.24867587E 04                     | 0.31501610E 00                     | 0.52268431E 01                    |                                   |                      |                                  |
| 21      | 4   | 0.13818733E 07                     | 0.15794278E 04                     | 0.91693105E 00                     | 0.52267586E 01                    |                                   |                      |                                  |
| 21      | 5   | 0.24089640E 07                     | 0.60488726E 03                     | 0.16519144E 01                     | 0.52266461E 01                    |                                   |                      |                                  |
| 21      | 6   | 0.21074272E 07                     | 0.47046787E 03                     | 0.21647161E 01                     | 0.52266052E 01                    |                                   |                      |                                  |
| 21      | 7   | 0.22244897E 07                     | 0.20162909E 03                     | 0.27566153E 01                     | 0.52265731E 01                    |                                   |                      |                                  |
| 21      | 7   | 0.38474022E 07                     | 0.67209696E 02                     | 0.35537726E 01                     | 0.52265083E 01                    |                                   |                      |                                  |
| 2       | 1   | 1                                  | -0.23931577E 10<br>0.              | -0.23463675E 10<br>0.26895355E 04  | -0.44902604E 10<br>0.             | -0.17388867E 08<br>0.52268645E 01 | 0.38069416E 09<br>0. | 0.27095852E 01<br>0.31264142E 07 |
| 1       | 2   | -0.18778176E 10<br>0.              | -0.18341485E 10<br>0.24502587E 04  | -0.34885065E 10<br>0.50000000E 00  | -0.26265162E 08<br>0.52268306E 01 | 0.33373193E 09<br>0.              | 0.27073992E 01<br>0. | 0.20671064E 07<br>0.             |
| 1       | 3   | -0.11574251E 10<br>-0.45982836E 02 | -0.11488164E 10<br>0.17018348E 04  | -0.22245737E 10<br>0.99999755E 00  | -0.12684503E 08<br>0.52267442E 01 | 0.11909564E 09<br>0.              | 0.27049672E 01<br>0. | 0.95834139E 06<br>0.             |
| 1       | 4   | -0.72166921E 09<br>-0.41458840E 02 | -0.72158350E 09<br>0.97621053E 03  | -0.14135468E 10<br>0.14999974E 01  | -0.10195242E 08<br>0.52266614E 01 | 0.47895798E 08<br>0.              | 0.27032357E 01<br>0. | 0.38928735E 06<br>0.             |
| 14      | 19  | 0.<br>0.                           | 0.<br>0.                           | 0.<br>0.90000000E 01               | 0.<br>0.54864995E 01              | 0.<br>0.                          | 0.27000000E 01<br>0. | 0.<br>0.                         |
| 14      | 20  | 0.<br>0.                           | 0.<br>0.                           | 0.<br>0.95000000E 01               | 0.<br>0.54864995E 01              | 0.<br>0.                          | 0.27000000E 01<br>0. | 0.<br>0.                         |
| 14      | 20  | 0.                                 | 0.                                 | 0.10000000E 02                     | 0.54864995E 01                    |                                   |                      |                                  |

END OF 01563-2

Figure 7 Example of Printed Output at Each Mesh

TR66-85

DEBRIS CLOUD TEST  
TIME=0.00992 USEC.



DEBRIS CLOUD TEST  
TIME=0.00992 USEC.

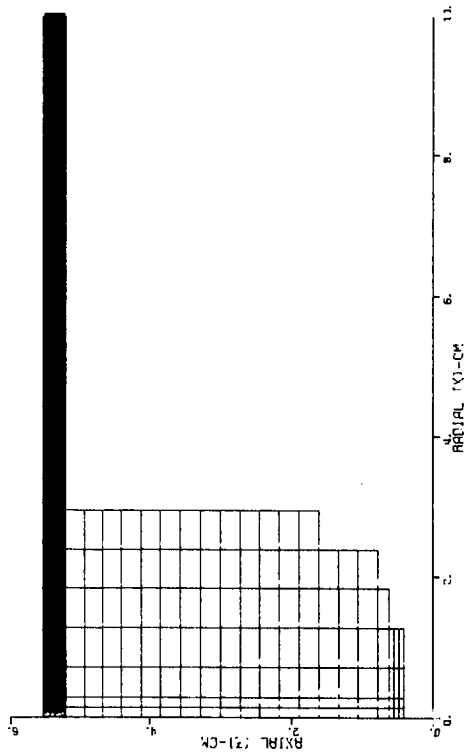


Figure 8 Example of Position Plot

TR66-85

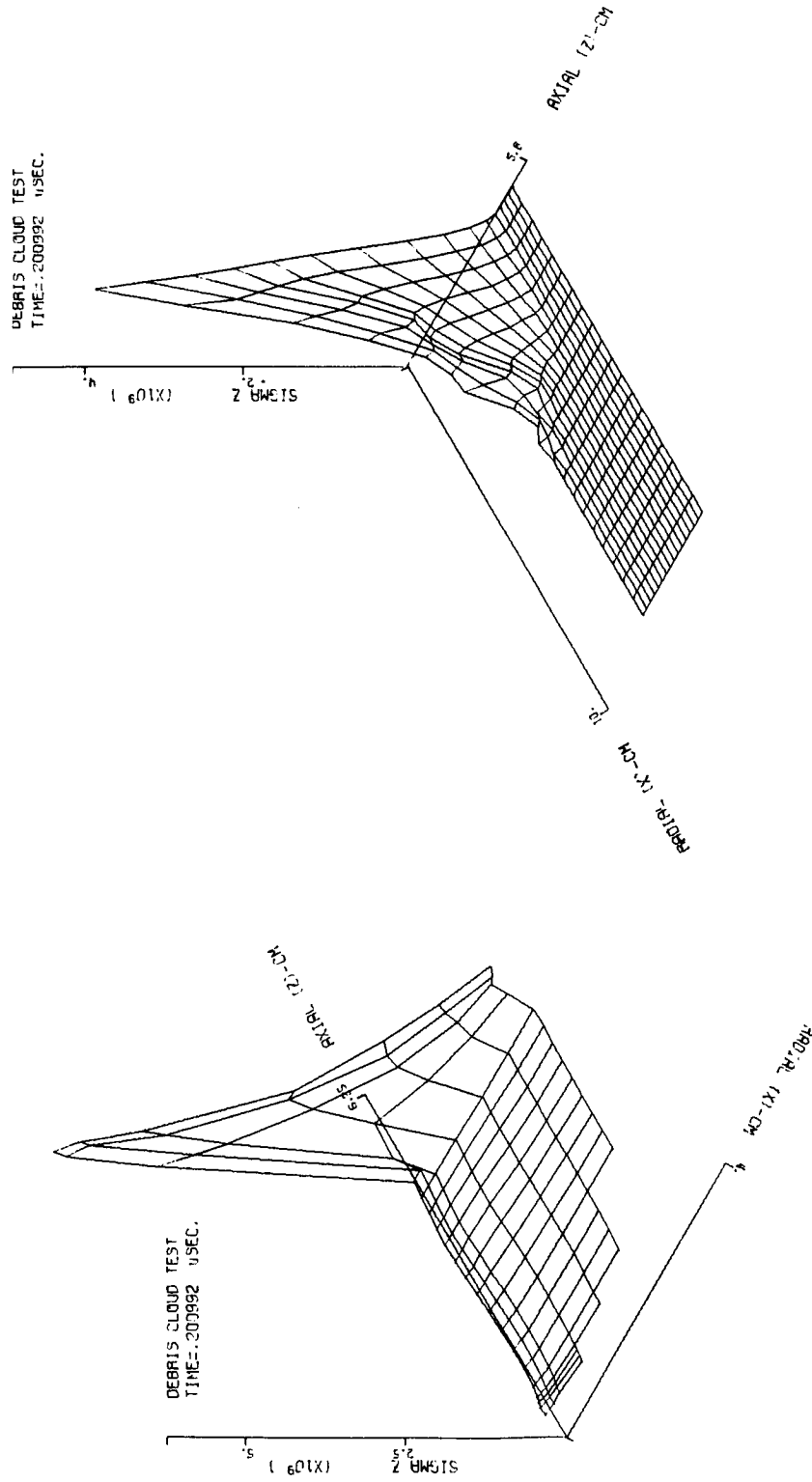


Figure 9 Example of Stress Plot

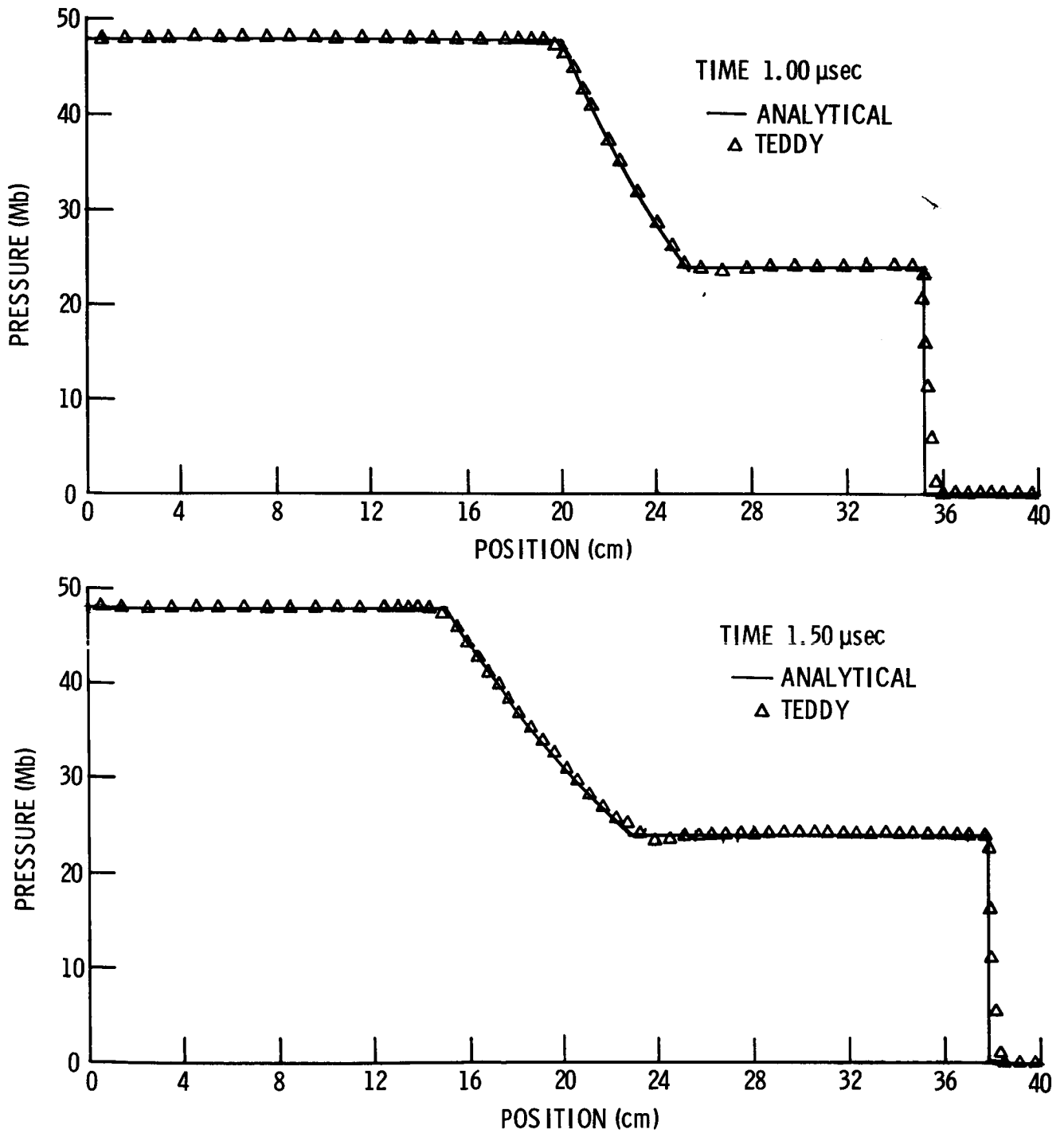


Figure 10 Example 1: Shock Tube Analysis



TR66-85

2. The second check solution is for an aluminum plate impacting a rigid wall at an impact velocity of 0.1262 km/sec. An elastic-plastic constitutive equation with  $\gamma = 2.76 kb$  is used. Again good agreement is obtained, as shown in Figure 11.
3. The third check problem involves the expansion of a spherical cavity within a sphere subject to time-dependent internal pressure. Exact solutions have been given by Blake.<sup>(8)</sup> Although this can be treated as a one-dimensional problem, the computer code treats it as an axially symmetric two-dimensional problem. Comparison of the analytical and computer solutions shows very good agreement; see Figure 12.

While these and other problems have given satisfactory answers, it should be emphasized that care must be exercised in using the program for other problems. Undetected errors may still exist in the program despite efforts to eliminate all of them. Also, other conditions may require a different choice of viscosity coefficients and stability criterion.

#### H. PROGRAM FLOW CHARTS

The flow charts for the program are presented in Appendix B. Not all the subroutines comprising the 2-D Lagrangian Code have been flow charted. Those that are very short and/or straight-forward in nature have been omitted. The source deck listings will explain fully these routines.

Also, in those routines that involve large amounts of "book-keeping" logic, an idealized approach has been taken in the flow charts so that the basic principles underlying these routines is made clear.

#### I. FORTRAN SOURCE DECKS

Appendix C is a listing of the Fortran Source Decks. In this appendix, the block of COMMON, DIMENSION, and EQUIVALENCE statements that is common to all the Fortran programs and subprograms, except three, is listed only once. It is to be remembered, though, that except for subroutines QUAD, PMU, and AX2, it is an integral part of all other programs.

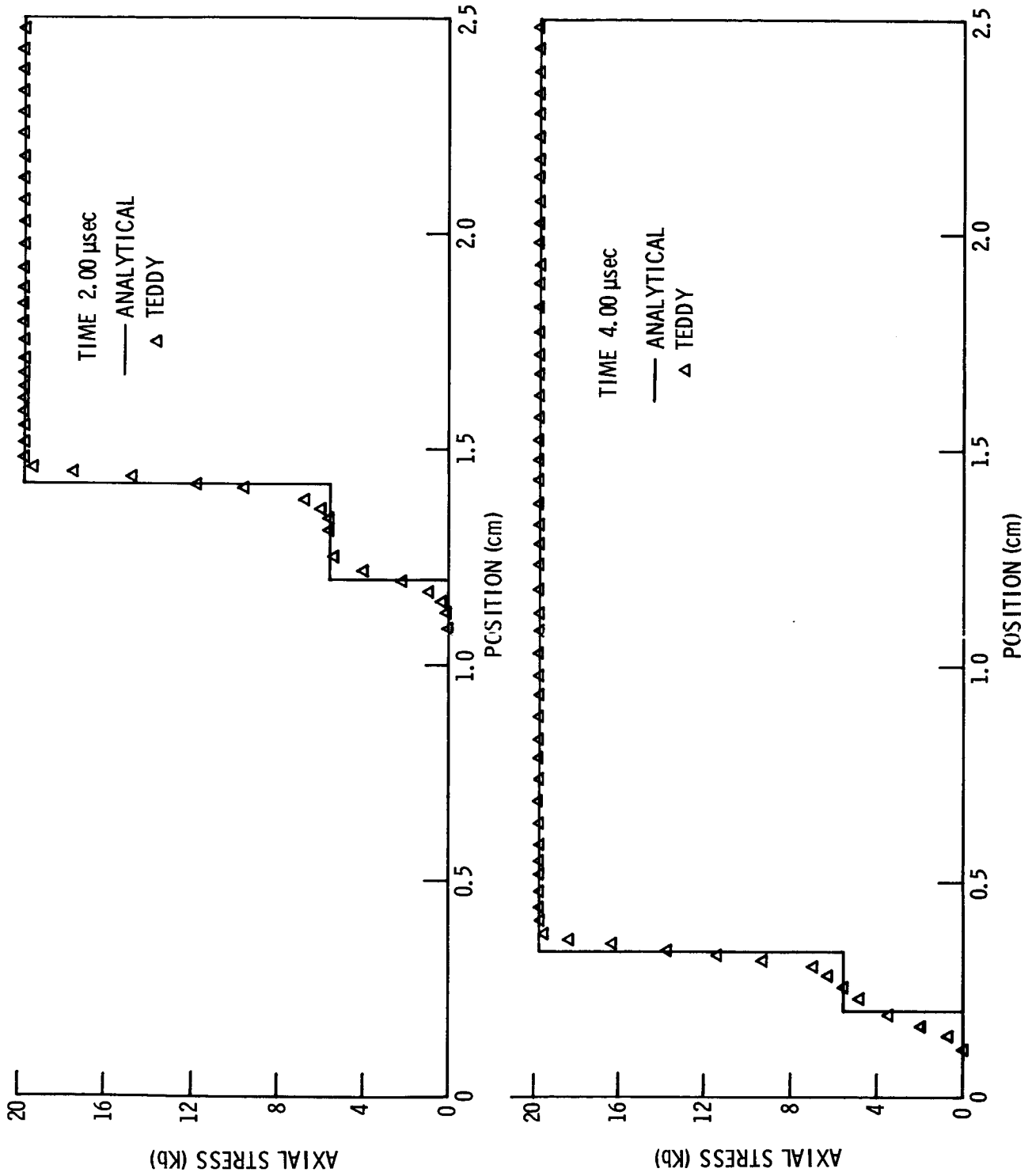
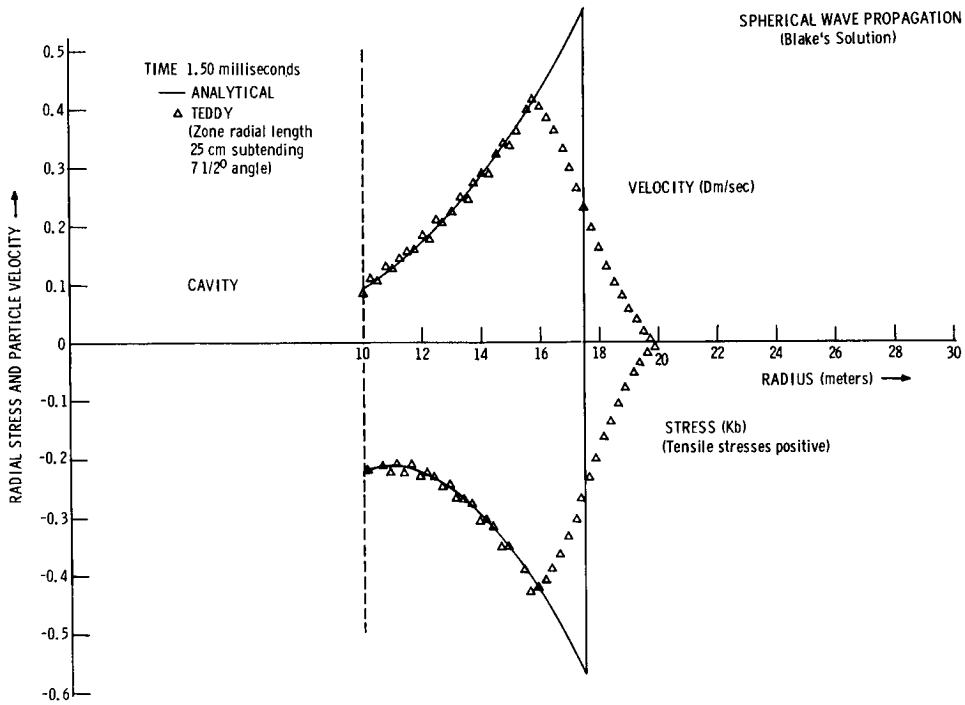
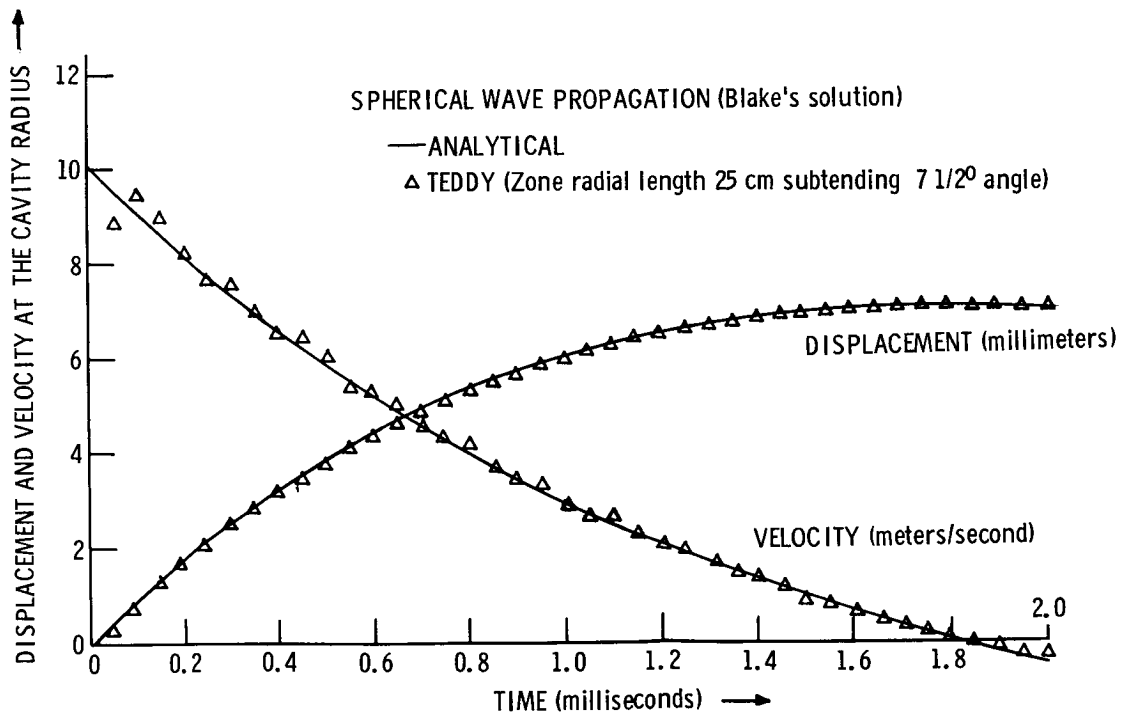


Figure 11 Example 2: Stress Wave Profiles due to the Elastic-Plastic Impact of an Aluminum Plate on a Rigid Wall

TR66-85



a. Velocity and Stress vs Radius



b. Velocity and Displacement vs Time

Figure 12 Example 3: Wave Propagation in an Elastic Medium Surrounding a Spherical Cavity Subjected to a Time-Varying Internal Pressure

## REFERENCES

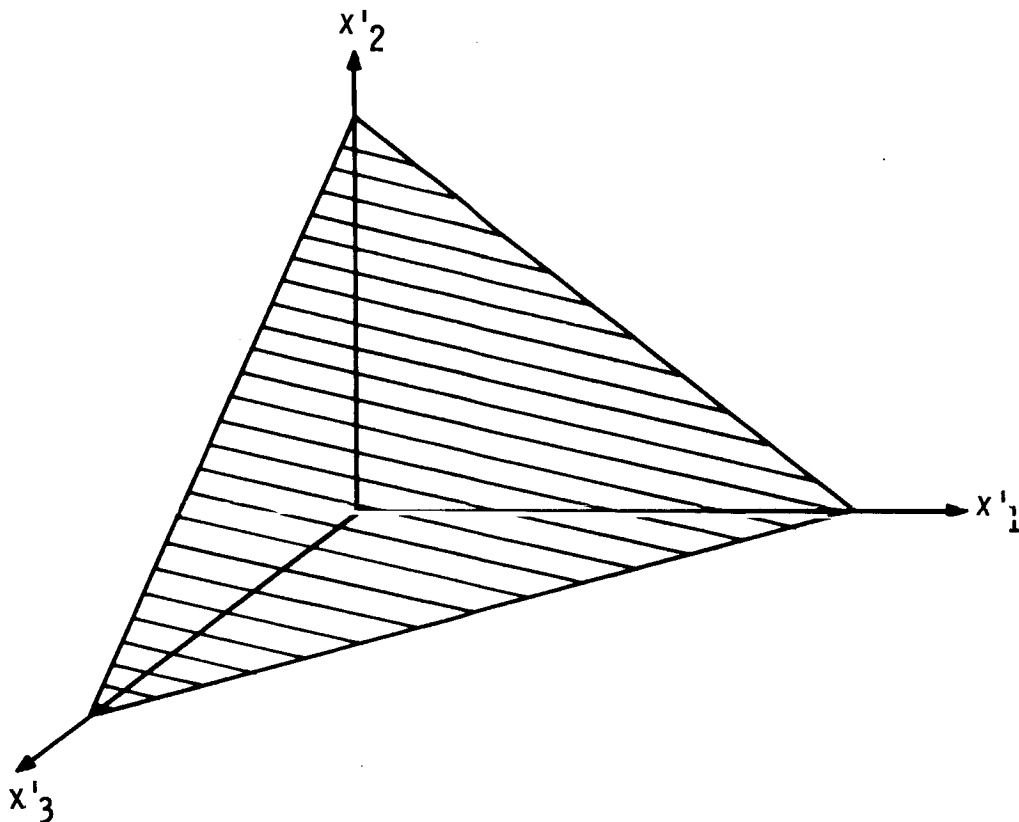
1. J.H. Tillotson, "Metallic Equations of State for Hypervelocity Impact," General Atomic Report GA-3216, July 18, 1962
2. G. Maenchen and S. Sack, "The Tensor Code," University of California, Lawrence Radiation Laboratory, UCRL-7316, April 9, 1963
3. M. L. Wilkins, "Calculation of Elastic-Plastic Flow," University of California, Lawrence Radiation Laboratory, UCRL-7322, April 19, 1963
4. C.S. Godfrey, D.J. Andrews, E.T. Trigg and M. Wilkins, "Prediction Calculations for Free Field Ground Motion," Air Force Weapons Laboratory Technical Documentary Report No. WL-TDR-64-27, May 1964
5. W. Herrmann and M. O'Brien, "RAVE I," Aeroelastic and Structures Research Laboratory, (Massachusetts Institute of Technology) Report No. 1021, June 1964
6. J. von Neumann and R.D. Richtmyer, "A Method for the Numerical Calculation of Hydrodynamic Shocks," J. Appl. Phys., Vol. 21, 1950, p. 232
7. M. L. Wilkins, Private correspondence
8. F.G. Blake, Jr., "Spherical Wave Propagation in Solid Media," J. of the Acoustical Society of America, Vol. 24, 1952, p. 211

## APPENDIXES

**Appendix A: Material Rotation**

**Appendix B: Program Flow Charts**

**Appendix C: Listings of Fortran Source Decks**

APPENDIX A  
MATERIAL ROTATION(1) Stresses at a Point

Consider the coordinate system whose axes lie in and normal to the shaded area above. Let  $x_1$   $x_2$   $x_3$  be the new coordinates with direction cosines  $a_{ij}$ . The stresses in the new coordinate system are given by

$$t^{ij} = a_{il} a_{jm} (t^{lm})'$$

TR66-85  
Appendix A

where  $(t^{lm})'$  are the stresses referred to the coordinate system  $x'_1 x'_2 x'_3$ .

Similarly

$$(t^{ij})' = a_{il} a_{jm} t^{lm}$$

## (2) Material Rotation

During a given time step an element of the body rotates which necessitates correcting the stresses to refer to the fixed coordinate system  $x_1 x_2 x_3$ . The system  $x'_1 x'_2 x'_3$  participates in the instantaneous rotation in the neighborhood of the considered particle P. The coordinate  $dx_i$  of this particle changes at the rate  $dv_i = \epsilon_{ijk} w_k dx'_j$  where  $w_k$  is the vorticity vector at P and  $\epsilon_{ijk}$  is such that

$$\epsilon_{ijk} = 0 \text{ when any two indices are equal}$$

$$\epsilon_{ijk} = 1 \text{ when } i, j, k \text{ are an even permutation of } 1, 2, 3$$

$$\epsilon_{ijk} = -1 \text{ when } i, j, k \text{ are an odd permutation of } 1, 2, 3.$$

At time  $t + dt$

$$dx_i = dx'_i - \epsilon_{ijk} w_k dt dx'_j$$

$$= (\delta_{ij} - \epsilon_{ijk} w_k dt) dx'_j$$

in which case the coefficients in the transformation are

$$a_{ij} = \delta_{ij} - \epsilon_{ijk} w_k dt$$

During the same time step the improved component of stress has increased to

$$t^{ij} + \dot{t}^{ij} dt$$

where  $t^{ij}$  and  $\dot{t}^{ij}$  are taken at particle P at time  $t$ .

Transforming

$$(t^{ij})' = a_{ip} a_{jq} (t^{pq} + \dot{t}^{pq} dt)$$

$$= (\delta_{ip} - \epsilon_{ipk} w_k dt)(\delta_{jq} - \epsilon_{jq\rho} w_\rho dt)(t^{pq} + \dot{t}^{pq} dt)$$

$$= t^{ij} + (\dot{t}^{ij} - t^{iq} \epsilon_{qj\rho} w_\rho - t^{jp} \epsilon_{pik} w_k) dt$$

to within higher-order terms in  $dt$ . The contents in brackets represent a suitable definition of the objective stress rate tensor  $\dot{t}^{ij}$

$$\dot{t}^{ij} = \dot{t}^{ij} - t^{iq} \epsilon_{qj\rho} w_{\rho} - t^{jp} \epsilon_{pik} w_k$$

Examples:

(1) For one dimensional motion Equations 6 reduces to

$$\dot{t}^{ii} = \dot{t}^{ii}$$

(2) For an axial-symmetric motion, using cylindrical coordinates (z axial and x radial coordinate)

$$\dot{t}^{xx} = \dot{t}^{xx} - 2w_{xz} t^{xz}$$

$$\dot{t}^{yy} = \dot{t}^{yy}$$

$$\dot{t}^{zz} = \dot{t}^{zz} + 2w_{xz} t^{xz}$$

$$\dot{t}^{xz} = \dot{t}^{xz} + w_{xz} (t^{xx} - t^{zz})$$

where  $w_{xz}$  is the spin given by

$$w_{xz} = \frac{1}{2} \left( \frac{\partial u^x}{\partial z} - \frac{\partial u^z}{\partial x} \right)$$

and  $u$  are the particle velocities in the direction indicated by the superscript.

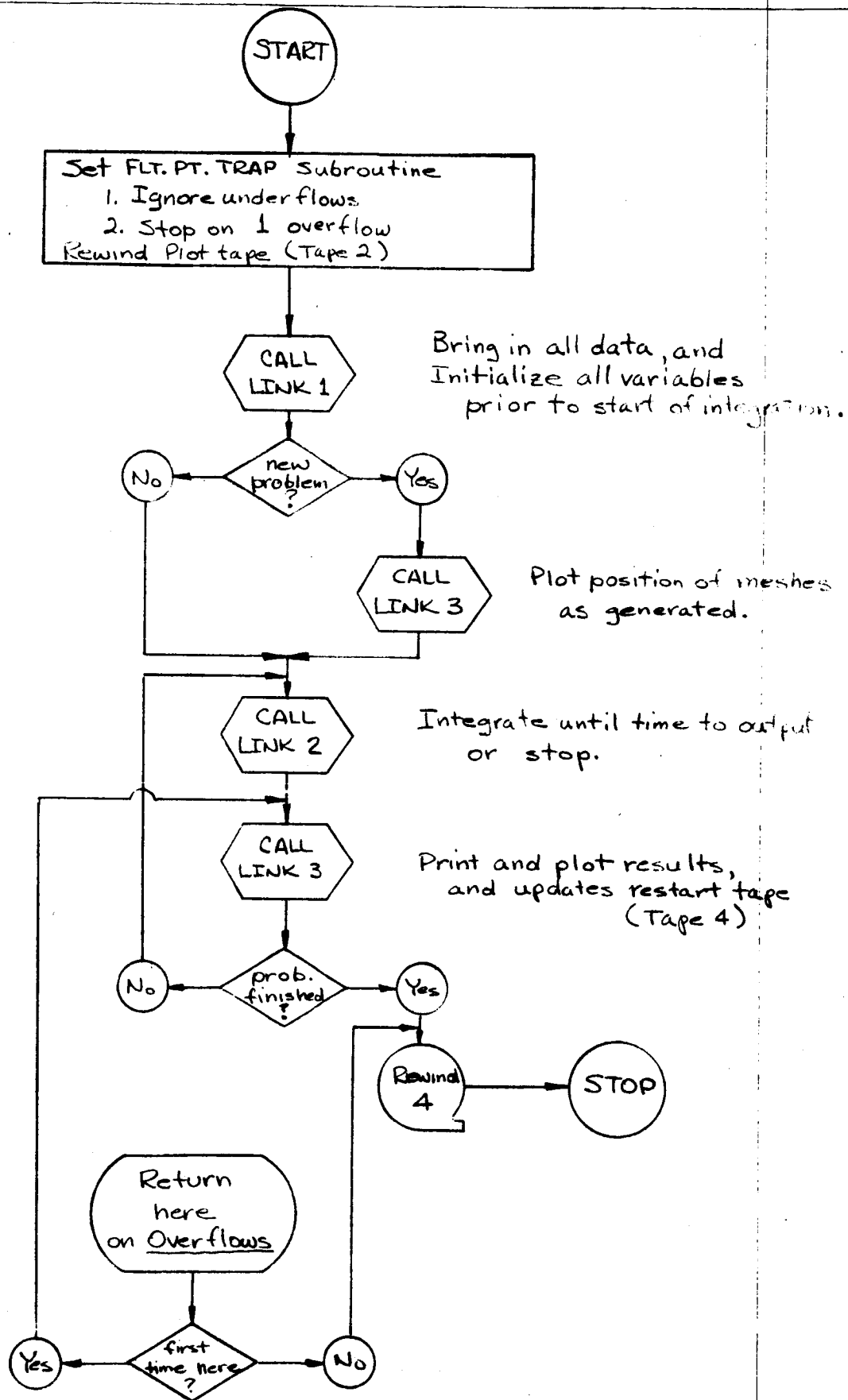


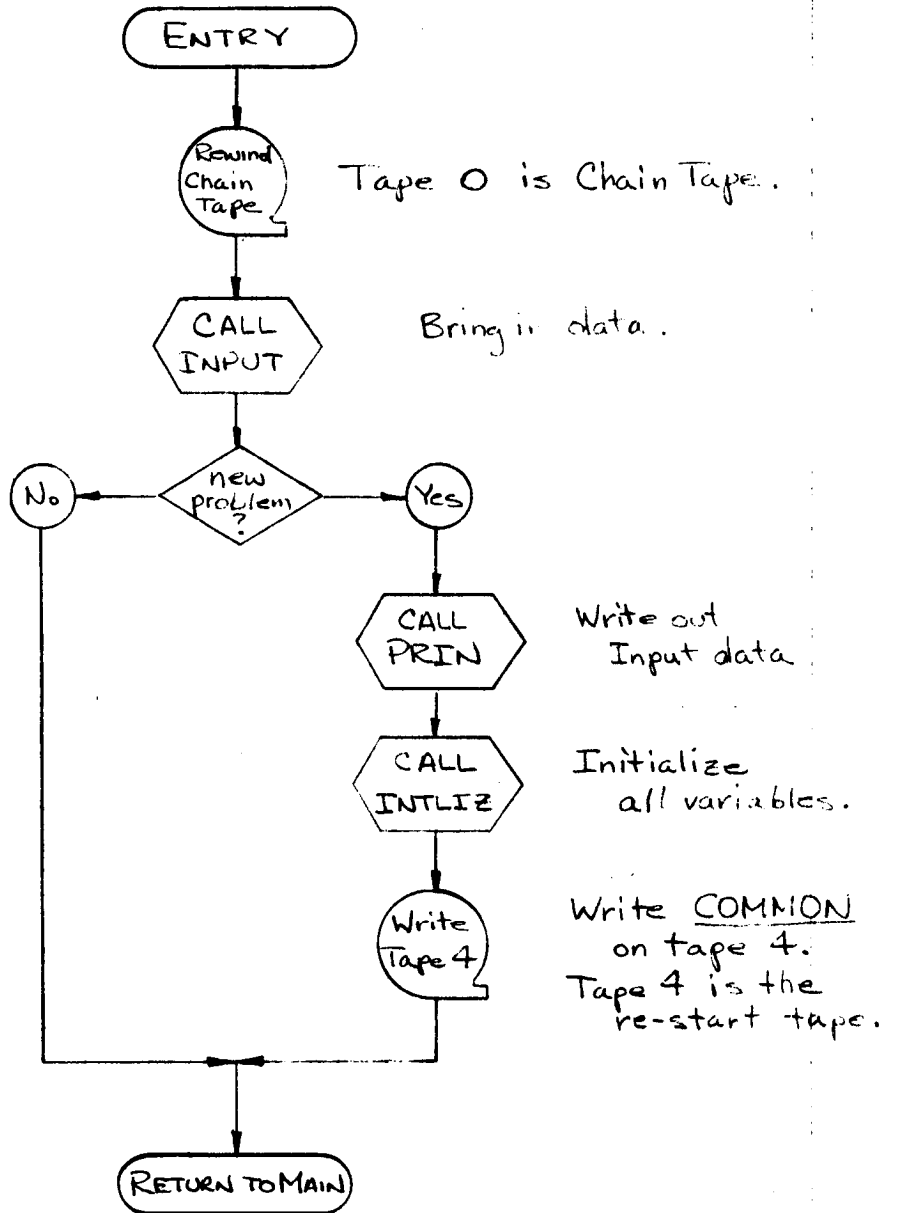
## APPENDIX B PROGRAM FLOW CHARTS

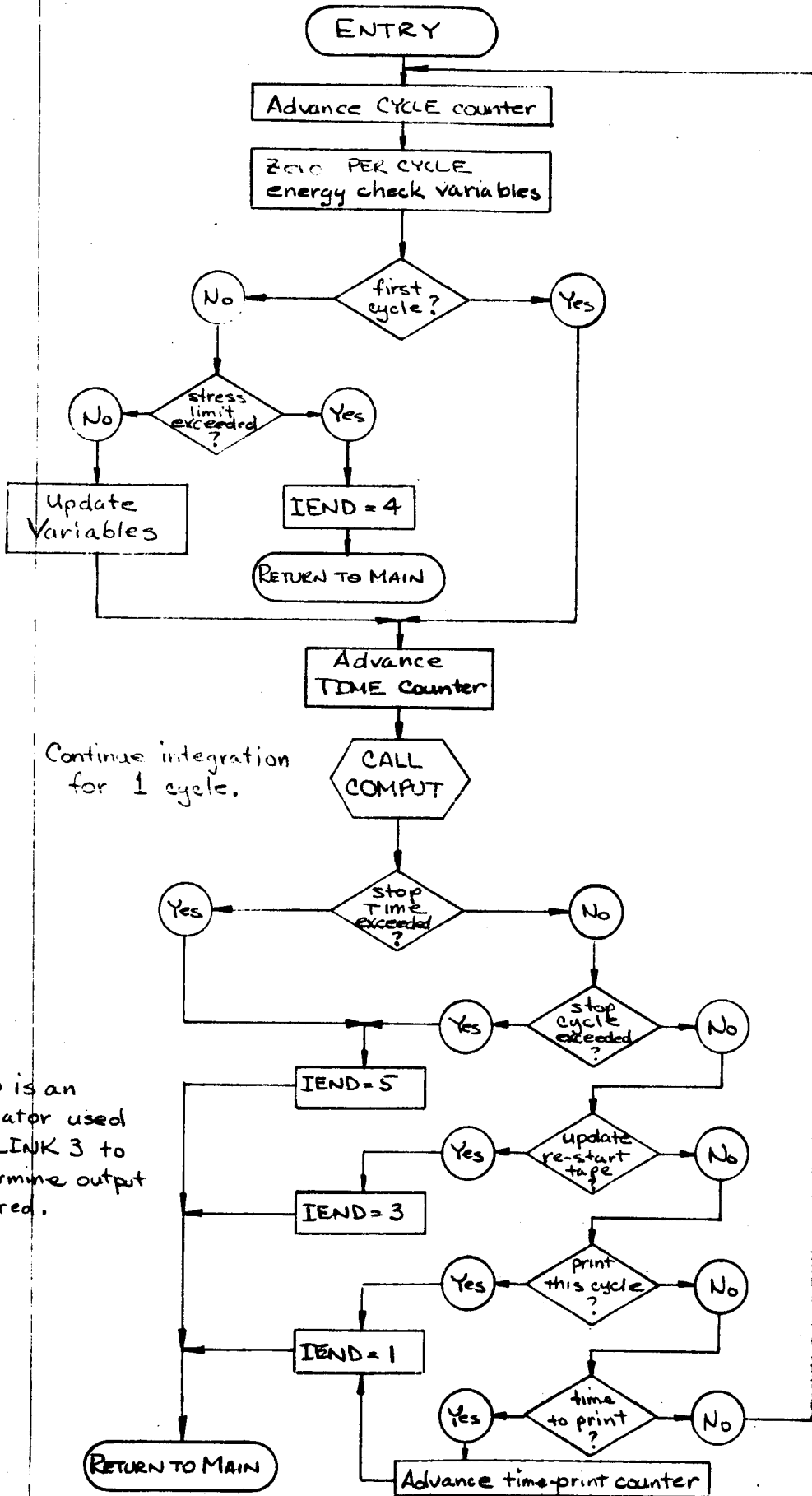
Flow Charts for the TEDDY Code Subroutines are presented in this Appendix as follows:

| <u>ITEM</u>         | <u>PAGE NO.</u> |
|---------------------|-----------------|
| MAIN PROGRAM        | B-1             |
| LINK 1              | B-2             |
| LINK 2              | B-3             |
| LINK 3              | B-4             |
| Subroutine COMPUT   | B-5             |
| Subroutine GETPOS   | B-6             |
| Subroutine GETCN    | B-7             |
| Subroutine COMPDT   | B-8             |
| Subroutine FINDP    | B-9             |
| Subroutine FINDP(2) | B-10            |
| Subroutine FINDP(3) | B-11            |
| Subroutine FINDP(4) | B-12            |
| Subroutine EVALQ    | B-13            |
| Subroutine SDST     | B-14            |
| Subroutine SDST(2)  | B-15            |
| Subroutine INTLIZ   | B-16            |
| Subroutine GNRT     | B-17            |
| Subroutine AMZRO    | B-18            |
| Subroutine CMPTIF   | B-19            |
| Subroutine FDUPTS   | B-20            |
| Subroutine AIFACE   | B-21            |
| Subroutine FDLPTS   | B-22            |

MAIN PROGRAM

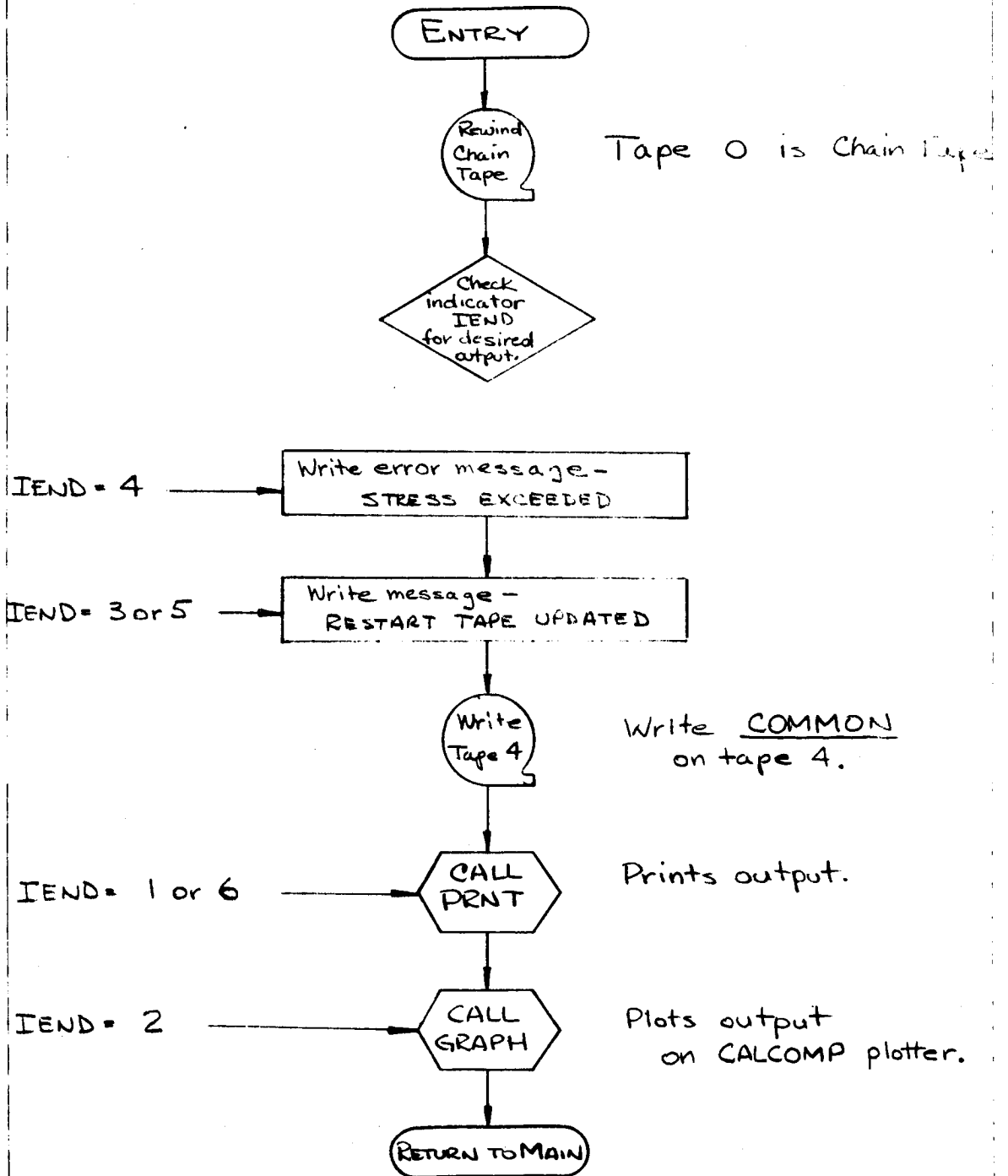




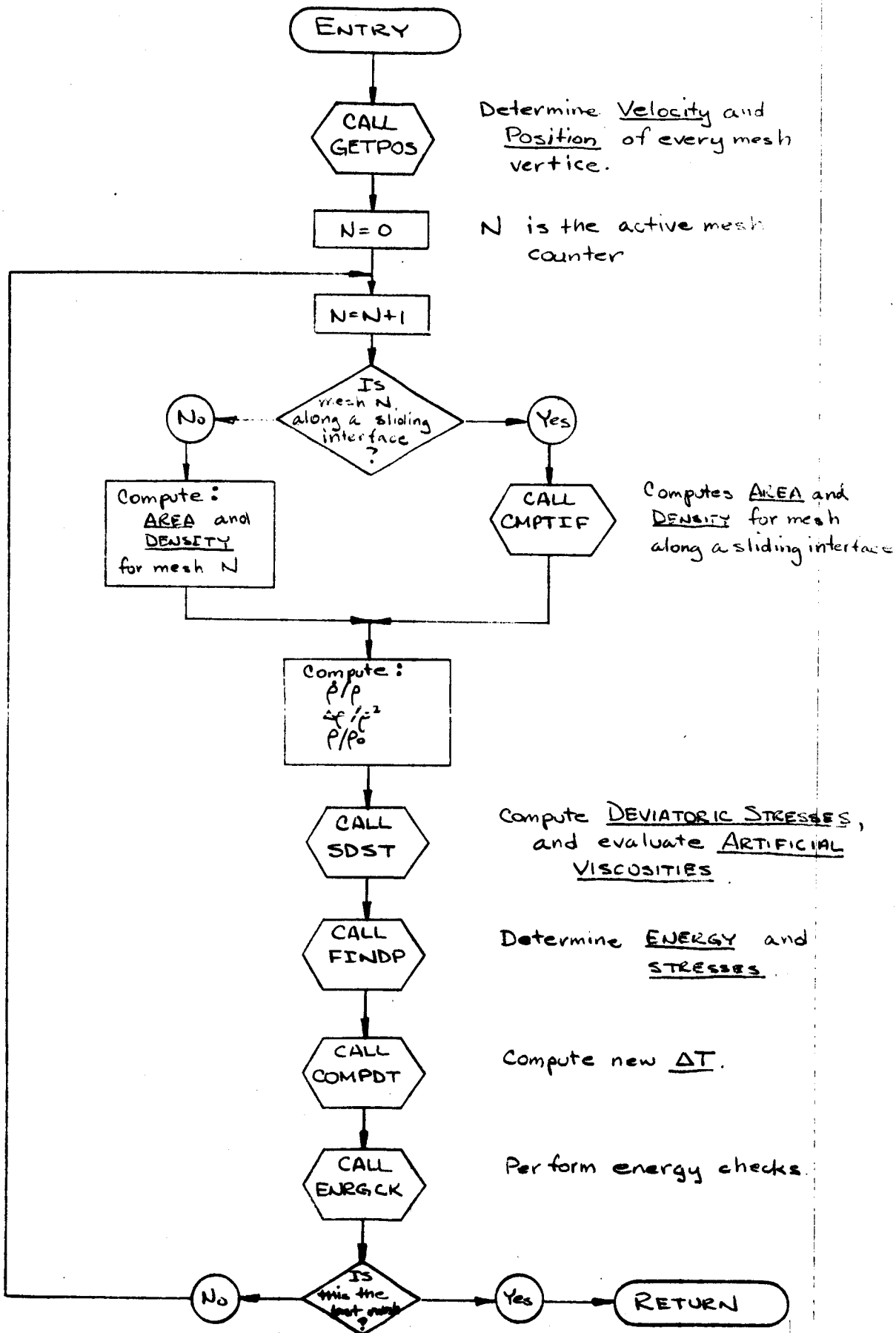


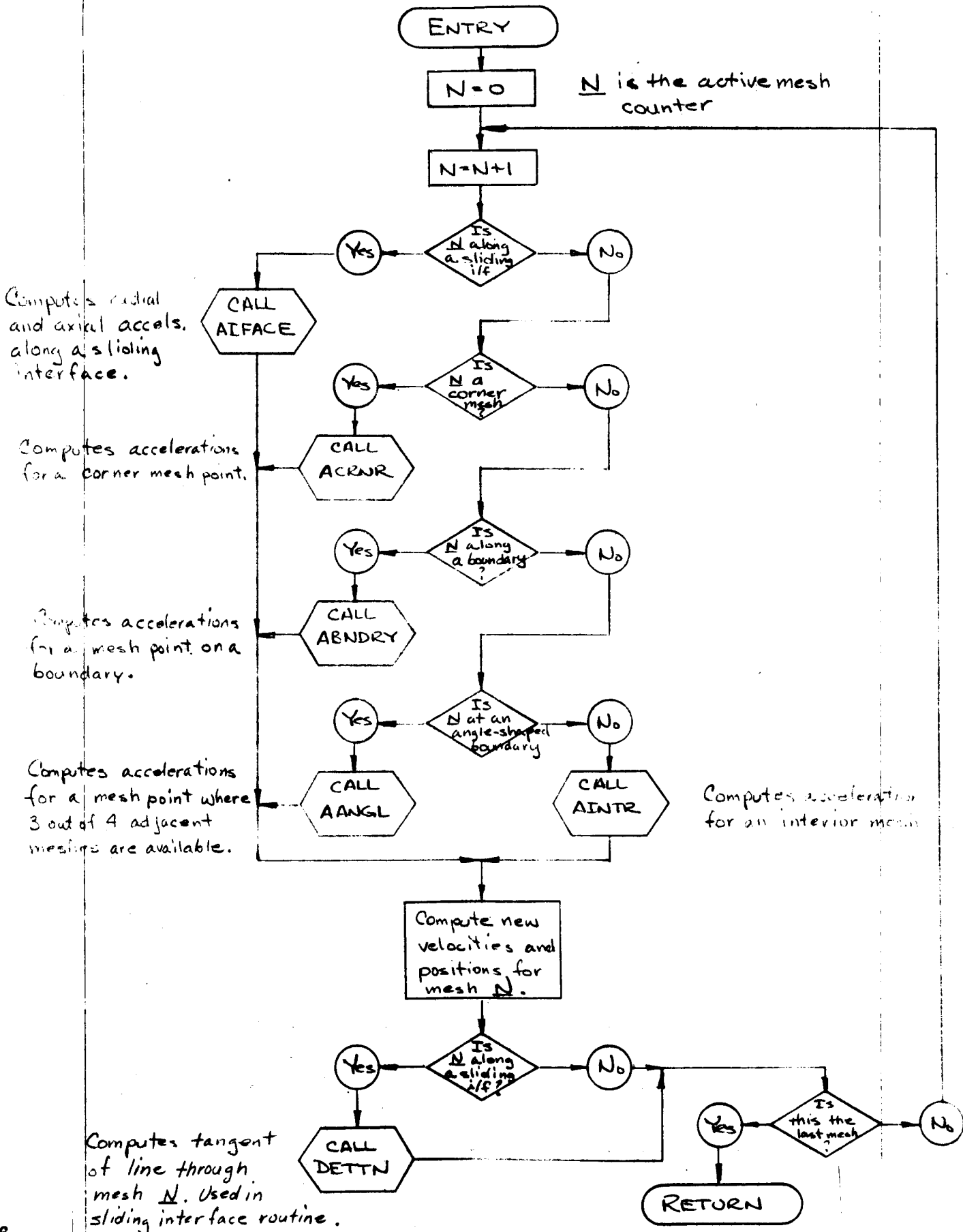
Continue integration for 1 cycle.

IEND is an indicator used by LINK 3 to determine output desired.

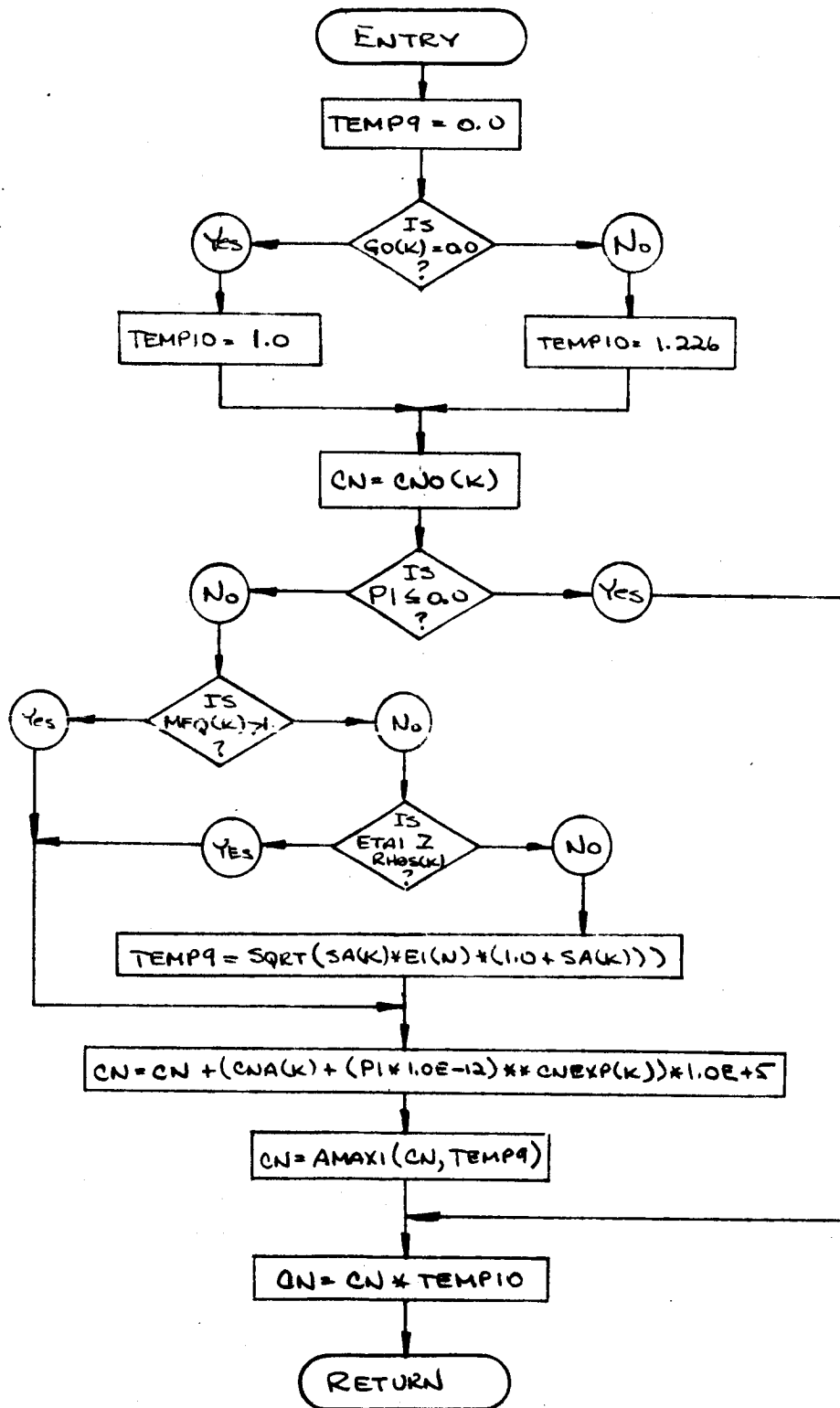


SUBROUTINE COMPUT

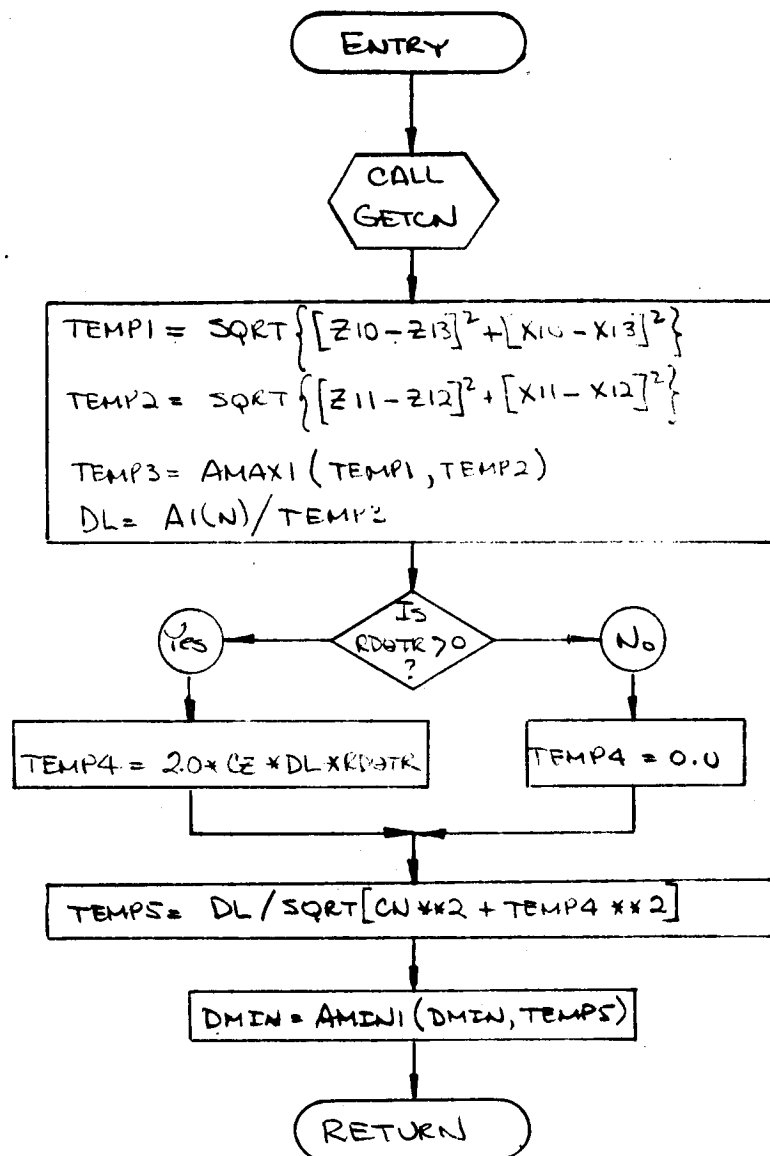




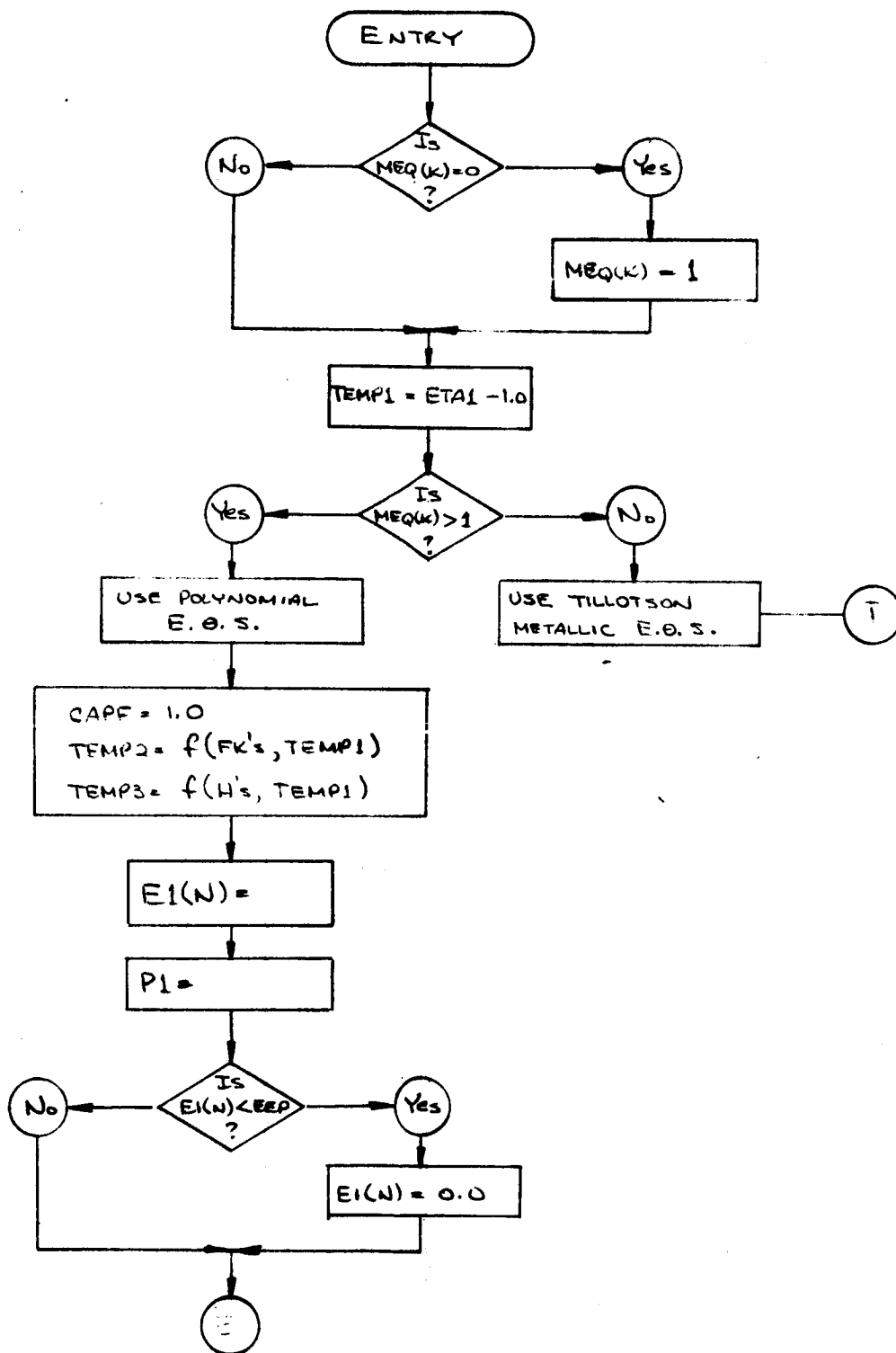
SUBROUTINE GETCN

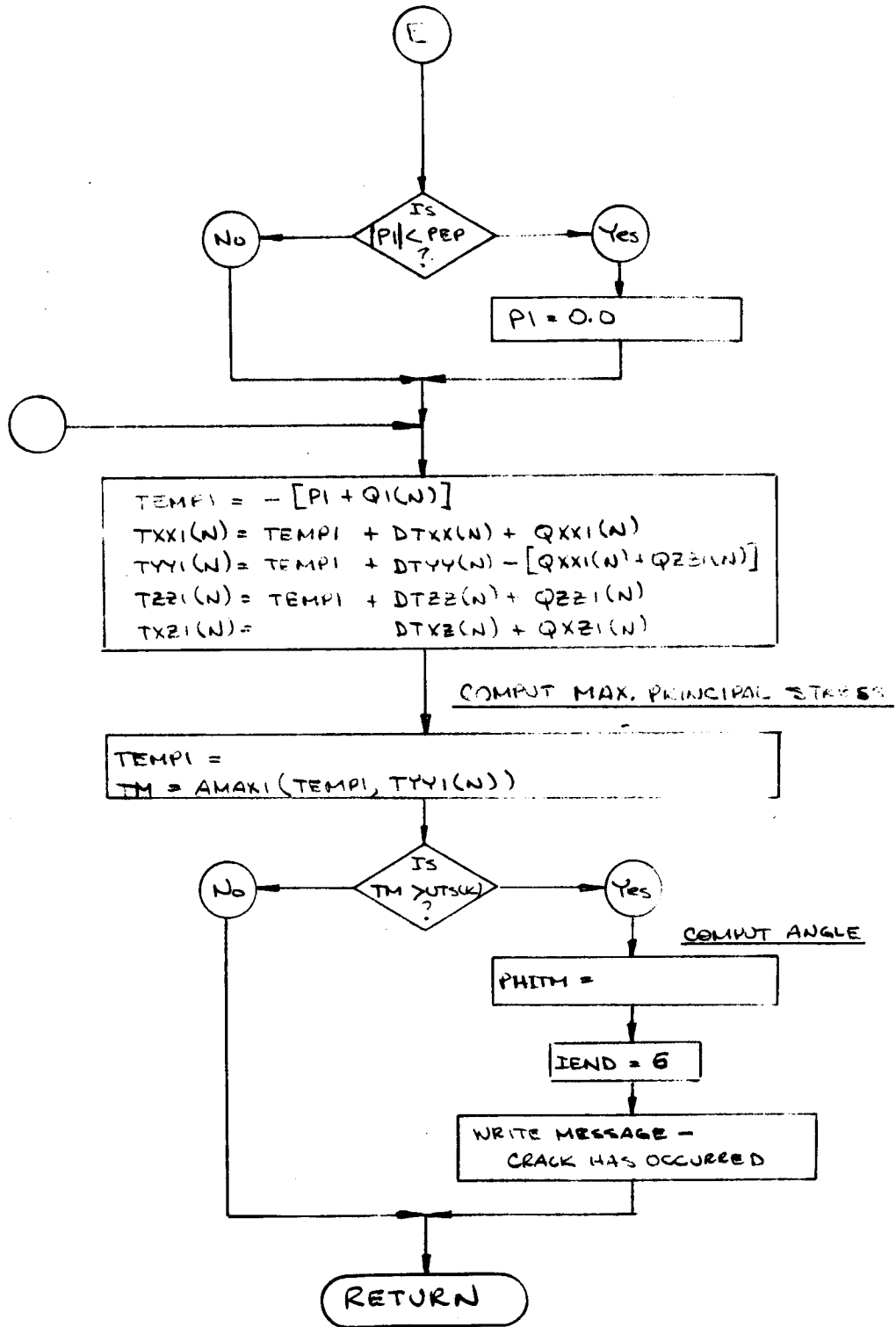


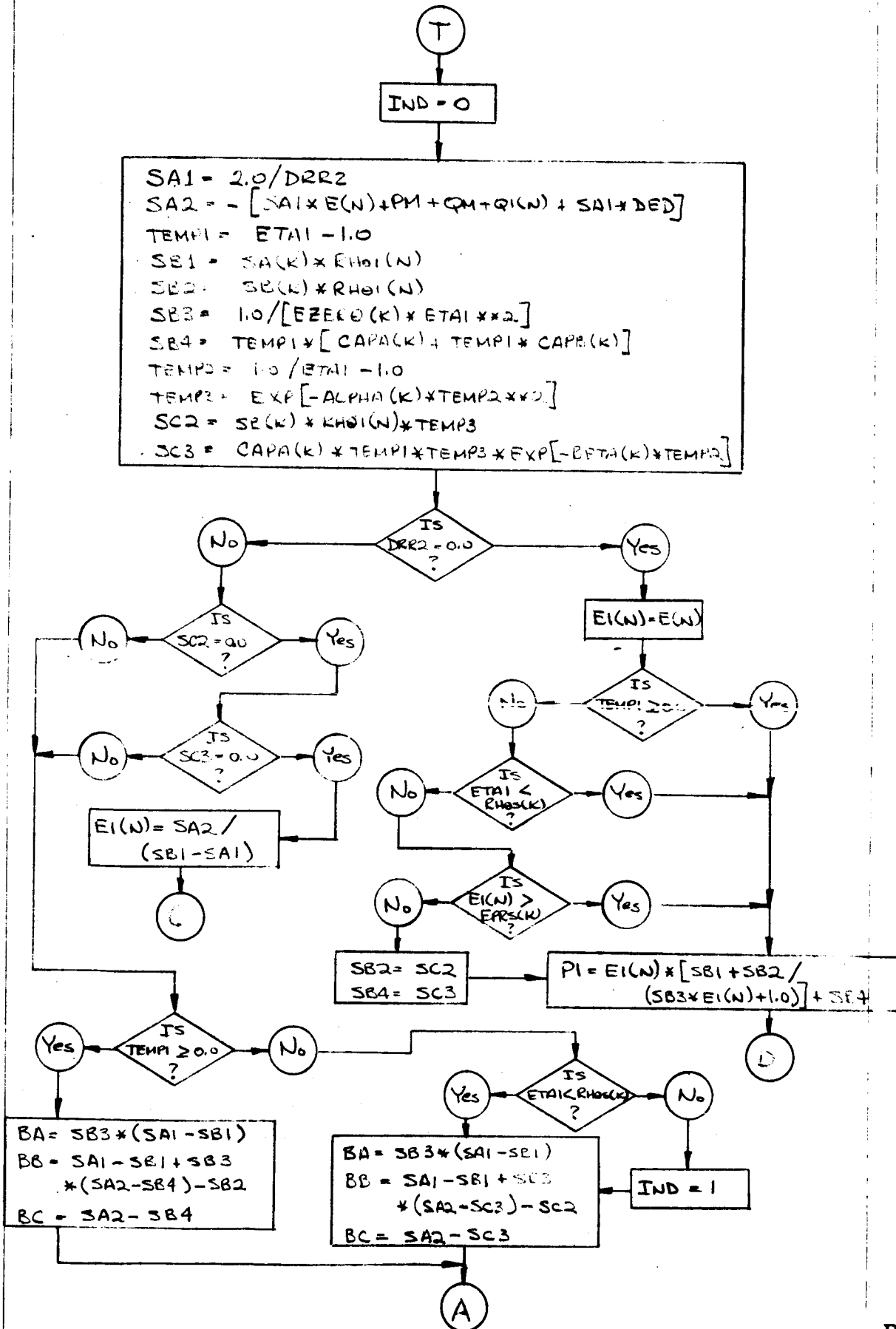


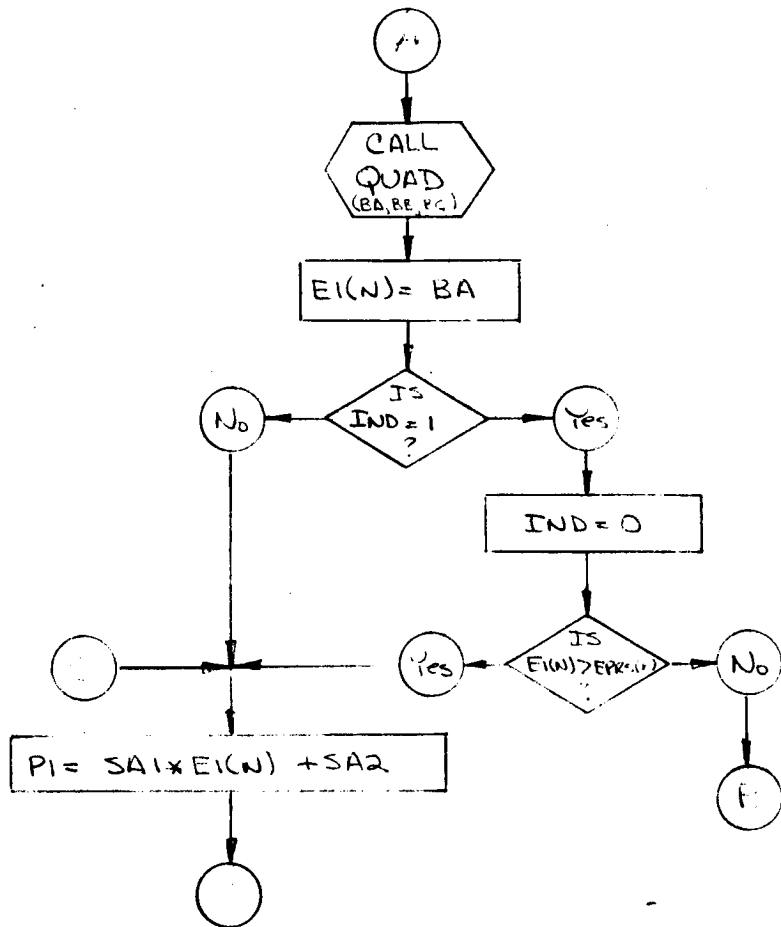


SUBROUTINE FINDP

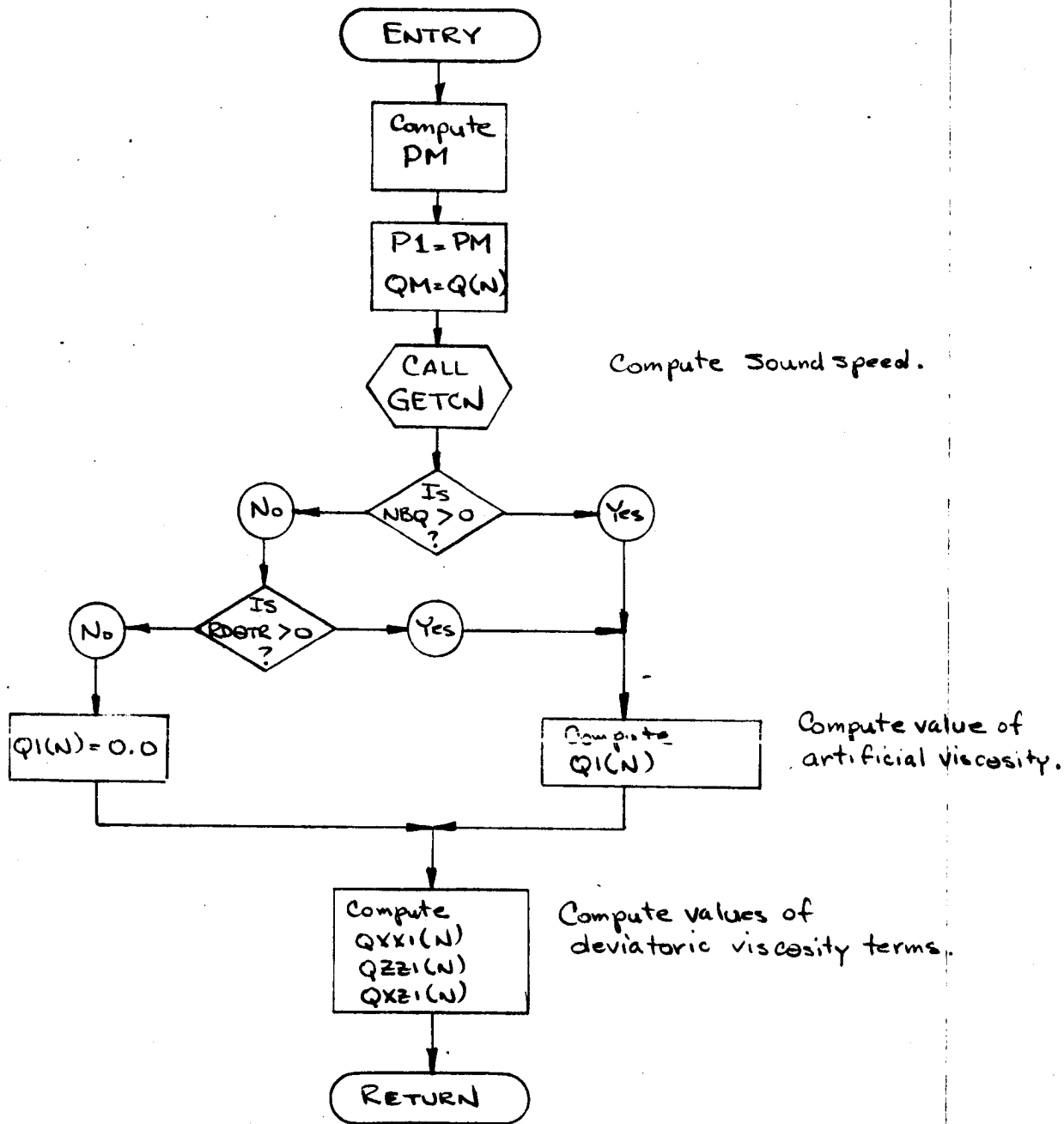








SUBROUTINE EVALQ



ENTRY

$TEMP1 = RD\theta TR / 3.0$   
 $TEMP2 = 2.0 * [A1(N) + AM]$   
 $TEMP3 = UX10 - UX11$   
 $TEMP4 = UX12 - UX11$   
 $TEMP5 = UZ10 - UZ11$   
 $TEMP6 = UZ12 - UZ11$   
 $TEMP7 = Z12 + Z12M - Z11 - Z11M$   
 $TEMP8 = Z10 + Z10M - Z11 - Z11M$   
 $TEMP9 = X12 + X12M - X11 - X11M$   
 $TEMP10 = X10 + X10M - X11 - X11M$

$$DXX = - (TEMP3 * TEMP7 - TEMP4 * TEMP5) / TEMP2 + TEMP1$$

$$DZZ = (TEMP5 * TEMP7 - TEMP6 * TEMP5) / TEMP2 + TEMP1$$

$$DYY = - (DXX + DZZ)$$

$$DXZ = (TEMP3 * TEMP9 - TEMP4 * TEMP10 - TEMP5 * TEMP7 + TEMP6 * TEMP5) / (2.0 * TEMP2)$$

$$WXZ = (TEMP3 * TEMP9 - TEMP4 * TEMP10 + TEMP5 * TEMP7 - TEMP6 * TEMP5) / (2.0 * TEMP2)$$

$$TEMP1 = 0.5 * [RH\theta 1(N) + RH\theta M] / RH\theta 2(K) - 1.0$$

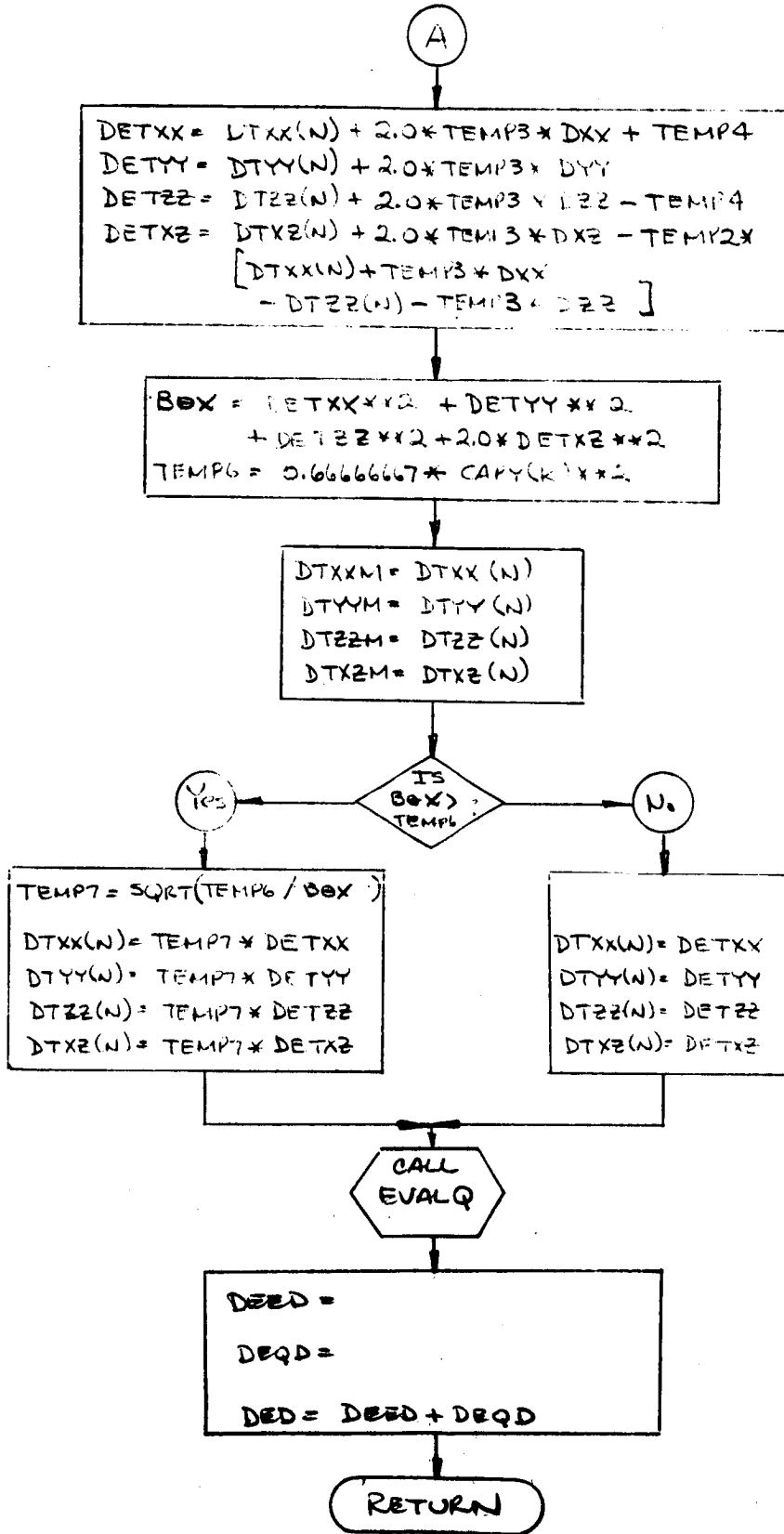
$$G = G0(K) * [1.0 + G1(K) * TEMP1 + G2(K) * TEMP1 * 2 + G3(K) * TEMP1 * 3]$$

$$TEMP2 = DELTA * WXZ$$

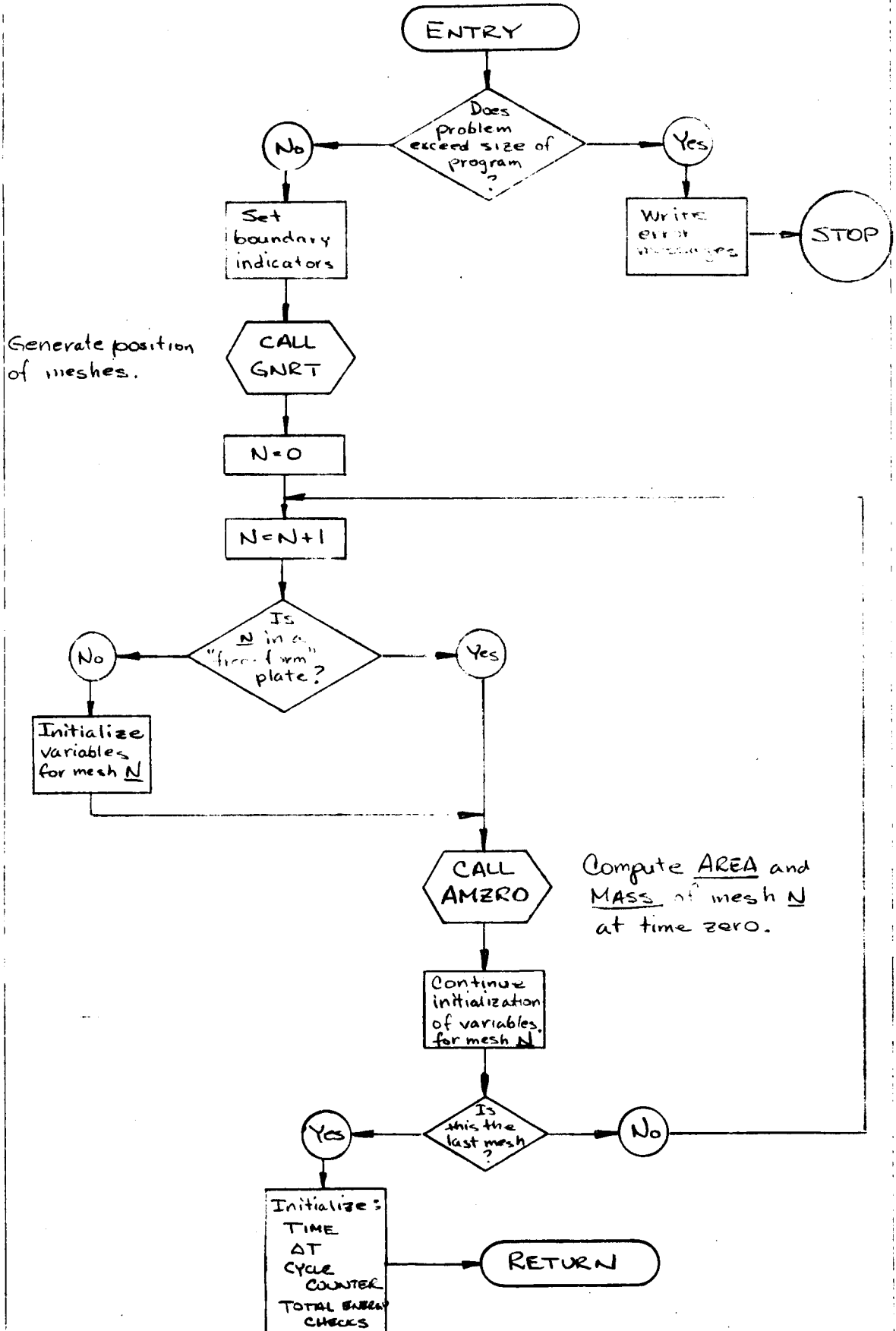
$$TEMP3 = G * DELTA$$

$$TEMP4 = 2.0 * TEMP2 * [DTXZ(N) + TEMP3 * DXZ]$$

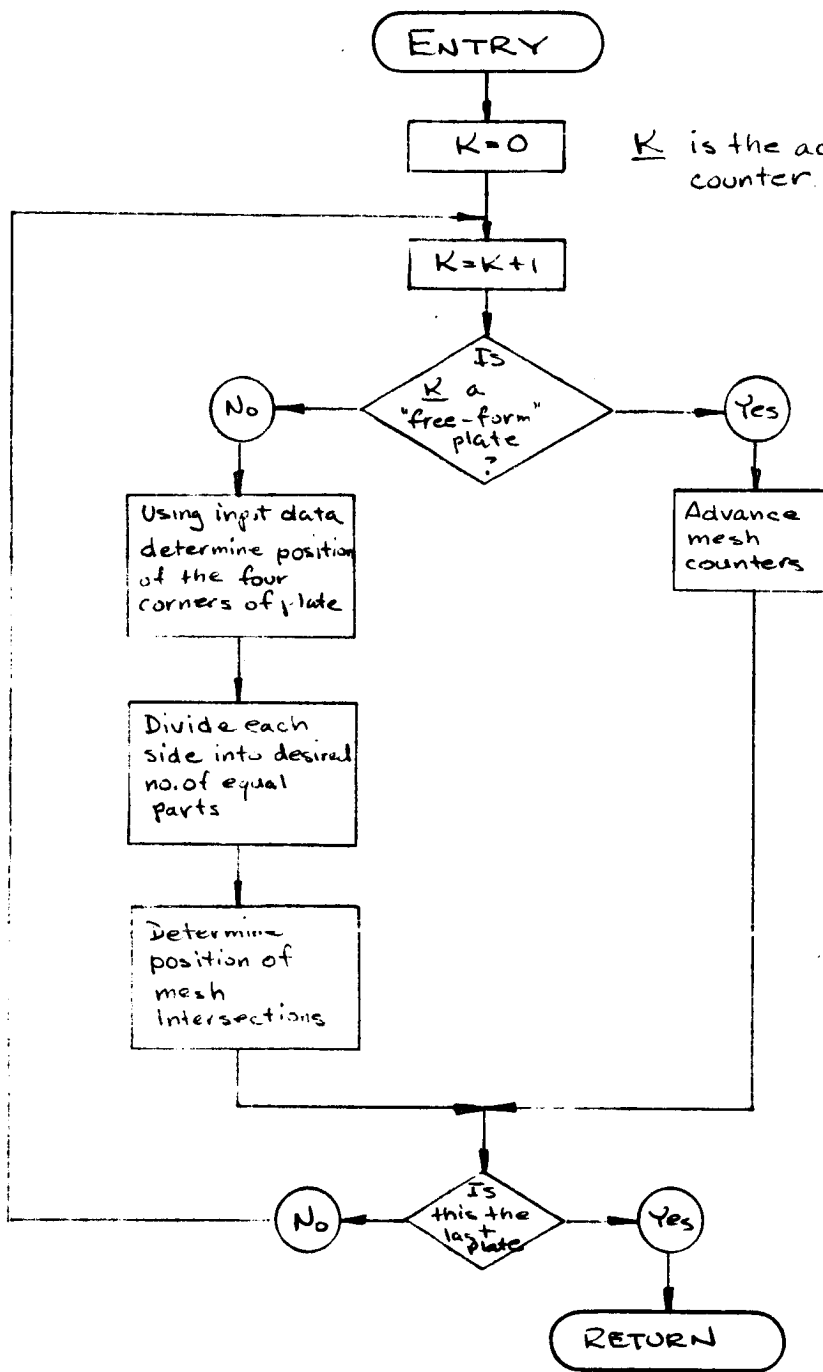
A





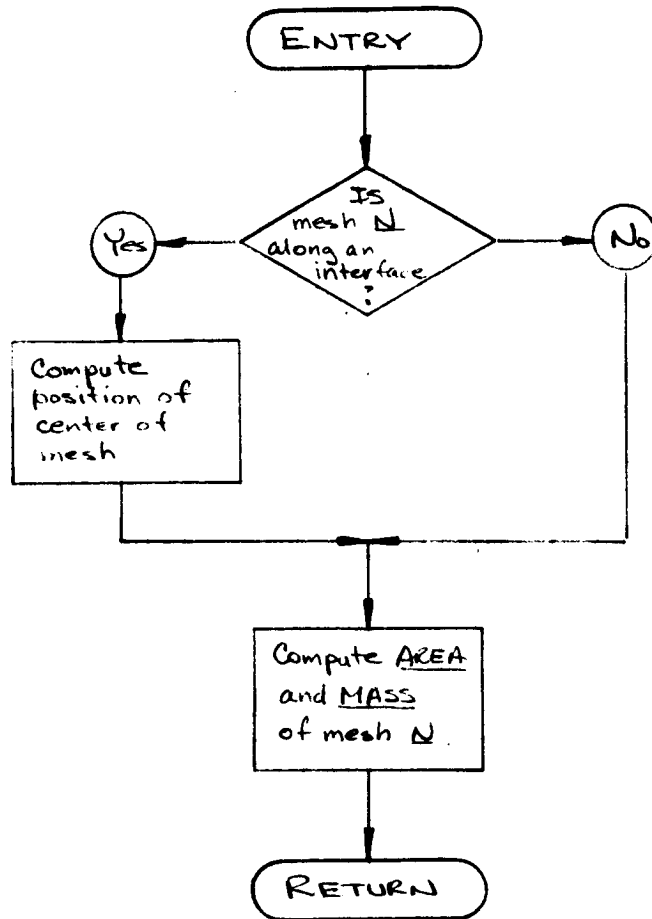


SUBROUTINE GNRT

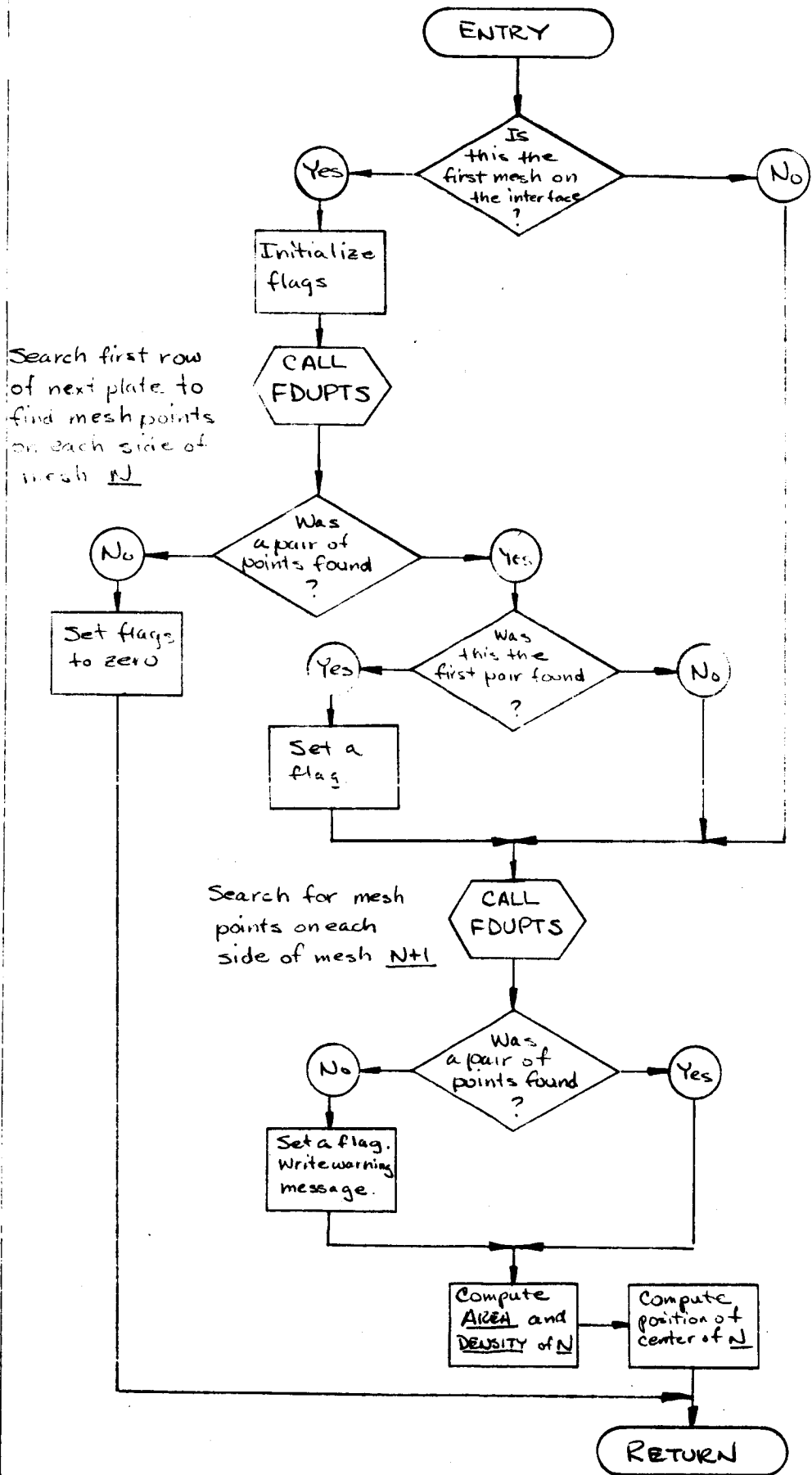


$K$  is the active plate counter.

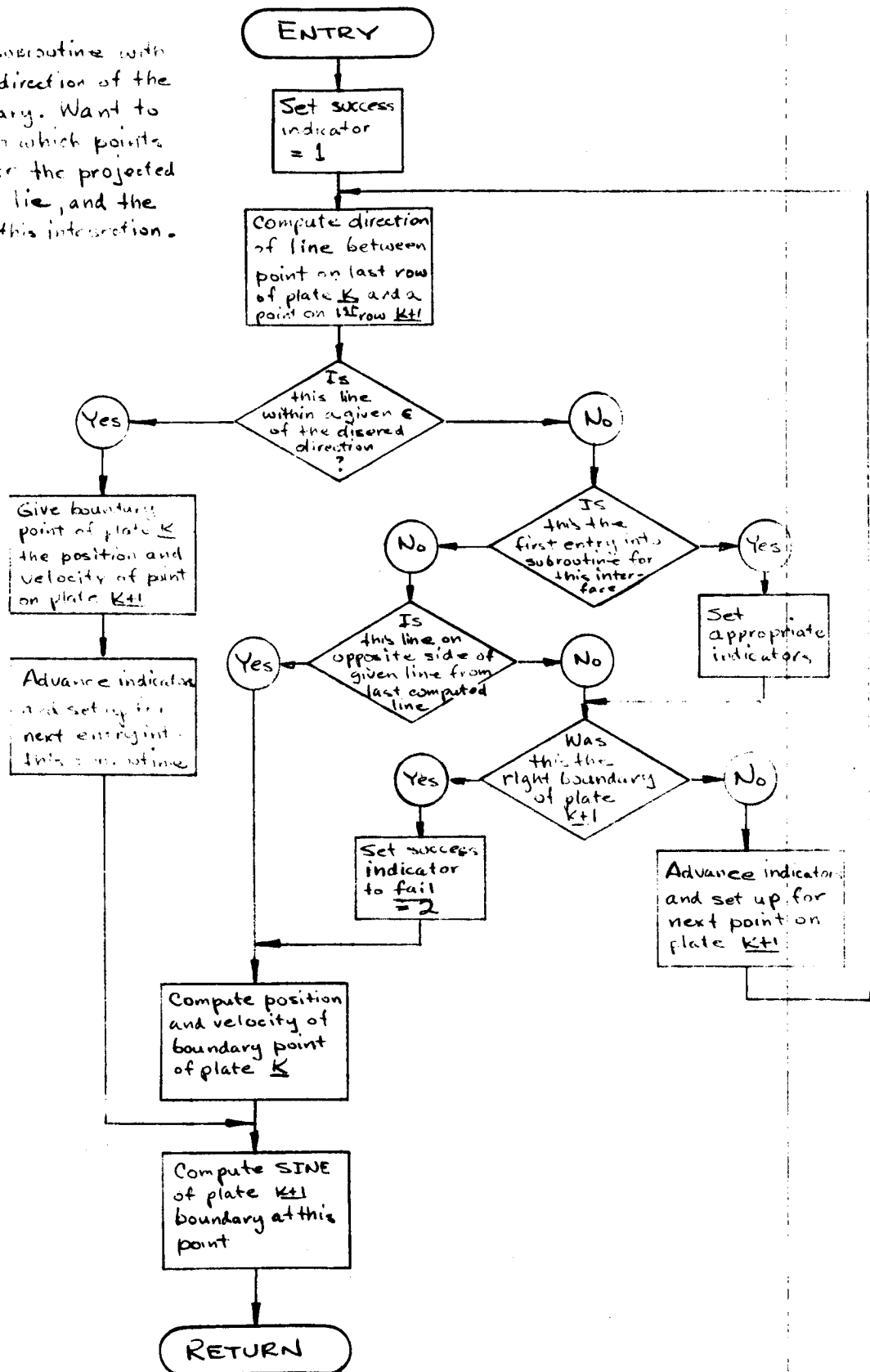
In a "free-form" plate, the positions of each mesh are given as input.



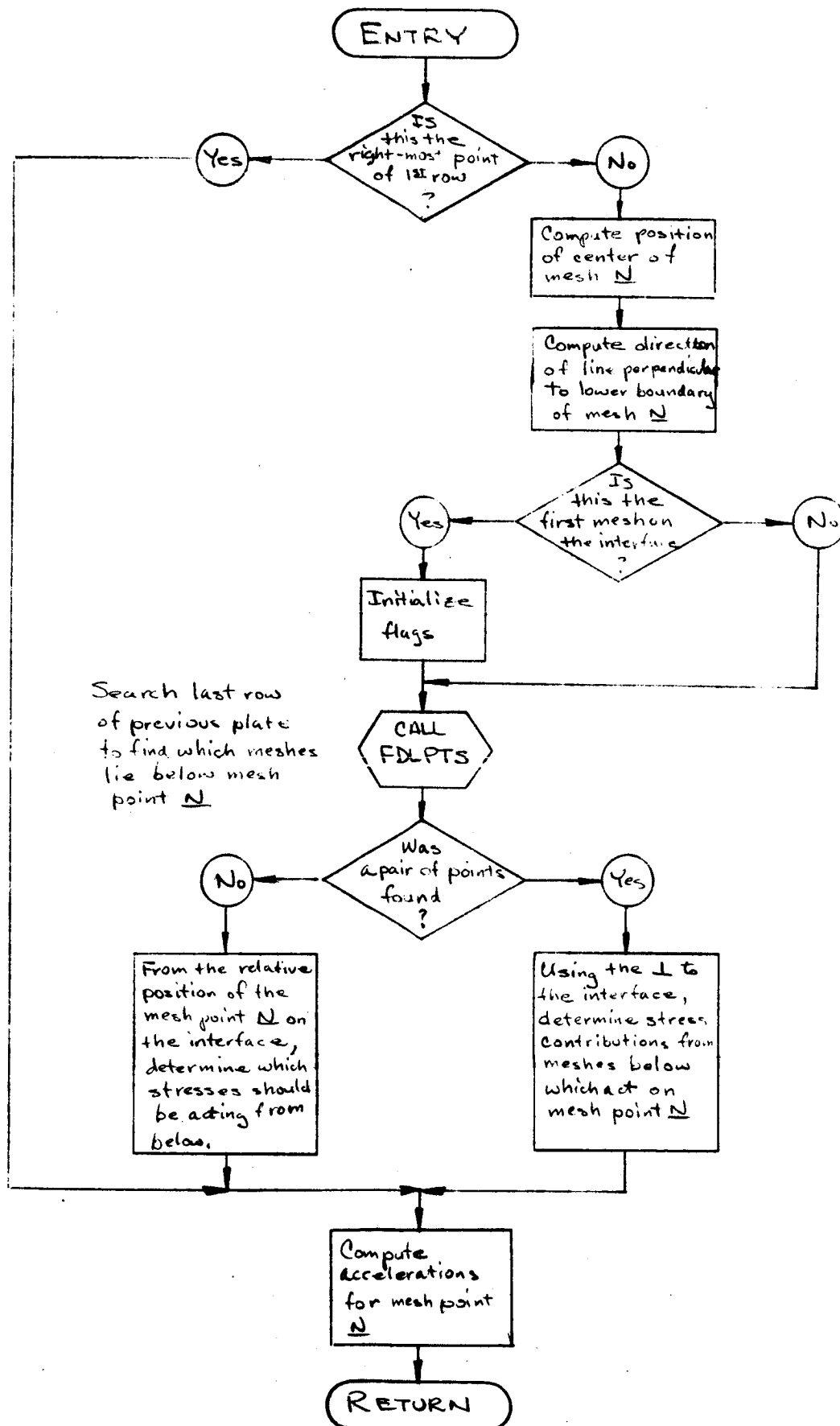
SUBROUTINE UMPTIF



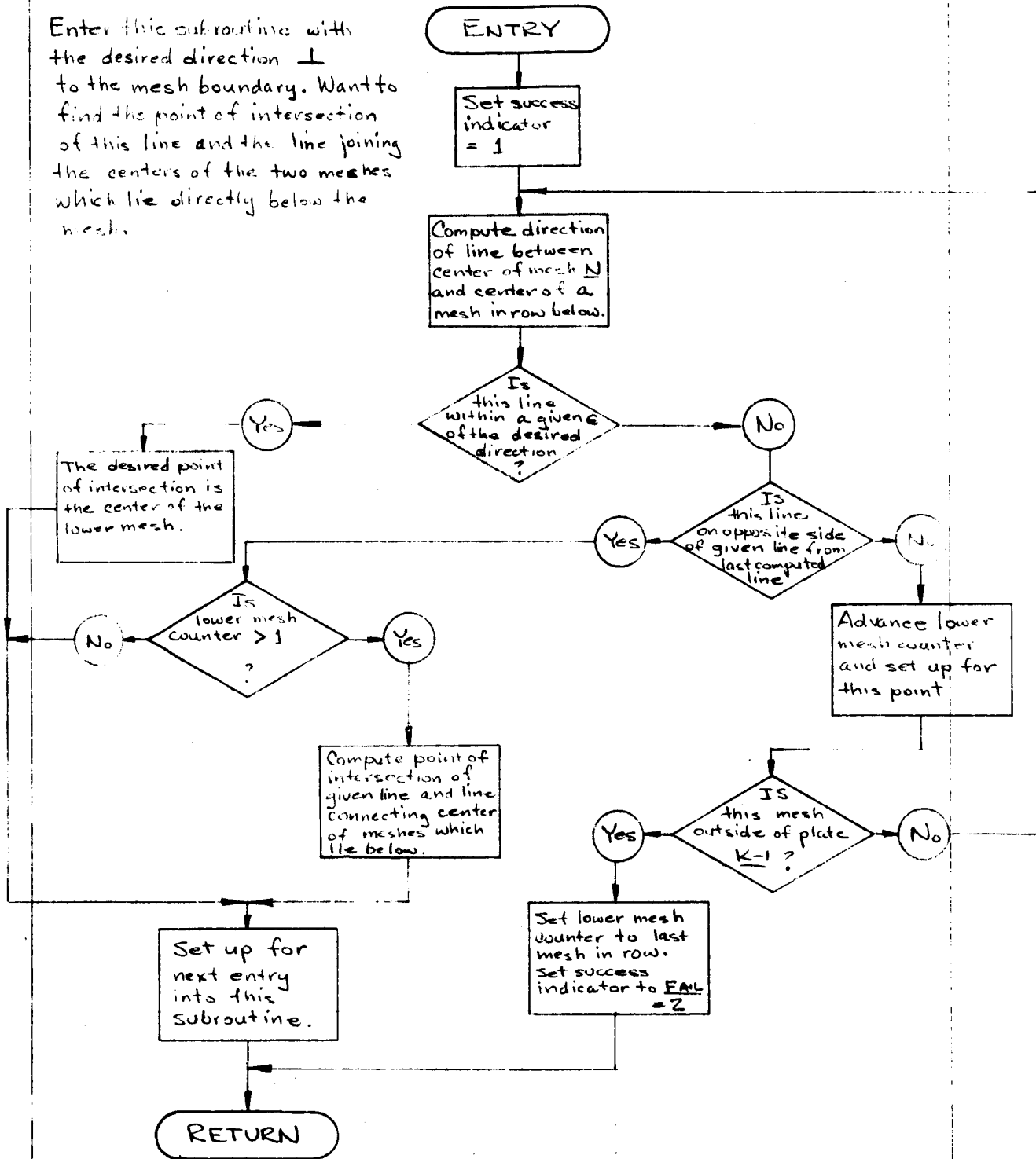
Enter this subroutine with the desired direction of the mesh boundary. Want to find between which points on next plate the projected point would lie, and the position of this intersection.



SUBROUTINE AIFACE



Enter this subroutine with the desired direction  $\perp$  to the mesh boundary. Want to find the point of intersection of this line and the line joining the centers of the two meshes which lie directly below the mesh.



# APPENDIX C

## LISTINGS OF FORTRAN SOURCE DECKS

The main program and subroutines are listed in this appendix as follows:

| <u>TITLE</u> | <u>PAGE NO.</u> |
|--------------|-----------------|
| MAIN PROGRAM | C-1             |
| LINK 1       | C-11            |
| INPUT        | C-13            |
| SPCLIN       | C-14            |
| PRIN         | C-15            |
| INTLIZ       | C-18            |
| GNRT         | C-22            |
| DOME         | C-25            |
| AMZRO        | C-27            |
| LINK 2       | C-29            |
| COMPUT       | C-30            |
| GETPOS       | C-34            |
| EXTPOS       | C-44            |
| AINTR        | C-45            |
| ACRNR        | C-46            |
| ABNDRY       | C-48            |
| AANGL        | C-50            |
| AIFACE       | C-51            |
| DETTN        | C-56            |
| GTTN         | C-58            |
| FDLPTS       | C-59            |
| CMPTIF       | C-60            |
| FDUPTS       | C-63            |
| SDST         | C-65            |
| EVALQ        | C-67            |
| FINDP        | C-68            |
| QUAD         | C-70            |
| COMPDT       | C-71            |
| GETCN        | C-72            |
| ENERGCK      | C-73            |
| LINK 3       | C-74            |
| PRNT         | C-75            |
| GRAPH        | C-77            |
| GRP 1        | C-78            |
| GRP 2        | C-81            |
| GRP 4        | C-86            |
| PMU          | C-88            |
| AX2          | C-89            |
| RWND2        | C-92            |
| RWDO         | C-92            |
| FPTMOD       | C-93            |

**NOTE:** The blocks of DIMENSION, COMMON, and EQUIVALENCE statements listed on pages C-11 and C-12 are common to all the subroutines except QUAD, PMU, and AX2. Although these blocks are repeated in the cards for the subroutines, they are listed here only once.



```

$IBFTC 2DMN00 LIST,REF
C ***** DECK 6A2  GMDRL *****
C 2-D LAGRANGIAN -- MAIN PROGRAM FOR IBM 7040 VERSION OF CHAIN
C
C
C      DEFINITION OF INPUT QUANTITIES                                FORMAT
C
C      CARD 1.                                                         (12A6)
C
C  TITLE  - 72 CHARACTERS OF ALPHA-NUMERICS FOR PROBLEM IDENTIFICATION.
C           PRINTS OUT ON FIRST SHEET OF OUTPUT, AND ON EACH PLOT.
C
C      CARD 2.                                                         (6E12.8)
C
C  STRTC  - STARTING TIME CYCLE.
C           IF      0.0 - SUBROUTINE INTLIZ CALLED.
C           IF NOT 0.0 - READS TAPE 4 FOR VALUES OF COMMON AND PROCEEDS.
C
C      CARD 3.                                                         (6E12.8)
C
C  CZ     - COEFFICIENT IN ARTIFICIAL VISCOSITY EQUATION
C  CL     - COEFFICIENT IN ARTIFICIAL VISCOSITY EQUATION
C  C3     - COEFFICIENT IN DEVIATORIC ARTIFICIAL VISCOSITY EQUATION
C  C4     - COEFFICIENT IN DEVIATORIC ARTIFICIAL VISCOSITY EQUATION
C  FK     - COEFFICIENT IN TIME INCREMENT EQUATION
C  SBAR   - UPPER LIMIT OF STRESS. COMPRESSION IS NEGATIVE.          (D/SQCM)
C           IF STRESS EXCEEDS THIS VALUE, PROGRAM WILL STOP.
C
C      CARD 4.                                                         (6E12.8)
C
C  TPRNT  - TIME INCREMENT FOR PRINTOUT AND PLOT                      (SEC)
C  TMAX   - PROBLEM TERMINATE TIME                                    (SEC)
C  PRNTC  - PRINT CYCLE INCREMENT.
C           ALLOWS PRINTOUT AND PLOT AFTER EACH PRNTC CYCLES, IF DESIRED.
C  WRTPC  - WRITE TAPE INCREMENT
C           UPDATES THE RESTART TAPE AFTER EACH WRTPC CYCLES.
C  STOPC  - STOP CYCLE.
C           ALLOWS PROGRAM TO BE TERMINATED AFTER STOPC CYCLES.
C
C      CARD 5 0.                                                       (12I6)
C
C  M      - NO. OF PLATES IN PROBLEM.
C  NBL    - BOUNDARY INDICATOR - LOWER X OF FIRST PLATE.
C           = 1 - FREE SURFACE
C           = 2 - FIXED SURFACE
C  NBU    - BOUNDARY INDICATOR - UPPER X OF LAST PLATE
C  IALF   - GEOMETRY INDICATOR
C           = 1 - PLANE
C           = 2 - CYLINDRICAL
C  NOSLID - BOUNDARY INTERFACE INDICATOR
C           = 0 - THE INTERFACES ARE ALLOWED TO SLIDE. ONLY LAST PLATE
C               IS ALLOWED TO INCLUDE STRENGTH.
C           = 1 - NO RELATIVE MOVEMENT ALONG INTERFACE BOUNDARIES.

```

C MESH SIZE IN X-DIRECTION MUST BE SAME FOR ALL PLATES.  
C N1D - ONE DIMENSIONAL CODE SIMULATOR INDICATOR  
C = 0 - NORMAL MODE - 2-D PROBLEM  
C = 1 - 1-D PROBLEM  
C NBQ - ARTIFICIAL VISCOSITY INDICATOR  
C = 0 - EVALUATE VISCOSITY ONLY IN COMPRESSION  
C = 1 - ALWAYS EVALUATE VISCOSITY  
C NPLOTT - PLOT INDICATOR  
C = 0 OR BLANK - PLOT X VS. Z  
C = 1 - PLOT TXX1 VS. X AND Z  
C = 2 - PLOT TYY1 VS. X AND Z  
C = 3 - PLOT TZZ1 VS. X AND Z  
C = 4 - PLOT TXZ1 VS. X AND Z  
C = 5 - DRAW ALL 5 PLOTS  
C = 6 - SKIP PLOTS  
C NCRVS - THE NUMBER OF COLUMNS (I VALUES) TO BE PLOTTED. MAX. OF 20.  
C  
C CARD 5 1. IF NCRVS = 0, OMIT THIS CARD. (1216)  
C  
C JCRVS(L),L=1,NCRVS  
C - THE COLUMN VALUES TO BE PLOTTED.  
C  
C CARD 6 0. THERE ARE 8 CARDS TO DEFINE EACH PLATE. THERE (1216)  
C WILL BE M SETS OF THE FOLLOWING 8 CARDS.  
C  
C IM - NO. OF ZONES IN RADIAL (X) DIRECTION  
C JM - NO. OF ZONES IN AXIAL (Z) DIRECTION  
C NBZL - INDICATOR FOR BOUNDARY CONDITION - LEFT Z OF EACH PLATE  
C NBZR - INDICATOR FOR BOUNDARY CONDITION - RIGHT Z OF EACH PLATE  
C MEQ - INDICATOR FOR EQUATION-OF-STATE  
C = 1 - TILLOTSON  
C = 2 - POLYNOMIAL  
C = 3 - POLYTROPIC (EXPLOSIVE)  
C  
C CARD 6 1. (6E12.8)  
C  
C XI - INITIAL POSITION IN X-DIRECTION OF LOWER-LEFT CORNER (CM)  
C XL - WIDTH OF PLATE IN X-DIRECTION (CM)  
C ZI - INITIAL POSITION IN Z-DIRECTION OF UPPER-LEFT CORNER (CM)  
C ZL - LENGTH OF PLATE IN Z-DIRECTION (CM)  
C UXZ - INITIAL VELOCITY IN X-DIRECTION (CM/SEC)  
C UZZ - INITIAL VELOCITY IN Z-DIRECTION (CM/SEC)  
C  
C CARD 6 2. (6E12.8)  
C  
C UXINT - INITIAL INTERFACE VELOCITY IN X-DIRECTION (CM/SEC)  
C UZINT - INITIAL INTERFACE VELOCITY IN Z-DIRECTION (CM/SEC)  
C RHOI - INITIAL VALUE OF MATERIAL DENSITY (G/CC)  
C TXXZ - INITIAL VALUE OF STRESS IN X-DIRECTION (D/SQCM)  
C TYYZ - INITIAL VALUE OF STRESS IN Y-DIRECTION (D/SQCM)  
C TZZZ - INITIAL VALUE OF STRESS IN Z-DIRECTION (D/SQCM)  
C  
C CARD 6 3. (6E12.8)

C  
C TXZZ - INITIAL VALUE OF SHEAR STRESS (D/SQCM)  
C EZ - INITIAL VALUE OF ENERGY (ERG/GM)  
C PHIL - ANGLE OF INCLINATION OF LEFT Z-BOUNDARY (DEG)  
C PHIR - ANGLE OF INCLINATION OF RIGHT Z-BOUNDARY (DEG)  
C PSIL - ANGLE OF INCLINATION OF LOWER X-BOUNDARY (DEG)  
C PSIU - ANGLE OF INCLINATION OF UPPER X-BOUNDARY (DEG)

C  
C IF IM = 0, THE PLATE WILL BE --FREE-FORM-- IN SHAPE (EX. A  
C DEBRIS CLOUD). ALL QUANTITIES NEEDED TO DEFINE THE  
C PLATE ARE INPUTTED.

C  
C ONLY ONE FREE-FORM PLATE PER PROBLEM.

C  
C CARDS 6 1, 6 2, 6 3 ARE OMITTED AND REPLACED WITH  
C THE FOLLOWING CARDS.

C  
C THERE WILL BE JM SETS OF CARDS. EACH SET CONSISTS  
C OF THREE TYPES OF CARDS, AND DEFINES A ROW.

C  
C CARD 1. (16)

C IM1(J) - J IS THE ROW INDICATOR. NO. OF ZONES IN ROW J. UP TO 20 ROWS  
C ARE ALLOWED.

C  
C CARD 2. EACH PAIR OF CARDS (ONE CARD 2 AND ONE CARD 3) (5E15.8)  
C DEFINES A ZONE. THEREFORE, THERE WILL BE IM1 PAIRS  
C OF CARDS.

C UX(N) - N IS THE ZONE COUNTER. RADIAL VELOCITY (CM/SEC)  
C UZ(N) - AXIAL VELOCITY (CM/SEC)  
C P(N) - HYDRODYNAMIC PRESSURE IN THE ZONE (D/SQCM)  
C E(N) - ENERGY (ERG/GM)  
C RHO(N) - DENSITY (G/CC)

C  
C CARD 3. (5E15.8)

C X(N) - RADIAL POSITION (CM)  
C Z(N) - AXIAL POSITION (CM)

C  
C AFTER ALL ZONE-DEFINING CARDS HAVE BEEN READ IN,  
C CARDS DEFINING THE POSITION AND VELOCITIES OF THE  
C RIGHT-HAND AND INTERFACE BOUNDARY MESH POINTS ARE  
C READ.

C  
C CARD 4. EACH CARD DEFINES ONE BOUNDARY MESH POINT. (5E15.8)

C UXS(NS) - RADIAL VELOCITY (CM/SEC)  
C UZS(NS) - AXIAL VELOCITY (CM/SEC)  
C XS(NS) - RADIAL POSITION (CM)  
C ZS(NS) - AXIAL POSITION (CM)

C  
C THE FOLLOWING QUANTITIES DESCRIBE THE MATERIAL PROPERTIES OF EACH PLATE

```
C
C   FOR MEQ = 1
C   CARD 6 4. (6E12.8)
C
C   RHOZ - DENSITY (G/CC)
C   SA - SMALL A IN E.O.S.
C   SB - SMALL B IN E.O.S.
C   CAPA - BIG A IN E.O.S. (D/SQCM)
C   CAPB - BIG B IN E.O.S. (D/SQCM)
C   ALPHA - ALPHA IN E.O.S.
C
C   FOR MEQ = 2
C   CARD 6 4. (6E12.8)
C
C   RHOZ - DENSITY (G/CC)
C   K0 - COEFFICIENT OF ARG**0 IN F1
C   K1 - COEFFICIENT OF ARG**1 IN F1
C   K2 - COEFFICIENT OF ARG**2 IN F1
C   K3 - COEFFICIENT OF ARG**3 IN F1
C   H0 - COEFFICIENT OF ARG**0 IN F2
C
C   FOR MEQ = 1
C   CARD 6 5. (6E12.8)
C
C   BETA - BETA IN E.O.S.
C   EZERO - E SUB ZERO IN E.O.S. (ERG/GM)
C   RHOS - RHO SUB S IN E.O.S.
C   EPRS - E SUB S IN E.O.S. (ERG/GM)
C
C   FOR MEQ = 2
C   CARD 6 5. (6E12.8)
C   H1 - COEFFICIENT OF ARG**1 IN F2
C   H2 - COEFFICIENT OF ARG**2 IN F2
C   H3 - COEFFICIENT OF ARG**3 IN F2
C
C   CARD 6 6. (6E12.8)
C
C   CNO - CONSTANT TERM IN SOUND SPEED EQUATION (CM/SEC)
C   CNA - PRESSURE MULTIPLIER TERM IN SOUND SPEED EQUATION (CMPS/MB)
C   CNEXP - EXPONENT OF PRESSURE TERM IN SOUND SPEED EQUATION
C
C   CARD 6 7. (6E12.8)
C
C   G0 - COEFFICIENT IN MODULUS OF RIGIDITY FIT (D/SQCM)
C   G1 - COEFFICIENT IN MODULUS OF RIGIDITY FIT (D/SQCM)
C   G2 - COEFFICIENT IN MODULUS OF RIGIDITY FIT (D/SQCM)
C   G3 - COEFFICIENT IN MODULUS OF RIGIDITY FIT (D/SQCM)
C   CAPY - YIELD STRENGTH (D/SQCM)
C   UTS - ULTIMATE TENSILE STRENGTH OF MATERIAL (D/SQCM)
C
C   END OF INPUT QUANTITIES.
C
C   ALL QUANTITIES DEFINED BELOW REFER TO INDIVIDUAL ZONES.
```

C  
C DX - INITIAL ZONE SIZE IN X-DIRECTION (CM)  
C DZ - INITIAL ZONE SIZE IN Z-DIRECTION (CM)  
C X - RADIAL POSITION AT TIME N (CM)  
C X1 - RADIAL POSITION AT TIME N+1 (CM)  
C Z - AXIAL POSITION AT TIME N (CM)  
C Z1 - AXIAL POSITION AT TIME N+1 (CM)  
C UX - RADIAL VELOCITY AT TIME N-1/2 (CM/SEC)  
C UX1 - RADIAL VELOCITY AT TIME N+1/2 (CM/SEC)  
C UZ - AXIAL VELOCITY AT TIME N-1/2 (CM/SEC)  
C UZ1 - AXIAL VELOCITY AT TIME N+1/2 (CM/SEC)  
C RHO - DENSITY AT TIME N (G/CC)  
C RHO1 - DENSITY AT TIME N+1 (G/CC)  
C A - AREA AT TIME N (SQ CM)  
C A1 - AREA AT TIME N+1 (SQ CM)  
C TXX - STRESS IN X-DIRECTION AT TIME N (D/SQCM)  
C TXX1 - STRESS IN X-DIRECTION AT TIME N+1 (D/SQCM)  
C TYY - STRESS IN Y-DIRECTION AT TIME N (D/SQCM)  
C TYY1 - STRESS IN Y-DIRECTION AT TIME N+1 (D/SQCM)  
C TZZ - STRESS IN Z-DIRECTION AT TIME N (D/SQCM)  
C TZZ1 - STRESS IN Z-DIRECTION AT TIME N+1 (D/SQCM)  
C TXZ - SHEAR STRESS AT TIME N (D/SQCM)  
C TXZ1 - SHEAR STRESS AT TIME N+1 (D/SQCM)  
C Q - ARTIFICIAL VISCOSITY VALUE AT TIME N (D/SQCM)  
C Q1 - ARTIFICIAL VISCOSITY VALUE AT TIME N+1 (D/SQCM)  
C QXX1 - X COMPONENT OF DEVIATORIC ARTIFICIAL VISCOSITY (D/SQCM)  
C QZZ1 - Z COMPONENT OF DEVIATORIC ARTIFICIAL VISCOSITY (D/SQCM)  
C QXZ1 - SHEAR COMP. OF DEVIATORIC ARTIFICIAL VISCOSITY (D/SQCM)  
C F - ENERGY AT TIME N (ERG)  
C E1 - ENERGY AT TIME N+1 (ERG)  
C ZM - MASS (GM)  
C DTXX - DEVIATORIC STRESS IN X-DIRECTION AT TIME N+1 (D/SQCM)  
C DTYY - DEVIATORIC STRESS IN Y-DIRECTION AT TIME N+1 (D/SQCM)  
C DTZZ - DEVIATORIC STRESS IN Z-DIRECTION AT TIME N+1 (D/SQCM)  
C DTXZ - DEVIATORIC SHEAR STRESS AT TIME N+1 (D/SQCM)  
C TM - LARGEST TENSILE STRESS OF THE 3 PRINCIPAL STRESSES (D/SQCM)  
C PHITM - ANGLE OF ROTATION OF THE PRINCIPAL STRESS COORDINATES (DEG)  
C  
C THE FOLLOWING QUANTITIES REFER TO SPECIAL RIGHT AND INTER-FACE  
C BOUNDARY MESH POINTS.  
C  
C XS - RADIAL POSITION AT TIME N (CM)  
C XS1 - RADIAL POSITION AT TIME N+1 (CM)  
C ZS - AXIAL POSITION AT TIME N (CM)  
C ZS1 - AXIAL POSITION AT TIME N+1 (CM)  
C UXS - RADIAL VELOCITY AT TIME N-1/2 (CM/SEC)  
C UXS1 - RADIAL VELOCITY AT TIME N+1/2 (CM/SEC)  
C UZS - AXIAL VELOCITY AT TIME N-1/2 (CM/SEC)  
C UZS1 - AXIAL VELOCITY AT TIME N+1/2 (CM/SEC)  
C TN - TANGENT FOR SLIDING INTERFACE  
C SNAB - SINE OF LINE AB FOR SLIDING INTERFACE  
C  
C OTHER TERMS WHICH APPEAR IN COMMON

|   |  |          |
|---|--|----------|
| C |  |          |
| C | NCYCLE - CYCLE COUNTER   |          |
| C | TIME - TIME  | (SEC)    |
| C | DELPR - SAME AS TPRNT  | (SEC)    |
| C | DELTA - TIME INCREMENT AT $N+1/2$                                    | (SEC)    |
| C | DELTAM - TIME INCREMENT AT $N-1/2$                                   | (SEC)    |
| C | DMIN - USED IN TIME INCREMENT CALCULATION                            |          |
| C | ALF - FLOATED IALF   |          |
| C | IEND - END OF PROBLEM INDICATOR                                      |          |
| C | N - ACTIVE ZONE SEQUENTIAL COUNTER                                   |          |
| C | NS - ACTIVE SPECIAL MESH SEQUENTIAL COUNTER                          |          |
| C | NSR - RIGHT BOUNDARY MESH POINT INDICATOR                            |          |
| C | NMAX - TOTAL NO. OF ZONES  |          |
| C | NMAX1 - TOTAL NO. OF ZONES IN THE FREE-FORM PLATE                    |          |
| C | NSMAX - TOTAL NO. OF SPECIAL MESH POINTS                             |          |
| C | I - X-DIRECTION (COLUMN) INDICATOR                                   |          |
| C | J - Z-DIRECTION (ROW) INDICATOR                                      |          |
| C | K - PLATE INDICATOR  |          |
| C | IMX - IM OF ACTIVE PLATE   |          |
| C | JMX - JM OF ACTIVE PLATE   |          |
| C | IMXM - NO. OF ZONES IN LARGEST ROW IN FREE-FORM PLATE                |          |
| C | LSTN - ZONE NUMBER AFTER WHICH ALL ZONE VALUES ARE ZERO. STOP        |          |
| C | CALCULATION OF PRESENT CYCLE, AND PROCEED TO NEXT CYCLE.             |          |
| C | L - INDICATOR IN ACCELERATION SUBROUTINES                            |          |
| C | LL - INDICATOR IN ACCELERATION SUBROUTINES                           |          |
| C | PI - HYDRODYNAMIC PRESSURE AT TIME $N+1$                             | (D/SQCM) |
| C | PM - HYDRODYNAMIC PRESSURE AT TIME $N$                               | (D/SQCM) |
| C | RDOTR - $\rho \dot{\rho} / \rho$ AT TIME $N+1/2$                     |          |
| C | DRR2 - $\Delta \rho / \rho^2$ AT TIME $N+1/2$                        |          |
| C | DED - TERM IN ENERGY EQUATION  |          |
| C | ETA - $\rho / \rho_z$  |          |
| C | ETA1 - $\rho_{01} / \rho_{0z}$                                       |          |
| C | AM - A OF ACTIVE ZONE  |          |
| C | RHOM - $\rho$ OF ACTIVE ZONE   |          |
| C | EM - ENERGY OF ACTIVE ZONE AT TIME $N$                               |          |
| C | G - MODULUS OF RIGIDITY  | (D/SQCM) |
| C | DXX - X COMPONENT OF STRETCHING DEVIATOR AND SPIN TENSOR ( $N+1/2$ ) | (1/SEC)  |
| C | DZZ - Z COMPONENT OF STRETCHING DEVIATOR AND SPIN TENSOR ( $N+1/2$ ) | (1/SEC)  |
| C | DXZ - SHEAR COMP. OF STRETCHING DEVIATOR AND SPIN TENSOR ( $N+1/2$ ) | (1/SEC)  |
| C | BOX - SUM OF STRESS DEVIATORS  | (D/SQCM) |
| C | XP - TOTAL RADIAL MOMENTUM FOR EACH CYCLE                            | (G/CMPS) |
| C | ZP - TOTAL AXIAL MOMENTUM FOR EACH CYCLE                             | (G/CMPS) |
| C | XK - TOTAL RADIAL KINETIC ENERGY FOR EACH CYCLE                      | (ERG)    |
| C | ZK - TOTAL AXIAL KINETIC ENERGY FOR EACH CYCLE                       | (ERG)    |
| C | TIE - INTERNAL ENERGY FOR EACH CYCLE                                 | (ERG)    |
| C | TE - TOTAL ENERGY FOR EACH CYCLE                                     | (ERG)    |
| C | TDEE - ACCUMULATED TOTAL SPHERICAL STRESS WORK                       | (ERG)    |
| C | TDEQ - ACCUMULATED TOTAL SPHERICAL VISCOUS STRESS WORK               | (ERG)    |
| C | TDEED - ACCUMULATED TOTAL DEVIATORIC STRESS WORK                     | (ERG)    |
| C | TDEQD - ACCUMULATED TOTAL DEVIATORICVISCOUS STRESS WORK              | (ERG)    |
| C | TDEPD - ACCUMULATED TOTAL PLASTIC WORK                               | (ERG)    |
| C | TDEQPD - ACCUMULATED TOTAL ENERGY                                    | (ERG)    |
| C | DEED - DEVIATORIC STRESS WORK FOR ACTIVE ZONE                        | (ERG)    |

|   |       |  |          |
|---|-------|--|----------|
| C | DEGD  | - DEVIATORIC VISCOUS STRESS WORK FOR ACTIVE ZONE           | (ERG)    |
| C | UEP   | - CALCULATION CUTOFF FOR VELOCITIES                        | (CM/SEC) |
| C | REP   | - CALCULATION CUTOFF FOR DENSITIES                         | (G/CC)   |
| C | EEP   | - CALCULATION CUTOFF FOR ENERGIES                          | (ERG/GM) |
| C | PEP   | - CALCULATION CUTOFF FOR PRESSURES                         | (D/SQCM) |
| C | NONE  | - FLAG IN INTERFACE CALCULATION                            |          |
| C | XRT   | - USED IN INTERFACE CALCULATION                            |          |
| C | ZRT   | - USED IN INTERFACE CALCULATION                            |          |
| C | XH    | - USED IN INTERFACE CALCULATION                            |          |
| C | ZH    | - USED IN INTERFACE CALCULATION                            |          |
| C | NH    | - USED IN INTERFACE CALCULATION                            |          |
| C | TNRT  | - USED IN INTERFACE CALCULATION                            |          |
| C | TNR   | - USED IN INTERFACE CALCULATION                            |          |
| C | TNP   | - USED IN INTERFACE CALCULATION                            |          |
| C | LK    | - USED IN INTERFACE CALCULATION                            |          |
| C | NFRST | - USED IN INTERFACE CALCULATION                            |          |
| C | NLST  | - USED IN INTERFACE CALCULATION                            |          |
| C | NSLST | - USED IN INTERFACE CALCULATION                            |          |
| C | XSD   | - USED IN INTERFACE CALCULATION                            |          |
| C | ZSD   | - USED IN INTERFACE CALCULATION                            |          |
| C | TNNS  | - USED IN INTERFACE CALCULATION                            |          |
| C | TN    | - USED IN INTERFACE CALCULATION                            |          |
| C | NL    | - USED IN INTERFACE CALCULATION                            |          |
| C | SINAB | - USED IN INTERFACE CALCULATION                            |          |
| C | COSAB | - USED IN INTERFACE CALCULATION                            |          |
| C | UXM   | - USED IN INTERFACE CALCULATION                            |          |
| C | UZM   | - USED IN INTERFACE CALCULATION                            |          |
| C | DSUM  | - DELTA + DELTAM   |          |
| C | AX    | - ACCELERATION IN X-DIRECTION                              |          |
| C | AZ    | - ACCELERATION IN Z-DIRECTION                              |          |
| C | NBXL  | - INDICATOR FOR BOUNDARY CONDITION - LOWER X OF EACH PLATE |          |
| C | NBXU  | - INDICATOR FOR BOUNDARY CONDITION - UPPER X OF EACH PLATE |          |
| C | AIN   | - TEMPORARY STORAGE FOR AREA. ALSO FOR ACCELERATION.       |          |
| C | AIN1  | - TEMPORARY STORAGE FOR AREA.                              |          |
| C | AIN2  | - TEMPORARY STORAGE FOR AREA.                              |          |
| C | N1    | - WORKING STORAGE  |          |
| C | N2    | - WORKING STORAGE  |          |
| C | N3    | - WORKING STORAGE  |          |
| C | X10   | - WORKING STORAGE  |          |
| C | Z10   | - WORKING STORAGE  |          |
| C | X11   | - WORKING STORAGE  |          |
| C | Z11   | - WORKING STORAGE  |          |
| C | X12   | - WORKING STORAGE  |          |
| C | Z12   | - WORKING STORAGE  |          |
| C | X13   | - WORKING STORAGE  |          |
| C | Z13   | - WORKING STORAGE  |          |
| C | X21   | - WORKING STORAGE  |          |
| C | Z21   | - WORKING STORAGE  |          |
| C | X22   | - WORKING STORAGE  |          |
| C | Z22   | - WORKING STORAGE  |          |
| C | X10M  | - WORKING STORAGE  |          |
| C | Z10M  | - WORKING STORAGE  |          |
| C | X11M  | - WORKING STORAGE  |          |

C Z11M - WORKING STORAGE  
 C X12M - WORKING STORAGE  
 C Z12M - WORKING STORAGE  
 C X13M - WORKING STORAGE  
 C Z13M - WORKING STORAGE  
 C UX10 - WORKING STORAGE  
 C UZ10 - WORKING STORAGE  
 C UX11 - WORKING STORAGE  
 C UZ11 - WORKING STORAGE  
 C UX12 - WORKING STORAGE  
 C UZ12 - WORKING STORAGE  
 C UX13 - WORKING STORAGE  
 C UZ13 - WORKING STORAGE  
 C DTXXM - WORKING STORAGE  
 C DTYM - WORKING STORAGE  
 C DTZZM - WORKING STORAGE  
 C DTXZM - WORKING STORAGE  
 C DL - WORKING STORAGE  
 C IFNT - WORKING STORAGE  
 C SPACE - WORKING STORAGE  
 C ISPACE - WORKING STORAGE  
 C  
 C  
 C

|   |             |             |             |             |
|---|-------------|-------------|-------------|-------------|
|   | DIMENSION   | X( 600),    | X1( 600),   | Z( 600),    |
| 1 | Z1( 600),   | UX( 600),   | UX1( 600),  | UZ( 600),   |
| 2 | UZ1( 600),  | RHO( 600),  | RHO1( 600), | A( 600),    |
| 3 | A1( 600),   | TXX( 600),  | TXX1( 600), | TY( 600),   |
| 4 | TY1( 600),  | TZZ( 600),  | TZZ1( 600), | TXZ( 600),  |
| 5 | TXZ1( 600), | Q( 600),    | Q1( 600),   | E( 600),    |
| 6 | E1( 600),   | DTXX( 600), | DTYY( 600), | DTZZ( 600), |
| 7 | DTXZ( 600), | ZM( 600),   | QXX1( 600), | QZZ1( 600), |
| 8 | QXZ1( 600)  |             |             |             |
|   | DIMENSION   | XS( 210),   | XS1( 210),  | ZS( 210),   |
| 1 | ZS1( 210),  | UXS( 210),  | UXS1( 210), | UZS( 210),  |
| 2 | UZS1( 210), | TN( 210),   | SNAB( 210), |             |
| 3 | NH( 42, 5), | XH( 42, 5), | ZH( 42, 5)  |             |
|   | DIMENSION   | TITLE(12),  | IM( 5),     | JM( 5),     |
| 1 | NBZL( 5),   | NBZR( 5),   | XI( 5),     | XL( 5),     |
| 2 | ZI( 5),     | ZL( 5),     | UXZ( 5),    | UZZ( 5),    |
| 3 | UXINT( 5),  | UZINT( 5),  | RHOI( 5),   | TXXZ( 5),   |
| 4 | TYYZ( 5),   | TZZZ( 5),   | TXZZ( 5),   | EZ( 5),     |
| 5 | RHOZ( 5),   | SA( 5),     | SB( 5),     | CAPA( 5),   |
| 6 | CAPB( 5),   | ALPHA( 5),  | BETA( 5),   | EZERO( 5),  |
| 7 | RHOS( 5),   | EPRS( 5),   | CNO( 5),    | CNA( 5),    |
| 8 | CNEXP( 5),  | GO( 5),     | G1( 5),     | G2( 5),     |
| 9 | G3( 5),     | CAPY( 5),   | UTS( 5),    | MEQ( 5)     |
|   | DIMENSION   | PHIL( 5),   | PHIR( 5),   | PSIL( 5),   |
| 1 | PSIU( 5),   | FK0( 5),    | FK1( 5),    | FK2( 5),    |
| 2 | FK3( 5),    | H0( 5),     | H1( 5),     | H2( 5),     |
| 3 | H3( 5),     | GAM( 5),    | CJP( 5),    | CJR( 5),    |
| 4 | CJD( 5),    | DETV( 5),   | CBRN( 5),   | XD( 5),     |
| 5 | ZD( 5),     | NFRST( 5),  | NLST( 5),   | NSLST( 5),  |
| 6 | NBXL( 5),   | NBXU( 5)    |             |             |



```

DIMENSION          SPACE(10),    ISPACE(10),    JCRVS(20)
DIMENSION          IM1( 50)

C
COMMON  NCYCLE, STOPC, PRNTC, STRTC, TIME, TMAX, TPRNT,
1  DELPR, DELTA, DELTAM, DMIN, FK, IEND, NMAX, NSMAX,
2  SBAR, N, NS, TITLE, CZ, CL, M, NBL,
3  NBU, IALF, IM, JM, NBZL, NBZR, XI, XL,
4  ZI, ZL, UXZ, UZZ, UXINT, UZINT, RHO1, TXXZ,
5  TYYZ, TZZZ, TXZZ, EZ, RHOZ, SA, SB, CAPA,
6  CAPB, ALPHA, BETA, EZERO, RHOS, EPRS, CNO, CNA,
7  CNEXP, GO, G1, G2, G3, CAPY, IMX, JMX,
8  NPLOT, CN, P1, RDOTR, DRR2, PM, DED, ETA,
9  ETA1, NONE, ZRT, XRT, ZH, XH, TNRT, TNR
COMMON  TNP, LK, NL, XSD, ZSD, TNNS, AX,
1  AZ, DSUM, AIN, COSAB, SINAB, NH, ALF, AIN1,
2  AIN2, TEMP1, TEMP2, TEMP3, TEMP4, TEMP5, TEMP6, TEMP7,
3  TEMP8, TEMP9, TEMP10, X10, Z10, X11, Z11, X12,
4  Z12, X13, Z13, X21, Z21, X22, Z22, LL,
5  N2, WRTPC, L, X10M, Z10M, X11M, Z11M, X12M,
6  Z12M, X13M, Z13M, UX10, UZ10, UX11, UZ11, UX12,
7  UZ12, UX13, UZ13, DTXXM, DTYYM, DTZZM, DTXZM, DL,
8  NFRST, NLST, IENT, N1, NSLST, NBXL, NBXU, NBQ,
9  NFN, AM, RHOM, EM, C3, C4, DXX, DZZ
COMMON  DXZ, XP, ZP, XK, ZK, TIE, TE,
1  TDEE, TDEQ, TDEED, TDEQD, TDEPD, TDEQPD, DEED, DEQD,
2  BOX, G, NCRVS, JCRVS, CAPF, UEP, REP, EEP,
3  PEP, IJK, NOSLID, MEQ, PHIL, PHIR, PSIL, PSIU,
4  UTS
COMMON  I, J, K, X, X1, Z, Z1,
1  UX1, UZ1, RHO1, A1, TXX1, TYY1, TZZ1, TXZ1,
2  DTXX, DTYY, DTZZ, DTXZ, Q1, QXX1, QZZ1, QXZ1,
3  E1, ZM, XS, XS1, ZS, ZS1, UXS1, UZS1,
4  TN
COMMON  SPACE, ISPACE
COMMON  IM1

C
EQUIVALENCE          (UX,UX1),          (UZ,UZ1),          (UXS,UXS1),
1  (UZS,UZS1),          (TXX,TXX1),          (TZZ,TZZ1),          (TYY,TYY1),
2  (TXZ,TXZ1),          (E,E1),          (A,A1),          (RHO,RHO1),
3  (Q,Q1),          (TN,SNAB)
EQUIVALENCE          (SA,FK0,GAM),          (SB,FK1,CJP),          (CAPA,FK2,CJR),
1  (CAPB,FK3,CJD),          (ALPHA,H0,DETV),          (BETA,H1,CBRN),          (EZERO,H2,XD),
2  (RHOS,H3,ZD)
EQUIVALENCE          (ISPACE(4),NSR)

C
C  SET TRAP ROUTINE TO IGNORE UNDERFLOWS, BUT TRANSFER TO STATEMENT
C  200 ON ANY OVERFLOW.
C
NTMS = 0
ASSIGN 200 TO ITRA
CALL FPT (.TRUE.,ITRA)

C
C  SETUP FOR UTILITY TAPES USED IN OTHER LINKS

```

```
C
  CALL RWND2
  IFAKE = 1
  IF (IFAKE .EQ. 1) GO TO 140
  READ (5,100)
100 FORMAT ( 1H )
  WRITE(6,100)

C
C   LINK1 READS INPUT CARDS, WRITES INPUT VALUES.
C   IF A NEW PROBLEM, INITIALIZES ALL VARIABLES AND WRITES TAPE 4.
C   IF A CONTINUATION, READS TAPE 4.
C
140 CALL CHAIN(3)

C
C   PLOT SHAPE AT CYCLE 0.
C
  IF (STRTC .GT. 0.0) GO TO 150
  DO 142 L=1,NMAX
  X1(L) = X(L)
  Z1(L) = Z(L)
142 CONTINUE
  DO 144 L=1,NSMAX
  XS1(L) = XS(L)
  ZS1(L) = ZS(L)
144 CONTINUE
  IPLOT = NPLOT
  NPLOT = 0
  IEND = 2
  CALL CHAIN(2)
  NPLOT = IPLOT

C
C   LINK2 INTEGRATES UNTIL TIME TO OUTPUT RESULTS. DETERMINES WHEN
C   PROBLEM IS FINISHED.
C
150 CALL CHAIN(1)

C
C   LINK3 PRINTS AND PLOTS RESULTS.
C
160 CALL CHAIN(2)

C
C   CHECK TO SEE IF PROBLEM FINISHED.
C
  IF (IEND .LT. 4) GO TO 150
180 REWIND 4
  CALL EXIT

C
200 IEND = 5
  NTMS = NTMS+1
  IF (NTMS .GT. 1) GO TO 180
  GO TO 160
  END
```

Appendix C

\$IBFTC LNK110 LIST,REF  
C 2-D LAGRANGIAN -- LINK1 FOR IBM 7040 CHAIN  
C

|               |             |             |             |
|---------------|-------------|-------------|-------------|
| DIMENSION     | X( 600),    | X1( 600),   | Z( 600),    |
| 1 Z1( 600),   | UX( 600),   | UX1( 600),  | UZ( 600),   |
| 2 UZ1( 600),  | RHO( 600),  | RHO1( 600), | A( 600),    |
| 3 A1( 600),   | TXX( 600),  | TXX1( 600), | TTY( 600),  |
| 4 TYY1( 600), | TZZ( 600),  | TZZ1( 600), | TXZ( 600),  |
| 5 TXZ1( 600), | Q( 600),    | Q1( 600),   | E( 600),    |
| 6 E1( 600),   | DTXX( 600), | DTYY( 600), | DTZZ( 600), |
| 7 DTXZ( 600), | ZM( 600),   | GXX1( 600), | QZZ1( 600), |
| 8 QXZ1( 600)  |             |             |             |

|               |             |             |            |
|---------------|-------------|-------------|------------|
| DIMENSION     | XS( 210),   | XS1( 210),  | ZS( 210),  |
| 1 ZS1( 210),  | UXS( 210),  | UXS1( 210), | UZS( 210), |
| 2 UZS1( 210), | TN( 210),   | SNAB( 210), |            |
| 3 NH( 42, 5), | XH( 42, 5), | ZH( 42, 5)  |            |

|              |            |           |            |
|--------------|------------|-----------|------------|
| DIMENSION    | TITLE(12), | IM( 5),   | JM( 5),    |
| 1 NBZL( 5),  | NBZR( 5),  | XI( 5),   | XL( 5),    |
| 2 ZI( 5),    | ZL( 5),    | UXZ( 5),  | UZZ( 5),   |
| 3 UXINT( 5), | UZINT( 5), | RHOI( 5), | TXXZ( 5),  |
| 4 TYYZ( 5),  | TZZZ( 5),  | TXZZ( 5), | EZ( 5),    |
| 5 RHOZ( 5),  | SA( 5),    | SB( 5),   | CAPA( 5),  |
| 6 CAPB( 5),  | ALPHA( 5), | BETA( 5), | EZERO( 5), |
| 7 RHOS( 5),  | EPRS( 5),  | CNO( 5),  | CNA( 5),   |
| 8 CNEXP( 5), | G0( 5),    | G1( 5),   | G2( 5),    |
| 9 G3( 5),    | CAPY( 5),  | UTS( 5),  | MEQ( 5)    |

|             |            |             |            |
|-------------|------------|-------------|------------|
| DIMENSION   | PHIL( 5),  | PHIR( 5),   | PSIL( 5),  |
| 1 PSIU( 5), | FK0( 5),   | FK1( 5),    | FK2( 5),   |
| 2 FK3( 5),  | H0( 5),    | H1( 5),     | H2( 5),    |
| 3 H3( 5),   | GAM( 5),   | CJP( 5),    | CJP( 5),   |
| 4 CJD( 5),  | DETV( 5),  | CBRN( 5),   | XD( 5),    |
| 5 ZD( 5),   | NFRST( 5), | NLST( 5),   | NSLST( 5), |
| 6 NBXL( 5), | NBXU( 5)   |             |            |
| DIMENSION   | SPACE(10), | ISPACE(10), | JCRVS(20)  |

|   |        |         |         |        |        |        |        |        |
|---|--------|---------|---------|--------|--------|--------|--------|--------|
| C | COMMON | NCYCLE, | STOPC,  | PRNTC, | STRTC, | TIME,  | TMAX,  | TPRNT, |
| 1 | DELPR, | DELTA,  | DELTAM, | DMIN,  | FK,    | IEND,  | NMAX,  | NSMAX, |
| 2 | SBAR,  | N,      | NS,     | TITLE, | CZ,    | CL,    | M,     | NBL,   |
| 3 | NBU,   | IALF,   | IM,     | JM,    | NBZL,  | NBZR,  | XI,    | XL,    |
| 4 | ZI,    | ZL,     | UXZ,    | UZZ,   | UXINT, | UZINT, | RHOI,  | TXXZ,  |
| 5 | TYYZ,  | TZZZ,   | TXZZ,   | EZ,    | RHOZ,  | SA,    | SB,    | CAPA,  |
| 6 | CAPB,  | ALPHA,  | BETA,   | EZERO, | RHOS,  | EPRS,  | CNO,   | CNA,   |
| 7 | CNEXP, | G0,     | G1,     | G2,    | G3,    | CAPY,  | IMX,   | JMX,   |
| 8 | NPLOT, | CN,     | P1,     | RDOTR, | DRR2,  | PM,    | DED,   | ETA,   |
| 9 | ETA1,  | NONE,   | ZRT,    | XRT,   | ZH,    | XH,    | TNRT,  | TNR    |
|   | COMMON | TNP,    | LK,     | NL,    | XSD,   | ZSD,   | TNNS,  | AX,    |
| 1 | AZ,    | DSUM,   | AIN,    | COSAB, | SINAB, | NH,    | ALF,   | AIN1,  |
| 2 | AIN2,  | TEMP1,  | TEMP2,  | TEMP3, | TEMP4, | TEMP5, | TEMP6, | TEMP7, |
| 3 | TEMP8, | TEMP9,  | TEMP10, | X10,   | Z10,   | X11,   | Z11,   | X12,   |
| 4 | Z12,   | X13,    | Z13,    | X21,   | Z21,   | X22,   | Z22,   | LL,    |
| 5 | N2,    | WRTPC,  | L,      | X10M,  | Z10M,  | X11M,  | Z11M,  | X12M,  |
| 6 | Z12M,  | X13M,   | Z13M,   | UX10,  | UZ10,  | UX11,  | UZ11,  | UX12,  |
| 7 | UZ12,  | UX13,   | UZ13,   | DTXXM, | DTYYM, | DTZZM, | DTXZM, | DL,    |

```

8  NFRST, NLST, IENT, N1, NSLST, NBXL, NBXU, NBQ,
9  NFN, AM, RHOM, EM, C3, C4, DXX, DZZ
COMMON DXZ, XP, ZP, XK, ZK, TIE, TE,
1  TDEE, TDEQ, TDEED, TDEQD, TDEPD, TDEQPD, DEED, DEQD,
2  BOX, G, NCRVS, JCRVS, CAPF, UEP, REP, EEP,
3  PEP, IJK, NOSLID, MEQ, PHIL, PHIR, PSIL, PSIU,
4  UTS
COMMON I, J, K, X, X1, Z, Z1,
1  UX1, UZ1, RHO1, A1, TXX1, TYY1, TZZ1, TXZ1,
2  DTXX, DTYY, DTZZ, DTXZ, Q1, QXX1, QZZ1, QXZ1,
3  E1, ZM, XS, XS1, ZS, ZS1, UXS1, UZS1,
4  TN
COMMON SPACE, ISPACE

```

```

C
EQUIVALENCE (UX,UX1), (UZ,UZ1), (UXS,UXS1),
1 (UZS,UZS1), (TXX,TXX1), (TZZ,TZZ1), (TYY,TYY1),
2 (TXZ,TXZ1), (E,E1), (A,A1), (RHO,RHO1),
3 (Q,Q1), (TN,SNAB)
EQUIVALENCE (SA,FK0,GAM), (SB,FK1,CJP), (CAPA,FK2,CJR),
1 (CAPB,FK3,CJD), (ALPHA,H0,DETV), (BETA,H1,CBRN), (EZERO,H2,XD),
2 (RHOS,H3,ZD)

```

```

C
DIMENSION ZZ(15778)
C
EQUIVALENCE (ZZ(1),NCYCLE)
C
REWIND CHAIN TAPE (TAPE 0)
C
CALL RWND0
C
READ INPUT DATA FROM CARDS
C
CALL INPUT
C
CHECK VALUE OF STARTING CYCLE -
= 0, INITIALIZE VARIABLES, AND WRITE ON TAPE 4.
NOT = 0, EXIT.
C
IF ( STRTC .GT. 0.0) GO TO 150
C
WRITE OUT INPUT VALUES
C
CALL PRIN
CALL INTLIZ
WRITE (4) ZZ
150 IEND = 0
CALL CHNXIT
END

```

TR66-85  
Appendix C

INPUT  
Page 1 of 1

```

$IBFTC INPT21 LIST,REF
      SUBROUTINE INPUT
C
C 2-D LAGRANGIAN -- READS INPUT DATA FROM CARDS
C
      DIMENSION          ZZ(15778)
C
      EQUIVALENCE        (ZZ(1),NCYCLE)
      EQUIVALENCE        (ISPACE(4),NSR)
      EQUIVALENCE        (ISPACE(9),N1D)
C
      READ (5,120) (TITLE(L),L=1,12)
120  FORMAT (12A6)
      READ (5,140) CSTRT
140  FORMAT (6E12.8)
C
      CHECK VALUE OF STARTING CYCLE. IF NOT = 0, READ COMMON FROM TAPE 4.
C
      IF (CSTRT .LE. 0.0) GO TO 170
      READ (4) ZZ
170  STRTC = CSTRT
      READ (5,140) CZ,CL,C3,C4,FK,SBAR
      READ (5,140) DELPR,TMAX,PRNTC,WRTPC,STOPC
      READ (5,200) M,NBL,NBU,IALF,NOSLID,N1D,NBQ,NPLOT,NCRVS
200  FORMAT (12I6)
      IF ((NPLOT .EQ. 0) .OR. (NPLOT .GE. 6)) GO TO 210
      IF (NCRVS .EQ. 0) GO TO 210
      READ (5,200) (JCRVS(L),L=1,NCRVS)
210  IF (STRTC .GT. 0.0) RETURN
      DO 290 L=1,M
      READ (5,200) IM(L),JM(L),NBZL(L),NBZR(L),MEQ(L)
      IF (IM(L) .GT. 0) GO TO 230
C
C INPUT FOR THIS PLATE IS IN A SPECIAL FORM
C
      CALL SPCLIN
      GO TO 250
C
230  READ (5,140) X1(L),XL(L),Z1(L),ZL(L),UXZ(L),UZZ(L),UXINT(L),UZINT(L),
1L),RHOI(L),TXXZ(L),TYYZ(L),TZZZ(L),TXZZ(L),EZ(L)
2,PHIL(L),PHIR(L),PSIL(L),PSIU(L)
250  READ (5,140) RHOZ(L),SA(L),SB(L),CAPA(L),CAPB(L),ALPHA(L),BETA(L),
1EZERO(L),RHOS(L),EPRS(L)
      READ (5,140) CN0(L),CNA(L),CNEXP(L)
      READ (5,140) G0(L),G1(L),G2(L),G3(L),CAPY(L),UTS(L)
290  CONTINUE
      RETURN
      END

```

INPT 100

INPT 110

INPT 120

INPT 130

INPT 140

INPT 150

INPT 160

INPT 170

INPT 171

INPT 180

INPT 190

INPT 200

INPT 201

INPT 202

INPT 203

INPT 210

INPT 211

INPT 220

INPT 221

INPT 222

INPT 223

INPT 230

INPT 240

INPT 241

INPT 250

INPT 260

INPT 270

INPT 280

INPT 290

INPT 300

INPT 310

-

```
$IBFTC SPCIN3 LIST,REF
SUBROUTINE SPCLIN
C
C 2-D LAGRANGIAN -- READS IN DATA WHEN PLATE INPUT IS IN A SPECIAL FORM
EQUIVALENCE (ISPACE(4),NSR)
EQUIVALENCE (ISPACE(5),IMXM),(ISPACE(6),NMAX1)
C
NS = 0
N = 0
IMXM = 0
C
C DETERMINE MESH NO. OF FIRST MESHES IN THIS PLATE.
C
LL = L-1
IF (LL .LT. 1) GO TO 150
DO 140 I=1,LL
N = N+IM(I)*JM(I)
NS = NS+IM(I)+JM(I)+1
140 CONTINUE
C
150 JMX = JM(L)
DO 400 J=1,JMX
READ (5,160) IM1(J)
160 FORMAT ( 12I6 )
IMX = IM1(J)
IMXM = MAX0(IMXM,IMX)
DO 300 I=1,IMX
N = N+1
READ (5,220) UX(N),UZ(N),TXX(N),E(N),RHO(N),X(N),Z(N)
220 FORMAT ( 5E15.8 )
TXX(N) = -TXX(N)
TYY(N) = TXX(N)
TZZ(N) = TXX(N)
C
300 CONTINUE
C
400 CONTINUE
NMAX1 = N
NSMAX = JMX+IMXM+1
C
C READ POSITION AND VELOCITIES OF RIGHT-SIDE SPECIAL STATIONS
C
DO 500 I=1,NSMAX
NS = NS+1
READ (5,220) UXS(NS),UZS(NS),XS(NS),ZS(NS)
500 CONTINUE
READ (5,220) UXINT(L),UZINT(L)
C
PHIL(L) = 90.0
PHIR(L) = 90.0
PSIL(L) = 0.0
PSIU(L) = 0.0
RETURN
END
```

\$IBFTC PRIN20 LIST,REF  
SUBROUTINE PRIN

```

C
C 2-D LAGRANGIAN -- PRINTS INPUT DATA READ IN SUBR. INPUT
C
      EQUIVALENCE      ( ISPACE(4),NSR)
      EQUIVALENCE      ( ISPACE(5),IMXM),( ISPACE(6),NMAX1)
      EQUIVALENCE      ( ISPACE(9),NID)
C
      WRITE (6,120) (TITLE(L),L=1,12)
120  FORMAT (1H1,42X,47HA.H,JONES MATERIAL SCIENCE GROUP GM DRL X470
1/1H0,61X,10HINPUT DATA//1H0,29X,12A6 )
      WRITE (6,160) CZ,CL,C3,C4,FK,DELPR,TMAX,SBAR,STRTC,PRNTC,WRTPC,STO
1PC,M,NBL,NBU, IALF,NOSLID,NBQ
160  FORMAT (1H0, 2X, 2HC0, 5X, 2HCL, 5X, 2HC3, 5X, 2HC4, 5X, 2HFK, 6X,
1 9HTPRNT-SEC, 8X, 8HTMAX-SEC, 6X,10HSBAR-D/CSQ , 3X, 7HSTART C, 1X
2, 7HPRINT C, 1X, 6HW RTP C, 2X, 6HSTOP C /1H ,F6.3,F7.3,F7.3,F7.3,F
37.3,E16.8,E16.8,E16.8,F7.0,F8.0,F7.0,F8.0 /1H0, 9HNO.PLATES, 1X, 8
4HX-LOW.B., 1X, 7HX-UP.B., 1X, 5HALPHA, 1X,10HSLDNG.IFCE, 1X, 8H1-D
5 IND., 1X, 9HVISC.IND. /1H , 6X,I2, 6X,I1, 8X,I1, 6X,I1, 6X,I1, 9X
6,I1, 8X,I1 )
      WRITE (6,220)
220  FORMAT (1H0,5HPLATE, 1X,10HX-STATIONS, 1X,10HZ-STATIONS, 1X, 8HX-L
1FT.B., 1X, 9HZ-RGHT.B., 2X, 6HE.O.S. /)
      DO 270 L=1,M
      WRITE (6,260) L,IM(L),JM(L),NBZL(L),NBZR(L),MEQ(L)
260  FORMAT (1H , 2X,I2, 5X,I6, 5X,I6, 7X,I1, 8X,I1, 7X, I1)
270  CONTINUE
      WRITE (6,290)
290  FORMAT (1H0,5HPLATE, 4X, 5HXI-CM,11X, 5HXL-CM,11X, 5HZI-CM,11X, 5H
1ZL-CM, 9X,10HUXZ-CM/SEC, 6X,10HUZZ-CM/SEC, 5X,12HUXINT-CM/SEC, 4X,
212HUZINT-CM/SEC /)
      DO 360 L=1,M
      IF (IM(L) .LE. 0) GO TO 360
      WRITE (6,350) L,XI(L),XL(L),ZI(L),ZL(L),UXZ(L),UZZ(L),UXINT(L),
1UZINT(L)
350  FORMAT (1H ,1X,I2,8E16.8)
360  CONTINUE
      WRITE (6,362)
362  FORMAT (1H0,5HPLATE, 3X, 8HPHIL-DEG, 8X, 8PHIR-DEG, 8X, 8HPSIL-DE
1G, 8X, 8HPSIU-DEG / )
      DO 366 L=1,M
      IF (IM(L) .LE. 0) GO TO 366
      WRITE (6,350) L,PHIL(L),PHIR(L),PSIL(L),PSIU(L)
366  CONTINUE
      WRITE (6,380)
380  FORMAT (1H0,5HPLATE, 2X, 9HRHOZ-G/CC, 7X,10HTXXZ-D/CSQ, 6X,10HTYYZ
1-D/CSQ, 6X,10HTZZZ-D/CSQ, 6X,10HTXZZ-D/CSQ, 6X, 9HEZ-ERG/GM /)
      DO 420 L=1,M
      IF (IM(L) .LE. 0) GO TO 420
      WRITE (6,350) L,RHOI(L),TXXZ(L),TYYZ(L),TZZZ(L),TXZZ(L),EZ(L)
420  CONTINUE
      DO 431 L=1,M
      IF (IM(L) .GT. 0) GO TO 431
      WRITE (6,421) L

```

```
421 FORMAT (1H0, 6HPLATE ,12,19H IS A DEBRIS CLOUD. /1H0,      4HMESH,
13X, 9HUX-CM/SEC, 7X, 9HUZ-CM/SEC, 6X,12HSTRESS-D/CSQ, 6X, 7HE-ERG/
2G, 9X, 8HRHO-G/CC,10X, 4HX-CM,12X, 4HZ-CM / )
N = 0
LL = L-1
IF (LL .LT. 1) GO TO 424
DO 423 I=1,LL
N = N+IM(I)*JM(I)
423 CONTINUE
424 JMX = JM(L)
DO 428 J=1,JMX
IMX = IM1(J)
DO 427 I=1,IMX
N = N+1
WRITE (6,426) N,UX(N),UZ(N),TXX(N),E(N),RHO(N),X(N),Z(N)
426 FORMAT(1H ,13,8E16.8)
427 CONTINUE
428 CONTINUE
WRITE (6,429)
429 FORMAT (1H0, 5HPOINT, 2X,10HUXS-CM/SEC, 6X,10HUZS-CM/SEC, 8X, 5HXS
1-CM,11X, 5HZS-CM / )
LL = JMX+IMXM+1
DO 430 I=1,LL
WRITE (6,426) I,UXS(I),UZS(I),XS(I),ZS(I)
430 CONTINUE
431 CONTINUE
WRITE (6,440)
440 FORMAT (1H0,56X,18HMATERIAL CONSTANTS )
DO 770 L=1,M
MEQT = MEQ(L)
GO TO (480,520,560),MEQT

480 WRITE (6,490)
490 FORMAT (1H0, 5HPLATE, 3X, 8HRHO-G/CC,11X, 2HSA,14X, 2HSB,14X, 1HA,
115X, 1HB,13X, 5HALPHA,12X, 4HBETA,11X, 5HEZERO )
GO TO 590
520 WRITE (6,530)
530 FORMAT (1H0, 5HPLATE, 3X, 8HRHO-G/CC,11X, 2HK0,14X, 2HK1,14X, 2HK2
1,14X, 2HK3,14X, 2HH0,14X, 2HH1,14X, 2HH2 )
GO TO 590
560 WRITE (6,570)
570 FORMAT (1H0, 5HPLATE, 3X, 8HRHO-G/CC,10X, 5HGAMMA,11X, 3HCJP,12X,
15HCJRHO,12X, 3HCJD,13X, 4HDETV,12X, 4HCBRN,13X, 2HXD )
590 WRITE (6,350) L,RHOZ(L),SA(L),SB(L),CAPA(L),CAPB(L),ALPHA(L),BETA(
1L),EZERO(L)
GO TO (620,660,690),MEQT
620 WRITE (6,630)
630 FORMAT (1H0, 10X, 4HRHOS,13X, 2HES,13X, 3HCNO,13X, 3HCNA,12X, 5HCN
1EXP )
GO TO 710
660 WRITE (6,670)
670 FORMAT (1H0, 11X, 2HH3, 29X, 3HCNO,13X, 3HCNA,12X, 5HCNEXP )
GO TO 710
690 WRITE (6,700)
700 FORMAT (1H0, 11X, 2HZD, 29X, 3HCNO,13X, 3HCNA,12X, 5HCNEXP )
710 WRITE (6,720) RHOS(L),EPRS(L),CNO(L),CNA(L),CNEXP(L)
```



```
720 FORMAT (1H , 3X, 8E16.8)
      WRITE (6,740)
740 FORMAT (1H0,          11X, 2HG0,14X, 2HG1,14X, 2HG2,14X, 2HG3,13X-
      14HCAPY,13X, 3HUTS   )
      WRITE (6,720)  GO(L),G1(L),G2(L),G3(L),CAPY(L),UTS(L)
770 CONTINUE

C
C   CONVERT ALL ANGLES FROM DEGREES TO RADIANS.
C
      DO 830 L=1,M
      PHIL(L) = PHIL(L)/57.2957795
      PHIR(L) = PHIR(L)/57.2957795
      PSIL(L) = PSIL(L)/57.2957795
      PSIU(L) = PSIU(L)/57.2957795
830 CONTINUE
      RETURN
      END
```

\$IBFTC INTL50 LIST,REF  
SUBROUTINE INTLIZ

INTL 100

```
C
C 2-D LAGRANGIAN -- INITIALIZES ALL VARIABLES PRIOR TO CYCLE 1.
C
EQUIVALENCE      ( ISPACE(4),NSR)
EQUIVALENCE      ( ISPACE(5),IMXM),( ISPACE(6),NMAX1)
EQUIVALENCE      ( ISPACE(7),IMP),( ISPACE(8),IDP)
C
NMAX = 600
NSMAX = 210
K = 5
C
CHECK TO SEE IF PROBLEM EXCEEDS SIZE OF PROGRAM
C
IF (M .GT. K) GO TO 20
N = 0
NS = 0
DO 10 I=1,M
IF (IM(I) .GT. 0) GO TO 9
N = N+NMAX1
NS= NS+JM(I)+IMXM+1
GO TO 10
9 N = N + IM(I)*JM(I)
NS = NS + IM(I) + JM(I) + 1
10 CONTINUE
IF (N .GT. NMAX) GO TO 40
IF (NS .GT. NSMAX) GO TO 60
GO TO 110
C
WRITE ERROR MESSAGES
C
20 WRITE (6, 30) K
30 FORMAT (1H0,35HERROR - YOU MAY NOT HAVE MORE THAN ,I2, 8H PLATES.)
GO TO 80
40 WRITE (6, 50) NMAX
50 FORMAT (1H0,35HERROR - YOU MAY NOT HAVE MORE THAN ,I4, 7H ZONES.)
GO TO 80
60 WRITE (6, 70) NSMAX
70 FORMAT (1H0,41HERROR - NO. OF SPECIAL STATIONS EXCEEDED ,I4, 2H .)
80 CALL EXIT
C
SET CUTOFF VALUES FOR VELOCITY, DENSITY, ENERGY, AND PRESSURE
C
110 UEP = 1.0
RFP = 1.0E-6
EEP = 1.0E+3
PEP = 1.0E+6
C
SET UP BOUNDARY CONDITION INDICATORS
C
1 = FREE
2 = FIXED
C
IF (IALF .EQ. 1) GO TO 120
```

```
C
DO 111 L=1,M
111 NBZL(L) = 2
120 NBXL(1) = NBL
    IF (M - 2) 130,200,150
130 NBXU(1) = NBU
    GO TO 230
150 K = M-1
    DO 190 L=2,K
        NBXL(L) = 1
        NBXU(L) = 1
        NFRST(L) = 0
        NLST(L) = 9999
190 CONTINUE
200 NBXU(1) = 1
    NBXL(M) = 1
    NBXU(M) = NBU
    NFRST(1) = 0
    NLST(1) = 9999

C
C   GENERATE POSITION OF THE MESHES
C
230 CALL GNRT

C
C   INITIALIZE VARIABLES FOR EACH PLATE
C
    DMIN = 1.0
    ALF = IALF
    N = 0
    NS = 0
    IF (IM(1) .GT. 0) GO TO 270
    JMX = JM(1)

    N3 = JMX+IMXM+1
    DO 260 N2=1,N3
260 SNAB(N2) = 0.0
    N2 = N3-IM1(JMX)
    GO TO 280
270 N2 = JM(1)+1
    N3 = N2+IM(1)

C
280 DO 810 K=1,M
    IFLG = 0
    IMP = 0
    IDP = 0
    IF (K .LE. 2) GO TO 300
    N2 = N2 + JM(K-1)+IM(K-1)+1
300 JMX=JM(K)
```

```
C
DO 720 J=1,JMX
NS = NS + 1
NSR = NS
IMX = IM(K)
IMP = IMX
IF (IMX .GT. 0) GO TO 440
IMX = IMI(J)
IFLG = 1
IF (IDP .LT. 0) NSR = NSR+IDP
IF (J .LT. JMX) GO TO 400
IMP = 0
IDP = 0
GO TO 460
400 IMP = IMI(J+1)
IDP = IMP-IMX
GO TO 460
440 UXS(NS) = UXZ(K)
UZS(NS) = UZZ(K)

C
460 DO 710 I=1,IMX
N = N + 1
IF (IFLG .GT. 0) GO TO 462
RHO(N) = RHOI(K)
TXX(N) = TXX7(K)
TYY(N) = TYY7(K)
TZZ(N) = TZZ7(K)
TXZ(N) = TXZZ(K)
E(N) = EZ(K)

C
462 CALL AMZRO

C
C DETERMINE MESH NUMBERS WHICH SHOW RELATIVE PLACEMENT OF THE PLATES
C
IF (K .EQ. 1) GO TO 480
IF (J .NE. 1) GO TO 480
IF (X(N)+1.0E-6 .LT. XS(N2)) GO TO 480
IF (NFRST(K-1) .NE. 0) GO TO 470
NFRST(K-1) = N
GO TO 480
470 IF (X(N)+.0E-6 .LT. XS(N3)) GO TO 480
IF (NLST(K-1) .NE. 9999) GO TO 480
NLST(K-1) = N

C
480 TEMP1 = SQRT((Z10-Z13)**2 + (X10-X13)**2)
TEMP2 = SQRT((Z11-Z12)**2 + (X11-X12)**2)
TEMP3 = AMAX1(TEMP1,TEMP2)
DL = A(N)/TEMP3
TEMP5 = SQRT(UXZ(1)**2 + UZZ(1)**2)
TEMP4 = DL/(CNO(K) + 0.5*TEMP5) /3.0
DMIN = AMINI(DMIN,TEMP4)
IF (IFLG .GT. 0) GO TO 710
IF ((K .EQ. 1) .OR. (J .GT. 1)) GO TO 500
IF ((N .LT. NFRST(K-1)) .OR. (N .GT. NLST(K-1))) GO TO 500
UX(N) = UXINT(K-1)
```

```
      UZ(N) = UZINT(K-1)
      IF (I .LT. IMX) GO TO 520
      UXS(NS) = UXINT(K-1)
      UZS(NS) = UZINT(K-1)
      GO TO 520
500  UX(N) = UXZ(K)
      UZ(N) = UZZ(K)
520  IF (J .LT. JMX) GO TO 710
      UXS(NS) = UXINT(K)
      UZS(NS) = UZINT(K)
      SNAB(NS) = SIN(PSIU(K))
C
710  CONTINUE
C
720  CONTINUE
C
      NS = NS+1
      IF (IFLG .GT. 0) GO TO 810
      UXS(NS) = UXINT(K)
      UZS(NS) = UZINT(K)
      SNAB(NS) = SIN(PSIU(K))
C
810  CONTINUE
C
      NMAX = N
      NSMAX = NS
C
      TPRNT = DELPR
      DELTA = DMIN*FK
      DELTAM = DELTA
      TIME = 0.0
      NCYCLE = 0
      TDEE  = 0.0
      TDEQ  = 0.0
      TDEED = 0.0
      TDEQD = 0.0
      TDEPD = 0.0
      TDEQPD = 0.0
      RETURN
      END
```

\$IBFTC GNRT01 LIST,REF  
SUBROUTINE GNRT

```
C
C 2-D LAGRANGIAN - GENERATES POSITIONS OF MESHES FROM INPUT DATA
C THIS VERSION IS FOR THE COUPLED CAMEO-2-D SERIES.
C
EQUIVALENCE      (ISPACE(4),NSR)
EQUIVALENCE      (ISPACE(5),IMXM),(ISPACE(6),NMAX1)
EQUIVALENCE      (ISPACE(10),NS1)
C
ONE = 0.9999995
ZRO = 0.0000005
C
N = 0
NS = 0
NS1 = 0
C
DO 800 K=1,M
IMX = IM(K)
JMX = JM(K)
IF (IMX .GT. 0) GO TO 160
C
N = N+NMAX1
NS1 = NS1+JMX+IMXM+1
GO TO 800
C
160 IF ((PSIU(K) .LE. 6.2831853) .OR. (PSIL(K) .LE. 6.2831853)) GO TO
1 200
CALL DOME
GO TO 800
C
200 FIMX = IMX
FJMX = JMX
NS = NS1
NS1 = NS1+JMX
CPSL = COS(PSIL(K))
IF (ABS(CPSL) .GT. ONE) CPSL = SIGN( 1.0,CPSL)
IF (ABS(CPSL) .LT. ZRO) CPSL = 0.0
SPSL = SIN(PSIL(K))
IF (ABS(SPSL) .GT. ONE) SPSL = SIGN( 1.0,SPSL)
IF (ABS(SPSL) .LT. ZRO) SPSL = 0.0
CPSU = COS(PSIU(K))
IF (ABS(CPSU) .GT. ONE) CPSU = SIGN( 1.0,CPSU)
IF (ABS(CPSU) .LT. ZRO) CPSU = 0.0
SPSU = SIN(PSIU(K))
IF (ABS(SPSU) .GT. ONE) SPSU = SIGN( 1.0,SPSU)
IF (ABS(SPSU) .LT. ZRO) SPSU = 0.0
CPHL = COS(PHIL(K))
IF (ABS(CPHL) .GT. ONE) CPHL = SIGN( 1.0,CPHL)
IF (ABS(CPHL) .LT. ZRO) CPHL = 0.0
SPHL = SIN(PHIL(K))
IF (ABS(SPHL) .GT. ONE) SPHL = SIGN( 1.0,SPHL)
IF (ABS(SPHL) .LT. ZRO) SPHL = 0.0
```

```

CPHR = COS(PHIR(K))
IF (ABS(CPHR) .GT. ONE) CPHR = SIGN( 1.0,CPHR)
IF (ABS(CPHR) .LT. ZRO) CPHR = 0.0
SPHR = SIN(PHIR(K))
IF (ABS(SPHR) .GT. ONE) SPHR = SIGN( 1.0,SPHR)
IF (ABS(SPHR) .LT. ZRO) SPHR = 0.0

```

```

C
C      (XIF,ZIF) - POSITION OF LOWER RIGHT CORNER OF PLATE K.
C      (XJ ,ZJ ) -          UPPER LEFT
C      (XJF,ZJF) -          UPPER RIGHT
C      (XI ,ZI ) -          LOWER LEFT
C

```

```

XIF = XI(K)+XL(K)*CPSL
ZIF = ZI(K)+XL(K)*SPSL
XJ  = XI(K)+ZL(K)*CPHL
ZJ  = ZI(K)+ZL(K)*SPHL
XJF = ((ZIF-ZJ)*CPSU*CPHR + XJ*SPSU*CPHR - XIF*CPSU*SPHR)/(SPSU*CPH
1R - CPSU*SPHR)

```

```

IF (ABS(PSIU(K)) .GT. ABS(PHIR(K))) GO TO 290
ZJF = ZJ + (XJF-XJ)*SPSU/CPSU
GO TO 300

```

```

290 ZJF = ZIF + (ZJF-XIF)*SPHR/CPHR

```

```

C
C      OBJECT OF THE GAME IS TO DIVIDE EACH SIDE INTO IMX, OR JMX EQUAL PARTS.
C

```

```

300 XUL = DSQRT((DBLE(XJF-XJ))**2 + (DBLE(ZJF-ZJ))**2)
DXU = XUL/FIMX
ZRL = DSQRT((DBLE(XJF-XIF))**2 + (DBLE(ZJF-ZIF))**2)
DZR = ZRL/FJMX
DXL = XL(K)/FIMX
DZL = ZL(K)/FJMX

```

```

C
DO 700 J=1,JMX
NS = NS+1
FJM1 = J-1
XS(NS) = XIF+FJM1*CPHR*DZR
ZS(NS) = ZIF+FJM1*SPHR*DZR
TEMP1 = XI(K)+FJM1*CPHL*DZL
TEMP2 = ZI(K)+FJM1*SPHL*DZL
TNJ = (ZS(NS)-TEMP2)/(XS(NS)-TEMP1)
IF (J .GT. 1) GO TO 560
DO 550 L=1,IMX
NS1 = NS+1
FIM1 = L-1
XS1(L) = XI(K)+FIM1*CPSL*DXL
ZS1(L) = ZI(K)+FIM1*SPSL*DXL
XS(NS1) = XJ+FIM1*CPSU*DXU
ZS(NS1) = ZJ+FIM1*SPSU*DXU
TEMP3 = XS(NS1)-XS1(L)
IF (ABS(TEMP3/ZL(K)) .GT. 1.0E-7) GO TO 540
TN(L) = 1.0E+7
GO TO 550

```

```
540 TN(L) = (ZS(NS1)-ZS1(L))/TEMP3
550 CONTINUE
C
560 DO 600 I=1,IMX
      N = N+1
      FIM1 = I-1
      IF (ABS(TN(I)) .LT. 1.0E+7) GO TO 580
      X(N) = XS1(I)
      GO TO 590
580 X(N) = (ZS1(I)-TEMP2+TEMP1*TNJ-XS1(I)*TN(I))/(TNJ-TN(I))
590 Z(N) = TEMP2+(X(N)-TEMP1)*TNJ
600 CONTINUE
C
700 CONTINUE
      NS1 = NS1+1
      XS(NS1) = XJF
      ZS(NS1) = ZJF
C
800 CONTINUE
C
      RETURN
      END
```



\$IBFTC DOME00 LIST,REF  
SUBROUTINE DOME

C  
C 2-D LAGRANGIAN -- GENERATES POSITION OF MESHES FOR ELLIPTICAL SURFACES  
C OF REVOLUTION  
C

EQUIVALENCE (ISPACE(4),NSR)  
EQUIVALENCE (ISPACE(10),NS1)

C  
NW = N  
NS = NS1  
NS1 = NS1+JM(K)

C  
SAL = XI(K)  
SAU = XI(K)+XL(K)  
SBL = ZI(K)  
SBU = ZI(K)+ZL(K)  
DPHI = (PHIL(K)-PHIR(K))/FLOAT(IM(K))

C  
PHI = PHIL(K)+DPHI  
IMX = IM(K)  
DO 500 I=1,IMX  
PHI = PHI-DPHI  
IF (ABS(PHI-1.5707963) .GT. 0.3E-6) GO TO 250  
XE = 0.0  
ZE = SBL  
XF = 0.0  
ZF = SBU  
GO TO 300

250 TANP = TAN(PHI)  
XE = (SAL\*SBL)/SQRT(SBL\*\*2+(SAL\*TANP)\*\*2)  
ZE = XE\*TANP  
XF = (SAU\*SBU)/SQRT(SBU\*\*2+(SBU\*TANP)\*\*2)  
ZF = XF\*TANP  
300 DXEF = (XF-XE)/FLOAT(JM(K))  
DZEF = (ZF-ZE)/FLOAT(JM(K))  
XW = XE-DXEF  
ZW = ZE-DZEF

C  
JMX = JM(K)  
DO 450 J=1,JMX  
XW = XW+DXEF  
ZW = ZW+DZEF  
N = I+IMX\*(J-1)+NW  
X(N) = XW  
Z(N) = ZW

450 CONTINUE  
NS1 = NS1+1  
XS(NS1) = XF  
ZS(NS1) = ZF

C  
500 CONTINUE

```
C
PHI = PHIR(K)
IF (ABS(PHI) .GT. 0.3E-7) GO TO 600
XE = SAL
ZE = 0.0
XF = SAU
ZF = 0.0
GO TO 650
600 TANP = TAN(PHI)
XE = (SAL*SBL)/SQRT(SBL**2+(SAL*TANP)**2)
ZE = XE*TANP
XF = (SAU*SBU)/SQRT(SBU**2+(SBU*TANP)**2)
ZF = XF*TANP
650 DXEF = (XF-XE)/FLOAT(JM(K))
DZEF = (ZF-ZE)/FLOAT(JM(K))
XW = XE-DXEF
ZW = ZE-DZEF

C
JMX = JM(K)
DO 750 J=1,JMX
XW = XW+DXEF
ZW = ZW+DZEF
NS = NS+1
XS(NS) = XW
ZS(NS) = ZW
750 CONTINUE

C
NS1 = NS1+1
XS(NS1) = XF
ZS(NS1) = ZF

C
PSIL(K) = 0.0
PSIU(K) = 0.0
RETURN
END
```

TR66-85  
Appendix C

\$IBFTC AMZR10 LIST,REF  
SUBROUTINE AMZRO

C  
C 2-D LAGRANGIAN -- COMPUTES AREA AND MASS OF EACH MESH AT TIME = 0.0  
C

EQUIVALENCE (ISPACE(4),NSR)  
EQUIVALENCE (ISPACE(5),IMXM),(ISPACE(6),NMAX1)  
EQUIVALENCE (ISPACE(7),IMP),(ISPACE(8),IDP)

C  
X10 = X(N)  
Z10 = Z(N)  
NW = N+1  
X11 = X(NW)  
Z11 = Z(NW)  
NW = N+IMX  
X12 = X(NW)  
Z12 = Z(NW)  
NW = NW+1  
X13 = X(NW)  
Z13 = Z(NW)

C  
IF (J .GE. JMX) GO TO 300

C  
IF (I .LT. IMX) GO TO 200

C  
X11 = XS(NSR)  
Z11 = ZS(NSR)

C  
200 IF (I-IMP) 450,250,210  
210 NS = NS+1  
X12 = XS(NS)  
Z12 = ZS(NS)  
250 NW = NS+1  
X13 = XS(NW)  
Z13 = ZS(NW)  
GO TO 450

C  
300 NS = NS+1  
X12 = XS(NS)  
Z12 = ZS(NS)  
NW = NS+1  
X13 = XS(NW)  
Z13 = ZS(NW)

C  
IF (I .LT. IMX) GO TO 410

C  
X11 = XS(NSR)  
Z11 = ZS(NSR)

C  
410 NH(I,K) = N  
XH(I,K) = 0.25\*(X10+X11+X12+X13)  
ZH(I,K) = 0.25\*(Z10+Z11+Z12+Z13)

C

```
C      COMPUTE AREA AND MASS OF MESH  N.
C
450  XIBAR = (X11+X12+X10)/3.0
      XOBAR = (X11+X12+X13)/3.0
      AIN1  = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
      AON1  = 0.5*((X12-X13)*(Z11-Z13) - (Z12-Z13)*(X11-X13))
      A(N)  = AIN1 + AON1
      TEMP9 = AIN1*XIBAR**(IALF-1) + AON1*XOBAR**(IALF-1)
      ZM(N) = TEMP9*RHO(N)
C
      RETURN
      END
```

TR66-85  
Appendix C

LINK 2  
Page 1 of 1

```
$IBFTC LNK211 LIST,REF
C 2-D LAGRANGIAN -- LINK2 FOR IBM 7040 CHAIN
C
C EQUIVALENCE (ISPACE(2),LSTN)
C
C LSTN = NMAX
C
C UPDATE VARIABLES EXCEPT ON FIRST CYCLE
C
10 NCYCLE = NCYCLE + 1
XP = 0.0
ZP = 0.0
XK = 0.0
7K = 0.0
TIE = 0.0
TF = 0.0
IF (NCYCLE .EQ. 1) GO TO 270
DO 160 N=1,NMAX
X(N) = X1(N)
7(N) = 71(N)
IF (TXX1(N) .LE. SBAR) GO TO 370
IF (TYY1(N) .LE. SBAR) GO TO 370
IF (TZZ1(N) .LE. SBAR) GO TO 370
IF (TXZ1(N) .LE. SBAR) GO TO 370
160 CONTINUE
DO 220 NS=1,NSMAX
XS(NS) = XS1(NS)
ZS(NS) = ZS1(NS)
220 CONTINUE
DELTAM = DELTA
DELTA = FK * DMIN
TEMP1 = 1.1*DELTAM
IF (DELTA .GT. TEMP1) DELTA = TEMP1
270 TIME = TIME + DELTA
C
C COMPUTE FOR 1 CYCLE
C
C CALL COMPUT
C
C CHECK FOR END OF PROBLEM, OR IF PRINT OUT DESIRED
C
IF (IFND .EQ. 5) GO TO 360
IF (TIME .LT. TMAX) GO TO 310
290 IFND = 5
GO TO 360
310 CYCLE = NCYCLE
IF ((STOPC .GT. 0.0) .AND. (CYCLE .GE. STOPC)) GO TO 290
IF ((WRTPC .GT. 0.0) .AND. (AMOD(CYCLE,WRTPC) .EQ. 0.0)) GO TO 390
IF ((PRNTC .GT. 0.0) .AND. (AMOD(CYCLE,PRNTC) .EQ. 0.0)) GO TO 350
IF (TIME .LT. TPRNT) GO TO 10
TPRNT = TPRNT + DELPR
350 IFND = 1
360 CALL CHNXIT
370 IFND = 4
GO TO 360
390 IFND = 3
GO TO 360
END
```

\$IBFTC CMPT10 LIST,REF  
SUBROUTINE COMPUT

CMPT 100

```
C
C 2-D LAGRANGIAN -- PERFORMS ALL COMPUTATIONS FOR 1 TIME CYCLE.
C
C      EQUIVALENCE      (ISPACE(2),LSTN)
C      EQUIVALENCE      (ISPACE(4),NSR)
C
C      GET VELOCITY AND POSITION OF EVERY MESH
C
C      CALL GETPOS
C
C      START COMPUTATION LOOPS
C
C      DMIN = 1.0
C      N = 0
C      NS = 0
C
C      DO 2300 K=1,M
C      JMX = JM(K)
C      NFRST(K) = 0
C      NSLST(K) = 9999
C      IENT = 0
C      NOIFCF = 0
C      IMP = 0
C      IDP = 0
C
C      DO 2290 J=1,JMX
C      NS = NS+1
C      NSR = NS
C      IMX = IM(K)
C      IMP = IMX
C
C      IF (IMX .GT. 0) GO TO 200
C      IMX = IM1(J)
C      IF (IDP .LT. 0) NSR = NSR+IDP
C      IF (J .LT. JMX) GO TO 150
C      IMP = 0
C      IDP = 0
C      GO TO 200
150 IMP = IM1(J+1)
C      IDP = IMP-IMX
C
C      200 DO 2280 I=1,IMX
C      N = N + 1
C      AM = A(N)
C      RHOM = RHO(N)
C      EM = E(N)
C
C      DEFINE VELOCITIES AND POSITIONS OF THE VERTICIES OF ZONE N.
C      10 = LOWER-LEFT CORNER (N)
C      11 = LOWER-RIGHT CORNER (N+1)
C      12 = UPPER-LEFT CORNER (N+IMX)
C      13 = UPPER RIGHT CORNER (N+IMX+1)
C
```

```

UX10 = UX1 (N)
UZ10 = UZ1 (N)
X10  = X1 (N)
Z10  = Z1 (N)
X10M = X (N)
Z10M = Z (N)
NW = N+1
UX11 = UX1 (NW)
UZ11 = UZ1 (NW)
X11  = X1 (NW)
Z11  = Z1 (NW)
X11M = X (NW)
Z11M = Z (NW)
NW = N+1MX
UX12 = UX1 (NW)
UZ12 = UZ1 (NW)
X12  = X1 (NW)
Z12  = Z1 (NW)
X12M = X (NW)
Z12M = Z (NW)
NW = NW+1
UX13 = UX1 (NW)
UZ13 = UZ1 (NW)
X13  = X1 (NW)
Z13  = Z1 (NW)
X13M = X (NW)
Z13M = Z (NW)
C
  IF (J .GE. JMX) GO TO 720
C
C   WE HAVE NOT REACHED THE FINAL ROW OF ZONES OF PLATE K.
C
  IF (I .LT. IMX) GO TO 500
C
C   WE ARE AT THE LAST ZONE IN THE ROW. RE-DEFINE RIGHT CORNERS.
C
  UX11 = UXS1 (NSR)
  UZ11 = UZS1 (NSR)
  X11  = XS1 (NSR)
  Z11  = ZS1 (NSR)
  X11M = XS (NSR)
  Z11M = ZS (NSR)
C
500 IF (I-IMP) 640,800,510
510 NS = NS+1
    GO TO 740
C
C   COMPUTE AREA AND DENSITY FOR ZONE N.
C
640 XIBAR = (X11+X12+X10)/3.0
    XOBAR = (X11+X12+X13)/3.0
    AIN1  = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
    AON1  = 0.5*((X12-X13)*(Z11-Z13) - (Z12-Z13)*(X11-X13))
    A1 (N) = AIN1 + AON1
    TEMP1 = AIN1*XIBAR**(IALF-1) + AON1*XOBAR**(IALF-1)
    RHO1 (N) = ZM(N)/TEMP1
    GO TO 2180

```

```
C
C WE HAVE REACHED THE FINAL ROW OF ZONES FOR PLATE K.
C
720 NS = NS + 1
    IF (NOSLID .EQ. 1) GO TO 740
    IF (K .LT. M) GO TO 960
C
C WE ARE WORKING ON THE LAST PLATE. THERE IS NO INTERFACE.
C RE-DEFINE UPPER CORNERS
C
740 UX12 = UXS1(NS)
    UZ12 = UZS1(NS)
    X12 = XS1(NS)
    Z12 = ZS1(NS)
    X12M = XS(NS)
    Z12M = ZS(NS)
800 NW = NS+1
    UX13 = UXS1(NW)
    UZ13 = UZS1(NW)
    X13 = XS1(NW)
    Z13 = ZS1(NW)
    X13M = XS(NW)
    Z13M = ZS(NW)
C
870 IF ((I .LT. IMX) .OR. (J .LT. JMX)) GO TO 640
C
C WE ARE AT THE LAST ZONE IN THE PLATE. RE-DEFINE LOWER-RIGHT CORNER
    UX11 = UXS1(NSR)
    UZ11 = UZS1(NSR)
    X11 = XS1(NSR)
    Z11 = ZS1(NSR)
    X11M = XS(NSR)
    Z11M = ZS(NSR)
    NS = NS+1
    GO TO 640
C
C WE ARE NOT AT THE LAST PLATE. THEREFORE, THERE IS AN INTERFACE.
C
960 IF (NS .GE. NSLST(K)) GO TO 870
C
C WE ARE ALONG THE INTERFACE.
C
    NOIFCE = 1
    CALL CMPTIF
C
    IF (NONE .EQ. 2) GO TO 740
C
C CONTINUE COMPUTATION FOR ZONE N.
C
2180 TEMP2 = RHO1(N) - RHOM
    IF (ABS(TEMP2/RHOM) .GT. REP ) GO TO 2190
    TEMP2 = 0.0
    RHO1(N) = RHOM
    A1(N) = AM
    F1(N) = EM
```



```
2190 RHOBAR = 0.5*(RHO1(N)+RHOM)
      RDOTR = TEMP2 / (DELTA*RHOBAR)
      DRR2 = TEMP2 / RHOBAR**2
      ETA1 = RHO1(N)/RHOZ(K)
      ETA = RHOM/RHOZ(K)
C
C   COMPUTE DEVIATORIC STRESSES, AND EVALUATE ARTIFICIAL VISCOSITIES.
C
C   CALL SDST
C
C   DETERMINE STRESSES AND TOTAL ENERGY.
C
C   CALL FINDP
      IF (IEND .EQ. 5) RETURN
C
C   COMPUTE NEW DELTA T.
C
C   CALL COMPDT
C
C   PERFORM ENERGY CHECKS
C
C   CALL ENRGCK
C
C   END LOOPS

2280 CONTINUE
      IF (N .GT. LSTN) GO TO 2310
2290 CONTINUE
2300 CONTINUE
2310 RETURN
      END
```

\$IBFTC GTPS40 LIST,REF  
SUBROUTINE GETPOS

C  
C 2-D LAGRANGIAN -- COMPUTES VELOCITIES AND POSITION OF ALL MESHES.

C  
EQUIVALENCE (ISPACE(1),N3)  
EQUIVALENCE (ISPACE(2),LSTN)  
EQUIVALENCE (UXM,SPACE(5)), (UZM,SPACE(6))  
EQUIVALENCE (ISPACE(4),NSR)  
EQUIVALENCE (ISPACE(9),N1D)

C  
DSUM = DELTA + DELTAM

N = 0

NS = 0

JZRO = 0

IND = 0

IMX = 0

DO 2080 K=1,M

JMX = JM(K)

IENT = 0

NOIFCE = 0

TXXSD = 0.0

TZZSD = 0.0

TXZSD = 0.0

IMP = IM(K)

IMM = 0

IDM = 0

IDP = 0

C

C START PRIMARY COMPUTATION LOOPS.

C

DO 2075 J=1,JMX

TR66-85  
Appendix C

```
      NS = NS + 1
      NSR = NS
      IMM = IMX
      IMX = IM(K)
      IF (IMX .GT. 0) GO TO 200
      IMX = IM1(J)
      IF (IDP .LT. 0) NSR = NSR+IDP
      IDP = -IDP
      IF (J .LT. JMX) GO TO 150
      IMP = 0
      IDP = 0
      GO TO 200
150  IMP = IM1(J+1)
      IDP = IMP-IMX
200  LMX = IMX
      IFLG = 0
      IZRO = 0
      DO 1670 I=1,IMX
      N = N + 1
C
C   DEFINE POSITION OF VERTICES OF ZONE N.
C
      X10 = X(N)
      Z10 = Z(N)
      NW = N+1
      X11 = X(NW)
      Z11 = Z(NW)
      NW = N+IMX
      X12 = X(NW)
      Z12 = Z(NW)
      NW = NW+1
      X13 = X(NW)
      Z13 = Z(NW)
C
      IF (J .GT. 1) GO TO 640
C
C   WE ARE WORKING ON ROW 1 OF PLATE K.
C
      IF (K .GT. 1) GO TO 610
C
C   WE ARE IN PLATE 1.
C
380  NOIFCE = 0
      IFLG = 0
      IF (I .GT. 1) GO TO 440
C
C   WE ARE AT THE LOWER LEFT CORNER.
C
      LL = NBXL(K) + NBZL(K)
      IF (NBZL(K) .EQ. 1) LL=LL-1
      AINI = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
      CALL ACRNR
      GO TO 1620
```

```
C
C WE ARE PROGRESSING ALONG ROW 1.
C
440 N2 = N-1
    X21 = X12
    Z21 = Z12
    X22 = X(N2)
    Z22 = Z(N2)
    L = 2
    LL = NBXL(K)
    IF (I .LT. IMX) GO TO 570
C
C WE ARE AT THE LAST ZONE IN ROW 1. RE-DEFINE RIGHT VERTICES.
C
    X11 = XS(NS)
    Z11 = ZS(NS)
    IF (IDP .NE. 0) GO TO 570
    NW = NS+1
    X13 = XS(NW)
    Z13 = ZS(NW)
C
570 AIN2 = 0.5*((X21-X10)*(Z22-Z10) - (Z21-Z10)*(X22-X10))
    AIN1 = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
590 CALL ABNDRY
    GO TO 1620
C
C WE HAVE FINISHED WITH PLATE 1. WE ARE STILL WORKING WITH ROW 1.
C
610 IF ((N+1 .LE. NFRST(K-1)) .OR. (N .GT. NLST(K-1))) GO TO 380
    IF (NOSLID .EQ. 0) GO TO 630
C
C WE ARE ALONG AN INTERFACE WHICH DOES NOT SLIDE.
C
    IFLG = IFLG+1
    IMM = IMM
    IF (N .NE. NLST(K-1)) GO TO 620
    IF (I .LT. LMX) GO TO 612
C
C THE RIGHT VERTICES ARE ALONG THE RIGHT BOUNDARY . RE-DEFINE.
C
    X11 = XS(NS)
    Z11 = XS(NS)
    NW = NS+1
    X13 = XS(NW)
    Z13 = ZS(NW)
612 NW = NS-IMX-2
613 X22 = XS(NW)
    Z22 = ZS(NW)
    N1 = N
    N2 = N-1
    N3 = N2-IMX
    X21 = X(N2)
    Z21 = Z(N2)
```

TR66-85  
Appendix C

```
615 LL = 1
    CALL AANGL
    GO TO 1620
620 IF ((N .NE. NFRST(K-1)) .OR. (I .EQ. 1)) GO TO 640
    N1 = N-IMX
    N2 = N
    N3 = N-1
    X21 = X12
    Z21 = Z12
    X22 = X(N3)
    Z22 = Z(N3)
    X12 = X11
    Z12 = Z11
    X11 = X(N1)
    Z11 = Z(N1)
    GO TO 615
C
C    WE ARE ALONG A SLIDING INTERFACE.
C
630 NOIFCE = 1
    CALL AIFACE
    GO TO 1620
C
C    WE HAVE FINISHED WITH ROW1.
C
640 NOIFCE = 0
    IMX = IMM
    IF (J .GE. JMX) GO TO 830
    IF (I .GT. 1) GO TO 740
C
C    WE ARE ALONG A LEFT BOUNDARY.
C
660 N2 = N - IMX
    X21 = X(N2)
    Z21 = Z(N2)
    X22 = X11
    Z22 = Z11
    L = 1
    LL = NBZL(K)
    GO TO 570
C
C    WE HAVE A ZONE IN THE INTERIOR.
C
740 IF (I .LT. LMX) GO TO 750
C
C    THE RIGHT VERTICES ARE ALONG THE RIGHT BOUNDARY . RE-DEFINE.
C
    X11 = XS(NSR)
    Z11 = ZS(NSR)
C
750 IF (IDP .GT. 0) GO TO 760
    NW = NS+1
    X13 = XS(NW)
```

```
      713 = ZS(NW)
      IF (I-IMP-1) 760,752,970
752  NS = NS+1
      GO TO 1300
760  IF (IDM) 770,800,800
770  IF (I-IMM-1) 800,780,440
780  NW = NS-1
      GO TO 613
800  AIN1 = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
810  CALL AINTR
      GO TO 1620
```

C  
C WE ARE NOW WORKING ON THE LAST ROW IN PLATE K.

C  
C 830 IF (K .GE. M) GO TO 970

C  
C WE ARE NOT AT THE FINAL PLATE.

C  
C  
C NOIFCF = 1  
C NS = NS + 1  
C X12 = XS(NS)  
C Z12 = ZS(NS)  
C NW = NS+1  
C X13 = XS(NW)  
C Z13 = ZS(NW)

C  
C IF (I .LT. LMX) GO TO 950

C  
C WE ARE AT A RIGHT BOUNDARY. RE-DEFINE.

C  
C  
C NW = NS-IMX  
C X11 = XS(NW)  
C Z11 = ZS(NW)  
C GO TO 800

C  
C 950 IF (I .EQ. 1) GO TO 660  
C GO TO 800

C  
C WE HAVE REACHED THE FINAL PLATE. WE ARE STILL WORKING ON THE LAST  
C ROW.

C  
C 970 NS = NS+1

C  
C WE WILL FIND THE POSITIONS OF THE BOUNDARY MESH AND FINAL ROW MESH  
C CONCURRENTLY.  
C BOUNDARY MESH.

C  
C  
C X11 = X10  
C Z11 = Z10  
C X10 = XS(NS)  
C Z10 = ZS(NS)  
C NW = NS+1  
C X12 = XS(NW)

TR66-85  
Appendix C

```
      Z12 = ZS(NW)
      AIN1 = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
C
      IF (I .GT. 1) GO TO 1210
C
      WE ARE AT THE UPPER LEFT CORNER
C
      LL = NBXU(K) + NBZL(K)
      IF (NBZL(K) .EQ. 1) LL=LL-1
      CALL ACRNR
      CALL EXTPOS
C
      SET-UP FOR FINAL ROW MESH. LEFT BOUNDARY.
C
      X10 = X11
      Z10 = Z11
      NW = N+1
      X11 = X(NW)
      Z11 = Z(NW)
      X13 = X12
      Z13 = Z12
      X12 = XS(NS)
      Z12 = ZS(NS)
      GO TO 660
C
      PROCEED ALONG THE FINAL ROW OF LAST PLATE.
      BOUNDARY MESH
C
1210  NW = NS-1
      X21 = XS(NW)
      Z21 = ZS(NW)
      X22 = X11
      Z22 = Z11
      N2 = N-1
      L = 2
      LL = NBXU(K)
      AIN2 = 0.5*((X21-X10)*(Z22-Z10) - (Z21-Z10)*(X22-X10))
      CALL ABNDRY
      CALL EXTPOS
C
1300  IF (I .GE. LMX) GO TO 1370
      NW = N+1
      X11 = X(NW)
      Z11 = Z(NW)
      GO TO 1530
C
      WE ARE AT THE UPPER RIGHT CORNER OF PLATE K.
C
1370  X11 = XS(NS)
      Z11 = ZS(NS)
      NS = NS+1
      X10 = XS(NS)
      Z10 = ZS(NS)
```

```
NW = NSD
X12 = XS(NW)
Z12 = ZS(NW)
AIN1 = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
LL = NBXU(K) + NBZR(K)
IF (NBZR(K) .EQ. 1) LL=LL-1
CALL ACRNR
CALL FXTPOS
C
C   SET-UP FOR FINAL ROW MESH.
C
X11 = X12
Z11 = Z12
NS = NS-1
1530 X10 = X(N)
Z10 = Z(N)
X12 = XS(NS)
Z12 = ZS(NS)
NW = NS+1
X13 = XS(NW)
Z13 = ZS(NW)
GO TO 760
C
C   COMPUTE NEW VELOCITIES AND POSITIONS FOR MESH N.
C
1620 UXM = UX(N)
UZM = UZ(N)
UX1(N) = UX(N) + 0.5*AX*DSUM
UZ1(N) = UZ(N) + 0.5*AZ*DSUM
TEMP1 = UEP
IF (N1D .GT. 0) TEMP1 = 1.0E+10
IF (ABS(UX1(N)) .LT. TEMP1) UX1(N) = 0.0
IF (ABS(UZ1(N)) .LT. UEP) UZ1(N) = 0.0
X1(N) = X(N) + UX1(N)*DELTA
Z1(N) = Z(N) + UZ1(N)*DELTA
IMX = LMX
C
C   IF (NOSLID .EQ. 0) GO TO 1640
C   IF (IFLG .EQ. 0) GO TO 1660
C
C   THERE IS AN INTERFACE WHICH DOES NOT SLIDE
C
NW = NS+IFLG-IMM-2
UXS1(NW) = UX1(N)
UZS1(NW) = UZ1(N)
XS1(NW) = X1(N)
ZS1(NW) = Z1(N)
GO TO 1660
C
C   IF THERE IS AN INTERFACE, DETERMINE TANGENT OF LINE THROUGH MESH N.
C
1640 IF ((J .LT. JMX) .OR. (NOIFCE .EQ. 0)) GO TO 1660
CALL DETTN
```



TR66-85  
Appendix C

```
C
1660 IF (UX1(N) .NE. 0.0) GO TO 1661
      IF (UZ1(N) .EQ. 0.0) GO TO 1670
1661 IND = 1
      IZRO = 1
      JZRO = 0
C
C      END I LOOP
C
1670 CONTINUE
C
C      TAKE CARE OF RIGHT BOUNDARY MESHES.
C      ROTATE REFERENCE FRAME OF LAST MESH BY 90 DEG.
C
      NSC = NS
      NS = NSR
      X12 = X10
      Z12 = Z10
      X10 = X11
      Z10 = Z11
      X11 = X13
      Z11 = Z13
      IF (J .GT. 1) GO TO 1750
C
C      WE ARE AT THE LOWER RIGHT CORNER MESH.
C
      IF (K .EQ. 1) GO TO 1720
      IF ((NOIFCE .EQ. 0) .OR. (N .GE. NLST(K-1))) GO TO 1720
      I = I+1
      CALL AIFACE
      CALL EXTPOS
      GO TO 2070
C
1720 IF (N .LT. NLST(K-1)) GO TO 1780
      LL = NBXL(K) + NBZR(K)
      IF (NBZR(K) .EQ. 1) LL=LL-1
1730 AIN1 = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
      CALL ACRNR
1735 IF (J .LT. JMX) GO TO 1740
      NSC = NSC+1
      NW = NSC
      GO TO 2040
1740 CALL EXTPOS
      GO TO 2070
C
C      WE ARE ALONG RIGHT BOUNDARY.
C
1750 IF (IDM) 1751,1810,1752
1751 LL = 1
      GO TO 1730
1752 N1 = N
      N2 = N-1MM
      N3 = N2+1
```

X21 = X(N3)  
Z21 = Z(N3)  
NW = NS+1  
X22 = XS(NW)  
Z22 = ZS(NW)  
LL = 1  
CALL AANGL  
GO TO 1735

1780 LMX = IMX  
IMX = IMM

1810 N2 = N-IMX  
X21 = X12  
Z21 = Z12  
L = 1  
LL = NR7P(K)

C  
IF (J .GE. JMX) GO TO 1960  
IF ((J .GT. 1) .OR. (N .GE. NLST(K-1))) GO TO 1860  
IF (IMX .GT. LMX) GO TO 1840  
NW = NS-IMX-2  
GO TO 1870

1840 NW = N2+1  
X22 = X(NW)  
Z22 = Z(NW)  
GO TO 1890

1860 NW = NS-1  
1870 X22 = XS(NW)  
Z22 = ZS(NW)

1890 X10 = XS(NS)  
Z10 = ZS(NS)  
NW = NS+1  
X11 = XS(NW)  
Z11 = ZS(NW)  
NW = NS  
GO TO 2030

C  
C WE ARE NEXT TO THE UPPER RIGHT CORNER.  
C

1960 NSC = NSC+1  
NS = NSC  
NW = NS  
NS = NS-IMX-2  
X22 = XS(NS)  
Z22 = ZS(NS)  
NS = NS+1  
X10 = XS(NS)  
Z10 = ZS(NS)  
X11 = XS(NW)  
Z11 = ZS(NW)

C  
2030 AIN1 = 0.5\*((X11-X10)\*(Z12-Z10) - (Z11-Z10)\*(X12-X10))  
AIN2 = 0.5\*((X12-X10)\*(Z22-Z10) - (Z12-Z10)\*(X22-X10))  
CALL ABNDRY

TR66-85  
Appendix C

```
2040 UXM = UXS(NS)
      UZM = UZS(NS)
      CALL EXTPOS
      IF ((J .GT. 1) .OR. (N .GE. NLST(K-1))) GO TO 2050
      N1 = NS-1
      UXS1(N1) = UXS1(NS)
      UZS1(N1) = UZS1(NS)
      XS1(N1) = XS1(NS)
      ZS1(N1) = ZS1(NS)
2050 NS = NW
      IMX = LMX
C
      IF (NOSLID .EQ. 1) GO TO 2070
      IF ((J .LT. JMX) .OR. (NOIFCE .EQ. 0)) GO TO 2070
      I = I+1
      CALL DETTN
C
2070 NS = NSC
      IF ((K .EQ. 1) .AND. (M .GT. 1)) GO TO 2075
      IF (NCYCLE .EQ. 1) GO TO 2075
      IF (UXS1(NS) .NE. 0.0) GO TO 2071
      IF (UZS1(NS) .EQ. 0.0) GO TO 2072
2071 IND = 1
      IZRO = 1
      JZRO = 0
      GO TO 2075
2072 IF (IND .EQ. 0) GO TO 2075
      IF (IZRO .NE. 0) GO TO 2075
      JZRO = JZRO+1
      IF (JZRO .LT. 2) GO TO 2075
      GO TO 2100
C
C      END J LOOP
C
2075 CONTINUE
C
C      END K LOOP
C
2080 CONTINUE
2100 LSTN = N
      RETURN
      END
```

```
$IBFTC EXTPOS LIST,REF
SUBROUTINE EXTPOS                                XTP 100

C 2-D LAGRANGIAN -- COMPUTES VELOCITY AND POSITION OF A BOUNDARY MESH
C
C EQUIVALENCE      (ISPACE(9),NID)
C
UXS1(NS) = UXS(NS) + 0.5*AX*DSUM                XTP 110
UZS1(NS) = UZS(NS) + 0.5*A7*DSUM                XTP 120
TEMP1 = UEP                                     XTP 121
IF (NID .GT. 0) TEMP1 = 1.0E+10                 XTP 122
IF (ABS(UXS1(NS)) .LT. TEMP1) UXS1(NS)=0.0      XTP 123
IF (ABS(UZS1(NS)) .LT. UEP) UZS1(NS)=0.0       XTP 124
XS1(NS) = XS(NS) + UXS1(NS)*DELTA              XTP 130
ZS1(NS) = ZS(NS) + UZS1(NS)*DELTA              XTP 140
RETURN                                          XTP 150
END                                             XTP 160
```

```

$IBFTC AINTR1 LIST,REF
SUBROUTINE AINTR
                                                    AINT 100

C 2-D LAGRANGIAN -- COMPUTES THE ACCELERATION OF AN INTERIOR MESH
C
N2 = N-1
                                                    AINT 110
X21 = X(N2)
                                                    AINT 120
Z21 = Z(N2)
                                                    AINT 130
N4 = N-IMX
                                                    AINT 140
X22 = X(N4)
                                                    AINT 150
Z22 = Z(N4)
                                                    AINT 160
N3 = N4-1
                                                    AINT 170
AIN2 = 0.5*((X12-X10)*(Z21-Z10) - (Z12-Z10)*(X21-X10))
                                                    AINT 180
AIN3 = 0.5*((X21-X10)*(Z22-Z10) - (Z21-Z10)*(X22-X10))
                                                    AINT 190
AIN4 = 0.5*((X22-X10)*(Z11-Z10) - (Z22-Z10)*(X11-X10))
                                                    AINT 200
TEMP1 = AIN1*RHO(N) + AIN2*RHO(N2) + AIN3*RHO(N3) + AIN4*RHO(N4)
                                                    AINT 210
TEMP2 = 0.25*(ALF-1.0)
                                                    AINT 220
IF (TEMP2 .NE. 0.0) GO TO 270
                                                    AINT 230
TEMP3 = 0.0
                                                    AINT 240
TEMP4 = 0.0
                                                    AINT 250
GO TO 320
                                                    AINT 260
270 NW = IALF-1
                                                    AINT 270
TEMP5 = RHO(N)*(0.33333333*(X10+X11+X12))**NW
                                                    AINT 271
TEMP6 = RHO(N2)*(0.33333333*(X10+X12+X21))**NW
                                                    AINT 272
TEMP7 = RHO(N3)*(0.33333333*(X10+X21+X22))**NW
                                                    AINT 273
TEMP8 = RHO(N4)*(0.33333333*(X10+X22+X11))**NW
                                                    AINT 274
TEMP3 = TEMP2*((TXX(N)-TYY(N))/TEMP5 + (TXX(N2)-TYY(N2))/TEMP6 +
1(TXX(N3)-TYY(N3))/TEMP7 + (TXX(N4)-TYY(N4))/TEMP8)
                                                    AINT 280
TEMP4 = TEMP2*(TXZ(N)/TEMP5 + TXZ(N2)/TEMP6 + TXZ(N3)/TEMP7 + TXZ(N4)/TEMP8)
                                                    AINT 300
320 AZ = -(TXZ(N)*(Z11-Z12) + TXZ(N2)*(Z12-Z21) + TXZ(N3)*(Z21-Z22) +
1TXZ(N4)*(Z22-Z11))/TEMP1 + (TZZ(N)*(X11-X12) + TZZ(N2)*(X12-X21) +
2TZZ(N3)*(X21-X22) + TZZ(N4)*(X22-X11))/TEMP1 + TEMP4
                                                    AINT 320
AX = -(TXX(N)*(Z11-Z12) + TXX(N2)*(Z12-Z21) + TXX(N3)*(Z21-Z22) +
1TXX(N4)*(Z22-Z11))/TEMP1 + (TXZ(N)*(X11-X12) + TXZ(N2)*(X12-X21) +
2TXZ(N3)*(X21-X22) + TXZ(N4)*(X22-X11))/TEMP1 + TEMP3
                                                    AINT 360
RETURN
                                                    AINT 380
END
                                                    AINT 390

```

\$IBFTC ACRNR5 LIST,REF  
SUBROUTINE ACRNR

```
C
C 2-D LAGRANGIAN -- COMPUTES THE ACCELERATION OF A CORNER MESH
C          LL = 1 - Z FREE, X FREE
C          = 2 - Z FREE, X FIXED
C          = 3 - Z FIXED, X FREE
C          = 4 - Z FIXED, X FIXED
C
C          EQUIVALENCE      (SINB,SPACE(3)),(COSB,SPACE(4))
C
C          ZRO = 0.0000005
C          ONE = 0.9999995
C
C          IF (LL .LT. 4) GO TO 150
120  AZ = 0.0
      AX = 0.0
      GO TO 290
150  TEMP1 = AIN1*RHO(N)
      TEMP2 = (ALF-1.0)/(RHO(N)*(0.33333333*(X10+X11+X12))**(IALF-1))
      TEMP3 = X11-X12
      TEMP4 = Z11-Z12
      GO TO (200,210,210,120),LL
200  AZ = (-TXZ(N)*TEMP4 + TZZ(N)*TEMP3)/TEMP1 + TEMP2*TXZ(N)
      AX = (-TXX(N)*TEMP4 + TXZ(N)*TEMP3)/TEMP1 + TEMP2*(TXX(N)-TYY(N))
      GO TO 290
210  IF (IALF .LT. 2) GO TO 260
C
C          WHEN CYLINDRICAL SYMMETRY IS REQUESTED, A FIXED BOUNDARY MAY ONLY
C          BE AT 0.0 OR 90.0 DEG.
C
C          GO TO (200,242,244,120),LL
242  IF (J .GT. 1) GO TO 243
      TEMP9 = PSIL(K)
      GO TO 250
243  TEMP9 = PSIU(K)
      GO TO 250
244  IF (I .GT. 1) GO TO 245
      TEMP9 = PHIL(K)
      GO TO 250
245  TEMP9 = PHIR(K)
250  IF (TEMP9 .GT. 0.7854) GO TO 255
      SINB = 0.0
      COSB = 1.0
      AZ = 0.0
      AX = (-TXX(N)*TEMP4 + TXZ(N)*TEMP3)/TEMP1 + TEMP2*(TXX(N) - TY
1Y(N))
      GO TO 290
255  AZ = (-TXZ(N)*TEMP4 + TZZ(N)*TEMP3)/TEMP1
      AX = 0.0
      SINB = 1.0
      COSB = 0.0
      GO TO 290
```

```
C
C      PLANE GEOMETRY REQUESTED.
C
260 GO TO (200,262,270,120), LL
262 IF (J .GT. 1) GO TO 266
    TEMP9 = PSIL(K)+1.5707963
    GO TO 272
266 TEMP9 = PSIU(K)+1.5707963
    GO TO 272
270 IF (I .GT. 1) GO TO 271
    TEMP9 = PHIL(K)
    GO TO 272
271 TEMP9 = PHIR(K)
272 COSB = COS(TEMP9)
    IF (ABS(COSB) .GT. ONE) COSB = SIGN( 1.0,COSB)
    IF (ABS(COSB) .LT. ZRO) COSB = 0.0
    COSB2 = COSB**2
    SINB = SIN(TEMP9)
    IF (ABS(SINB) .GT. ONE) SINB = SIGN( 1.0,SINB)
    SINB2 = SINB**2
    IF (ABS(SINB) .LT. ZRO) SINB = 0.0
    TIIN = TXX(N)*SINB2+TZZ(N)*COSB2-2.0*TXZ(N)*SINB*COSB
    TJJN = TXX(N)*COSB2+TZZ(N)*SINB2+2.0*TXZ(N)*SINB*COSB
    TIJN = -SINB*COSB*(TZZ(N)-TXX(N))+TXZ(N)*(SINB2-COSB2)
    TEMP4 = (X11-X12)*COSB + (Z11-Z12)*SINB
    TEMP3 = (X11-X12)*SINB - (Z11-Z12)*COSB
    GO TO (200,284,286,120),LL
284 AW = (-TIIN*TEMP4 + TIJN*TEMP3)/TEMP1
    TEMP10 = SINB
    SINB = -COSB
    COSB = TEMP10
    GO TO 287
286 AW = (-TIJN*TEMP4 + TJJN*TEMP3)/TEMP1
287 AZ = AW*SINB
    AX = AW*COSB
290 RETURN
    END
```

§IBFTC ABNDR5 LIST,REF  
SUBROUTINE ABNDRY

```

C
C 2-D LAGRANGIAN -- COMPUTES THE ACCELERATION OF A BOUNDARY MESH
C          LL = 1 - DENOTES FREE BOUNDARY
C          = 2 - DENOTES FIXED BOUNDARY
C
C          L = 1 - PARALLEL TO Z-AXIS
C          = 2 - PARALLEL TO X-AXIS
C
C
C      ZR0 = 0.0000005
C      ONE = 0.9999995
C
C      TEMP1 = AIN1*RHO(N) + AIN2*RHO(N2)
C      TEMP2 = 0.50*(ALF - 1.0)
C      TEMP3 = X11-X12
C      TEMP4 = Z11-Z12
C      TEMP5 = X21-X22
C      TEMP6 = Z21-Z22
C      NW = 1/ALF-1
C      TEMP7 = 1.0/(RHO(N)*(0.33333333*(X10+X11+X12))*NW)
C      TEMP8 = 1.0/(RHO(N2)*(0.33333333*(X10+X22+X21))*NW)
C      GO TO (200,240),LL
200  AZ = -((TXZ(N)*TEMP4 + TXZ(N2)*TEMP6)/TEMP1) + ((TZZ(N)*TEMP3 + TZ
1Z(N2)*TEMP5)/TEMP1) + TEMP2*(TXZ(N)*TEMP7 + TXZ(N2)*TEMP8)
      AX = -((TXX(N)*TEMP4 + TXX(N2)*TEMP6)/TEMP1) + ((TXZ(N)*TEMP3 + TX
1Z(N2)*TEMP5)/TEMP1) + TEMP2*((TXX(N)-TYY(N))*TEMP7 + (TXX(N2)-TYY(
2N2))*TEMP8)
      GO TO 350
240  IF (IALF .LT. 2) GO TO 260
C
C      WHEN CYLINDRICAL SYMMETRY IS REQUESTED, A FIXED BOUNDARY MAY ONLY
C      BE AT 0.0 OR 90.0 DEG.
C
C
C      GO TO (244,242),L
242  IF (J .GT. 1) GO TO 243
      TEMP9 = PSIL(K)
      GO TO 250
243  TEMP9 = PSIU(K)
      GO TO 250
244  IF (I .GT. 1) GO TO 245
      TEMP9 = PHIL(K)
      GO TO 250
245  TEMP9 = PHIR(K)
250  IF (TEMP9 .GT. 0.7854) GO TO 255
C
      AZ = 0.0
      AX = -(((TXX(N)-TXX(N2))*TEMP4)/TEMP1) + ((TXZ(N)*TEMP3 + TXZ(N2)*
1TEMP5)/TEMP1) + TEMP2*((TXX(N)-TYY(N))*TEMP7 + (TXX(N2)-TYY(N2)
2))*TEMP8)
      GO TO 350
255  AZ = -((TXZ(N)*TEMP4 + TXZ(N2)*TEMP6)/TEMP1) + ((TZZ(N)*TEMP3 + TZ
1Z(N2)*TEMP5)/TEMP1)
      AX = 0.0
      GO TO 350

```



```
C
C      PLANE GEOMETRY REQUESTED.
C
260 GO TO (270,261),L
261 IF (J .GT. 1) GO TO 264
    TEMP9 = PSIL(K)+1.5707963
    GO TO 272
264 TEMP9 = PSIU(K)+1.5707963
    GO TO 272
270 IF (I .GT. 1) GO TO 271
    TEMP9 = PHIL(K)
    GO TO 272
271 TEMP9 = PHIR(K)
272 COSB = COS(TEMP9)
    IF (ABS(COSB) .GT. ONE) COSB = SIGN( 1.0,COSB)
    IF (ABS(COSB) .LT. ZRO) COSB = 0.0
    COSB2 = COSB**2
    SINB = SIN(TEMP9)
    IF (ABS(SINB) .LT. ZRO) SINB = 0.0
    IF (ABS(SINB) .GT. ONE) SINB = SIGN( 1.0,SINB)
    SINB2 = SINB**2
    TIIN = TXX(N)*SINB2+TZZ(N)*COSB2-2.0*TXZ(N)*SINB*COSB
    TJJN = TXX(N)*COSB2+TZZ(N)*SINB2+2.0*TXZ(N)*SINB*COSB
    TIJN =-SINB*COSB*(TZZ(N)-TXX(N))+TXZ(N)*(SINB2-COSB2)
    TIIN2 = TXX(N2)*SINB2+TZZ(N2)*COSB2-2.0*TXZ(N2)*SINB*COSB
    TJJN2 = TXX(N2)*COSB2+TZZ(N2)*SINB2+2.0*TXZ(N2)*SINB*COSB
    TIJN2 =-SINB*COSB*(TZZ(N2)-TXX(N2))+TXZ(N2)*(SINB2-COSB2)
    TEMP4 = (X11-X12)*COSB + (Z11-Z12)*SINB
    TEMP6 = (X21-X22)*COSB + (Z21-Z22)*SINB
    TEMP3 = (X11-X12)*SINB - (Z11-Z12)*COSB
    TEMP5 = (X21-X22)*SINB - (Z21-Z22)*COSB
    GO TO (291,310),L

291 AW = -((TIJN*TEMP4 + TIJN2*TEMP6)/TEMP1) + ((TJJN*TEMP3 + TJJN2*
1TEMP5)/TEMP1)
    GO TO 330
310 AW = -(((TIIN-TIIN2)*TEMP4)/TEMP1) + ((TIJN*TEMP3 + TIJN2*TEMP5)/
1TEMP1)
    TEMP10 = SINB
    SINB = -COSB
    COSB = TEMP10
330 AZ = AW *SINB
    AX = AW *COSB
350 RETURN
    END
```

\$IBFTC AANGL2 LIST,REF  
SUBROUTINE AANGL

```

C
C 2-D LAGRANGIAN -- COMPUTES THE ACCELERATION OF A MESH AT THE BOUNDARY
C OF TWO PLATES OF UNEQUAL LENGTH. THAT IS, THEY FORM
C AN ANGLE.
C LL = 1 - DENOTES FREE BOUNDARY
C = 2 - DENOTES FIXED BOUNDARY
C
C EQUIVALENCE (ISPACE(1),N3)
C
GO TO (120,320),LL AANG 110
120 AIN1 = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10)) AANG 120
AIN2 = 0.5*((X12-X10)*(Z21-Z10) - (Z12-Z10)*(X21-X10)) AANG 130
AIN3 = 0.5*((X21-X10)*(Z22-Z10) - (Z21-Z10)*(X22-X10)) AANG 140
TEMP1 = AIN1*RHO(N1) + AIN2*RHO(N2) + AIN3*RHO(N3) AANG 150
TEMP2 = 0.33333333*(ALF-1.0) AANG 160
IF (TEMP2 .NE. 0.0) GO TO 210 AANG 170
TEMP3 = 0.0 AANG 180
TEMP4 = 0.0 AANG 190
GO TO 250 AANG 200
210 NW = IALF-1 AANG 210
TEMP5 = RHO(N1)*(0.33333333*(X10+X11+X12))**NW AANG 211
TEMP6 = RHO(N2)*(0.33333333*(X10+X12+X21))**NW AANG 212
TEMP7 = RHO(N3)*(0.33333333*(X10+X21+X22))**NW AANG 213
TEMP3 = TEMP2*((TXX(N1)-TYY(N1))/TEMP5 + (TXX(N2)-TYY(N2))/TEMP6 + AANG 220
1(TXX(N3)-TYY(N3))/TEMP7) AANG 230
TEMP4 = TEMP2*(TXZ(N1)/TEMP5 + TXZ(N2)/TEMP6 + TXZ(N3)/TEMP7) AANG 240
250 AX = -(TXX(N1)*(Z11-Z12) + TXX(N2)*(Z12-Z21) + TXX(N3)*(Z21-Z22))/AANG 250
1TEMP1 + (TXZ(N1)*(X11-X12) + TXZ(N2)*(X12-X21) + TXZ(N3)*(X21-X22)AANG 260
2)/TEMP1 + TEMP3 AANG 270
AZ = -(TXZ(N1)*(Z11-Z12) + TXZ(N2)*(Z12-Z21) + TXZ(N3)*(Z21-Z22))/AANG 280
1TEMP1 + (TZZ(N1)*(X11-X12) + TZZ(N2)*(X12-X21) + TZZ(N3)*(X21-X22)AANG 290
2)/TEMP1 + TEMP4 AANG 300
GO TO 340 AANG 310
320 AX = 0.0 AANG 320
AZ = 0.0 AANG 330
340 RETURN AANG 340
END AANG 350

```

```
$IBFTC AIFC40 LIST,REF
      SUBROUTINE AIFACE
C
C 2-D LAGRANGIAN -- DETERMINES ACCELERATIONS OF MESHES ON THE INTERFACE
C                   BETWEEN TWO PLATES.
C
C      EQUIVALENCE      (SINB,SPACE(3)),(COSB,SPACE(4))
C
C      KM = K-1
C      IF (I .GT. IMX) GO TO 820
C
C      IF (I .LT. IMX) GO TO 100
C
C      X11 = XS(NS)
C      Z11 = ZS(NS)
C      NW = NS + 1
C      X13 = XS(NW)
C      Z13 = ZS(NW)
C
C 100 XRT = 0.25*(X11+X13+X12+X10)
C      ZRT = 0.25*(Z11+Z13+Z12+Z10)
C      TEMP10 = Z11-Z10
C      IF (TEMP10 .NE. 0.0) GO TO 120
C      TNRT = SIGN(1.0E+8,(-(X11-X10)))
C      GO TO 130
C 120 TNRT = -(X11-X10)/TEMP10
C
C 130 IF (IENT .NE. 0) GO TO 170
C
C      UPON INTIAL ENTRY, SET-UP FLAGS.
C
C      LK = 1
```

```
      NL = 1
      TNP = +1.0
C
C 170 CALL FDLPTS
      IF (NONE .EQ. 1) GO TO 350
C
C     THE PROPER POINTS COULD NOT BE FOUND.
C
      IF (IM(KM) .GT. 0) GO TO 180
      KM2 = JM(KM)-1
      NW = 0
      DO 175 L=1,KM2
175  NW = NW + IM1(L)
      IML = IM1(KM2+1)
      GO TO 190
180  NW = IM(KM)*(JM(KM)-1) + 1
      IML = IM(KM)
190  IF (KM .LE. 1) GO TO 210
      KM2 = KM-1
      DO 200 L=1,KM2
200  NW = NW + IM(L)*JM(L)
210  IF (I .GT. 1) GO TO 250
C
      TXXSD = TXX(NW)
      TZZSD = TZZ(NW)
      TXZSD = TXZ(NW)
      ZMS = ZM(NW)
      IENT = 0
      GO TO 470
C
C 250 IF (IENT .NE. 0) GO TO 300
C
      TXXSE = TXX(NW)
      TZZSE = TZZ(NW)
      TXZSE = TXZ(NW)
      GO TO 620
C
C 300 NW = NW + IML - 1
      ISR = ISS
      ISS = NW
      TXXSD = TXXSE
      TZZSD = TZZSE
      TXZSD = TXZSE
      IF (N .GE. NLST(KM)) GO TO 320
C
      DS = SQRT((XS(NS-1)-X10   )**2 + (ZS(NS-1)-Z10   )**2)
      RS = SQRT((X10-X11   )**2 + (Z10-Z11   )**2)
      TEMP1 = DS/RS
      IF (TEMP1 .LE. 1.0E-4) TEMP1 = 0.0
      TXXSE = TXX(NW)*TEMP1
      TZZSE = TZZ(NW)*TEMP1
      TXZSE = TXZ(NW)*TEMP1
      GO TO 620
```

```
C
320 TXXSE = 0.0
    TZZSE = 0.0
    TXZSE = 0.0
    GO TO 620

C
C   POINTS WERE FOUND ON EITHER SIDE OF ZONE N.
C
350 NSS = NL
    ISS = NH(NSS,KM)
    IF (NL .GT. 1) GO TO 380
    ISR = ISS
    DR = 1.0
    DS = 0.0
    RS = 1.0
    GO TO 450
380 NSR = NL-1
    ISR = NH(NSR,KM)
    DR = SQRT((XSD-XH(NSR,KM))**2 + (ZSD-ZH(NSR,KM))**2)
    DS = SQRT((XSD-XH(NSS,KM))**2 + (ZSD-ZH(NSS,KM))**2)
    RS = SQRT((XH(NSR,KM)-XH(NSS,KM))**2 + (ZH(NSR,KM)-ZH(NSS,KM))**2)

C
450 IF (I .GT. 1) GO TO 550

C
    TXXSD = (TXX(ISR)*DS + TXX(ISS)*DR)/RS
    TZZSD = (TZZ(ISR)*DS + TZZ(ISS)*DR)/RS
    TXZSD = (TXZ(ISR)*DS + TXZ(ISS)*DR)/RS
    ZMS = ZM(ISS)

C
470 LL = 1
    IF (NBZL(K) .EQ. 1) GO TO 500
    LL = 3
    TEMP10 = PHIL(K)
500 XQ = X11
    7Q = 711
    XA = X10
    ZA = Z10
    AIN1 = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
    CALL ACRNR
    TEMP4 = (X10-X11)*COSB + (Z10-Z11)*SINB
    TEMP3 = (X10-X11)*SINB - (Z10-Z11)*COSB
    AW = AZ/SINB
    AW = AW - (TXZSD*TEMP4 - TZZSD*TEMP3)/TEMP1
    AZ = AW*SINB
    AX = AW*COSB
    GO TO 990

C
550 IF (I .LE. 2) GO TO 590
    TXXSD = TXXSE
    TZZSD = TZZSE
    TXZSD = TXZSE

C
590 TXXSE = (TXX(ISR)*DS + TXX(ISS)*DR)/RS
```

```
TZZSE = (TZZ(ISR)*DS + TZZ(ISS)*DR)/RS
TXZSE = (TXZ(ISR)*DS + TXZ(ISS)*DR)/RS
```

C

```
620 L = 2
    LL = 1
    N2 = N-1
    X22 = X(N2)
    Z22 = Z(N2)
    X21 = X12
    Z21 = Z12
    XQ = X11
    ZQ = Z11
    XA = X22
    NW = NS-IM(K-1)-2+1
    ZA = Z22
    IF (ABS((X10-XS(NW))/X10) .GT. 1.0E-4) GO TO 700
    ZMS = ZM(ISR)
    GO TO 750
700 ZMS = 0.5*(ZM(ISR)+ZM(ISS))
750 AIN2 = 0.5*((X21-X10)*(Z22-Z10) - (Z21-Z10)*(X22-X10))
    AIN1 = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
    CALL ABNDRY
    AX = AX - ((TXXSD*(Z22-Z10)+TXXSE*(Z10-Z11))/TEMP1) + ((TXZSD*(X2
12-X10)+TXZSE*(X10-X11))/TEMP1)
    AZ = AZ - ((TXZSD*(Z22-Z10)+TXZSE*(Z10-Z11))/TEMP1) + ((TZZSD*(X2
12-X10)+TZZSE*(X10-X11))/TEMP1)
    GO TO 990
```

C

```
820 X10 = XS(NS)
    Z10 = ZS(NS)
    NW = NS+1
    X11 = XS(NW)
    Z11 = ZS(NW)
    X12 = X(N)
    Z12 = Z(N)
    XQ = X10
    ZQ = Z10
    XA = X12
    ZA = Z12
    ZMS = ZM(ISS)
    LL = 1
    IF (NBZR(K) .EQ. 1) GO TO 940
    LL = 3
    TEMP10 = PHIR(K)
940 AIN1 = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
    CALL ACRNR
    TEMP4 = (X12-X10)*COSB + (Z12-Z10)*SINB
    TEMP3 = (X12-X10)*SINB - (Z12-Z10)*COSB
    AW = AZ/SINB
    AW = AW - (TXZSE*TEMP4 - TZZSE*TEMP3)/TEMP1
    AZ = AW*SINB
    AX = AW*COSB
```

C

```
990 IF (LL .NE. 3) GO TO 1000
    SINAP = SIN(TEMP10-1.5707963)
    GO TO 1060
1000 TEMP4 = XQ-XA
    TEMP5 = ZQ-ZA
    IF (ABS(TEMP5/TEMP4) .GE. 1.0E-4) GO TO 1040
1020 SINAP = 0.0
    GO TO 1060
1040 TEMP6 = SQRT(TEMP4**2 + TEMP5**2)
    SINAP = TEMP5/TEMP6
1060 COSAP = SQRT(1.0 - SINAP**2)
    AXW = AX*COSAP + AZ*SINAP
    AZW = ZM(N)/(ZM(N)+ZMS)*(-AX*SINAP + AZ*COSAP)
    AX = AXW*COSAP - AZW*SINAP
    AZ = AXW*SINAP + AZW*COSAP
    RETURN
    END
```

\$IBFTC DTTN10 LIST,REF  
SUBROUTINE DETTN

C  
C 2-D LAGRANGIAN -- DETERMINES THE TANGENT OF THE LINE THROUGH MESH N.

C  
C EQUIVALENCE (ISPACE(4),NSR)

C  
C IF (I .GT. IMX) GO TO 1940  
C AIN1 = A(N) - 0.5\*((X13-X11)\*(Z10-Z11) - (Z13-Z11)\*(X10-X11))  
C IF (I .GT. 1) GO TO 1750

C  
C WE ARE AT THE UPPER LEFT CORNER ZONE OF PLATE K.

C  
C X11 = X10  
C Z11 = Z10  
C X12 = X13  
C Z12 = Z13  
C IF (NBZL(K) .EQ. 1) GO TO 1670  
C LL = 3  
C GO TO 1680

1670 LL = 1  
1680 CALL ACRNR  
C TEMP1 = X1(N)  
C TEMP2 = Z1(N)  
C CALL GTTN

1700 X11 = X(N+1)  
C Z11 = Z(N+1)  
C X12 = XS(NS)  
C Z12 = ZS(NS)  
C GO TO 2170

C  
C WE ARE ALONG THE UPPER BOUNDARY OF PLATE K.

C  
C 1750 X11 = X10  
C Z11 = Z10  
C X12 = X13  
C Z12 = Z13  
C NW = NS-1  
C X21 = XS(NW)  
C Z21 = ZS(NW)  
C X22 = X11  
C Z22 = Z11  
C N2 = N-1  
C L = 2  
C LL = 1  
C AIN2 = A(N2) - 0.5\*((X22-X(N2))\*(Z21-Z(N2)) - (Z22-Z(N2))\*(X21-X(N  
C 12)))  
C CALL ABNDRY  
C TEMP1 = X1(N)  
C TEMP2 = Z1(N)  
C CALL GTTN  
C GO TO 1700

C  
C WE ARE AT THE LAST MESH IN THE PLATE.



```
C
1940 X11 = XS(NS-1)
      Z11 = ZS(NS-1)
      X12 = X10
      Z12 = Z10
      AIN1 = A(N) - 0.5*((X12-X(N))*(Z11-Z(N)) - (Z12-Z(N))*(X11-X(N)))
      IF (NBZR(K) .EQ. 1) GO TO 2020
      LL = 3
      GO TO 2030
2020 LL = 1
2030 CALL ACRNR
      TEMP1 = XS(NSR)
      TEMP2 = ZS(NSR)
      CALL GTTN
2170 RETURN
      END
```

```
$IBFTC GTTN5 LIST,REF
      SUBROUTINE GTTN

C 2-D LAGRANGIAN -- COMPUTES TANGENT OF MESH BOUNDARIES AT INTERFACE
C
C EQUIVALENCE          (UXM,SPACE(5)), (UZM,SPACE(6))
C
      COSAB = SQRT(1.0-SNAB(NS)**2)
      AIN = AX*COSAB + AZ*SNAB(NS)
      TEMP8 = UXM + 0.5*AIN*DSUM*COSAB
      TEMP9 = UZM + 0.5*AIN*DSUM*SNAB(NS)
      IF (ABS(TEMP8) .LT. UEP) TEMP8 = 0.0
      IF (ABS(TEMP9) .LT. UEP) TEMP9 = 0.0
      XT = XS(NS) + TEMP8 *DELTA
      ZT = ZS(NS) + TEMP9 *DELTA
      TN(NS) = ATAN2(ZT-TEMP2,XT-TEMP1)
220 IF ((I .GT. 1) .AND. (I .LE. IMX)) GO TO 400
      IF (I .GT. 1) GO TO 300
      TEMP9 = PHIL(K)
      IF (TN(NS).GT. TEMP9) GO TO 320
      GO TO 400
300 TEMP9 = PHIR(K)
      IF (TN(NS).GT. TEMP9) GO TO 400
320 TN(NS) = TEMP9
400 RETURN
      END
```

\$IBFTC FDLPT0 LIST,REF  
SUBROUTINE FDLPTS

LPT 100

```
C
C 2-D LAGRANGIAN -- FIND POINTS ON LOWER BOUNDARY OF INTERFACE
C
C      EQUIVALENCE          (ISPACE(4),NSR)
C
      NONE = 1
      L = K-1
      IMM = IM(L)
      IF (IM(L) .GT. 0) GO TO 120
      JMM = JM(L)
      IMM = IM1(JMM)
120  TEMP10 = XH(NL,L)-XRT
      TEMP9 = ZH(NL,L)-ZRT
      IF (TEMP10 .NE. 0.0) GO TO 130
      TNR = SIGN(1.0E+8, TEMP9)
      GO TO 140
130  TNR = TEMP9/TEMP10
140  IF (ABS(TNRT) .GE. 1.0E+4) GO TO 190
      IF (ABS(TNR) .GE. 1.0E+4) GO TO 170
      TEMP8 = TNRT-TNR
      TNR = TEMP8 / (1.0 + TNRT*TNR)
      IF (ABS(TNR) .LT. 1.0E-4) GO TO 220
      GO TO 230
170  TNR = -1.0/TNRT
      GO TO 230
190  IF (ABS(TNR) .GE. 1.0E+4) GO TO 220
      TNR = +1.0/TNR
      GO TO 230
220  TNR = 0.0
      XSD = XH(NL,L)
      ZSD = ZH(NL,L)
      GO TO 350
230  IF ((TNR*TNP) .LE. 0.0) GO TO 320
270  TNP = TNR
      NL = NL + 1
      IF (NL .LE. IMM) GO TO 120
      NL = IMM
      NONE = 2
      GO TO 360
320  IF (NL .LE. 1) GO TO 350
      TEMP1 = (ZH(NL-1,L)-ZH(NL,L))/(XH(NL-1,L)-XH(NL,L))
      IF ((ABS(TEMP1).GT.1.0E-4) .OR. (ABS(TNRT).LT.1.0E+4)) GO TO 330
      XSD = XRT
      ZSD = ZH(NL,L)
      GO TO 350
330  TEMP2 = TNRT - TEMP1
      XSD = (ZH(NL-1,L)-ZRT+TNRT*XRT-TEMP1*XH(NL-1,L))/TEMP2
      ZSD = (TNRT*(ZH(NL-1,L)-TEMP1*(XH(NL-1,L)-XRT))-TEMP1*ZRT)/TEMP2
350  IENT = 1
      TNP = +1.0
360  RETURN
      END
```

5IBFTC CMPI20 LIST,REF  
SUBROUTINE CMPTIF

C  
C 2-D LAGRANGIAN -- COMPUTES AREA AND DENSITY OF ZONES ALONG LOWER  
C EDGE OF INTERFACE.

C EQUIVALENCE (XW,TEMP4), (ZW,TEMP5), (UXRT,SPACE(1)),  
1 (UZRT,SPACE(2))  
C EQUIVALENCE (ISPACE(4),NSR)

C IF (IENT .NE. 0) GO TO 1170

C THIS IS THE FIRST ZONE ON THE INTERFACE. INITIALIZE FLAGS.

C  
C LK = 1  
C IJK = 1  
C NL = 1  
C DO 990 L=1,K  
C IF (IM(L) .GT. 0) GO TO 980  
C NW = JM(L)  
C DO 970 LL=1,NW  
970 NL = NL+IM1(LL)  
C GO TO 990  
980 NL = NL + IM(L)\*JM(L)  
990 CONTINUE  
C XRT = X1(NL)  
C ZRT = Z1(NL)  
C UXRT = UX1(NL)  
C UZRT = UZ1(NL)  
C NFN = NL-1  
C IENT = 1

```

1100 TNNS = TN(NS)
C
C SEARCH J=1 MESH OF PLATE K+1 TO FIND POINTS ON EACH SIDE OF N.
C
XW = X1(N)
ZW = Z1(N)
CALL FDUPTS
IF (NONE .EQ. 1) GO TO 1160
C
C NO PAIR OF POINTS WERE FOUND.
C
IFNT = 0
NOIFCE = 0
GO TO 1600
C
C A PAIR OF POINTS WERE FOUND. IF THIS WAS THE FIRST PAIR, SET FLAG.
C
1160 IF (NFRST(K) .GT. 0) GO TO 1170
NFRST(K) = NL-1
IF (NL-1 .LT. NFN+1) NFRST(K) = NFN+1
1170 X12 = XS1(NS)
Z12 = ZS1(NS)
X12M = XS(NS)
Z12M = ZS(NS)
UX12 = UXS1(NS)
UZ12 = UZS1(NS)
N1 = NL
NS = NS + 1
C
C CONTINUE SEARCH FOR POINTS ON EACH SIDE OF N+1. IF NO POINTS FOUND,
C SET FLAG.
C
TNNS = TN(NS)
XW = X1(N+1)
ZW = Z1(N+1)
IF (N+1 .LE. NFN) GO TO 1220
XW = XS1(NSR)
ZW = ZS1(NSR)
1220 CALL FDUPTS
IF (NONE .EQ. 1) GO TO 1250
NSLST(K) = NS
WRITE (6,1240) K,I,J,N,NS,NL,LK,IJK,NFN,TN(NS),TNR,XRT,ZRT,XW,ZW
1240 FORMAT (1H0,45HCAUTION - INTERFACE POINT OUTSIDE UPPER PLATE /1H0,
1 9I6 / 1H0,6E16.8 )
1250 X13 = XS1(NS)
Z13 = ZS1(NS)
X13M = XS(NS)
Z13M = ZS(NS)
UX13 = UXS1(NS)
UZ13 = UZS1(NS)
N5 = NL
NS = NS-1
IF (I .LT. IMX) GO TO 1280

```

```

C
C WE ARE AT THE LAST ZONE IN THE PLATE. RE-DEFINE LOWER-RIGHT CORNER
C AND SET A FLAG
C
NLST(K) = N5
NW = NSR
X11 = XS1(NW)
Z11 = ZS1(NW)
X11M = XS(NW)
Z11M = ZS(NW)
UX11 = UXS1(NW)
UZ11 = UZS1(NW)
NS = NS+1
C
C COMPUTE AREA AND DENSITY FOR ZONE N.
C
1280 A11 = 0.5*((X11-X10)*(Z12-Z10) - (Z11-Z10)*(X12-X10))
IF (N1 .LE. NFN+IM(K+1)) GO TO 1290
XINT = (X10+X12+X13)/3.0
A1(N) = 0.5*((X13-X10)*(Z12-Z10) - (Z13-Z10)*(X12-X10))
RDEN = A1(N)*XINT**((IALF-1)
GO TO 1470
1290 XINT = (X10+X12+X1(N1))/3.0
AIN = 0.5*((X1(N1)-X10)*(Z12-Z10) - (Z1(N1)-Z10)*(X12-X10))
IF (XINT .NE. 0.0) GO TO 1310
RDEN = 0.0
GO TO 1320
1310 RDEN = AIN *XINT**((IALF-1)
1320 A1(N)= AIN
N3 = N5-1
IF (N3-N1) 1470,1430,1350
1350 N3 = N3-1
DO 1420 N4=N1,N3
XINT = (X10+X1(N4)+X1(N4+1))/3.0
AIN = 0.5*((X1(N4+1)-X10)*(Z1(N4)-Z10) - (Z1(N4+1)-Z10)*(X1(N4)-
1X10))
RDEN = RDEN + AIN *XINT**((IALF-1)
A1(N) = A1(N) + AIN
1420 CONTINUE
1430 XINT = (X10+X1(N5-1)+X13)/3.0
AIN = 0.5*((X13-X10)*(Z1(N5-1)-Z10) - (Z13-Z10)*(X1(N5-1)-X10))
RDEN = RDEN + AIN *XINT**((IALF-1)
A1(N) = A1(N) + AIN
1470 AIN1 = A1(N)
XINT = (X11+X10+X13)/3.0
AIN = 0.5*((X11-X10)*(Z13-Z10) - (Z11-Z10)*(X13-X10))
RDEN = RDEN + AIN *XINT**((IALF-1)
A1(N) = A1(N) + AIN
RHO1(N) = ZM(N)/RDEN
C
NH(I,K) = N
XH(I,K) = 0.25*(X10+X11+X12+X13)
ZH(I,K) = 0.25*(Z10+Z11+Z12+Z13)
1600 RETURN
END

```

TR66-85  
Appendix C

FDUPTS  
Page 1 of 2

\$IBFTC FDUP80 LIST,REF  
SUBROUTINE FDUPTS

UPT 100

C  
C 2-D LAGRANGIAN -- FIND POINTS ON UPPER BOUNDARY OF INTERFACE  
C THIS VERSION IS FOR THE COUPLED CAMEO-2-D SERIES.

C  
C EQUIVALENCE (XW,TEMP4), (ZW,TEMP5), (UXRT,SPACE(1)),  
1 (UZRT,SPACE(2))  
C EQUIVALENCE (ISPACE(4),NSR)

C  
NONE = 1  
120 TEMP10 = XRT-XW  
TEMP9 = ZRT-ZW  
TNR = ATAN2(TEMP9,TEMP10)  
IF (ABS(TNR-TNNS) .LT. 1.0E-4) GO TO 220  
TEMP7 = SIN(TNNS)  
TEMP8 = COS(TNNS)  
TNR = TEMP10\*TEMP7 - TEMP9\*TEMP8  
GO TO 230  
220 TNR = 0.0  
XS1(NS) = XRT  
ZS1(NS) = ZRT  
UXS1(NS) = UXRT  
UZS1(NS) = UZRT  
IF (LK .GT. 1) GO TO 400  
LK = 2  
TNP = -1.0  
XLFT = XRT  
ZLFT = ZRT  
NL = NL+1  
XRT = X1(NL)  
ZRT = Z1(NL)

```
GO TO 400
230 IF (LK .NE. 1) GO TO 260
    LK = 2
    TNP = -1.0
    GO TO 270
260 IF ((TNR*TNP) .LE. 0.0) GO TO 320
    TNP = TNR
270 IF (IJK .GF. 2) GO TO 310
    XLFT = XRT
    ZLFT = ZRT
    NL = NL + 1
    XRT = X1(NL)
    ZRT = Z1(NL)
    UXRT = UX1(NL)
    UZRT = UZ1(NL)
    IF (NL .LE. NFN+IM(K+1)) GO TO 120
    NW = 1
    DO 300 L=1,K
    IF (IM(L) .GT. 0) GO TO 290
    N3 = JM(L)
    DO 280 LL = 1,N3
280 NW = NW+IM1(LL)
    NW = NW+1
    GO TO 300
290 NW = NW+JM(L)+IM(L)+1
300 CONTINUE
    XRT = XS1(NW)
    ZRT = ZS1(NW)
    UXRT = UXS1(NW)
    UZRT = UZS1(NW)
    IJK = IJK+1
    GO TO 120
310 NONF = 2
320 TEMP1 = (ZLFT-ZRT)/(XLFT-XRT)
    IF (ABS(TNNS-1.5707963) .LE. 0.3E-7) GO TO 340
    TNNS = TAN(TNNS)
    IF (ABS(TNNS) .LE. 1.0E+9) GO TO 350
340 TNNS = SIGN(1.0E+9,TNNS)
350 TEMP2 = TNNS - TEMP1
    XS1(NS) = (ZLFT-ZW +TNNS*XW-TEMP1*XLFT)/TEMP2
    ZS1(NS) = (TNNS*(ZLFT-TEMP1*(XLFT-XW))-TEMP1*ZW)/TEMP2
    TEMP1 = XS1(NS)-XS(NS)
    TEMP2 = ZS1(NS)-ZS(NS)
    UXS1(NS) = TEMP1/DELTA
    UZS1(NS) = TEMP2/DELTA
    IF (ABS(UXS1(NS)) .LT. UEP) UXS1(NS)=0.0
    IF (ABS(UZS1(NS)) .LT. UEP) UZS1(NS)=0.0
400 TEMP1 = XRT-XLFT
    TEMP2 = ZRT-ZLFT
420 IF (ABS(TEMP2/TEMP1) .GE. 1.0E-4) GO TO 490
430 SNAB(NS) = 0.0
    GO TO 500
490 TEMP1 = SQRT(TEMP1**2 + TEMP2**2)
    SNAB(NS) = TEMP2/TEMP1
500 RETURN
    FND
```



```

$IBFTC SDST LIST,REF
SUBROUTINE SDST
C
C 2-D LAGRANGIAN -- DETERMINES DEVIATORIC STRESSES
C
C DDD IS DEVIATORIC STRESS FUNCTION
C
C DDD(TEMP1,TEMP2,TEMP3) = (DELTA/ TEMP4 )*(TEMP1*(2.0*DXX+DZ
1Z) + TEMP2*(2.0*DZZ+DXX) + TEMP3*(2.0*DXZ))
C
TEMP1 = RDOTR/3.0
TEMP2 = 2.0*(A1(N)+AM)
TEMP3 = UX10-UX13
TEMP4 = UX12-UX11
TEMP5 = UZ10-UZ13
TEMP6 = UZ12-UZ11
TEMP7 = Z12+Z12M-Z11-Z11M
TEMP8 = Z10+Z10M-Z13-Z13M
TEMP9 = X12+X12M-X11-X11M
TEMP10= X10+X10M-X13-X13M
DXX = -(TEMP3*TEMP7 - TEMP4*TEMP8)/TEMP2 + TEMP1
DZZ = (TEMP5*TEMP9 - TEMP6*TEMP10)/TEMP2 + TEMP1
DYY = -(DXX+DZZ)
DXZ = (TEMP3*TEMP9 - TEMP4*TEMP10 - TEMP5*TEMP7 + TEMP6*TEMP8)/
1(2.0*TEMP2)
WXZ = (TEMP3*TEMP9 - TEMP4*TEMP10 + TEMP5*TEMP7 - TEMP6*TEMP8)/
1(2.0*TEMP2)
TEMP1 = 0.5*(RH01(N)+RH0M)/RH0Z(K)-1.0
G = G0(K)*(1.0+TEMP1*(G1(K)+TEMP1*(G2(K)+TEMP1*(G3(K))))))
TEMP2 = DELTA*WXZ
TEMP3 = G*DELTA
TEMP4 = 2.0*TEMP2*(DTXZ(N)+TEMP3*DXZ)
DETXX = DTXX(N) + 2.0*TEMP3*DXX + TEMP4
DETYX = DTYX(N) + 2.0*TEMP3*DYY

```

```
DETZZ = DTZZ(N) + 2.0*TEMP3*DZZ - TEMP4
DETZX = DTZX(N) + 2.0*TEMP3*DXZ - TEMP2*(DTXX(N)+TEMP3*DXX-DTZZ(N)
1-TEMP3*DZZ)
BOX    = DETXX**2 +DETY**2 +DETZZ**2 +2.0*DETZX**2
TEMP6  = 0.66666667*CAPY(K)**2
DTXXM  = DTXX(N)
DTYYM  = DTY(N)
DTZZM  = DTZZ(N)
DTXZM  = DTZX(N)
IF (BOX .GT. TEMP6) GO TO 500
DTXX(N) = DETXX
DTYY(N) = DETY
DTZZ(N) = DETZZ
DTXZ(N) = DETZX
GO TO 550
500 TEMP7 = SQRT(TEMP6/BOX )
DTXX(N) = DETXX*TEMP7
DTYY(N) = DETY*TEMP7
DTZZ(N) = DETZZ*TEMP7
DTXZ(N) = DETZX*TEMP7
C
C EVALUATE ARTIFICIAL VISCOSITY.
C
550 QXXM = QXX1(N)
QZZM = QZZ1(N)
QXZM = QXZ1(N)
CALL EVALQ
C
TEMP4 = RHO1(N)+RHOM
TEMP1 = DTXX(N)+DTXXM
TEMP2 = DTZZ(N)+DTZZM
TEMP3 = DTXZ(N)+DTXZM
DEED = DDD(TEMP1,TEMP2,TEMP3)
TEMP1 = QXX1(N)+QXXM
TEMP2 = QZZ1(N)+QZZM
TEMP3 = QXZ1(N)+QXZM
DEQD = DDD(TEMP1,TEMP2,TEMP3)
DED = DEED + DEQD
RETURN
END
```

```
$IRFTC EVALQ2 LIST,REF
SUBROUTINE EVALQ
C
EVLQ 100
EQUIVALENC (SPACE(7),QM)
C
PM = -TXX(N)+DTXXM-Q(N)-QXX1(N)
EVLQ 110
QM = Q(N)
EVLQ 111
P1 = PM
EVLQ 120
CALL GETCN
EVLQ 121
IF (NBQ .GT. 0) GO TO 160
EVLQ 122
IF (RDOTR .GT. 0.0) GO TO 160
EVLQ 130
Q1(N) = 0.0
EVLQ 140
GO TO 200
EVLQ 150
160 TEMP1 = 0.5*(RH01(N)+RHOM)
EVLQ 170
TEMP2 = 0.5*(A1(N)+AM)
EVLQ 180
Q1(N) = CZ**2*TEMP1*TEMP2*RDOTR**2 + CL*TEMP1*CN*RDOTR*SQRT(TEMP2)
EVLQ 190
200 TEMP3 = C3*TEMP1*TEMP2*CN
EVLQ 200
QXX1(N) = TEMP3*DXX
EVLQ 210
QZZ1(N) = TEMP3*DZZ
EVLQ 220
QXZ1(N) = TEMP3*DXZ
EVLQ 230
RETURN
EVLQ 240
END
EVLQ 250
```

\$IBFTC FDP53 LIST,REF  
SUBROUTINE FINDP

C  
C 2-D LAGRANGIAN -- DETERMINES E AND P, ACCORDING TO E.O.S. SELECTED.  
C IF MEQ(K) = 1 - TILLOTSON METALLIC E.O.S.  
C = 2 - POLYNOMIAL E.O.S.  
C = 3 - OTHER

EQUIVALENCE (SPACE(7),QM)

C  
C IF (MEQ(K) .LE. 0) MEQ(K)=1  
C TEMP1 = ETA1-1.0  
C IF (MEQ(K) .GT. 1) GO TO 500

C  
C TILLOTSON METALLIC EQUATION OF STATE  
C 2-D LAGRANGIAN -- DETERMINES E AND P BY SOLVING A QUADRATIC. USES  
C MOST POSITIVE VALUF.  
C

IND = 0  
SA1 = 0.0  
IF (DRR2 .EQ. 0.0) GO TO 120  
SA1 = 2.0/DRR2  
120 SA2 = -(SA1\*E(N)+PM+ QM+Q1(N)+ SA1\*DED)  
SB1 = SA(K)\*RHO1(N)  
SB2 = SB(K)\*RHO1(N)  
SB3 = 1.0/(FZERO(K)\*ETA1\*\*2)  
SB4 = TEMP1\*(CAPA(K) + TEMP1\*CAPB(K))  
TEMP2 = 1.0/ETA1 - 1.0  
TEMP3 = EXP(-ALPHA(K)\*TEMP2\*\*2)  
SC2 = SB(K)\*RHO1(N)\*TEMP3  
SC3 = CAPA(K)\*TEMP1\*TEMP3\*EXP(-BETA(K)\*TEMP2)  
IF (DRR2 .NE. 0.0) GO TO 230  
F1(N) = E(N)  
IF (TEMP1 .GE. 0.0) GO TO 210  
IF ((ETA1 .GE. RHOS(K)) .AND. (E1(N) .LE. EPRS(K))) GO TO 210  
SB2 = SC2  
SB4 = SC3  
210 P1 = F1(N)\*(SB1+SB2/(SB3\*F1(N) +1.0)) + SB4  
GO TO 460  
230 IF (SC2 .NE. 0.0) GO TO 270  
IF (SC3 .NE. 0.0) GO TO 270  
F1(N) = SA2/(SB1-SA1)  
GO TO 450  
270 IF (TEMP1 .LT. 0.0) GO TO 360  
320 BA = SB3\*(SA1-SB1)  
BB = SA1-SB1+SB3\*(SA2-SB4)-SB2  
BC = SA2-SB4  
GO TO 400  
360 IF (ETA1 .GE. RHOS(K)) IND=1  
BA = SB3\*(SA1-SB1)  
BB = SA1-SB1+SB3\*(SA2-SC3)-SC2  
BC = SA2-SC3

```

400 CALL QUAD(RA,RR,RC)
      IF (RC .NE. (-999.)) GO TO 410
      WRITE (6,403) N
403  FORMAT (1H0,32HERROR - ROOTS IMAGINARY. ZONE = ,I4)
      IFND = 5
      RETURN
410  E1(N) = BA
      IF (IND .NE. 1) GO TO 450
      IND = 0
      IF (E1(N) .LE. EPRS(K)) GO TO 320
450  P1 = SA1*E1(N)+SA2
      IF (E1(N) .LT. EEP ) E1(N)=0.0
460  IF (ABS(P1) .LT. PEP ) P1=0.0
      GO TO 800
C
C   POLYNOMIAL EQUATION OF STATE
C
500  CONTINUE
      CAPF = 1.0
      TEMP2 = FK0(K)*TEMP1*(1.0+TEMP1*(FK1(K)+TEMP1*(FK2(K)+TEMP1*(FK3(K)
1))))))
      TEMP3 = H0(K)*(1.0+TEMP1*(H1(K)+TEMP1*(H2(K)+TEMP1*(H3(K))))))
700  E1(N) = (EM + (0.5*(TEMP2*CAPF+PM)+Q1(N))*DRR2 + DED)/(1.0-0.5*TEM
1P3*CAPF*DRR2)
      P1 = (TEMP2 + TEMP3*E1(N))*CAPF
      IF (E1(N) .LT. EEP) E1(N)=0.0
      IF (ABS(P1) .LT. PEP) P1=0.0
C
C   800  TEMP1 = -(P1+Q1(N))
      TXX1(N) = TEMP1+DTXX(N)+QXX1(N)
      TYY1(N) = TEMP1+DTYY(N)-(QXX1(N)+QZZ1(N))
      TZZ1(N) = TEMP1+DTZZ(N)+QZZ1(N)
      TXZ1(N) =          DTXZ(N)+QXZ1(N)
C
C   COMPUTE PRINCIPAL STRESSES, AND DIRECTION OF STRESSES.
C   LARGEST POSITIVE VALUE IS COMPARED WITH ULTIMATE TENSILE STRENGTH
C   (UTS) OF MATERIAL. IF THE STRESS EXCEEDS THE UTS, A FRACTURE WILL
C   OCCUR.
C
      TEMP1 = 0.5*(TXX1(N)+TZZ1(N)+SQRT((TXX1(N)-TZZ1(N))**2+(2.0*TXZ1(N)
1)**2))
      TM = AMAX1(TEMP1,TYY1(N))
      IF (TM .LT. UTS(K)) GO TO 950
      PHITM = 0.5*ATAN2(2.0*TXZ1(N),TXX1(N)-TZZ1(N))*57.2957795
      IFND = 6
      WRITE (6,920) N,K,TM ,PHITM
920  FORMAT (1H0,30HFRACTURE HAS OCCURRED AT MESH ,I4, 8H, PLATE ,I2, 1
1H, / 11X,18HVALUE OF STRESS = ,E25.8, 8H D/SQCM. / 11X,21HDIRECTIO
2N OF CRACK = ,F6.1, 5H DEG. )
C
950  RETURN
      FND

```

```
$IBFTC QUAD2D LIST
SUBROUTINE QUAD (SA,SR,SC)
C
C DETERMINES THE ROOTS OF A QUADRATIC EQUATION -  $A*X**2 + B*X + C = 0$ 
C UPON RETURNING, A = THE MOST POSITIVE ROOT
C B = THE REMAINING ROOT
C
DOUBLE PRECISION A,B,C,DISC,ALPHA,BETA
A = SA
B = SR
C = SC
DISC = B**2 - 4.0*A*C
IF (DISC) 50, 70,100
50 SC = -999.0
RETURN
70 SA = -B/(2.0*A)
SR = SA
RETURN
100 IF (C) 103,101,103
101 DISC = DABS(B)
GO TO 104
103 DISC = DSQRT(DISC)
104 IF (B) 110,130,150
110 ALPHA = (-B + DISC)/(2.0*A)
GO TO 160
130 ALPHA = DSQRT(-C/A)
GO TO 160
150 ALPHA = (-B - DISC)/(2.0*A)
160 BETA = C/(A*ALPHA)
IF (ALPHA-BETA) 180,210,240
180 SA = BETA
SR = ALPHA
RETURN
210 SA = ALPHA
SR = -SA
RETURN
240 SA = ALPHA
SR = BETA
RETURN
END
```

QUAD 010  
QUAD 011  
QUAD 012  
QUAD 013  
QUAD 014  
QUAD 020  
QUAD 030  
QUAD 050  
QUAD 060  
QUAD 070  
QUAD 080  
QUAD 090  
QUAD 100  
QUAD 101  
QUAD 102  
QUAD 103  
QUAD 104  
QUAD 110  
QUAD 120  
QUAD 130  
QUAD 140  
QUAD 150  
QUAD 160  
QUAD 170  
QUAD 180  
QUAD 190  
QUAD 200  
QUAD 210  
QUAD 220  
QUAD 230  
QUAD 240  
QUAD 250  
QUAD 260  
QUAD 270

TR66-85  
Appendix C

COMPDT  
Page 1 of 1

```

$IRFTC COMPDT LIST,REF
      SUBROUTINE COMPDT
                                                    CDT 100
C
C 2-D LAGRANGIAN -- DETERMINES MINIMUM DT FROM DT OF EACH MESH
C
      CALL GETCN
                                                    CDT 110
C
      TEMP1 = SQRT((Z10-Z13)**2 + (X10-X13)**2)
                                                    CDT 120
      TEMP2 = SQRT((Z11-Z12)**2 + (X11-X12)**2)
                                                    CDT 130
      TEMP3 = AMAX1(TEMP1,TEMP2)
                                                    CDT 140
      DL = A1(N)/TEMP3
                                                    CDT 150
      IF (DL .LE. 0.0) WRITE (6,300) N,DL,A1(N)
300  FORMAT (1H0,I4,2E16.8)
      IF (RDOTR .LE. 0.0) GO TO 190
      TEMP4 = 2.0*CZ*DL*RDOTR
                                                    CDT 160
      GO TO 200
                                                    CDT 170
190  TEMP4 = 0.0
                                                    CDT 180
200  TEMP5 = DL/SQRT(CN**2 + TEMP4**2)
                                                    CDT 190
      DMIN = AMIN1(DMIN,TEMP5)
                                                    CDT 200
      RETURN
                                                    CDT 210
      END
                                                    CDT 220
                                                    CDT 230

```

\$IRFIC GETCN1 LIST,REF  
SUBROUTINE GETCN

C  
C 2-D LAGRANGIAN -- DETERMINES SPEED OF SOUND IN MESH

C  
C

|  |          |
|--|----------|
| TEMP9 = 0.0  | GTCN 100 |
| IF (G0(K) .EQ. 0.0) GO TO 140                        | GTCN 110 |
| TEMP10 = 1.226                                       | GTCN 120 |
| GO TO 150  | GTCN 130 |
| 140 TEMP10 = 1.0                                     | GTCN 140 |
| 150 CN = CNO(K)                                      | GTCN 150 |
| IF (P1 .LE. 0.0) GO TO 180                           | GTCN 160 |
| IF (MEQ(K) .GT. 1) GO TO 170                         | GTCN 161 |
| IF (ETA1 .GE. RHOS(K)) GO TO 170                     | GTCN 162 |
| TEMP9 = SQRT(SA(K)*E1(N)*(1.0+SA(K)))                | GTCN 163 |
| 170 CN = CN + (CNA(K)*(P1*1.0E-12)**CNEXP(K))*1.0E+5 | GTCN 170 |
| CN = AMAX1(CN,TEMP9)                                 | GTCN 171 |
| 180 CN = CN * TEMP10                                 | GTCN 180 |
| RETURN   | GTCN 190 |
| END  | GTCN 200 |

-



```

$IBFIC ENRGCK LIST,REF
SUBROUTINE ENRGCK
C
C 2-D LAGRANGIAN -- COMPUTES TOTAL ENERGY AND ENERGY DISTRIBUTION
C          XP      = TOTAL RADIAL MOMENTUM FOR EACH CYCLE
C          ZP      = TOTAL AXIAL MOMENTUM FOR EACH CYCLE
C          ZK      = TOTAL AXIAL KINETIC ENERGY
C          XK      = TOTAL RADIAL KINETIC ENERGY
C          TIE     = TOTAL INTERNAL ENERGY
C          TE      = TOTAL ENERGY FOR EACH CYCLE
C          TDEE    = TOTAL SPHERICAL STRESS WORK
C          TDEQ    = TOTAL SPHERICAL VISCOUS STRESS WORK
C          TDFED   = TOTAL DEVIATORIC STRESS WORK
C          TDEQD   = TOTAL DEVIATORIC VISCOUS STRESS WORK
C          TDFPD   = TOTAL PLASTIC WORK
C          TDFQPD  = TOTAL ENERGY
C
C          EQUIVALENCE      (SPACE(7),QM)
C
TEMP1 = (6.2831853)**(IALP-1)*ZM(N)
TEMP2 = 0.25*(UX10+UX11+UX12+UX13)
TEMP3 = 0.25*(UZ10+UZ11+UZ12+UZ13)
XP = XP + TEMP1*TEMP2
ZP = ZP + TEMP1*TEMP3
TEMP4 = 0.5*TEMP1*TEMP2**2
TEMP5 = 0.5*TEMP1*TEMP3**2
TEMP6 = 0.5*TEMP1*(E1(N)+EM)
TE = TE + TEMP4+TEMP5+TEMP6
XK = XK + TEMP4
ZK = ZK + TEMP5
TIE = TIE + TEMP6
TEMP2 = 0.5*TEMP1*(P1+PM)*DRR2
TEMP3 = TEMP1*DRR2*0.5*(Q1(N)+QM)
TEMP4 = TEMP1*DEFD
TEMP5 = TEMP1*DEQD
TEMP7 = 0.0
IF (G.FQ.0.0) GO TO 300
TEMP7 = -TEMP1*(CAPY(K)*(SQRT(0.66666667*BOX)-0.66666667*CAPY(K)))
1/(G*(RH01(N)+RHOM))
300 TDEQPD = TDFQPD + TEMP2+TEMP3+TEMP4+TEMP5+TEMP7
TDFE = TDFE + TEMP2
TDEQ = TDEQ + TEMP3
TDFED = TDFED + TEMP4
TDEQD = TDEQD + TEMP5
TDFPD = TDFPD + TEMP7
RETURN
END

```

```
$IRBTC LNK320 LIST,REF
C 2-D LAGRANGIAN -- LINK3 FOR IBM 7040 CHAIN
C
C DIMENSION ZZ(15778)
C
C EQUIVALENCE (ZZ(1),NCYCLE)
C
C REWIND CHAIN TAPE (TAPE 0)
C
C CALL RWIND0
C
C CHECK ERROR INDICATOR, IEND, AND WRITE MESSAGE -
C =1 - CONTINUE COMPUTING AFTER OUTPUT
C =2 - PLOT ONLY AND CONTINUE COMPUTING
C =3 - UPDATE RESTART TAPE AND OUTPUT
C =4 - ERROR STOP - STRESS VALUES EXCEED LIMIT
C =5 - NORMAL STOP AFTER OUTPUT
C =6 - ERROR STOP, PRINT ONLY.
C
C GO TO (200,210,130,110,130,200),IEND
110 WRITE (6,120) N
120 FORMAT (1H0,38HERROR - STRESS LIMIT EXCEEDED AT ZONE ,I4 )
GO TO 200
C
C UPDATE RESTART TAPE
C
C 130 WRITE (6,140) NCYCLE
140 FORMAT (1H0,30HRESTART TAPE UPDATED AT CYCLE ,I6)
REWIND 4
WRITE (4) 77
C
C PRINT OUTPUT, PLOT OUTPUT
C
C 200 CALL PRNT
210 CALL GRAPH
C
C CALL CHNXIT
END
```

```

$IBFTC PRNT10 LIST,REF
      SUBROUTINE PRNT
C
C 2-D LAGRANGIAN --
C
      EQUIVALENCE      ( ISPACE(2),LSTN)
      EQUIVALENCE      ( ISPACE(4),NSR)
C
      WRITE (6,120) (TITLE(L),L=1,12)
120  FORMAT (1H1,29X,12A6/ )
      WRITE (6,140) NCYCLE,TIME,DELTA
140  FORMAT (1H0,6H CYCLE, 5X, 8HTIME-SEC, 6X,11HDELTA T-SEC /1H ,16,
12E16.8/ )
      TEMP1 = TDFQ + TDEQD
      TEMP2 = XK + ZK
      WRITE (6,153) TIE,TEMP2,TE,XP,ZP,TEMP1,TDEPD
153  FORMAT (1H0,35H PER CYCLE ENERGIES - INTERNAL-ERG, 4X,11HKINETIC-
1ERG, 6X, 9HTOTAL-ERG, 9X,20HRADIAL MOM.-G CM/SEC, 1X,19HAXIAL MOM.
2-G CM/SEC /1H ,20X,3E16.8, 4X,2E20.8 /
*          1H0,35HACCUMULATED ENERGIES - VISC.ST.-ERG, 4X,
311HPLASTIC-ERG /1H ,20X,2E16.8 / )
      WRITE (6,170)
170  FORMAT (1H , 5HPLATE, 1X, 1HZ, 4X, 1HX, 6X,13HSIGMA X-D/CSQ, 3X,13
1HSIGMA Y-D/CSQ, 3X,13HSIGMA Z-D/CSQ, 3X,14HSIGMA XZ-D/CSQ, 5X, 7HQ
2-D/CSQ, 9X, 8HRHO-G/CC, 5X,13HENERGY-ERG/GM / 6X,3HST., 2X,3HS
3T., 5X,13HX VEL.-CM/SEC, 3X,13HZ VEL.-CM/SEC, 8X, 4HX-CM,12X, 4HZ-
4CM, 9X, 9HQ X-D/CSQ, 8X, 9HQ Z-D/CSQ, 5X,10HQ XZ-D/CSQ /)
      LCM = 53
      LC = 14
      NS = 0
      N = 0
      DO 780 K=1,M
      KJ = 1
      JMX = JM(K)
      IMX = 0
      IDM = 0
      DO 771 J=1,JMX
      LL = 0
      IMM = IMX
      IMX = IM(K)
      IF (IMX .GT. 0) GO TO 300
      IMX = IM1(J)
      IDM = IMM - IMX
300  NS = NS+1
      DO 470 I=1,IMX
      N = N + 1
      IF (LC .LE. LCM) GO TO 380
      WRITE (6,340)
340  FORMAT (1H1)
      WRITE (6,170)
      LC = 1
      KJ = 1
380  GO TO (390,430), KJ

```

```
390 WRITE (6,410) K,J,I,TXX1(N),TYY1(N),TZZ1(N),TXZ1(N),Q1(N),RHO1(N),
    1E1(N),UX1(N),UZ1(N),X1(N),Z1(N),QXX1(N),QZZ1(N),QXZ1(N)
410 FORMAT (1H0, 1X,I2,I5,I5, 3X, 7E16.8/17X,7E16.8)
    KJ = 2
    GO TO 460
430 WRITE (6,450) J, I,TXX1(N),TYY1(N),TZZ1(N),TXZ1(N),Q1(N),RHO1(N),
    1F1(N),UX1(N),UZ1(N),X1(N),Z1(N),QXX1(N),QZZ1(N),QXZ1(N)
450 FORMAT (1H0, 3X,I5,I5, 3X, 7E16.8/17X, 7E16.8)
460 LC = LC + 3
470 CONTINUE
480 IF (LC .LE. LCM) GO TO 550
    WRITE (6,340)
    WRITE (6,170)
    LC = 1
    KJ = 1
550 GO TO (560,590), KJ
560 WRITE (6,570) K, J, I,UXS1(NS),UZS1(NS),XS1(NS),ZS1(NS)
570 FORMAT ( 1H0, 1X,I2,I5,I5,3X, 7E16.8 )
    KJ = 2
    GO TO 610
590 WRITE (6,600) J, I,UXS1(NS),UZS1(NS),XS1(NS),ZS1(NS)
600 FORMAT (1H0, 3X,I5,I5, 3X, 7E16.8)
610 LC = LC + 2
    IF (IDM .LE. 0) GO TO 620
    IF (LL .GE. IDM) GO TO 620
    LL = LL+1
    NS = NS+1
    GO TO 480
620 IF (J .NF. JMX) GO TO 770
    J = J + 1
    DO 750 I=1,IMX
    IF (LC .LE. LCM) GO TO 700
    WRITE (6,340)
    WRITE (6,170)
    LC = 1
    KJ = 1
700 GO TO (710,730), KJ
710 WRITE (6,570) K,J,I,UXS1(NS),UZS1(NS),XS1(NS),ZS1(NS)
    KJ = 2
    GO TO 740
730 WRITE (6,600) J, I,UXS1(NS),UZS1(NS),XS1(NS),ZS1(NS)
740 LC = LC + 2
750 CONTINUE
    NS = NS + 1
    GO TO 480
770 IF (N .GE. LSTN) GO TO 790
771 CONTINUE
780 CONTINUE
790 RETURN
    END
```

```
$IBFTC GRAPH1 LIST,REF
      SUBROUTINE GRAPH
C
C 2-D LAGRANGIAN --
C
      DIMENSION DATA(100)
C
      CALL PLOTS (DATA,100)
C
      IF (NPLOT .GT. 5) GO TO 300
      IF (NPLOT .GT. 0) GO TO 150
      CALL GRP1
      GO TO 300
150  IPLIT = NPLOT
      IF (NPLOT .LT. 5) GO TO 180
      IPLIT = 1
180  GO TO (190,210,230,250),IPLIT
190  CALL GRP2 (TXX1, 8HSIGMA X )
      GO TO 260
210  CALL GRP2 (TYY1, 8HSIGMA Y )
      GO TO 260
230  CALL GRP2 (TZZ1, 8HSIGMA Z )
      GO TO 260
250  CALL GRP2 (TXZ1, 8HSIGMA XZ)
260  IF (NPLOT .LT. 5) GO TO 300
      IF (IPLIT .GE. 4) GO TO 300
      IPLIT = IPLIT + 1
      GO TO 180
300  RETURN
      END
```

GRAF 100  
GRAF 120  
GRAF 130  
GRAF 140  
GRAF 141  
GRAF 150  
GRAF 160  
GRAF 170  
GRAF 180  
GRAF 190  
GRAF 200  
GRAF 210  
GRAF 220  
GRAF 230  
GRAF 240  
GRAF 250  
GRAF 260  
GRAF 270  
GRAF 280  
GRAF 290  
GRAF 300  
GRAF 310  
-

```
$IBFTC GRP101 LIST,REF
SUBROUTINE GRP1
C
C 2-D LAGRANGIAN -- DRAWS PICTURE OF MESHES - X VS. Z
C
EQUIVALENCF      (ISPACE(4),NSR)
EQUIVALENC      (ISPACE(5),IMXM),(ISPACE(6),NMAX1)
EQUIVALENC      (TEMP1,XMN),      (TEMP2,ZMN)
C
DIMENSION XT(2),ZT(2)
C
C DETERMINE RANGE OF VALUES OF X AND Z
C
XMIN = X1(1)
XMAX = XMIN
ZMIN = Z1(1)
ZMAX = ZMIN
DO 200 N=2,NMAX
IF (X1(N) .LT. XMIN) XMIN = X1(N)
IF (X1(N) .GT. XMAX) XMAX = X1(N)
IF (Z1(N) .LT. ZMIN) ZMIN = Z1(N)
IF (Z1(N) .GT. ZMAX) ZMAX = Z1(N)
200 CONTINUE
DO 260 NS=1,NSMAX
IF (XS1(NS) .LT. XMIN) XMIN = XS1(NS)
IF (XS1(NS) .GT. XMAX) XMAX = XS1(NS)
IF (ZS1(NS) .LT. ZMIN) ZMIN = ZS1(NS)
IF (ZS1(NS) .GT. ZMAX) ZMAX = ZS1(NS)
260 CONTINUE
C
C SCALE SO THAT Z DOES NOT EXCEED 10 IN.,DRAW AXIS, LABEL
C
XT(1) = XMIN
```

```
      XT(2) = XMAX
      ZT(1) = ZMIN
      ZT(2) = ZMAX
      CALL SCALE (ZT,2,10.0,ZMN,DZZ, 1)
      ZLENGTH = AINT(ZT(2)+0.99)
      XMN = AINT(XT(1)/DZZ)
      XLNGTH= AINT((XT(2)-XMN )/DZZ+ 0.99)
      IF (XLNGTH .LE. 10.0) GO TO 340
      ZMN = ZMN*DZZ
      CALL SCALE (XT,2,10.0,XMN,DZZ,1)
      XLNGTH = AINT(XT(2)+0.99)
      ZLENGTH = AINT((ZMAX-ZMN)/DZZ+0.99)
      ZMN = ZMN/DZZ
340  PLN = AMINI(XLNGTH,2.0)
      CALL AX2 (0.0,0.0,13HRADIAL (X)-CM,-13,XLNGTH, 0.0,XMN,DZZ, PLN)
      PLN = AMINI(ZLENGTH,2.0)
      CALL AX2 (0.0,0.0, 12HAXIAL (Z)-CM, 12,ZLENGTH, 90.0,ZMN,DZZ, PLN)
      CALL SYMBOL (0.5,10.0,0.14,TITLE,0.0, 72)
      CALL SYMBOL (0.5, 9.75,0.14,5HTIME=,0.0,5)
      T = TIME*1.0E+6
      CALL NUMBER (1.10,9.75,0.14,T ,0.0, 6)
      CALL PMU (2.18, 9.75,0.14)
      CALL SYMBOL (2.30, 9.75, 0.14,4HSEC.,0.0, 4)
C
C  START PLOT OF X US. Z
C
      NS = 0
      N = 0
      N1 = 0
      DO 900 K=1,M
      IF (IM(K) .GT. 0) GO TO 440
C
C  THIS PLATE IS A DEBRIS CLOUD.
C
      CALL GRP4
      GO TO 900
C
C  FIRST DRAW LINES OF CONSTANT J
C
440  JMX = JM(K)
      DO 580 J=1,JMX
      NS = NS + 1
      IMX = IM(K)
      IC = 3
      DO 540 I=1,IMX
      N = N + 1
      XPLT = (X1(N)-XMN)/DZZ
      YPLT = (Z1(N)-ZMN)/DZZ
      CALL PLOT (XPLT,YPLT,IC)
      IC = 2
540  CONTINUE
      XPLT = (XS1(NS)-XMN)/DZZ
      YPLT = (ZS1(NS)-ZMN)/DZZ
```

```
      CALL PLOT (XPLT,YPLT,IC)
580 CONTINUE
C
C   NOW DRAW LINES OF CONSTANT I
C
      K1 = K-1
      IF (K1 .LE. 0) GO TO 630
      IF (IM(K1) .GT. 0) GO TO 620
      N1 = NMAX1+N1
      GO TO 630
620 N1 = IM(K1)*JM(K1)+N1
630 DO 760 I=1,IMX
      IC = 3
      NS = NS + 1
      DO 720 J=1,JMX
      N2 = N1 + (J-1)*IMX + I
      XPLT = (X1(N2)-XMN)/DZZ
      YPLT = (Z1(N2)-ZMN)/DZZ
      CALL PLOT (XPLT,YPLT,IC)
      IC = 2
720 CONTINUE
      XPLT = (XS1(NS)-XMN)/DZZ
      YPLT = (ZS1(NS)-ZMN)/DZZ
      CALL PLOT (XPLT,YPLT,IC)
760 CONTINUE
      IC = 3
      N2 = NS - IMX - JMX
      DO 850 J=1,JMX
      N2 = N2 + 1
      XPLT = (XS1(N2)-XMN)/DZZ
      YPLT = (ZS1(N2)-ZMN)/DZZ
      CALL PLOT (XPLT,YPLT,IC)
      IC = 2
850 CONTINUE
      NS = N2 + IMX + 1
      XPLT = (XS1(NS)-XMN)/DZZ
      YPLT = (ZS1(NS)-ZMN)/DZZ
      CALL PLOT (XPLT,YPLT,IC)
900 CONTINUE
      DO 1000 I=1,IMX
      NS = NS-1
      XPLT = (XS1(NS)-XMN)/DZZ
      YPLT = (ZS1(NS)-ZMN)/DZZ
      CALL PLOT (XPLT,YPLT,IC)
1000 CONTINUE
      CALL PLOT (XLNGTH+2.0, 0.0,-3)
      RETURN
      END
```



```
$IBFTC GRP2E2 LIST,REF
      SUBROUTINE GRP2 ( S1,LABL)
C
C 2-D LAGRANGIAN -- PLOTS SIGMA VS. X AND Z
C
      EQUIVALENCE      ( ISPACE(5),IMXM),( ISPACE(6),NMAX1)
C
      DIMENSION      XT(2),ZT(2),ST(2),S1( 600),LABL(2)
C
      THETA = -30.0
      TH = THETA*0.017453294
      SNTH = SIN(TH)
      CSTH = COS(TH)
      GAMMA = +30.0
      GM = GAMMA*0.017453294
      SNGM = SIN(GM)
      CSGM = COS(GM)
      YOFF = 3.5
C
      NSV = 0
      NSSV = 0
      N1SV = 0
      NSTRT = 0
      NLAST = 0
      NSSTRT = 0
      NSLAST = 0
C
      DO 1000 K=1,M
C
      IF (K .LE. 1) GO TO 100
      THETA = -150.0
      TH = THETA*0.017453294
      SNTH = SIN(TH)
```

```
CSTH = COS(TH)
GAMMA = -30.0
GM = GAMMA*0.017453294
SNGM = SIN(GM)
CSGM = COS(GM)
YOFF = 5.5
```

```
C
100 NSTRT = NLAST+1
    NSSTRT = NSLAST+1
    N1 = IM(K)*JM(K)
    N2 = IM(K)+JM(K)+1
    IF (N1 .GT. 0) GO TO 130
    N1 = NMAX1
    N2 = JM(K)+IMXM+1
130 NLAST = NLAST+N1
    NSLAST = NSLAST+N2
```

```
C
C
C DETERMINE RANGE OF VALUES OF X AND Z
```

```
XMIN = X1(NSTRT)
XMAX = XMIN
ZMIN = Z1(NSTRT)
ZMAX = ZMIN
SMIN = 0.0
SMAX = ABS(S1(NSTRT))
DO 200 N=NSTRT,NLAST
  IF (X1(N) .LT. XMIN) XMIN = X1(N)
  IF (X1(N) .GT. XMAX) XMAX = X1(N)
  IF (Z1(N) .LT. ZMIN) ZMIN = Z1(N)
  IF (Z1(N) .GT. ZMAX) ZMAX = Z1(N)
  IF (ABS(S1(N)) .GT. SMAX) SMAX = ABS(S1(N))
200 CONTINUE
DO 260 NS=NSSTRT,NSLAST
  IF (XS1(NS) .LT. XMIN) XMIN = XS1(NS)
  IF (XS1(NS) .GT. XMAX) XMAX = XS1(NS)
  IF (ZS1(NS) .LT. ZMIN) ZMIN = ZS1(NS)
  IF (ZS1(NS) .GT. ZMAX) ZMAX = ZS1(NS)
260 CONTINUE
XT(1) = XMIN
XT(2) = XMAX
ZT(1) = ZMIN
ZT(2) = ZMAX
ST(1) = SMIN
ST(2) = SMAX
```

```
C
C
C SCALE SO THAT X DOES NOT EXCEED 5 IN.,DRAW AXIS, LABEL
CALL SCALE (ST,2, 5.0,SMN,DSS,1)
SLNGTH = AINT(ST(2)+0.99)
CALL SCALE (XT,2, 5.0,XMN,DXX,1)
XLNGTH = AINT(XT(2)+0.99)
DZZ = DXX
IF (K .GE.2) DZZ = DXX/10.0
```

```

ZMN = AINT(ZT(1))
ZLNTH = AINT((ZT(2)-ZMN)/DZZ+0.99)
IF (ZLNTH .LE. 5.0) GO TO 380
CALL SCALE (ZT,2, 5.0,ZMN,DZZ,1)
ZLNTH = AINT(ZT(2)+0.99)
C
380 CALL PLOT ( 4.0,YOFF,-3)
CALL AX2 (0.0,0.0,LABL(1), 12,SLNTH, 90.0,SMN,DSS,-2.0)
CALL SYMBOL (0.5,4.85,0.14,TITLE,0.0, 72)
CALL SYMBOL (0.5, 4.60,0.14,5HTIME=,0.0,5)
T = TIME*1.0E+6
CALL NUMBER (1,10,4.60,0.14,T ,0.0, 6)
CALL PMU (2,18, 4.60,0.14)
CALL SYMBOL (2,30, 4.60, 0.14,4HSEC.,0.0, 4)
THR = 180.0+THETA
CALL AX2 (XLNTH*CSTH,XLNTH*SNTH,1H , 1,-XLNTH,THR ,XLNTH* DXX
1,-DXX, XLNTH)
CALL SYMBOL ((XLNTH+2.0)*CSTH,(XLNTH+2.0)*SNTH,0.14,13HRADIAL (X
1)-CM,THR , 13)
CALL AX2 ( 0.0, 0.0, 1H , 1,ZLNTH,GAMMA,ZMN,DZZ,-ZLNTH)
C
ISWTCH = 2
500 NS = NSSV
N = NSV
N1 = N1SV
C
C IF ISWTCH = 1, PLOT AT CONSTANT Z
C IF ISWTCH = 2, PLOT AT CONSTANT X
C
GO TO (520,720),ISWTCH
520 JMX = JM(K)
DO 680 J=1,JMX
NS = NS+1
IC = 3
IMXT = IM(K)
IMX = IM(K)
IF (IMX .GT. 0) GO TO 580
IMX = IM1(J)
IMXT = IMXM
580 DO 640 I=1,IMX
N = N+1
TEMP1 = X1(N)-XMN
TEMP2 = Z1(N)-ZMN
XPLT = (TEMP2*CSGM)/DZZ + (TEMP1*CSTH)/DXX
YPLT = -(S1(N)-SMN)/DSS + (TEMP2*SNGM)/DZZ + (TEMP1*SNTH)/DXX
CALL PLOT (XPLT,YPLT,IC)
IC = 2
640 CONTINUE
C
IF (K .LE. M) GO TO 680
C
650 TEMP1 = XS1(NS)-XMN
TEMP2 = ZS1(NS)-ZMN

```

```
XPLT = (TEMP2*CSGM)/DZZ + (TEMP1*CSTH)/DXX
YPLT = (TEMP2*SNGM)/DZZ + (TEMP1*SNTH)/DXX
CALL PLOT (XPLT,YPLT,IC)
680 CONTINUE
IF (K .LT. M) GO TO 700
IC = 3
DO 690 I=1,IMX
NS = NS+1
TEMP1 = XS1(NS)-XMN
TEMP2 = ZS1(NS)-ZMN
XPLT = (TEMP2*CSGM)/DZZ + (TEMP1*CSTH)/DXX
YPLT = (TEMP2*SNGM)/DZZ + (TEMP1*SNTH)/DXX
CALL PLOT (XPLT,YPLT,IC)
IC = 2
690 CONTINUE
NS = NS+1
GO TO 960
700 NS = NS+IMXT+1
GO TO 960
C
720 N1 = N
IMX = IM(K)
IF (IMX .GT. 0) GO TO 740
IMX = IMXM
740 DO 850 I=1,IMX
IC = 3
JMX = JM(K)
MXT = 0
IF (NCRVS .EQ. 0) GO TO 780
DO 773 L=1,NCRVS
IF (I .EQ. JCRVS(L)) GO TO 780
773 CONTINUE
GO TO 850
780 DO 840 J=1,JMX
IMXT = (J-1)*IMX
IF (IM(K) .GT. 0) GO TO 800
IF (J .LE. 1) GO TO 790
MXT = MXT+IM1(J-1)
790 IMXT = MXT
800 N = N1+IMXT+I
IF ((I .LE. IM1(J)) .OR. (IM(K) .NE. 0)) GO TO 820
IC = 3
GO TO 840
820 TEMP1 = X1(N)-XMN
TEMP2 = Z1(N)-ZMN
XPLT = (TEMP2*CSGM)/DZZ + (TEMP1*CSTH)/DXX
YPLT = -(S1(N)-SMN)/DSS + (TEMP2*SNGM)/DZZ + (TEMP1*SNTH)/DXX
CALL PLOT (XPLT,YPLT,IC)
IC = 2
840 CONTINUE
IF (K .LT. M) GO TO 850
L = JMX+I+NS
TEMP1 = XS1(L)-XMN
```

```
      TEMP2 = ZS1(L)-ZMN
      XPLT = (TEMP2*CSGM)/DZZ + (TEMP1*CSTH)/DXX
      YPLT = (TEMP2*SNGM)/DZZ + (TEMP1*SNTH)/DXX
      CALL PLOT (XPLT,YPLT,IC)
850 CONTINUE
C
      IF (K .LE. M) GO TO 950
C
860 IC = 3
      DO 930 J=1,JMX
      NS = NS+1
      TEMP1 = XS1(NS)-XMN
      TEMP2 = ZS1(NS)-ZMN
      XPLT = (TEMP2*CSGM)/DZZ + (TEMP1*CSTH)/DXX
      YPLT = (TEMP2*SNGM)/DZZ + (TEMP1*SNTH)/DXX
      CALL PLOT (XPLT,YPLT,IC)
      IC = 2
930 CONTINUE
      NS = NS+IMX+1
950 CONTINUE
      ISWTCH = 1
      GO TO 500
C
960 CALL SYMBOL((ZLNTH+0.75)*CSGM,(ZLNTH+0.75)*SNGM,0.14,12HAXIAL (Z
1)-CM, GAMMA, 12)
      CALL PLOT (ZLNTH+4.50,-YOFF,-3)
C
      NSSV = NS
      NSV = N
      N1SV = N1
C
1000 CONTINUE
      RETURN
      END
```

```
$IBFTC GRP400 LIST,REF
      SUBROUTINE GRP4
C
C 2-D LAGRANGIAN -- DRAWS MESHES FOR A DEBRIS CLOUD
C
      EQUIVALENCE      (ISPACE(4),NSR)
      EQUIVALENCE      (TEMP1,XMN),      (TEMP2,ZMN)
C
      JMX = JM(K)
      IMX = 0
      IDP = 0
      DO 600 J=1,JMX
      IMM = IMX
      IMX = IM1(J)
      IDM = -IDP
      IF (J .LT. JMX) GO TO 150
      IMP = 0
      IDP = 0
      GO TO 200
150  IMP = IM1(J+1)
      IDP = IMP-IMX
200  NS = NS+1
      NSR = NS
      IF (IDM .GT. 0) NSR = NSR-IDM
      IC = 3
      DO 400 I=1,IMX
      N = N+1
      XPLT = (X1(N)-XMN)/DZZ
      YPLT = (Z1(N)-ZMN)/DZZ
      CALL PLOT (XPLT,YPLT,IC)
      IC = 2
      IF (I .LF. IMP) GO TO 320
      NS = NS+1
```

TR66-85  
Appendix C

GRP 4  
Page 2 of 2

```
      XPLTW= (XS1(NS)-XMN)/D77
      YPLTW= (ZS1(NS)-ZMN)/DZZ
      GO TO 350
320  NW = N+IMX
      XPLTW= (X1(NW)-XMN)/DZZ
      YPLTW= (Z1(NW)-ZMN)/DZZ
350  CALL PLOT (XPLTW,YPLTW,IC)
      IC = 3
      CALL PLOT (XPLT,YPLT,IC)
      IC = 2
400  CONTINUE
      XPLT = (XS1(NSR)-XMN)/D77
      YPLT = (ZS1(NSR)-ZMN)/DZZ
      CALL PLOT (XPLT,YPLT,IC)
      IF (IDP .GT. 0) GO TO 440
      NW = NS+1
      XPLTW= (XS1(NW)-XMN)/DZZ
      YPLTW= (ZS1(NW)-ZMN)/DZZ
      GO TO 470
440  NW = NW+1
      XPLTW = (X1(NW)-XMN)/D77
      YPLTW = (Z1(NW)-ZMN)/D77
470  CALL PLOT (XPLTW,YPLTW,IC)
      IC = 3
      CALL PLOT (XPLT,YPLT,IC)
      IC = 2
      IF (IDM .LE. 0) GO TO 600
      NW = NSR
      DO 550 LL=1, IDM
      NW = NW+1
      XPLT = (XS1(NW)-XMN)/D77
      YPLT = (ZS1(NW)-ZMN)/DZZ
      CALL PLOT (XPLT,YPLT,IC)
550  CONTINUE
C
600  CONTINUE
C
      NS = NS+1
C
      RETURN
      END
```

```
41BETC PMU      LIST,REF  
SUBROUTINE PMU (X,Y,H)  
C  
HU = H*0.715  
CALL SYMBOL (X,Y,HU,1HU, 0.0,1)  
RETURN  
END
```



```

$IBFTC AX2M      LIST
C      SUBROUTINE AX2 - MODIFIED AXIS TO ALLOW AXIS ANNOTATION AT OTHER AX200010
C      THAN 1.0 INCH INCREMENTS. Z IS THE NUMBER OF AX200020
C      INCHES BETWEEN AXIS ANNOTATIONS. AX200030
      SUBROUTINE AX2 (X,Y,BCD,NC,SIZE,THETA,YMIN,ZDY,Z ) AX200040
      CALL OVERFL (KOV) AXS40031
      SIG = 1.0 AXIS2050
      IF(NC) 1,2,2 AXIS2060
1 SIG=-1.0 AXIS2070
2 NAC=IABS(NC) AXIS2080
      TH=THETA*0.017453294 AXIS2081
      ZSIG = Z AX2M 090
      Z = ABS(Z) AX2M 091
      SSIG = SIZE AX2M 092
      SIZE = ABS(SIZE) AX2M 093
      CTH = COS (TH) AXIS2100
      STH = SIN (TH) AXIS2110
      DY = Z *ZDY AX200120
      SIZE = SIZE + 0.0001 AX200121
      N = SIZE / Z AX200130
      WN = N AX200140
      XB = X AXIS2143
      YB = Y AXIS2144
      XA = X - 0.1 * SIG * STH AXIS2150
      YA = Y + 0.1 * SIG * CTH AXIS2160
      CALL PLOT (XA,YA,3) AXIS2170
      IC = 2 AX2M 171
      IF (ZSIG .LT. 0.0) IC = 3 AX2M 172
      DO 20 I =1,N AXIS2180
      CALL PLOT (XB,YB,IC) AX2M 190
      IC = 2 AX2M 191
      XC = XB + CTH * Z AXIS2200
      YC = YB + STH * Z AXIS2210
      CALL PLOT (XC,YC,2) AXIS2220
      XA = XA + CTH * Z AXIS2230
      YA = YA + STH * Z AXIS2240
      IF ((SSIG .LT. 0.0) .AND. (I .EQ. N)) GO TO 226 AX2M2241
      CALL PLOT (XA,YA,2) AXIS2250
226 XB = XC AXIS2260
20 YB = YC AXIS2270
      KEXP = 0 AXS40271
      IF ( SIZE - ABS(WN * Z ) ) 28, 28,271 AX200270
271 CALL PLOT (XB,YB, 2) AX200271
      XC = CTH * SIZE + X AX200272
      YC = STH * SIZE + Y AX200273
      CALL PLOT (XC,YC, 2) AX100274
      XA = XC - 0.1 * SIG * STH AX200275
      YA = YC + 0.1 * SIG * CTH AX200276
      CALL PLOT (XA,YA, 2) AX100277
28 CHAR = ABS (YMIN) AXS40280
      ARSV = ABS (YMIN + WN * DY) AXS40290
      CALL OVERFL (KOV) AXS40291
      GO TO (293, 13,295), KOV AXS40292

```

|     |   |          |
|-----|---|----------|
| 293 | KEXP = KEXP + 38  | AXS40293 |
|     | GO TO 296   | AXS40294 |
| 295 | KEXP = KEXP - 38  | AXS40295 |
| 296 | CHAR = ABS (YMIN/ 10.0 ** KEXP)                               | AXS40296 |
|     | ABSV = ABS (YMIN/(10.0 ** KEXP)+ (WN * DY) / (10.0 ** KEXP))  | AXS40297 |
| 13  | IF (ABSV .GE. CHAR) GO TO 32                                  | AXS40300 |
|     | ABSV = CHAR   | AXS40310 |
| 32  | DO 35 J = 1,40  | AXS40320 |
|     | K = J - 1   | AXS40330 |
|     | IF (ABSV .LT. (10.0 ** K)) GO TO 39                           | AXS40340 |
| 35  | CONTINUE  | AXS40350 |
| 36  | WRITE (6,37)  | AXS40360 |
| 37  | FORMAT (1H0,24HERROR IN SUBROUTINE AXIS)                      | AXS40370 |
| 39  | IF (K) 402,391, 41  | AXS40390 |
| 391 | DO 40 J = 1,40  | AXS40391 |
|     | K = -J  | AXS40392 |
|     | IF (ABSV .GT. (10.0 ** K )) GO TO 402                         | AXS40393 |
| 40  | CONTINUE  | AXS40400 |
|     | GO TO 36  | AXS40401 |
| 402 | IEXP = K + 1  | AXS40402 |
|     | KM = 5  | AXS40403 |
| 404 | ABSV = ABSV / (10.0 ** (IEXP - 3))                            | AXS40404 |
|     | GO TO 422   | AXS40405 |
| 41  | IF (K .LT. 4) GO TO 42  | AXS40410 |
|     | KM = 4  | AXS40411 |
|     | IFXP = K - 1  | AXS40412 |
|     | GO TO 404   | AXS40413 |
| 42  | KM = 5 - K  | AXS40420 |
|     | IEXP = 0  | AXS40421 |
| 422 | NSIG = 3  | AXS40422 |
| 423 | NT = ABSV   | AXS40423 |
|     | CHAR = NT   | AXS40430 |
|     | IF (ABSV .EQ. CHAR) GO TO 93                                  | AXS40431 |
|     | NSIG = NSIG + 1   | AXS40432 |
|     | IF (NSIG .LT. KM) GO TO 423                                   | AXS40433 |
|     | NSIG = KM   | AXS40434 |
| 93  | IEXP = IEXP + KEXP  | AXS40435 |
|     | ADY = DY * 10.0 ** (-IEXP)                                    | AX200446 |
|     | ABSV= YMIN * 10.0 ** (-IEXP) + WN * ADY                       | AX200450 |
|     | XA = XB - (.20 * SIG - .05) *STH - .0857 * CTH                | AXIS2460 |
|     | YA = YB + (.20 * SIG - .05) * CTH - .0857 * STH               | AXIS2470 |
|     | N = N + 1   | AXIS2480 |
|     | DO 30 I = 1,N   | AXIS2490 |
|     | IF ((ZSIG .LT. 0.0) .AND. (I .EQ. N)) GO TO 51                | AX2M 491 |
|     | IF (SSIG .LT. 0.0) GO TO 51                                   | AX2M 492 |
|     | CALL NUMBER (XA,YA,0.1,ABSV,THETA,NSIG)                       | AX200500 |
| 51  | ABSV = ABSV - ADY   | AXIS2510 |
|     | SSIG = ABS(SSIG)  | AX2M 511 |
|     | XA = XA - CTH * Z   | AXIS2520 |
| 30  | YA = YA - STH * Z   | AXIS2530 |
|     | TNC = NAC + 7   | AXIS2540 |
|     | XA = X + (SIZE / 2.0 -.06 * TNC)*CTH - (-.07 + SIG *.36)* STH | AXIS2550 |
|     | YA = Y + (SIZE / 2.0 -.06 * TNC)*STH + (-.07 + SIG *.36)* CTH | AXIS2560 |

```
CALL SYMBOL (XA,YA,0.14,BCD,THETA,NAC)
XA = XA +((TNC -6.0) * 0.12)* CTH
YA = YA +((TNC -6.0) * 0.12)* STH
IF(IEXP) 61,50,61
61 CALL SYMBOL (XA,YA,0.14,7H(X10 ),THETA,7)
XA = XA + .48 * CTH -.07 * STH
YA = YA + .48 * STH +.07 * CTH
IF(IEXP) 65,50,65
65 EXP = IEXP
CALL NUMBER (XA,YA,0.10,EXP,THETA,-1)
50 RETURN
END
```

AXIS2570  
AXIS2580  
AXIS2590  
AXS40600  
AXS40610  
AXS40620  
AXS40630  
AXS40640  
AXS40641  
AXS40650  
AXIS2640

|         |        |          |                          |      |    |
|---------|--------|----------|--------------------------|------|----|
| \$IBMAP | RWND2  | LIST     |                          | RWD2 | 10 |
| *       |        |          | CALL RWND2               | RWD2 | 20 |
|         | ENTRY  | RWND2    |                          | RWD2 | 30 |
| RWND2   | SAVE   | 1,2,4    |                          | RWD2 | 40 |
|         | TSX    | S.I00P,4 | REWIND TAPE 2            | RWD2 | 50 |
|         | IOSKP  | RWND     | WITH NO END-OF-FILE MARK | RWD2 | 60 |
|         | PZE    | S.SU02   |                          | RWD2 | 70 |
|         | RETURN | RWND2    |                          | RWD2 | 80 |
| RWND    | PZE    | 0,,-1    |                          | RWD2 | 90 |
|         | END    |          |                          |      | -  |

|         |        |          |                          |      |    |
|---------|--------|----------|--------------------------|------|----|
| \$IBMAP | RWND0  | LIST     |                          | RWD0 | 10 |
| *       |        |          | CALL RWND0               | RWD0 | 20 |
|         | ENTRY  | RWND0    |                          | RWD0 | 30 |
| RWND0   | SAVE   | 1,2,4    |                          | RWD0 | 40 |
|         | TSX    | S.I00P,4 | REWIND TAPE 0.           | RWD0 | 50 |
|         | IOSKP  | RWND     | WITH NO END-OF-FILE MARK | RWD0 | 60 |
|         | PZE    | S.SU00   |                          | RWD0 | 70 |
|         | RETURN | RWND0    |                          | RWD0 | 80 |
| RWND    | PZE    | 0,,-1    |                          | RWD0 | 90 |
|         | END    |          |                          |      | -  |

```

$IRMAP FPTMOD LIST,REF
      PMC      ON
DPOPT SET      1          NOW SET TO GIVE D P SIMULATION
*DPOPT SET      0          USED TO REMOVE D P SIMULATION
*                                     ROUTINE FOR DP INSTRUCTIONS
*                                     EXECUTED ON SP MACHINES.
FPTLIM EQU      1000      FLOATING POINT TRAP LIMIT
*
      ENTRY    SFTFP.
      ENTRY    OVERF.
      ENTRY    FPT          CALL FPT(SILNCE,I)
      EXTERN   FEXEM.,EXIT,FIL06.,HNLIO.,FILIO.,SLOIO.,STHIO.,IOHEF.,
      ETC      IOHHC.,IOHOC.,IOHAC.,S.SLOC,S.PLOC,S.FBOU
* * * * *
* ENTRY FROM MAIN PROGRAM TO SET UP SPECIAL ENTRY FOR FLOATING PT. TRAPS.
SFTFP. TRA      **
      CAL      FLPTRP      SET RETURN TO THIS ROUTINE IN
      SLW      8           CASE OF FLT-PT TRAPS
      CAL      INTTIM      IGNORE INTERVAL TIMER RESET TRAP, CON-
      SLW      31          TINUE WITH EXECUTION.
      TRA      SETFP.      RETURN TO CALLING PROGRAM
INTTIM TRA*     30
* * * * *
* ENTRY FROM FORTRAN PROGRAMS TO SET UP SPECIAL HANDLING OF FLT-PT TRAPS.
MINUS STA      FPTLMM      SET NEW LIMIT
      MSM      LIM         SET LIMIT SW ON
XR4   AXT      **,4        RESTORE INDEX 4
FPT   TRA      **          CALL FPT (SILNCE,I)
* IF SILNCE=.TRUE.= -, THEN DONT WRITE OUT MESSAGE WHEN UNDERFLOW OR
* D.P. TRAPS OCCUR.
* IF I IS + THEN EXECUTE I UPON LEAVING FLPTRP ON OVERFLOW, NO LIMIT
* TO UNDERFLOWS OR D.P. TRAPS, CONTINUE PROGRAM EXECUTION.
* IF I IS - THEN MAG(I) = FPTLIM
* IF I IS 0 THEN TREAT OVERFLOWS THE SAME AS UNDERFLOW AND D.P. TRAPS
* AND ALLOW AN INFINITE NUMBER OF TRAPS.
      SXA      XR4,4        SAVE INDEX 4
      LAC      FPT,4
      MSP      SILNCE
      PLT*     2,4          IF 1ST ARGUMENT IS TRUE, THEN
      MSM      SILNCE      DON-T WRITE MESSAGE
AR2   CLA*     3,4
      TMI     MINUS
      TZE     ZERO
      STA     IRT          I=+, SO SET OVERFL EXIT'
      MSP     IGNORE
NOLIM MSP     LIM
      TRA     XR4          RETURN
ZERO  MSM     IGNORE
      TRA     NOLIM
* * * * *
*

```

\* SPECIAL ENTRY FOR FLT-PT TRAPS OCCURRING IN FORTRAN PROGRAMS.

|        |     |                |   |
|--------|-----|----------------|---|
| FLPTRP | TRA | *+1            |   |
|        | SXA | FPTX,4         | SAVE XR4                                  |
|        | STO | SAC            | SAVE AC                                   |
|        | STO | SMQ            | AND MQ                                    |
|        | CLA | 0              | SAVE CONTENTS OF TRAP CELL                |
|        | SLW | OVERF.         | FOR OVERF. ROUTINE                        |
|        | IFT | DPOPT=1        |   |
|        | TMI | DPFLPT         | DP INSTRUCTION ON SP MACHINE              |
|        | IFF | DPOPT=1        |   |
|        | TMI | SFTAC          | IGNORE THIS TRAP                          |
|        | ANA | =000004000000  | WAS THIS A DOUBLE PRECISION ERROR         |
|        | TZE | CAL            | NO  |
|        | CLA | D              | YES, WRITE COMMENT, TEST LIMIT, CONTINUE  |
| SILNCE | TXL | TESTLM,0,      | IF SILNCE=.TRUE. DONT WRITE MESSAGE       |
|        | TSL | WRITE          |   |
|        | TRA | TESTLM         |   |
| CAL    | CAL | OVERF.         |   |
|        | ANA | =0000004000000 | WAS TRAP CONDITION OVERFLOW               |
|        | TZE | UNFL           | NO, MUST BE UNDERFLOW                     |
|        | CLA | 0              | YES, WRITE COMMENT                        |
|        | PLT | IGNORE         |   |
|        | MIT | SILNCF         |   |
|        | TSL | WRITE          |   |
|        | MSM | OVERF.         | SET OVERFLOW SW. ON                       |
| LIM    | TXL | FPTLMM,0,      | TXL A,0 OR TXH A,0 =BRN A =BRANCH TO A IF |
| *      |     |                | THIS CELL IS NEGATIVE                     |
|        | TRA | CAL1           |   |
| UNFL   | CLA | U              | WRITE UNDERFL COMMENT                     |
|        | XEC | SILNCE         |   |
|        | TSL | WRITE          |   |
| TESTLM | MIT | LIM            | IS THERE A LIMIT                          |
|        | TRA | CAL1           | NO  |
| FPTLMM | AXT | FPTLIM,4       | HAS TRAP LIMIT BEEN REACHED               |
|        | TNX | FPTOUT,4,1     | YES, THEN RESTORE TRAPS, CALL FEXEM.      |
|        | SXA | *-2,4          | NO, REDUCE COUNTER                        |
| CAL1   | CAL | OVERF.         |   |
|        | ANA | =0000002000000 | WAS CONDITION IN AC                       |
|        | TZE | MQDB           | NO, ONLY IN MQ OR DOUBLE PRECISION        |
|        | MSP | ACMQSW         | INITIALIZE BYPASS SW                      |
|        | CAL | OVERF.         |   |
|        | ANA | =0000001000000 | WAS CONDITION IN BOTH AC AND MQ           |
|        | TN7 | *+2            | YES                                       |
|        | MSM | ACMQSW         | NO, ONLY IN AC, SET BYPASS SW ON          |
|        | ZAC |                | UNDERFL, MAKE AC ALL ZEROS                |
|        | PLT | OVERF.         |   |
|        | COM |                | OVERFLOW, MAKE AC ALL ONES                |
|        | PLT | SAC            | RESTORE AC SIGN                           |
|        | CHS |                |   |
|        | STO | SAC            | SAVE AC SETTING                           |
| ACMQSW | BRN | SETAC          | TRA IF CONDITION ONLY IN AC               |
| SETMQ  | LDQ | SMQ            |   |
|        | LLS | 0              | RESTORE MQ SIGN                           |

TR66-85  
Appendix C

|         |      |                  |                                      |
|---------|------|------------------|--------------------------------------|
|         | STO  | SMQ              | SAVE MQ SETTING                      |
| SFTAC   | CLA  | SAC              | RESTORE AC                           |
|         | LDQ  | SMQ              | AND MQ SETTINGS                      |
| FPTX    | AXT  | ** , 4           | RESTORE XR4 ,                        |
|         | MIT  | LIM              |                                      |
|         | MIT  | OVERF .          |                                      |
|         | TRT* | OVERF .          | RETURN TO TRAPPED LOCATION +1        |
| IGNORE  | BRN  | *-1              | IGNORE PRESET EXIT                   |
| TRT     | TRT  | **               | GO TO PRESET EXIT SET BY CALL TO FPT |
| *       |      |                  |                                      |
| MOQB    | CAL  | OVERF .          |                                      |
|         | ANA  | =0000040000000   | WAS THIS A DOUBLE PRECISION ERROR    |
|         | INZ  | SFTAC            |                                      |
|         | PLT  | OVERF .          | AC =0 NOW                            |
|         | COM  |                  | OVERFLOW , MAKE AC ALL ONES          |
|         | TRA  | SFTMQ            |                                      |
| FPTOUT  | TRT  | *+1              |                                      |
|         | TSX  | S , SLOC , 4     | PURGE OUTPUT BUFFER                  |
|         | PZE  | S , FROU , , 149 |                                      |
|         | TSL  | FEXEM .          | ERROR TRAP LIMIT REACHED             |
|         | PZE  | EXIT , , 31      | NO OPTIONAL RETURN                   |
| *       |      |                  |                                      |
| SAC     | D7E  |                  |                                      |
| SMQ     | D7E  |                  |                                      |
| OVERF . | D7E  |                  | INDICATORS FROM LAST TRAP            |
| D       | D7E  | DQUR , , 2       |                                      |
| O       | D7E  | OV , , 2         |                                      |
| U       | D7E  | UN , , 2         |                                      |
| DQUR    | RCI  | 2 , D.P. ADDRESS |                                      |
| OV      | RCI  | 2 , OVERFLOW     |                                      |
| UN      | RCI  | 2 , UNDERFLOW    |                                      |
| *       |      |                  |                                      |
| X2      | AXT  | ** , 2           | RESTORE XR2                          |
| WRITE   | TRA  | **               |                                      |
|         | SXA  | X2 , 2           | SAVE XR2                             |
|         | STO  | TYPE             |                                      |
|         | LXA  | S , SLOC , 4     | FIND NAME OF ROUTINE IN              |
|         | TXL  | M1 , 4 , 0       | WHICH TRAP OCCURED , NAME            |
|         | CLA  | 0 , 4            | OF CALLING ROUTINE AND               |
|         | PAC  | , 2              | CALLING STATEMENT                    |
|         | CLA  | -6 , 2           |                                      |
|         | STO  | CULPRT           |                                      |
|         | CLA  | 2 , 4            |                                      |
|         | STO  | CALLS            |                                      |
|         | CLA  | -3 , 2           |                                      |
|         | PAX  | , 4              |                                      |
|         | TXL  | M2 , 4 , 0       |                                      |
|         | CLA  | 0 , 4            |                                      |
|         | PAC  | , 2              |                                      |
|         | CLA  | -6 , 2           |                                      |
|         | STO  | CALLR            |                                      |
|         | TRA  | STH              |                                      |
| M1      | STZ  | CULPRT           | S , SLOC = 0 WHEN CONTROL IS IN      |





|                            |      |                |                                     |
|----------------------------|------|----------------|-------------------------------------|
| ADD1                       | CLA  | **             | CLA Y / CLA Y,T / CLA* Y / CLA* Y,T |
|                            | STO  | ARG1           | STASH Y                             |
| INDRK                      | BRA  | ADD2           | BRANCH IF ADDRESS IS DIRECT         |
|                            | XFC  | DPINS          |                                     |
|                            | ANA  | =0000000777777 |                                     |
|                            | ORA  | CLA            |                                     |
|                            | STO  | **+1           |                                     |
|                            | CLA  | **             | CLA Y / CLA Y,T                     |
|                            | ANA  | =0000000777777 |                                     |
|                            | ORA  | CLA            | CLA* Y / CLA* Y,T                   |
|                            | TRA  | **+2           |                                     |
| ADD2                       | CLA  | ADD1           |                                     |
|                            | ADD  | =1             |                                     |
|                            | STO  | **+1           |                                     |
|                            | CLA  | **             | CLA Y+1 / CLA Y+1,T                 |
|                            | STO  | ARG2           | STASH Y+1                           |
|                            | XFC  | DPINS          |                                     |
|                            | ANA  | =0777700000000 | GET OP CODE                         |
|                            | LAS  | DFDP           |                                     |
|                            | TRA  | SFTAC          |                                     |
|                            | TRA  | DPDVP          | -0241 (DP DIVIDE)                   |
|                            | LAS  | DFSR           |                                     |
|                            | TRA  | SFTAC          |                                     |
|                            | TRA  | DPSUB          | +0303 (DP SUBTRACT)                 |
|                            | LAS  | DFAD           |                                     |
|                            | TRA  | SFTAC          |                                     |
|                            | TRA  | DPADD          | +0301 (DP ADD)                      |
|                            | TRA  | DPMPY          | +0234 (DP MULTIPLY)                 |
| *                          | DFSR | SUBTRACT       | SAME AS DFAD WITH SIGN CHANGE       |
| *                          |      |                |                                     |
| DPSUB                      | CLA  | ARG1           |                                     |
|                            | CHS  |                |                                     |
|                            | STO  | ARG1           |                                     |
|                            | CLA  | ARG2           |                                     |
|                            | CHS  |                |                                     |
|                            | STO  | ARG2           |                                     |
| *                          | DFAD | ADD            | (A+B)+(C+D)=B+D+(A+C)LOW+A+C HIGH   |
| DPADD                      | CLA  | AC             | FORM A+C HIGH AND LOW               |
|                            | FAD  | ARG1           |                                     |
|                            | STO  | TEMP           | A+C HIGH                            |
|                            | STO  | SMCHAR         | A+C LOW                             |
|                            | CLA  | MO             | FORM B+C                            |
|                            | FAD  | ARG2           | LOSE MQ LOW ORDER                   |
|                            | FAD  | SMCHAR         | A+C LOW                             |
|                            | FAD  | TEMP           | A+C HIGH ANSWER NOW IN AC MQ        |
|                            | TRA  | THER           | EXIT WITH ANSWER                    |
| *DOUBLE PRECISION MULTIPLY |      |                | (A + B) * (C + D) = A*C + B*C + A*D |
| DPMPY                      | LDO  | AC             | A*C                                 |
|                            | FMP  | ARG1           |                                     |
|                            | STO  | MARG1          |                                     |
|                            | STO  | MARG2          |                                     |
|                            | LDO  | MO             | B*C                                 |
|                            | FMP  | ARG1           |                                     |

|                          |        |         |                                    |
|--------------------------|--------|---------|------------------------------------|
|                          | STO    | MQ      | B*C HIGH ORDER B*C NOT SIGNIFICANT |
|                          | LDQ    | AC      | A*D                                |
|                          | FMP    | ARG2    |                                    |
|                          | FAD    | MQ      | ADD IN B*C HIGH                    |
|                          | FAD    | MARG2   | ADD IN A*C LOW                     |
|                          | FAD    | MARG1   | ADD IN A*C HIGH                    |
|                          | TRA    | THFR    | RETURN                             |
| *DOUBLE PRECISION DIVIDE |        |         | (A + B) / (C + D) = Q1 + Q2        |
| DPDVP                    | CLA    | AC      | BEGIN DIVIDE SIMULATION            |
|                          | FDP    | ARG1    | (A + B) / C = Q1 + R1              |
|                          | STQ    | Q1      |                                    |
|                          | STO    | R1      |                                    |
|                          | FMP    | ARG2    | Q1 * D                             |
|                          | CHS    |         |                                    |
|                          | FAD    | R1      | R1 - Q1 * D                        |
|                          | FAD    | MQ      | ADD IN B TO BE DIVIDED             |
|                          | FDP    | ARG1    |                                    |
|                          | STQ    | TFMP    | (R1 - Q1 * D) / C = Q2 + R2        |
|                          | CLA    | Q1      |                                    |
|                          | FAD    | TEMP    |                                    |
| THFR                     | AXT    | ** ,4   | RESTORE INDEX ON EXIT              |
|                          | TRT*   | THEND   |                                    |
| *                        |        |         |                                    |
| AC                       | PZE    | 0       |                                    |
| MQ                       | PZE    | 0       |                                    |
| THEND                    | PZE    | 0       |                                    |
| CLA                      | CLA    | **      |                                    |
| ARG1                     | PZE    | 0       | HIGH-ORDER OPERATOR                |
| ARG2                     | PZE    | 0       | LOW-ORDER OPERATOR                 |
| DFDP                     | DFDP   | **      |                                    |
| DFSR                     | DFSR   | **      |                                    |
| DFAD                     | DFAD   | **      |                                    |
| SMCHAR                   | PZE    | 0       | SMALLER CHARACTERISTIC             |
| TEMP                     | PZE    | 0       | TEMPORARY STORAGE                  |
| MARG1                    | FQU    | SMCHAR  |                                    |
| MARG2                    | FQU    | TEMP    |                                    |
| Q1                       | FQU    | SMCHAR  |                                    |
| R1                       | FQU    | TEMP    |                                    |
|                          | ENDM   | DPTRAN  |                                    |
|                          | IFT    | DPOPT=1 |                                    |
|                          | DPTRAN |         |                                    |
|                          | END    |         |                                    |

The term  $(\rho A)$  appearing in these equations is required at 0, where  $\rho$  and  $A$  are quantities averaged over the meshes 1, 2, 3 and 4. This term is well represented by the expression

$$(\rho A)_0 = A_{I1}\rho_1 + A_{I2}\rho_2 + A_{I3}\rho_3 + A_{I4}\rho_4$$

where  $A_{I1}$  is the area of triangle AOD,  $A_{I2}$  of AOB,  $A_{I3}$  of BOC, and  $A_{I4}$  of COD.

The last term on the right of the momentum equation is written

$$\left( \frac{t^{xx} - t^{yy}}{\rho x} \right)_0 = \frac{1}{4} \left\{ \frac{[(t^{xx})_1 - (t^{yy})_1]}{\rho_1 \bar{x}_{I1}} + \frac{[(t^{xx})_2 - (t^{yy})_2]}{\rho_2 \bar{x}_{I2}} \right. \\ \left. + \frac{[(t^{xx})_3 - (t^{yy})_3]}{\rho_3 \bar{x}_{I3}} + \frac{[(t^{xx})_4 - (t^{yy})_4]}{\rho_4 \bar{x}_{I4}} \right\}$$

where

$$\bar{x}_{I1} = \frac{1}{3} (x_O + x_D + x_A)$$

$$\bar{x}_{I2} = \frac{1}{3} (x_O + x_A + x_B)$$

$$\bar{x}_{I3} = \frac{1}{3} (x_O + x_B + x_C)$$

$$\bar{x}_{I4} = \frac{1}{3} (x_O + x_C + x_D)$$

Similarly for the second momentum equation

$$\left( \frac{1}{\rho} \frac{\partial t^{xz}}{\partial x} \right)_0 = \frac{-1}{(\rho A)_0} \left\{ (t^{xz})_1 (z_D - z_A) + (t^{xz})_2 (z_A - z_B) \right. \\ \left. + (t^{xz})_3 (z_B - z_C) + (t^{xz})_4 (z_C - z_D) \right\}$$

$$\left( \frac{1}{\rho} \frac{\partial t^{zz}}{\partial z} \right)_0 = \frac{1}{(\rho A)_0} \left\{ (t^{zz})_1 (x_D - x_A) + (t^{zz})_2 (x_A - x_B) \right. \\ \left. + (t^{zz})_3 (x_B - x_C) + (t^{zz})_4 (x_C - x_D) \right\}$$

TR66-85

and

$$\left( \frac{t^{zz}}{\rho x} \right)_0 = \frac{1}{4} \left\{ \frac{(t^{xz})_1}{\rho_1 \bar{x}_{I1}} + \frac{(t^{xz})_2}{\rho_2 \bar{x}_{I2}} + \frac{(t^{xz})_3}{\rho_3 \bar{x}_{I3}} + \frac{(t^{xz})_4}{\rho_4 \bar{x}_{I4}} \right\}$$

Substituting these finite difference quantities in the momentum equation enables us to evaluate the accelerations  $a^x$  and  $a^z$ :

$$\begin{aligned} (a^x)_0^n &= - \frac{(t^{xx})_1^n (z_D^n - z_A^n) + (t^{xx})_2^n (z_A^n - z_B^n) + (t^{xx})_3^n (z_B^n - z_C^n) + (t^{xx})_4^n (z_C^n - z_D^n)}{A_{I1}^n \rho_1^n + A_{I2}^n \rho_2^n + A_{I3}^n \rho_3^n + A_{I4}^n \rho_4^n} \\ &+ \frac{(t^{xz})_1^n (x_D^n - x_A^n) + (t^{xz})_2^n (x_A^n - x_B^n) + (t^{xz})_3^n (x_B^n - x_C^n) + (t^{xz})_4^n (x_C^n - x_D^n)}{A_{I1}^n \rho_1^n + A_{I2}^n \rho_2^n + A_{I3}^n \rho_3^n + A_{I4}^n \rho_4^n} \\ &+ \frac{(\alpha-1)}{4} \left\{ \frac{[(t^{xx})_1^n - (t^{yy})_1^n]}{\rho_1^n \bar{x}_{I1}^n} + \frac{[(t^{xx})_2^n - (t^{yy})_2^n]}{\rho_2^n \bar{x}_{I2}^n} + \frac{[(t^{xx})_3^n - (t^{yy})_3^n]}{\rho_3^n \bar{x}_{I3}^n} \right. \\ &\left. + \frac{[(t^{xx})_4^n - (t^{yy})_4^n]}{\rho_4^n \bar{x}_{I4}^n} \right\} \\ (a^z)_0^n &= - \frac{(t^{zz})_1^n (z_D^n - z_A^n) + (t^{zz})_2^n (z_A^n - z_B^n) + (t^{zz})_3^n (z_B^n - z_C^n) + (t^{zz})_4^n (z_C^n - z_D^n)}{A_{I1}^n \rho_1^n + A_{I2}^n \rho_2^n + A_{I3}^n \rho_3^n + A_{I4}^n \rho_4^n} \\ &+ \frac{(t^{zz})_1^n (x_D^n - x_A^n) + (t^{zz})_2^n (x_A^n - x_B^n) + (t^{zz})_3^n (x_B^n - x_C^n) + (t^{zz})_4^n (x_C^n - x_D^n)}{A_{I1}^n \rho_1^n + A_{I2}^n \rho_2^n + A_{I3}^n \rho_3^n + A_{I4}^n \rho_4^n} \\ &+ \frac{(\alpha-1)}{4} \left\{ \frac{(t^{xz})_1^n}{\rho_1^n \bar{x}_{I1}^n} + \frac{(t^{xz})_2^n}{\rho_2^n \bar{x}_{I2}^n} + \frac{(t^{xz})_3^n}{\rho_3^n \bar{x}_{I3}^n} + \frac{(t^{xz})_4^n}{\rho_4^n \bar{x}_{I4}^n} \right\} \end{aligned}$$