ANL-7498

ANL-7498

Argonne National Laboratory

TWO-DIMENSIONAL HYDRODYNAMICS ANALYSIS FOR PRIMARY CONTAINMENT

by

Yao-Wen Chang, Joseph Gvildys,
and Stanley H. Fistedis

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Atomic Energy Commission, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona
Carnegie-Mellon University
Case Western Reserve University
The University of Chicago
University of Cincinnati
Illinois Institute of Technology
University of Illinois
Indiana University
Iowa State University
The University of Iowa
Kansas State University
The University of Kansas
Loyola University
Marquette University
Michigan State University
The University of Michigan
University of Minnesota
University of Missouri
Northwestern University
University of Notre Dame
The Ohio State University
Ohio University
The Pennsylvania State University
Purdue University
Saint Louis University
Southern Illinois University
University of Texas
Washington University
Wayne State University
The University of Wisconsin

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U. S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.65

ANL-7498
Reactor Technology

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

TWO-DIMENSIONAL HYDRODYNAMICS ANALYSIS FOR PRIMARY CONTAINMENT

by

Yao-Wen Chang, Joseph Gvildys,
and Stanley H. Fistedis

Reactor Engineering Division

November 1969

TABLE OF CONTENTS

	Page
NOMENCLATURE	6
ABSTRACT	7
I. INTRODUCTION	7
II. ANALYTICAL DEVELOPMENT	9
A. Hydrodynamic Equations of Shock Waves	9
1. Basic Equations	9
2. Elimination of Shock Discontinuities	12
3. Form of Pseudoviscosity Term q	13
a. One-dimensional Flow	13
b. Two-dimensional Flow	14
B. Equation of State of the Media	14
C. Stability of the Differential Equations	16
D. Finite-difference Equations	18
III. COMPUTER PROGRAM	26
A. Input Information Required	26
B. Computer Output	34
1. Standard Program Printout	34
a. Full-accuracy Output	35
b. Limited 2-D Output	36
2. Pictorial Display	37
a. Grid Displacement	37
b. Pressure	37
3. Calcomp Display	38
C. Program Limitations and Subroutines	38
1. Limitations	38
2. Subroutines	39
IV. SAMPLE PROBLEM	39
A. Reactor Configuration	39
B. Excursion Model	39
C. Equations of State	39

TABLE OF CONTENTS

Page

D. Boundary Conditions 41

E. Initial Conditions. 41

F. Results and Discussion. 42

APPENDIXES

A. Momentum Equations for Solid Materials Having Tensile Strength. 47

1. Cylindrical Shell 47
2. Bottom Plate 48

B. Equation of State for a Group of Mixed Materials 52

1. Mixtures Containing Liquids and Solids. 52
2. Mixtures Containing Gases, Liquids, and Solids 53

C. FORTRAN Program Listing 55

REFERENCES. 93

LIST OF FIGURES

No.	Title	Page
1.	Experimental Hugoniot of Iron in the Region of the Phase Transformation	16
2.	A Typical Mesh Point, (I,J), with the Adjacent Zones and Points, Illustrating the Notation Used in Finite-difference Equations	18
3.	Dimensions and Lagrangian Meshes of the Reactor Configuration Selected as Sample Problem	40
4.	Deformation of Lagrangian Grids from 0 to 442 μsec after Start of a Power Excursion in a "Pancake" Core Configuration	43
5.	Deformation of Lagrangian Grids from 582 to 810.75 μsec after Start of a Power Excursion in a "Pancake" Core Configuration	44
6.	Pressure Profiles along Core Vertical Centerline at Various Times	45
7.	Pressure Profiles along Core Horizontal Axis at Various Times	45
8.	Pressure-Time Curve at Mesh Zone (2.49)	46
9.	Total Force Acting on the Rotating Plug as a Function of Time	46

NOMENCLATURE

Symbol	Description
A	Area
a	Shock-width constant
b	Radius of the circular plate
c	Speed of sound
c_0	Speed of sound at standard conditions
DI	Distortion index
E	Specific internal energy
E_H	Specific internal energy along the Hugoniot
E_s	Modulus of elasticity
h_1, h_2	Thicknesses
IE	Total internal energy
I, J	Lagrangian grid lines
K	Variable $= \left(\frac{2}{\gamma} + 1\right)$
J	Jacobian of the transformation
KE	Total kinetic energy
ℓ	Constant, having dimensions of length
ℓ_1, ℓ_2	Lengths
m	Mass
P	Sum of pressure p and pseudoviscosity q
p	Pressure
p_B	Material constant
p_H	Pressure along the Hugoniot
q	Pseudoviscosity
R, Z	Lagrangian coordinates
r, z	Eulerian coordinates
R_r, R_z	Remainder terms of the pressure gradient
$\bar{r}_{I,J}$	Radius of the centroid of the area $A_{I,J}$
t	Time
$\bar{u}$	Velocity vector
u, w	Radial and axial velocities
u_s	Shock-wave velocity
u_p	Particle velocity
$\bar{u}, \bar{w}$	Radial and axial displacements
V	Specific Volume
$\mathcal{V}, \mathcal{v}$	Volume
W	White stability number
$\bar{w}_0$	Axial displacement at center of plate
γ	Mie-Grüneisen coefficient
Δt	Time interval
ΔV	Change of specific volume
δ_{r57}, δ_{z57}, ...	Measures of the asymmetry of the mesh
ϵ_{rr}, ϵ_{zz}, $\epsilon_{\theta\theta}$	Strains
$\dot{\epsilon}_{rr}$, $\dot{\epsilon}_{zz}$, $\dot{\epsilon}_{\theta\theta}$	Strain rates
η	Isentropic exponent
ν	Poisson's ratio
ρ	Density
σ_{rr}, $\sigma_{\theta\theta}$	Stresses

TWO-DIMENSIONAL HYDRODYNAMICS ANALYSIS FOR PRIMARY CONTAINMENT

by

Yao-Wen Chang, Joseph Gvildys,
and Stanley H. Fistedis

ABSTRACT

This report describes a numerical method for calculating the two-dimensional hydrodynamic response of a primary reactor containment system to a high-energy excursion. Equations of hydrodynamics and equations of state of reactor materials are expressed in Lagrangian form and then set into finite-difference equations. Shock discontinuities are eliminated by the use of the von Neumann-Richtmyer pseudoviscosity, q. These equations, along with pressure pulse and other pertinent input data, are programmed for solution on the IBM-360 computer. Propagation of shock waves, loads imposed on different parts of the reactor components, and the resulting damage are determined at every time step until the steel vessel ruptures or the force acting on the rotating shield plug exceeds the strength of the plug holddown device. Calculated displacements and pressures at all spatial points at any instant of time also are given in pictorial form.

I. INTRODUCTION

Large, liquid-metal-cooled, fast reactors probably cannot be designed to completely rule out the possibility of a core meltdown. Therefore, the reactor designer must rigorously analyze the capability of the primary containment system to sustain the consequences of such an accident. He should then make appropriate adjustments, using to best advantage the strength of the system structures and components. The objective in this report is to effect a sequence of component failures that will dissipate the energy of the ensuing excursion without breaching overall system integrity and releasing fission products and other harmful radioactive materials to the surrounding environs.

To accomplish this objective, the designer must know in detail the propagation of shock waves emanating from the reactor core, the loads imposed on adjacent structures and components, and the damage produced by

these loads. Most important, he must know the sequence of failures of the various components, since early failure of one component may completely change the loadings on other parts of the system.

For example, if the reactor vessel fails under the shock loading, the pressure in the core oxide vapor will be reduced substantially by venting of the coolant. This, in turn, will reduce the duration of pressure loading on the rotating shield plug in the vessel cover, and the degree to which the plug is dislodged. On the other hand, if the vessel is strong enough to survive the shock loading, the kinetic energy of the coolant will be redirected upwards, lifting the coolant and compressing the inert-gas blanket. In this case, the pressure loading on the rotating plug will be higher and of longer duration, with a greater degree of dislodgement.

This report describes a numerical method for calculating the two-dimensional hydrodynamic response of primary reactor containment to shock waves emanating from the core after the neutronics have ceased. The one-dimensional case has been investigated by Sorensen and Fistedis.[1] Briefly, the analysis uses conservation equations (mass, momentum, and energy) and equations of state of reactor materials expressed in finite-difference form. These equations, along with the pressure pulse and other pertinent input data, are programmed for solution on the IBM-360 computer, using the REXCO-H code developed at Argonne.*

For each problem, the code computes the displacements, velocities, pressures, specific internal energies, densities, and strains at every spatial point for a specific time interval. (The output of displacements and pressures also can be given pictorially, showing the movements and deformations or magnitudes of pressures at all spatial points at any instant of time.) These computations are repeated for every time interval until the reactor vessel ruptures or the force acting on the rotating shield plug exceeds the strength of the plug holddown mechanism, whereupon the computation ceases. Thus this containment analysis and code give in detail the propagation of shock waves, the loads imposed on different parts of the reactor, and the consequent damage throughout the sequence of the problem.

*A power-excursion computer code, also developed at Argonne, provides initial values for the REXCO-H code. Both codes will be linked in sequence, constituting two modules of the Postburst Phenomena of the Fast Reactor Safety Program.

II. ANALYTICAL DEVELOPMENT

A. Hydrodynamic Equations of Shock Waves

1. Basic Equations

The partial differential equations that govern the flow of a nonviscous, non-heat-conducting, compressible fluid are:

$$\frac{D\rho}{Dt} + \rho \operatorname{div} \bar{u} = 0 \text{ (mass)},$$

$$\rho \frac{D\bar{u}}{Dt} = -\operatorname{grad} p \text{ (momentum)},$$

and

$$\frac{DE}{Dt} + p\frac{DV}{Dt} = 0 \text{ (energy)},$$

where

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \bar{u} \cdot \bar{\nabla},$$

$\bar{u}$ = velocity vector,

ρ = density,

p = pressure,

E = specific internal energy (energy per unit mass),

and

V = specific volume.

We assume that there are no external energy sources in the fluid. If the flow is axially symmetric, it is often advantageous to express these equations in the cylindrical coordinate system. Thus when the cylindrical coordinates are denoted by r and z, the equations take the form

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial r} + w\frac{\partial \rho}{\partial z} = -\rho\left(\frac{\partial u}{\partial r} + \frac{\partial w}{\partial z} + \frac{u}{r}\right), \tag{1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial r} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial r}, \tag{2}$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial r} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z}, \tag{3}$$

and

$$\frac{\partial E}{\partial t} + u\frac{\partial E}{\partial r} + w\frac{\partial E}{\partial z} = -p\left(\frac{\partial V}{\partial t} + u\frac{\partial V}{\partial r} + w\frac{\partial V}{\partial z}\right), \tag{4}$$

where u and w are the radial and axial velocities, respectively. Equations 1-4 are the Eulerian formulation of the fluid dynamics in which the independent space variables are referred to a coordinate system fixed in space. The alternative is to express the equations in material coordinates, and these equations are known as the Lagrangian equations of the fluid dynamics.

The advantage of using the Eulerian formulation is that the fluid can be distorted without limit. However, difficulty is encountered if there are nonrigid boundaries or interfaces between fluids of differing thermodynamic properties, because there is no simple way of telling what kind of fluid is to be found at a given instant and at a given point. This difficulty can be avoided in the Lagrangian approach by placing the coordinate lines along the nonrigid boundaries and interfaces, because the material coordinate is imbedded in the fluid and is undergoing all the motion and distortion of the fluid. Also, the limitation imposed on the distortion of Lagrangian mesh can be eliminated by using the rezoning process. Consequently, the present analysis uses the Lagrangian formulation of the fluid dynamics.

If the Lagrangian coordinates are denoted by R and Z, then for any function F, the Eulerian derivatives become

$$\frac{\partial F}{\partial r} = \frac{1}{J}\left(\frac{\partial F}{\partial R}\frac{\partial z}{\partial Z} - \frac{\partial F}{\partial Z}\frac{\partial z}{\partial R}\right)\frac{r}{R}$$

and

$$\frac{\partial F}{\partial z} = \frac{-1}{J}\left(\frac{\partial F}{\partial R}\frac{\partial r}{\partial Z} - \frac{\partial F}{\partial Z}\frac{\partial r}{\partial R}\right)\frac{r}{R},$$

where J is the Jacobian of the transformation

$$J = \begin{vmatrix} \frac{\partial r}{\partial R} & 0 & \frac{\partial r}{\partial Z} \\ 0 & \frac{r}{R} & 0 \\ \frac{\partial z}{\partial R} & 0 & \frac{\partial z}{\partial Z} \end{vmatrix} = \left(\frac{\partial r}{\partial R}\frac{\partial z}{\partial Z} - \frac{\partial r}{\partial Z}\frac{\partial z}{\partial R}\right)\frac{r}{R}. \tag{5}$$

Then on applying the transformation, the equation of mass conservation is simply

$$\frac{\rho_0}{\rho} = J = \frac{d\forall}{d\forall}, \tag{6}$$

where ρ_0 is the initial density at time $t = 0$, and $d\forall$ and $d\forall$ are the volume elements in the deformed and undeformed states, respectively. The momentum equations become

$$\ddot{r} = -\frac{1}{\rho_0}\left(\frac{\partial p}{\partial R}\frac{\partial z}{\partial Z} - \frac{\partial p}{\partial Z}\frac{\partial z}{\partial R}\right)\frac{r}{R}$$

and

$$\ddot{z} = \frac{1}{\rho_0}\left(\frac{\partial p}{\partial R}\frac{\partial r}{\partial Z} - \frac{\partial p}{\partial Z}\frac{\partial r}{\partial R}\right)\frac{r}{R}.$$

Although ρ_0, ∂R, and ∂Z are independent of time, the momentum equations have become considerably more complicated by the transformation. In the numerical analysis, it was easier to work with the quantities ρ, ∂r, and ∂z directly. Therefore the equations used are of the form

$$\ddot{r} = -\frac{1}{\rho}\frac{\partial p}{\partial r} \tag{7}$$

and

$$\ddot{z} = -\frac{1}{\rho}\frac{\partial p}{\partial z}, \tag{8}$$

where r and z are dependent variables; they are functions of (R, Z, t). Each fluid particle is now labeled with a set of Lagrangian coordinates $[R(I), Z(J)]$. Since the Lagrangian coordinates are imbedded in the fluid particle and move with the fluid, the total derivative is simply the time derivative, and the energy equation reduces to

$$dE = -pdV. \tag{9}$$

Equations 6-9 are the hydrodynamic equations in the Lagrangian formulation. There are five unknowns: density of the fluid, ρ; radial and axial accelerations of the fluid, $\ddot{r}$ and $\ddot{z}$; pressure in the fluid, p; and internal energy, E. To obtain a unique solution, a fifth equation relating the various unknowns is needed. If the thermodynamic properties of the fluid are known, a relationship between p, E, and ρ can be established. Such a relationship is known as the equation of state of the fluids. In the present analysis, this equation is assumed to have the form

$$p = f(V, E) \tag{10}$$

Equations 6-10 apply only to the smooth part of the flow, i.e., without shocks, or the flow between shocks. For flows at the shocks, the dependent variables ρ, u, w, E, and p are no longer continuous: The differential equations at the shock only have one-sided derivatives. Therefore, they must be supplemented by jump conditions that serve as internal boundary conditions. These special boundary conditions are provided by the well-known Rankine-Hugoniot equations. However the application of these equations is complicated because the surfaces on which the conditions are to be applied are in motion through the fluid and because the motion of the surfaces is determined by the equations themselves.

2. Elimination of Shock Discontinuities

To avoid the above-mentioned complication, von Neumann and Richtmyer[2] devised an approximate method for solving one-dimensional fluid-dynamics problems that eliminates discontinuities in the differential equations and dampens out spurious pressure oscillations in the numerical computations. Their method uses the well-known effect of dissipative mechanisms, such as viscosity and heat of conduction, on shocks. For example, when viscosity is taken into account, the shocks are smeared out and the mathematical surfaces of the discontinuities are replaced by thin transition layers in which quantities such as pressure, temperature, and density vary rapidly but continuously.

Accordingly, von Neumann and Richtmyer introduced a pseudoviscosity term, q, into the differential equations, thereby eliminating the shock discontinuities without jeopardizing the conservation laws on which the Hugoniot conditions are based. Also, the jump conditions across the transition layer still hold in the approximation in which the layer is regarded as thin, relative to other dimensions.

The differential equations employed in the pseudoviscosity method are

$$\left.\begin{aligned}
\frac{\rho_0}{\rho} &= J = \frac{d\mathcal{V}}{dV}, \\
\ddot{r} &= -\frac{1}{\rho}\left[\frac{\partial(p+q)}{\partial r}\right], \\
\ddot{z} &= -\frac{1}{\rho}\left[\frac{\partial(p+q)}{\partial z}\right], \\
dE &= -(p+q)\,dV, \\
&\text{and} \\
p &= f(V, E).
\end{aligned}\right\} \tag{11}$$

3. Form of Pseudoviscosity Term q

a. One-dimensional Flow. For one-dimensional fluid motion, the form of q proposed by von Neumann and Richtmyer is

$$\left.\begin{aligned} q &= \frac{(\rho_0 \ell)^2}{V}\left(\frac{\partial V}{\partial t}\right)^2 \quad \text{if } \frac{\partial V}{\partial t} < 0, \\ &= 0 \quad \text{if } \frac{\partial V}{\partial t} \geq 0, \end{aligned}\right\} \tag{12}$$

where ℓ is a constant having the dimensions of length, ρ_0 is the normal density of the fluid, and V is the specific volume of the fluid. Equation 12 can be written equivalently as

$$\left.\begin{aligned} q &= \frac{\ell^2}{V}\left(\frac{\partial u}{\partial R}\right)^2 \quad \text{if } \frac{\partial u}{\partial R} < 0, \\ &= 0 \quad \text{if } \frac{\partial u}{\partial R} \geq 0, \end{aligned}\right\} \tag{13}$$

where R is the Lagrangian coordinate, and u $(= \partial r/\partial t)$ and r are, respectively, the velocity and Eulerian coordinates at time t of a fluid element that was initially at position R. For some calculations, it is more convenient to use Eq. 13.

Von Neumann and Richtmyer showed that upon inclusion of q into the pressure p in the momentum and energy equations:

(1) The derivatives are no longer discontinuous.

(2) The shocks are immediately evident as near-discontinuities that move through the fluid with very nearly the correct speed.

(3) The quantities pressure, temperature, etc., have very nearly the correct jumps across the discontinuities, provided the thickness of the shock layers is small in comparison with other physically relevant dimensions of the system.

(4) The thickness of the shock transition layer is independent of the shock strength and of the condition of the material upon which it impinges.

(5) The effect of q is negligibly small in the smooth part of the flow between shocks.

b. Two-dimensional Flow. A number of different expressions for pseudoviscosity in two-dimensional flow have appeared in the literature,[3-5] but none offer any advantage over the von Neumann-Richtmyer's q on the pressure oscillations* that occur behind the shock wave. This is because q is not a physically real quantity, but, as mentioned above, merely a device to eliminate shock discontinuities in the equations and to dampen spurious pressure oscillations in the numerical calculations. Furthermore, since it is added into the pressure in the calculations, we can reasonably expect q to be a scalar function, like true pressure and energy.

For these reasons, Eq. 12 is modified and used directly in two-dimensional flows. (For two-dimensional flows, Eq. 13 is no longer equivalent to Eq. 12.) The modification consists of replacing ℓ^2 by a^2A_0R/r, where a is the shock-width constant and A_0 is the area of the original Lagrangian mesh, i.e., when the fluid is at the normal density ρ_0. The objective is to ensure that all shock widths cover the same number of mesh points (~3 to 5 times the mesh size).

When the equation of mass conservation is applied, Eq. 12 becomes

$$\left.\begin{aligned} q &= \frac{a^2\rho_0 A}{V^2}\left(\frac{\partial V}{\partial t}\right)^2 \quad \text{if } \frac{\partial V}{\partial t} < 0, \\ &= 0 \qquad\qquad\qquad\quad \text{if } \frac{\partial V}{\partial t} \geq 0, \end{aligned}\right\} \tag{14}$$

where A is the area of the Lagrangian mesh at time t when the fluid density changes to ρ. The value of a that yields the desired shock width ranges from 1.2 to 1.4. In this report, a is assumed to be 1.2.

B. Equation of State of the Media

As stated in Section A.1, solution of the four hydrodynamic equations requires an equation of state of the form given by Eq. 10. For gaseous materials, the pressure is of thermal origin; it is related to the transfer of momentum by particles participating in the thermal motion. Therefore, no appreciable difficulties are encountered in calculating the thermodynamic properties of gases.

In contrast, a theoretical description of similar properties of solids and liquids at high pressures generated by strong shocks presents a complex problem. Here the atoms or molecules are in proximity and interact strongly with each other. Thus strong compression of a condensed medium

*These pressure oscillations are not instabilities. They are analogous to the thermal agitation of the molecules, and they represent the internal energy that must appear in the shocked fluid according to the Hugoniot equations.

generates not only a thermal pressure, but also a significant internal pressure. The internal pressure is due to repulsive forces between the atoms; it differs from the thermal pressure associated with thermal motion of the atoms. Therefore, experimental methods play a major role in the study of condensed media in a compressed state. Accordingly, the following paragraphs describe briefly how the equation of state of the form $p = f(V, E)$ can be obtained from shock experiments.

A shock wave (one-dimensional) with velocity u_s proceeding through a condensed material changes the pressure, specific volume, and specific internal energy, respectively, from p_0, V_0, and E_0 to p_1, V_1, and E_1 and gives the material particles a velocity u_p in the direction of the shock. At the pressures of interest, the shear strength of the material is negligible and the material can be treated as a compressible fluid.

For example, consider a shock front having a time-independent pressure profile. Application of conservation of mass, momentum and energy laws across the shock front yields the Rankine-Hugoniot equations

$$\rho_0 u_s = \rho_1 (u_s - u_p), \tag{15}$$

$$\rho_0 u_s u_p = p_1 - p_0, \tag{16}$$

and

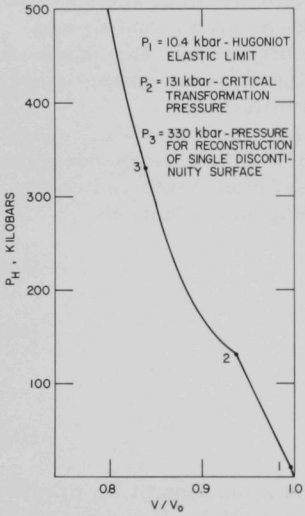
$$E_1 - E_0 = \tfrac{1}{2}(p_1 + p_0)(V_0 - V_1). \tag{17}$$

Since the specific internal energy of a material is a function of its pressure and volume, Eq. 17 may be regarded as the locus of all p_1, V_1 states attainable by propagating a shock wave into a fixed initial state p_0, V_0. This locus is defined as the Hugoniot curve (or p_H values) centered at p_0, V_0. There are "infinity squared" of curves for all possible values of p_0, V_0.

Equations 15 and 16 contain only four variables: p_1, ρ_1 $(= 1/V_1)$, u_s, and u_p. Therefore, if the values of u_s and u_p are measured experimentally, both equations may be used to find the pressure and the specific volume. These, in turn, can be substituted in Eq. 17 to calculate the internal energy. With advanced laboratory techniques, u_s and u_p values can be accurately measured for a wide range of pressures: from relatively weak shock pressures, up to 4×10^6 atm. Therefore, by varying the strength of the shock waves, we obtain a locus of all p_1, V_1 states. Thus the Hugoniot curve centered at p_0, V_0 can be obtained experimentally.

It is intended that p_H values used in the present analysis be based as much as possible on the experimental Hugoniot. If the experimental Hugoniot exhibits multiple steps similar to those attributed to an elastic

precursor or to phase changes, these steps will be included in the numerical representation of p_H curves or tables. For example, the experimental Hugoniot of iron is shown in Fig. 1; it is based on data from Ref. 6. Here, the Hugoniot curve is centered at p_0, V_0. Thus the p_H values are valid only for materials having the same initial condition as the experimental Hugoniot. During subsequent shock compressions, the material under consideration can no longer possess the same p_0, V_0 state. Therefore, it is necessary to find a general equation of state that will cover points along all possible Hugoniot curves with different initial conditions.



113-1940

Fig. 1. Experimental Hugoniot of Iron in the Region of the Phase Transformation

The most simple formulation that can be used for this purpose is the Mie-Grüneisen equation of state, which may be written

$$p = p_H + \frac{\gamma}{V}(E - E_H), \tag{18}$$

where p_H and E_H are the pressure and specific internal energy along the Hugoniot centered at p_0, V_0; and γ, the Grüneisen coefficient, is the ratio of the thermal component of pressure to the lattice vibrational energy density; it is a function only of volume. The value of E_H is obtained from Eq. 17. In Eq. 18, the neighboring states are determined from the Hugoniot centered at p_0, V_0 by adding an additional term $(\gamma/V)(E - E_H)$,* which acts like a thermal pressure rise. It is believed that Eq. 18 also can be used to compute pressures during the adiabatic expansion.

C. Stability of the Differential Equations

Another quantity of particular interest to the numerical solution of hyperbolic differential equations is the criterion for stability of a finite-difference mesh. A stability criterion derived by White[7] for the von Neumann-Richtmyer's pseudoviscosity method is of the form

$$W = \left[\frac{c^2(\Delta t)^2}{A} + 4a^2 \left|\frac{\Delta V}{V}\right|\right]^{\frac{1}{2}} < 1,$$

where W is White stability number, c is the speed of sound, and ΔV is the change of specific volume. To ensure additional computational stability, the value of W is limited to less than 0.45 and greater than 0.2245.

---

*During an initial compression ($E = E_H$), the Mie-Grüneisen equation of state reduces to the Hugoniot.

The White stability number for each zone is calculated by

$$\left(\frac{W}{1.2}\right)^2 = \frac{c^2}{A}\left(\frac{\Delta t}{1.2}\right)^2 + 4\left|\frac{\Delta V}{V}\right|, \tag{19}$$

where a is assumed to be 1.2, and the time interval Δt is chosen so that the maximum of the stability numbers for all zones satisfies

$$0.035 < \left(\frac{W}{1.2}\right)^2_{max} < 0.14. \tag{20}$$

The speed of sound is calculated from

$$c^2 = -V^2\left(\frac{\partial p}{\partial V}\right)_s. \tag{21}$$

For gaseous materials, the adiabatic equation of state has the form

$$pV^{\eta} = \text{constant}, \tag{22}$$

where η is the isentropic exponent. Therefore

$$c = (\eta pV)^{\frac{1}{2}}. \tag{23}$$

For condensed materials (solids and liquids), the adiabatic equation of state is assumed to have the form

$$p = p_B\left[\left(\frac{V_0}{V}\right)^{\eta} - 1\right]. \tag{24}$$

On applying Eq. 21, we obtain

$$c = [\eta V(p + p_B)]^{\frac{1}{2}}. \tag{25}$$

Here, the constants p_B and η are related by

$$c_0^2 = V_0 p_B \eta, \tag{26}$$

where c_0 is the speed of sound at standard conditions.

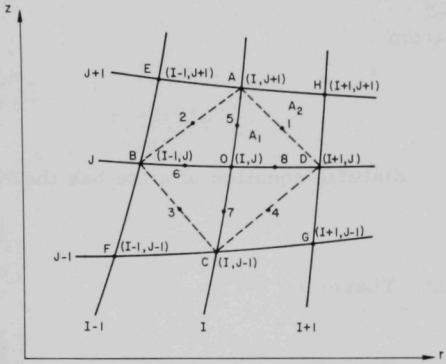
Substitution of Eqs. 23 and 25 into Eq. 19 yields a stability criterion of the form

$$\left(\frac{W}{1.2}\right)^2 = \frac{\eta V(p + p_B)}{A}\left(\frac{\Delta t}{1.2}\right)^2 + 4\left|\frac{\Delta V}{V}\right| \tag{27}$$

for both gaseous and condensed materials. (Note: For gaseous materials, p_B is taken to be zero.)

D. Finite-difference Equations

Consider a physical system of cylindrical symmetry, which is represented in a cylindrical (r, z) coordinate system by quadrilateral Lagrangian meshes as shown in Fig. 2. Because of symmetry, all quantities are dependent only on r and z. Since the Lagrangian coordinate system has been adopted, the mesh is imbedded in and moves with the fluid.



113-1939

Fig. 2

A Typical Mesh Point, (I, J), with the Adjacent Zones and Points, Illustrating the Notation Used in Finite-difference Equations

With reference to Fig. 2, the mesh points (the intersections of Lagrangian grid lines) are identified by a pair of integers (I, J), which are [R(I), Z(J)]. The mesh zones are identified by the integers on the lower left mesh point (in the undeformed Lagrangian meshes); for example, the mesh zone ODHA is identified as zone (I, J). Positions, velocities, and accelerations are assumed to be associated with the mesh points; they are identified by the mesh-point integers as subscripts. Specific volumes, densities, pressures, pseudoviscosities, strains, internal energies, and White stability numbers are assumed to be associated with zones; they are identified by the zone numbers as subscripts. Time is denoted by t. The times t, $t + \Delta t$, and $t + \frac{\Delta t}{2}$ at which quantities are assumed to be known or calculated are identified by superscripts n, n + 1, and $n + \frac{1}{2}$. The latter is also used to identify the change in a quantity between t^n and t^{n+1}. Where appearing, the superscript 0 denotes the initial value of a quantity at the start of the problem.

All difference equations used in this analysis are designed to be correct to first approximations only. For example, the lines connecting the mesh points in Fig. 2 are taken as straight lines. Use of the true curves given by the Jacobian of the transformation, assuming they were known, would result in second-order changes in the volume calculated.

Since the Lagrangian meshes are used, the mass equation is automatically satisfied. However, this equation is needed for the computation of the new density ρ, which occurs in the momentum equations. This computation is performed as follows:

The area of the deformed mesh ODHA is the sum of two triangles AOD and AHD:

$$A_{I,J} = A_1 + A_2,$$

where

$$\begin{aligned} A_1 &= \tfrac{1}{2}|\overline{OD} \times \overline{OA}| \\ &= \tfrac{1}{2}[(r_{I+1,J} - r_{I,J})(z_{I,J+1} - z_{I,J}) \\ &\quad - (r_{I,J+1} - r_{I,J})(z_{I+1,J} - z_{I,J})], \end{aligned} \tag{28}$$

and

$$\begin{aligned} A_2 &= \tfrac{1}{2}|\overline{HA} \times \overline{HD}| \\ &= \tfrac{1}{2}[(r_{I,J+1} - r_{I+1,J+1})(z_{I+1,J} - z_{I+1,J+1}) \\ &\quad - (r_{I+1,J} - r_{I+1,J+1})(z_{I,J+1} - z_{I+1,J+1})]. \end{aligned} \tag{29}$$

Substitution of Eqs. 28 and 29 for A_1 and A_2 gives

$$\begin{aligned} A_{I,J} &= \tfrac{1}{2}[(z_{I+1,J+1} - z_{I,J})(r_{I+1,J} - r_{I,J+1}) \\ &\quad - (r_{I+1,J+1} - r_{I,J})(z_{I+1,J} - z_{I,J+1})]. \end{aligned} \tag{30}$$

The volume $\forall$ of a quadrilateral is calculated from

$$\forall_{I,J} = A_{I,J} 2\pi \bar{r}_{I,J}, \tag{31}$$

where $\bar{r}_{I,J}$ is the radius of the centroid of the area $A_{I,J}$. A reasonable approximation for moderate distortions is

$$\bar{r}_{I,J} = \tfrac{1}{4}(r_{I,J} + r_{I,J+1} + r_{I+1,J} + r_{I+1,J+1}). \tag{32}$$

In Lagrangian coordinates, the mass of each volume is constant with time; therefore the density of a quadrilateral at time n is obtained by

$$\rho^n_{I,J} = \frac{(\rho_0 \forall^0)_{I,J}}{\forall^n_{I,J}}, \tag{33}$$

Next, the momentum equations are set into finite-difference form by the Midpoint Method.[8] Subsequent application of Taylor's expansion between points 0 and 5, 0 and 6, 0 and 7, and 0 and 8, yields four equations of the form

$$P_5 = P_0 + (z_5 - z_0)\frac{\partial P}{\partial z} + (r_5 - r_0)\frac{\partial P}{\partial r}$$

$$+ \tfrac{1}{2}(z_5 - z_0)^2 \frac{\partial^2 P}{\partial z^2} + (z_5 - z_0)(r_5 - r_0)\frac{\partial^2 P}{\partial z \partial r}$$

$$+ \tfrac{1}{2}(r_5 - r_0)^2 \frac{\partial^2 P}{\partial r^2} + \ldots,$$

where $P = p + q$, and terms of third and higher order in $(z_5 - z_0)$ and $(r_5 - r_0)$ have been omitted. The system is overdetermined for $\partial P/\partial z$ and $\partial P/\partial r$. One method of removing the overdeterminacy is to solve first for $(P_5 - P_7)$ and $(P_6 - P_8)$, which yields

$$P_5 - P_7 = (z_5 - z_7)\frac{\partial P}{\partial z} + (r_5 - r_7)\frac{\partial P}{\partial r}$$

$$+ \frac{\partial^2 P}{\partial z^2}(z_5 - z_7)\,\delta_{z57} + \frac{\partial^2 P}{\partial r^2}(r_5 - r_7)\,\delta_{r57}$$

$$+ \frac{\partial^2 P}{\partial z \partial r}\left[(z_5 - z_7)\,\delta_{r57} + (r_5 - r_7)\,\delta_{z57}\right] + \ldots$$

and

$$P_6 - P_8 = (z_6 - z_8)\frac{\partial P}{\partial z} + (r_6 - r_8)\frac{\partial P}{\partial r}$$

$$+ \frac{\partial^2 P}{\partial z^2}(z_6 - z_8)\,\delta_{z68} + \frac{\partial^2 P}{\partial r^2}(r_6 - r_8)\,\delta_{r68}$$

$$+ \frac{\partial^2 P}{\partial z \partial r}\left[(z_6 - z_8)\,\delta_{r68} + (r_6 - r_8)\,\delta_{z68}\right] + \ldots,$$

where

$$\delta_{z57} = \tfrac{1}{2}(z_5 + z_7) - z_0, \text{ etc.},$$

are measures of the asymmetry of the mesh. Second- and higher-order terms in the mesh size--i.e., $(z_5 - z_7)$, $(r_5 - r_7)$, etc.--are omitted.

The above equations may now be solved for the gradients

$$\left.\begin{aligned} \frac{\partial P}{\partial z} &= \frac{1}{2A_3}\left[(P_5 - P_7)(r_6 - r_8) - (P_6 - P_8)(r_5 - r_7)\right] + R_z \\ &\text{and} \\ \frac{\partial P}{\partial r} &= \frac{-1}{2A_3}\left[(P_5 - P_7)(z_6 - z_8) - (P_6 - P_8)(z_5 - z_7)\right] + R_r, \end{aligned}\right\} \tag{34}$$

where

$$A_3 = \tfrac{1}{2}\left[(z_5 - z_7)(r_6 - r_8) - (z_6 - z_8)(r_5 - r_7)\right]$$

represents the area of the quadrilateral 5678. The remaining terms, R_z and R_r, involve products of the mesh size and the δ's. If the mesh is nearly symmetric, these terms are negligible compared to those that have been retained.

On substituting Eq. 34 into the momentum equations, neglecting the remainder terms, and writing

$$P_5 = \tfrac{1}{2}(P_{I,J} + P_{I-1,J}), \quad P_6 = \tfrac{1}{2}(P_{I-1,J} + P_{I-1,J-1}),$$

$$P_7 = \tfrac{1}{2}(P_{I-1,J-1} + P_{I,J-1}), \quad P_8 = \tfrac{1}{2}(P_{I,J} + P_{I,J-1}),$$

$$r_5 = \tfrac{1}{2}(r_{I,J+1} + r_{I,J}), \quad r_6 = \tfrac{1}{2}(r_{I,J} + r_{I-1,J}),$$

$$r_7 = \tfrac{1}{2}(r_{I,J} + r_{I,J-1}), \quad r_8 = \tfrac{1}{2}(r_{I+1,J} + r_{I,J}),$$

$$z_5 = \tfrac{1}{2}(z_{I,J+1} + z_{I,J}), \quad z_6 = \tfrac{1}{2}(z_{I,J} + z_{I-1,J}),$$

$$z_7 = \tfrac{1}{2}(z_{I,J} + z_{I,J-1}), \text{ and } z_8 = \tfrac{1}{2}(z_{I,J} + z_{I+1,J}),$$

we obtain

$$\left.\begin{aligned} \ddot{r}_{I,J} = \frac{-1}{(A\rho)_{I,J}}\Big[&(P_{I,J} - P_{I-1,J-1})(z_{I,J+1} - z_{I,J-1} \\ &+ z_{I-1,J} - z_{I+1,J}) - (P_{I-1,J} - P_{I,J-1}) \\ &(z_{I,J+1} - z_{I,J-1} + z_{I+1,J} - z_{I-1,J})\Big] \\ &\text{and} \end{aligned}\right\} \tag{35}$$

(Contd.)

$$\ddot{z}_{I,J} = \frac{1}{(A\rho)_{I,J}} \Big[(P_{I,J} - P_{I-1,J-1})(r_{I,J+1} - r_{I,J-1} + r_{I-1,J} - r_{I+1,J}) - (P_{I-1,J} - P_{I,J-1}) (r_{I,J+1} - r_{I,J-1} + r_{I+1,J} - r_{I-1,J}) \Big], \qquad \text{(Contd.)} \quad (35)$$

where

$$(A\rho)_{I,J} = A_{I,J}\rho_{I,J} + A_{I-1,J}\rho_{I-1,J} + A_{I-1,J-1}\rho_{I-1,J-1} + A_{I,J-1}\rho_{I,J-1}.$$

No time index is specified in Eq. 35, the implication being that all variables are prescribed for the same time.

It is assumed that all calculations are performed up through cycle $n(t = t^n)$, so that for all points and zones there are now available the values of r^n, z^n, $\dot{r}^{n-\frac{1}{2}}$, $\dot{z}^{n-\frac{1}{2}}$, A^n, V^n, p^n, q^n, ϵ^n_{rr}, ϵ^n_{zz}, $\epsilon^n_{\theta\theta}$, and E^n.

Next, the accelerations at time n are calculated from Eq. 35. These accelerations may now be used to advance the velocities of all points to time $n + \frac{1}{2}$,

$$\dot{r}^{n+\frac{1}{2}} = \dot{r}^{n-\frac{1}{2}} + \ddot{r}^n \left[\tfrac{1}{2}(\Delta t^{n+\frac{1}{2}} + \Delta t^{n-\frac{1}{2}})\right]$$

and

$$\dot{z}^{n+\frac{1}{2}} = \dot{z}^{n-\frac{1}{2}} + \ddot{z}^n \left[\tfrac{1}{2}(\Delta t^{n+\frac{1}{2}} + \Delta t^{n-\frac{1}{2}})\right], \qquad (36)$$

where

$$\Delta t^{n+\frac{1}{2}} = t^{n+1} - t^n$$

and

$$\Delta t^{n-\frac{1}{2}} = t^n - t^{n-1}.$$

These velocities, in turn, are used to advance the coordinates of all points to time $n + 1$,

$$\left.\begin{aligned} r^{n+1} &= r^n + \dot{r}^{n+\frac{1}{2}} \Delta t^{n+\frac{1}{2}} \\ \text{and} \\ z^{n+1} &= z^n + \dot{z}^{n+\frac{1}{2}} \Delta t^{n+\frac{1}{2}}. \end{aligned}\right\} \tag{37}$$

Subsequent steps involve the calculation of V, p, E, etc., for the zones at time n + 1. The volume $\forall$ of a quadrilateral zone is calculated from Eq. 31; i.e.,

$$\forall_{I,J}^{n+1} = A_{I,J}^{n+1} 2\pi \bar{r}_{I,J}^{n+1},$$

where $A_{I,J}^{n+1}$ and $\bar{r}_{I,J}^{n+1}$ are calculated from Eqs. 30 and 32, respectively. The masses of the zones are calculated at the start of the problem by

$$m^0 = \rho_0 \forall^0.$$

Therefore, from Eq. 33, the density of a zone at time n + 1 becomes

$$\rho^{n+1} = \frac{m^0}{\forall^{n+1}}. \tag{38}$$

The von Neumann-Richtmyer pseudoviscosity, q, is calculated from

$$\left.\begin{aligned} q^{n+1} &= (1.2)^2 \rho_0 A^{n+1} \frac{1}{(V^{n+1})^2} \left(\frac{\Delta V^{n+\frac{1}{2}}}{\Delta t^{n+\frac{1}{2}}}\right)^2, && \text{if } \Delta V^{n+\frac{1}{2}} < 0, \\ &= 0 && \text{if } \Delta V^{n+\frac{1}{2}} \geq 0, \end{aligned}\right\} \tag{39}$$

where

$$\Delta V^{n+\frac{1}{2}} = V^{n+1} - V^n.$$

The pressure p and internal energy E of each zone are obtained from the equation of state,

$$p^{n+1} = f(V^{n+1}, E^{n+1}), \tag{40}$$

and from the equation of conservation of energy, $dE = -(p+q)\, dV$, in difference form,

$$E^{n+1} = E^n - \tfrac{1}{2}(P^n + p^{n+1} + q^{n+1})\, \Delta V^{n+\frac{1}{2}}. \tag{41}$$

If Eqs. 40 and 41 cannot be solved simultaneously for the two unknowns p^{n+1} and E^{n+1}, then iterative techniques are used.

The White stability number for each zone is calculated by

$$\left[\left(\frac{W}{1.2}\right)^2\right]^{n+1} = \frac{\eta V^{n+1}\left(P^{n+1} + p_B\right)}{A^{n+1}} \left(\frac{\Delta t^{n+\frac{1}{2}}}{1.2}\right)^2 + 4\left|\frac{\Delta V^{n+\frac{1}{2}}}{V^{n+1}}\right|, \tag{42}$$

and the time interval $\Delta t^{n+\frac{3}{2}}$ to be used for the next cycle is chosen so that the maximum of the stability numbers for all zones with

$$\Delta t^{n+\frac{3}{2}} \rightarrow \Delta t^{n+\frac{1}{2}}$$

satisfies Eq. 20.

Since the pressure and specific internal energy are determined from Eqs. 40 and 41, using $V\ (= 1/\rho)$ from Eq. 38, the calculation of strains is not needed. However, for materials such as steel, which have a limitation on tensile strain and which exhibit anisotropic stresses under shock compressions, the strain rates and the strains are calculated separately. The strain rates are given by

$$\left.\begin{aligned}
(\dot{\epsilon}_{rr})^{n+\frac{1}{2}}_{I,J} &= \frac{1}{2A^{n+1}_{I,J}}\left[\left(\dot{r}^{n+\frac{1}{2}}_{I+1,J+1} - \dot{r}^{n+\frac{1}{2}}_{I,J}\right)\left(z^{n+1}_{I,J+1} - z^{n+1}_{I+1,J}\right)\right. \\
&\qquad \left. - \left(\dot{r}^{n+\frac{1}{2}}_{I,J+1} - \dot{r}^{n+\frac{1}{2}}_{I+1,J}\right)\left(z^{n+1}_{I+1,J+1} - z^{n+1}_{I,J}\right)\right], \\
(\dot{\epsilon}_{zz})^{n+\frac{1}{2}}_{I,J} &= \frac{-1}{2A^{n+1}_{I,J}}\left[\left(\dot{z}^{n+\frac{1}{2}}_{I+1,J+1} - \dot{z}^{n+\frac{1}{2}}_{I,J}\right)\left(r^{n+1}_{I,J+1} - r^{n+1}_{I+1,J}\right)\right. \\
&\qquad \left. - \left(\dot{z}^{n+\frac{1}{2}}_{I,J+1} - \dot{z}^{n+\frac{1}{2}}_{I+1,J}\right)\left(r^{n+1}_{I+1,J+1} - r^{n+1}_{I,J}\right)\right], \\
\text{and} \\
(\dot{\epsilon}_{\theta\theta})^{n+\frac{1}{2}}_{I,J} &= \frac{\Delta V^{n+\frac{1}{2}}_{I,J}}{\Delta t^{n+\frac{1}{2}}} \cdot \frac{1}{V^{n+1}_{I,J}} - \left[(\dot{\epsilon}_{rr})^{n+\frac{1}{2}}_{I,J} + (\dot{\epsilon}_{zz})^{n+\frac{1}{2}}_{I,J}\right],
\end{aligned}\right\} \tag{43}$$

and the strains by

$$\left.\begin{aligned}
(\epsilon_{rr})^{n+1}_{I,J} &= (\epsilon_{rr})^{n}_{I,J} + (\dot{\epsilon}_{rr})^{n+\frac{1}{2}}_{I,J}\,\Delta t^{n+\frac{1}{2}}, \\
(\epsilon_{zz})^{n+1}_{I,J} &= (\epsilon_{zz})^{n}_{I,J} + (\dot{\epsilon}_{zz})^{n+\frac{1}{2}}_{I,J}\,\Delta t^{n+\frac{1}{2}}, \\
\text{and} \quad & \\
(\epsilon_{\theta\theta})^{n+1}_{I,J} &= (\epsilon_{\theta\theta})^{n}_{I,J} + (\dot{\epsilon}_{\theta\theta})^{n+\frac{1}{2}}_{I,J}\,\Delta t^{n+\frac{1}{2}}.
\end{aligned}\right\} \tag{44}$$

Calculation of the total energies serves as a check on the accuracy of the numerical computation. The internal energy is calculated by

$$(IE)^{n+1} = \sum_{I,J} \left(\frac{E^{n+1}_{I,J} + E^{n}_{I,J}}{2} \right) m^{0}_{I,J}, \tag{45}$$

and the kinetic energy by

$$\begin{aligned}
(KE)^{n+1} = \sum_{I,J} \tfrac{1}{8}\, m^{0}_{I,J} \Bigg[& \left(\dot{r}^{n+\frac{1}{2}}_{I,J}\right)^2 + \left(\dot{z}^{n+\frac{1}{2}}_{I,J}\right)^2 + \left(\dot{r}^{n+\frac{1}{2}}_{I,J+1}\right)^2 \\
& + \left(\dot{z}^{n+\frac{1}{2}}_{I,J+1}\right)^2 + \left(\dot{r}^{n+\frac{1}{2}}_{I+1,J+1}\right)^2 + \left(\dot{z}^{n+\frac{1}{2}}_{I+1,J+1}\right)^2 \\
& + \left(\dot{r}^{n+\frac{1}{2}}_{I+1,J}\right)^2 + \left(\dot{z}^{n+\frac{1}{2}}_{I+1,J}\right)^2 \Bigg].
\end{aligned} \tag{46}$$

Since the accuracy of the finite-difference equations depends largely on zone deformations, an extraordinarily large value of the distortion index can be used as a signal for stopping the computation. The index for each zone is calculated from

$$(DI)^{n+1}_{I,J} = \mathrm{Max}\left[\frac{\mathrm{Max}(D_1,D_3)}{\mathrm{Min}(D_1,D_3)}, \frac{\mathrm{Max}(D_2,D_4)}{\mathrm{Min}(D_2,D_4)}, \frac{\mathrm{Max}(D_5,D_6)}{\mathrm{Min}(D_5,D_6)} \right], \tag{47}$$

where

$$D_1 = \left(r^{n+1}_{I+1,J} - r^{n+1}_{I,J}\right)^2 + \left(z^{n+1}_{I+1,J} - z^{n+1}_{I,J}\right)^2,$$

$$D_2 = \left(r^{n+1}_{I+1,J+1} - r^{n+1}_{I+1,J}\right)^2 + \left(z^{n+1}_{I+1,J+1} - z^{n+1}_{I+1,J}\right)^2,$$

$$D_3 = \left(r^{n+1}_{I+1,J+1} - r^{n+1}_{I,J+1}\right)^2 + \left(z^{n+1}_{I+1,J+1} - z^{n+1}_{I,J+1}\right)^2,$$

$$D_4 = \left(r^{n+1}_{I,J+1} - r^{n+1}_{I,J}\right)^2 + \left(z^{n+1}_{I,J+1} - z^{n+1}_{I,J}\right)^2,$$

$$D_5 = \left(r^{n+1}_{I+1,J+1} - r^{n+1}_{I,J}\right)^2 + \left(z^{n+1}_{I+1,J+1} - z^{n+1}_{I,J}\right)^2,$$

and

$$D_6 = \left(r^{n+1}_{I,J+1} - r^{n+1}_{I+1,J}\right)^2 + \left(z^{n+1}_{I,J+1} - z^{n+1}_{I+1,J}\right)^2.$$

This completes one cycle of calculation. At this point, the finite-difference equations are programmed for numerical solution on the IBM 360 computer. A FORTRAN listing of the code is given in Appendix C.

III. COMPUTER PROGRAM

A. Input Information Required

The following card types are required as input information to the program and are listed in the order in which they must appear in the data deck.

Card Type	Columns	Format	FORTRAN Name	Description
1	1-72	(18A4)	TITLE	Title card: 71 characters of alphanumerics for problem identification. Column 1 must be blank.
2		(5I6,3F12.0)		
	1-6		IMAX	Number of zones in the radial (R) direction.
	7-12		JMAX	Number of zones in the axial (Z) direction.
	13-18		KB1	Boundary-condition indicator for the top surface (upper Z). KB1 = 0: Fixed surface. KB1 = 2: Free surface.
	19-24		KB2	Boundary-condition indicator for the cylindrical surface. KB2 = 0: Fixed surface. KB2 = 2: Free surface.
	25-30		KB3	Boundary-condition indicator for the bottom surface (lower Z). KB3 = 0: Fixed surface. KB3 = 2: Free surface.

Card Type	Columns	Format	FORTRAN Name	Description
2 (Contd.)	31-42		TIME	Initial problem starting time, in seconds.
	43-54		DELT	Initial time interval, in seconds.
	55-66		DELTM	Maximum time interval, in seconds.
3		(5F12.0)		
	1-12		CYCLM	Stop cycle; allows the problem to be terminated after stop cycle. (If CYCLM = 0 or blank, the program sets CYCLM = 10000.)
	13-24		TMAX	Maximum time, in seconds; allows the problem to be terminated after TMAX seconds. (If TMAX = 0 or blank, the program sets TMAX = 10000.)
	25-36		DISTM	Maximum distortion index; allows the problem to be terminated if the distortion index on any zone in the problem exceeds DISTM. (If DISTM = 0 or blank, the program sets DISTM = 10000.)
	37-48		DE1	Instability warning indicator. If the percentage change in the total energy is greater than DE1, the program ignores IOUA instruction on card of type 4 and prints full accuracy output every cycle. (If DE1 = 0 or blank, the program sets DE1 = 0.001.)
	49-60		DE2	Instability control parameter; allows the problem to be terminated if the percentage change in the total energy from the initial total energy exceeds DE2. (If DE2 = 0 or blank, the program sets DE2 = 0.005.)
4		(12I6)		
	1-6		IOUA	Parameter to determine the full accuracy printout. IOUA > 0: Full-accuracy output every IOUA cycle. IOUA = 0: No printout.
	7-12		INUMBA	Maximum number of full-accuracy printouts. After the number of full-accuracy printouts exceeds INUMBA, the program prints the full-accuracy printout for the last cycle only.
	13-18		IOUB	Parameter to determine the limited 2-dimensional (2-D) printout. IOUB > 0: Limited 2-D output every IOUB cycle. IOUB = 0: No printout.
	19-24		IOUC	Parameter to determine film output. (IBM 2280 Film Recorder Picture Display.) IOUC > 0: Film output every IOUC cycle. IOUC = 0: No film output.
	25-30		IOUT	Parameter to determine restart capability--usage of auxiliary tapes 8 and 9. IOUT = 0: Tapes 8 and 9 are not used.

Card Type	Columns	Format	FORTRAN Name	Description
4 (Contd.)	25-30		IOUT	IOUT = 1: After the computation is terminated, the program writes the output data on a binary tape 8 so that the problem may be continued later. IOUT = 2: Continuation of the problem from a previous run. Program reads the input data from a binary tape 9. IOUT = 3: Program combines both IOUT = 1 and IOUT = 2 capabilities: Program reads the input data from a binary tape 9 and also writes the output data on a binary tape 8 for later continuation.

Note: For IOUT = 2 and 3, the input to the program is on tape 9; thus cards of types 5 to 24 are omitted and only cards of types 1, 2, 3, 4, 25, 26, 27, 28, and 29 are required.

5		(8F9.0)		Initial grid dimensions--radial direction.
	1-9		R(2,2)	$R_{I,2}$; I = 2, 3, ..., n.
	10-18		R(3,2)	
	19-27		R(4,2)	
	Use as many cards of type 5 as required; n = IMAX + 2.			
6		(8F9.0)		Initial grid dimensions--axial direction.
	1-9		Z(2,2)	$Z_{2,J}$; J = 2, 3, ..., n.
	10-18		Z(2,3)	
	19-27		Z(2,4)	
	Use as many cards of type 6 as required; n = JMAX + 2.			
7	1-6	(I6)	NSEC	Number of rectangular sections into which the grid is subdivided.
8		(7I6,2F9.0)		Section Cards.
	1-6		$KR1_k$	Starting zone number in the radial direction.
	7-12		$KR2_k$	Final zone number in the radial direction.
	13-18		$KZ1_k$	Starting zone number in the axial direction.
	19-24		$KZ2_k$	Final zone number in the axial direction.
	25-30		$KT1_k$	Material indicator. $KT1_k$ = 1 (core) 2 (sodium) 3 (steel) 4 (argon) 5 (axial blanket) 6 (radial blanket) 7 (plenum) 8 (water) As used in the sample problem.
	31-36		$KT2_k$	Material phase indicator. $KT2_k$ = 1 (solids or liquids) 2 (vapor).
	37-42		KTM_k	Parameter describing the types of input used for zone properties (ρ,E,P). KTM_k = 1: Input for all zones in this section is on cards of type 10. KTM_k = 0: Input for each zone in this section is on cards of type 14.

Card Type	Columns	Format	FORTRAN Name	Description
8 (Contd.)	43-51		RDD	Initial radial velocity, in cm/sec.
	52-60		ZDD	Initial axial velocity, in cm/sec.
9	1-6	(I6)	NMAT	Number of different materials. NMAT must be equal or greater than any $KT1_k$ on cards of type 8.
10		(6F9.0,I3)		
	1-9		AA_k	a } Constants for equation of state (gaseous materials): Core $P = a \exp[-b/(E+c)]$.
	10-18		BB_k	b
	19-27		CC_k	c } Argon $P = a E \rho$; used only when $KT2_k = 2$ for the same material.
	28-36		$CRHO_k$	Initial density ρ_0, in g/cm^3.
	37-45		CE_k	Initial energy E_0, in dyne-cm/g.
	46-54		CP_k	Initial pressure P_0, in dynes/cm^2.
	55-57		KKK_k	Number of PP_I and VV_I values on cards of type 12. PP_I and VV_I represent Hugoniot curve for the material. If $KKK_k \leq 0$, cards of type 12 are not used.
11		(6F9.0)		
	1-9		CWN_k	η } Constants for stability criterion (see Eq. 27).
	10-18		CWB_k	p_B
	19-27		$CPLG_k$	Parameter for strain calculations. If $CPLG_k \leq 0$, no strain calculation is required. If $CPLG_k > 0$, radial, axial, and angular strains are calculated for this material. Value of $CPLG_k$ is the maximum allowable strain of the material. Problem will be terminated when the strain in any zone of the material exceeds CPLG.
12		(8F9.0)		Hugoniot-curve points; used only when $KKK_k > 0$.
	1-9		$PP_{k,1}$	$(P_H)_1$ } P_H are the pressure values in kilobars on Hugoniot curve for the specified material k. $(V/V_0)_I$ are the ratios of the specific volumes for the $(P_H)_I$ values. $I = 1, 2 \ldots, KKK_k$.
	10-18		$VV_{k,1}$	$(V/V_0)_1$
	19-27		$PP_{k,2}$	$(P_H)_2$
	28-36		$VV_{k,2}$	$(V/V_0)_2$
	Use as many cards of type 12 as required: $0 < KKK \leq 50$. Note: The Hugoniot curves on cards are in increasing order of V/V_0.			
13		(8F9.0)		Constants for materials that are using the Hugoniot table; used only when $KKK_k > 0$.
	1-9		PO_k	Initial pressure at which the Hugoniot curve is calculated, in dynes/cm^2.
	10-18		ROK	Initial density for the Hugoniot curve, in g/cm^3.
	19-27		EO_k	Initial specific energy for the Hugoniot curve, in dyne-cm/g.
	28-36		GO_k	γ_1: Mie-Grüneisen coefficient.
	37-45		CO_k	γ_2: Additional Mie-Grüneisen coefficient. Used when material is steel for $P_H > 131$ kb, or water for $P_H > 136$ kb.

Card Type	Columns	Format	FORTRAN Name	Description
14		(2I3,7F9.0,2I3)		Card input for individual zones; used only when $KTM_k = 0$ on card of type 8.
	1-3		$II_{I,J}$	Zone number in the radial direction.
	4-6		$JJ_{I,J}$	Zone number in the axial direction.
	7-15		$RC1_{I,J}$	Radial dimension, in cm.*
	16-24		$ZC1_{I,J}$	Axial dimension, in cm.*
	25-33		$RDOT_{I,J}$	Radial velocity, in cm/sec.*
	34-42		$ZDOT_{I,J}$	Axial velocity, in cm/sec.*
	43-51		$RHO_{I,J}$	Density, in g/cm^3.
	52-60		$E_{I,J}$	Specific energy, in dyne-cm/g.
	61-69		$P_{I,J}$	Pressure in $dynes/cm^2$.
	70-72		$KTX_{I,J}$	Material indicator; same identification as $KT1_k$ on card of type 8.
	73-75		$KTY_{I,J}$	Material phase indicator; same identification as $KT2_k$ on card of type 8.
15		(5I6,F12.0, 2I6,F12.0)		Parameters to describe the plug and platform calculations (upper surface).
	1-6		KPP	Parameter that determines whether platform motion is to be calculated. KPP = 1: Platform motion is calculated according to $M\ddot{Z} + C\dot{Z} + KZ = F(t)$, where M is the total mass of the platform, KZ is the spring force, $C\dot{Z}$ is the damping force, and F(t) is the total force applied by the system on the platform. KPP = 0: No platform calculations.
	7-12		KPP1	Radial-zone number at which the platform starts.
	13-18		KPP2	Radial-zone number at which the platform ends (KPP2 $\geq$ KPP1).
	19-24		KPPX	Number of CX_I and CXD_I values on cards of type 16, where the spring force KZ_I vs Z_I is tabulated. If KPPX $<$ 0, KZ is set to zero.
	25-30		KPPC	Number of CV_I and CVD_I values on cards of type 17, where the damping force $C\dot{Z}_I$ vs $\dot{Z}_I$ is tabulated. If KPPC $\leq$ 0, CZ is set to zero.
	31-42		PMASS	Total mass of platform, in grams.
	Note: KPP1, KPP2, KPPX, and KPPC are used only when KPP = 1.			
	43-48		KPL1	Parameter to determine plug calculations. If KPL1 $>$ 0, the total force applied by the system on the plug is calculated. KPL1 indicates the number of the first radial zone of the plug.

*These values are the displacements and velocities at the lower left corner of the individual zone.

Note: When $KTM_k = 0$ on card of type 8, cards of type 14 must be entered in the following order for each section: First, cards having $J = KZ1_k$ arranged in the increasing order of I, i.e., from $I = KR1_k$ to $I = KR2_k$; next, groups of cards having $J = KZ1_k + 1$, $KZ1_k + 2$, etc., until $J = KZ2_k$. In each group, cards are again arranged in increasing order of I, i.e., from $I = KR1_k$ to $I = KR2_k$. $KR1_k$, $KR2_k$, $KZ1_k$, and $KZ2_k$ are defined on cards of type 8.

Card Type	Columns	Format	FORTRAN Name	Description
15 (Contd.)	49-54		KPL2	Number of the last radial zone of the plug. (KPL2 ≥ KPL1.)
	55-66		PLUG	Allowable plug force, in dynes. Problems will be terminated when the force acting on the plug exceeds PLUG.

Note: KPL2 and PLUG are used only when KPL1 > 1.

16		(8F9.0)		KZ_I-vs-Z_I table; used only when KPPX > 0 on card of type 15.
	1-9		CX_1	KZ_1
	10-18		CXD_1	Z_1
	19-27		CX_2	KZ_2
	28-36		CXD_2	Z_2

KZ_I is the spring force, in dynes, for the displacement Z_I, in cm.

Use as many cards of type 16 as required.

Note: The KZ_I-vs-Z_I values are entered in pairs in increasing order of Z_I, starting with Z_1 (=0), Z_2, ..., to Z_{KPPX}.

17		(8F9.0)		$C\dot{Z}_I$-vs-$\dot{Z}_I$ table; used only when KPPC > 0 on card of type 15.
	1-9		CV_1	$C\dot{Z}_1$
	10-18		CVD_1	$\dot{Z}_1$
	19-27		CV_2	$C\dot{Z}_2$
	28-36		CVD_2	$\dot{Z}_2$

$C\dot{Z}_I$ is the damping force, in dynes, for the velocity $\dot{Z}_I$, in cm/sec.

Use as many cards of type 17 as required.

Note: The $C\dot{Z}_I$-vs-$\dot{Z}_I$ values are entered in pairs in increasing order of $\dot{Z}_I$, starting with $\dot{Z}_1$ (=0), $\dot{Z}_2$, ..., to $\dot{Z}_{KPPC}$.

18		[3(2I3,F12.0,2I3)]		Parameter card for the free-surface boundary condition.

The following parameters are associated with the top surface (upper Z); they are used only when KB1 = 2. (See card type 2.)

	1-3		KB11	Radial-zone number at which the free surface starts.
	4-6		KB12	Radial-zone number at which the free surface ends.

Note: Boundary condition for zones outside KB11 and KB12 is fixed surface.

	7-18		YMX	Limiting axial dimension, in cm. Mesh points on the free surface may move up to YMX.
	19-21		M11	Material indicator for space between free surface and YMX. Same identification as $KT1_k$ on card of type 8.
	22-24		M12	Material phase indicator for M11 material. Same identification as $KT2_k$ on card of type 8.

The following parameters are associated with the cylindrical surface; they are used only when KB2 = 2. (See card type 2.)

	25-27		KB21	Axial-zone number at which the free surface starts.

Card Type	Columns	Format	FORMAT Name	Description
18 (Contd.)	28-30		KB22	Axial-zone number at which the free surface ends.
	Note: The boundary condition for zones outside KB21 and KB22 is a fixed surface.			
	31-42		YMS	Limiting radial dimension, in cm. Mesh points on the free surface may move up to YMS.
	43-45		M21	Material indicator for space between free surface and YMS. Same identification as $KT1_k$ on card of type 8.
	46-48		M22	Material phase indicator for M21 material. Same identification as $KT2_k$ on card of type 8.
	The following parameters are associated with the bottom surface (lower Z); they are used only when KB3 = 2. (See card type 2.)			
	49-51		KB31	Radial-zone number at which the free surface starts.
	52-54		KB32	Radial-zone number at which the free surface ends.
	Note: Boundary condition for zones outside KB31 and KB32 is fixed surface.			
	55-66		YMN	Limiting axial dimension, in cm. Mesh points in the free surface may move up to YMN.
	67-69		M31	Material indicator for space between free surface and YMN. Same identification as $KT1_k$ on card of type 8.
	70-72		M32	Material phase indicator for M31 material. Same identification as $KT2_k$ on card of type 8.
19	1-6	(I6)	NPP	Number of zones for which pressures and/or displacements are printed after each cycle. These values also may be displayed using the Calcomp option. (See card type 29.)
20	1-6	(12I6)	KXP_1	First radial-zone number.
	7-12		KYP_1	First axial-zone number.
	13-18		KXP_2	Second radial-zone number.
	19-24		KYP_2	Second axial-zone number.
	KXP_j, KYP_j define each zone to be displayed; $j = 1, 2, \ldots, NPP$ $(0 \leq NPP \leq 6)$. The signs before KXP_j and KYP_j indicate the type of information to be displayed.			
				$\left.\begin{array}{l} KXP_j > 0 \\ KYP_j > 0 \end{array}\right\}$ Pressure.
				$\left.\begin{array}{l} KXP_j < 0 \\ KYP_j > 0 \end{array}\right\}$ Radial displacement of the lower left corner.
				$\left.\begin{array}{l} KXP_j > 0 \\ KYP_j < 0 \end{array}\right\}$ Axial displacement of the lower left corner.
	Note: Card type 20 is used only when NPP > 0.			
21	1-6	(2I6)	NTW	Number of reactor vessels.
	7-12		NTM	Number of different materials used for the vessels.

Card Type	Columns	Format	FORTRAN Name	Description
	The following card types are used only when NTW > 0.			
22	1-6	(5I6,2F12.0)	MI_j	Radial-zone number for the vessel wall.
	7-12		MJ_j	Axial-zone number for the vessel bottom plate.
	13-18		MTH_j	Material number for the vessel bottom plate.
	19-24		MTV_j	Material number for the vessel wall.
	25-30		MTC_j	Describes type of support for top of vessel:
				MTC_j = 4: simply supported.
				MTC_j = 5: fixed.
				MTC_j = 6: free.
	31-42		THH_j	Thickness of vessel bottom plate, in cm.
	43-54		THV_j	Thickness of vessel wall, in cm.
	Use as many of cards of type 22 as required; j = 1, 2, ..., NTW (0 ≤ NTW ≤ 10).			
23		(5F12.0,I6)		Vessel card properties.
	1-12		XME_k	Modulus of elasticity, in dynes/cm^2.
	13-24		XYP_k	Yield point, in dynes/cm^2.
	25-36		XEU_k	Ultimate strain, in cm/cm.
	37-48		$XRHO_k$	Material density, in g/cm^3.
	49-60		KPR_k	Poisson's ratio.
	61-66		KNP_k	Number of XSS_i and XSR_i values on cards of type 24.
	Use as many cards of type 23 as required; k = 1, 2, ..., NTM (0 ≤ NTM ≤ 10).			
24		(8F9.0)		Stress-strain curve points; used only when KNP_k > 0.
	1-9		$XSS_{k,1}$	$XSS_{k,i}$ are the stress values, in dynes/cm^2, for the specified material k.
	10-18		$XSR_{k,1}$	$XSR_{k,i}$ are the strain values for $XSS_{k,i}$ points.
	19-27		$XSS_{k,2}$	
	28-36		$XSR_{k,2}$	
	Use as many cards of type 24 as required; i = 1, 2, ..., KNP_k (0 ≤ KNP_k ≤ 50).			
	Note: The stress-strain curves are in decreasing order of stress values.			
	The following cards are used only when IOUC > 0.			
25	1-6	(I6)	N	Number of lines on the grid that are to be repeated to provide heavier outline for film output.
26		(12I6)		Specification of the lines that are to be repeated; used only when N > 0 on card of type 25.
	1-6		$IX1_1$	Starting mesh-point number of the first line (in radial direction).
	7-12		$IX2_1$	Final mesh-point number of the first line (in radial direction).
	13-18		$JX1_1$	Starting mesh-point number of the first line (in axial direction).

Card Type	Columns	Format	FORTRAN Name	Description
26 (Contd.)	19-24		$JX2_1$	Final mesh-point number of the first line (in axial direction).
	25-30		$IX1_2$	Starting mesh-point number of the second line (in radial direction).
	Use as many cards of type 26 as required, three lines per card.			
	Note: For each line J, either $IX1_J = IX2_J$ or $JX1_J = JX2_J$; $J = 1, 2, \ldots, N$ $(1 \leq N \leq 50)$.			
27	1-6	(I6,F12.0)	NNM	Number of pressure curves for each displayed cycle.
	7-18		PMAX	Maximum pressure, in dynes/cm^2, that can be plotted.
28		(12I6)		Specification of the lines for which the pressure is plotted; used only when NNM > 0 on card of type 27.
	1-6		$NX1_1$	For positive $NX1_i$, program plots the pressure for all axial zones (Z_J - J = 2, JMAX + 1) at the radial zone $R = NX1_i$. For negative $NX1_i$, program plots the pressure for all radial zones (R_I - 1 = 2, IMAX + 1) at the axial zone $Z = NX1_i$.
	7-12		$NX1_2$	
	13-18		$NX1_3$	
	Use as many cards of type 28 as required, 12 numbers per card; $i = 1, 2, \ldots, NNM$ $(1 \leq NNM \leq 50)$.			
29		[I6,(6F12.0)]		Calcomp option card.
	1-6		KCAL	KCAL > 0: Program draws Calcomp plots of plug force and the values specified on cards of types 19 and 20 vs time. KCAL ≤ 0: No Calcomp plot.
	7-18		SCT	Scale for time axis (horizontal). SCT is the number of seconds per inch of plot. If SCT = 0.0 or blank, the program sets SCT = 0.001.
	19-30		SCT1	Scale for plug force (vertical axis). SCT1 is the number of dynes per inch of plot. If SCT1 = 0.0 or blank, the program calculates the optimum scale.
	31-42		$SCTK_1$	Scale for first value described on input cards of types 19 and 20 (vertical axis). If $SCTK_1$ = 0.0 or blank, the program calculates the optimum scale.
	43-45		$SCTK_2$	Scale for second value described on input cards of types 19 and 20 (vertical axis). If $SCTK_2$ = 0.0 or blank, the program calculates the optimum scale.
	Use as many cards as required; $K = 1, 2, \ldots, NPP$ $(0 \leq NPP \leq 6)$.			

B. Computer Output

1. Standard Program Printout

(1) Title of problem

(2) Input data

(3) Total energy

(4) For each cycle

(a) Full accuracy output (see Subsection 1.a below)

(b) Limited 2-D output (see Subsection 1.b below)

(c) Time (sec); total internal energy (dyne-cm); total kinetic energy (dyne-cm); and total energy (dyne-cm)

(d) Cycle number; time; time interval (D-TIME); maximum distortion index (DISTORT); location of the maximum distorted zone; maximum (calculated) White stability number (WMAX) and its location

(e) Plug force and other values requested on input cards of types 19 and 20. (These values are also printed at the end of each run.)

(5) Reason for termination of run (if other than specified on input card type 3).

a. Full-accuracy Output

When IOUA > 0, a full-accuracy printout is given for each IOUA cycle, subject to the limitation in INUMBA. (See input card type 4 for explanation of IOUA and INUMBA.)

(1) For each cycle

(a) Title of problem

(b) Cycle number

(c) Time

(2) For each zone

Eleven columns, consisting of integers I and J; $R_{I,J}$; $Z_{I,J}$; $\dot{R}_{I,J}$; $\dot{Z}_{I,J}$; $P_{I,J}$; $VP_{I,J}$; $E_{I,J}$; $\rho_{I,J}$; and material and phase indicators.

$R_{I,J}$, $Z_{I,J}$ = position of lower left corner of zone I,J (in cm).

$\dot{R}_{I,J}$, $\dot{Z}_{I,J}$ = velocity of lower left corner of zone I,J (in cm/sec).

$P_{I,J}$ = total pressure (in dynes/cm^2).

$VP_{I,J}$ = viscous pressure (in dynes/cm^2).

$E_{I,J}$ = internal energy (in dyne-cm/g).

$\rho_{I,J}$ = density (in g/cm^3).

Note: For the initial data (cycle = 0), $M^0_{I,J}$ is printed instead of $VP_{I,J}$, where $M^0_{I,J}$ is the mass (in grams) of each zone.

b. Limited 2-D Output

This integer output is printed for every IOUB cycle, only when IOUB > 0. It is in the form of a matrix and includes the following properties:

(1) Initial radial position of the grid points ($R^0_{I,J}$).

(2) Initial axial position of the grid points ($Z^0_{I,J}$).

Note: Items 1 and 2 are printed only when cycle number = 0.

(3) Radial displacement of the grid points from the initial position.

(4) Axial displacement of the grid points from the initial position.

(5) Radial velocity of the grid points.

(6) Axial velocity of the grid points.

(7) Total pressure of the zones.

(8) Viscous pressure of the zones.

(9) Specific internal energy of the zones.

(10) Density of the zones.

(11) Radial strain of the zones.

(12) Axial strain of the zones.

(13) Angular strain of the zones.

More specifically, each page of printout is prefaced by the problem title, property definition, time, time interval, cycle number, maximum absolute value, and scale factor used. This is followed by a matrix (maximum 50 x 25) of the property values, where the rows indicate radial and the columns indicate axial direction. Numbers are printed on the top and left side of the matrix to indicate the position of each zone in the grid. If the grid size exceeds 50 x 25, the printout continues on successive pages until the grid is completed.

To obtain the printed integers, the program multiplies the calculated property values by the indicated scale factor and then truncates. The maximum number of integers for each property is limited to 4; thus the maximum number printed is ±9999. If all calculated values are zeros, the matrix output is omitted.

2. Pictorial Display (IBM-2280 Film Recorder)

Pictorial displays of the grid displacement and pressure can be obtained for every IOUC cycle, only when IOUC > 0. (See input card type 4.)

a. Grid Displacement. The program draws the grids for specified cycles according to the input on cards of types 25 and 26, and then obtains the required film output. Cycle number and time (in seconds) are drawn at the top of each grid. (Note: The lower left corner of the picture indicates the position of $R_{2,2}$ and $Z_{2,2}$.)

b. Pressure. If a pressure display in either the axial or radial direction is specified, the program plots the pressure of each zone (at the lower zone mesh point) in the specified direction, and then draws the interconnecting vectors. (Note: The zone number in the other direction is constant.)

To ensure uniform scaling of the plots, the expected maximum pressure must be specified. (See input cards of types 27 and 28.)

3. Calcomp Display

This display is executed at the end of a problem, but only when KCAL > 0. (See input card of type 29.)

When executed, the program plots, as a function of time elapsed, the total force applied on the plug by the system. It also plots, as a function of time, the values described on input cards of types 19 and 20.

Each of the above plots has its own vertical and horizontal axis. The latter always represents time, and its scale is specified on an input card of type 29. Scales for the vertical axis (10 in. allowed) may be specified on a card of type 29; if not, the program finds the optimum scale from the calculated values. On all plots, consecutive points are interconnected with vectors.

C. Program Limitations and Subroutines

1. Limitations

Number of	Not to exceed
(1) Grid zones	5000
(2) Different materials	20
(3) Different sections	20
(4) Points for Hugoniot curve(s)	50
(5) Points for KZ-vs-Z and $C\dot{Z}$-vs-$\dot{Z}$ tables	50
(6) Lines repeated for pictorial display	50
(7) Different pressure plots per cycle	50
(8) Reactor vessels	10
(9) Points for stress-vs-strain table for reactor material	50
(10) Cycles per run for Calcomp display	1000
(11) Different plots for Calcomp display	7

2. Subroutines

To execute the pictorial display, the program uses the subroutines described in, A Film-Plotting Subroutine Package (FSP) for the IBM Film Recorder, by D. Carson (Argonne AMD Technical Memorandum No. 167, June 17, 1968).

To execute the Calcomp display, the program uses the subroutines described in, S/360 Programming Techniques for the Calcomp 780, by R. F. Krupp (Argonne AMD Technical Memorandum No. 130, January 6, 1967).

IV. SAMPLE PROBLEM

A. Reactor Configuration

The reactor is of the oxide-fueled pancake type having a core height of 50 cm. The fuel is contained in stainless steel pins supported in a stainless steel grid plate and cooled by liquid sodium. With reference to Fig. 3, the core is surrounded by a radial blanket, an upper blanket and plenum, and a lower blanket. The sodium is blanketed with argon gas and is contained in a steel vessel. This vessel is installed within a concrete cavity that has a rotating shield plug in the platform at the top. Because of the axisymmetry, only half the cross section is shown in Fig. 3. The grid plate is not included in the analysis.

B. Excursion Model

It is assumed that at the start of the power excursion the core is molten and the sodium has been expelled from the core region, but the blankets are intact. The energy release during the excursion is so rapid that the molten oxide fuel is vaporized and superheated to a high temperature and pressure. At the end of the excursion, the core consists of only high-pressure oxide vapor, but is still surrounded by the core blankets and liquid sodium in their preexcursion state.

C. Equations of State

The equation of state for core oxide vapor is assumed to have the form

$$p = B \exp\left(-\frac{C}{E+D}\right), \tag{48}$$

where

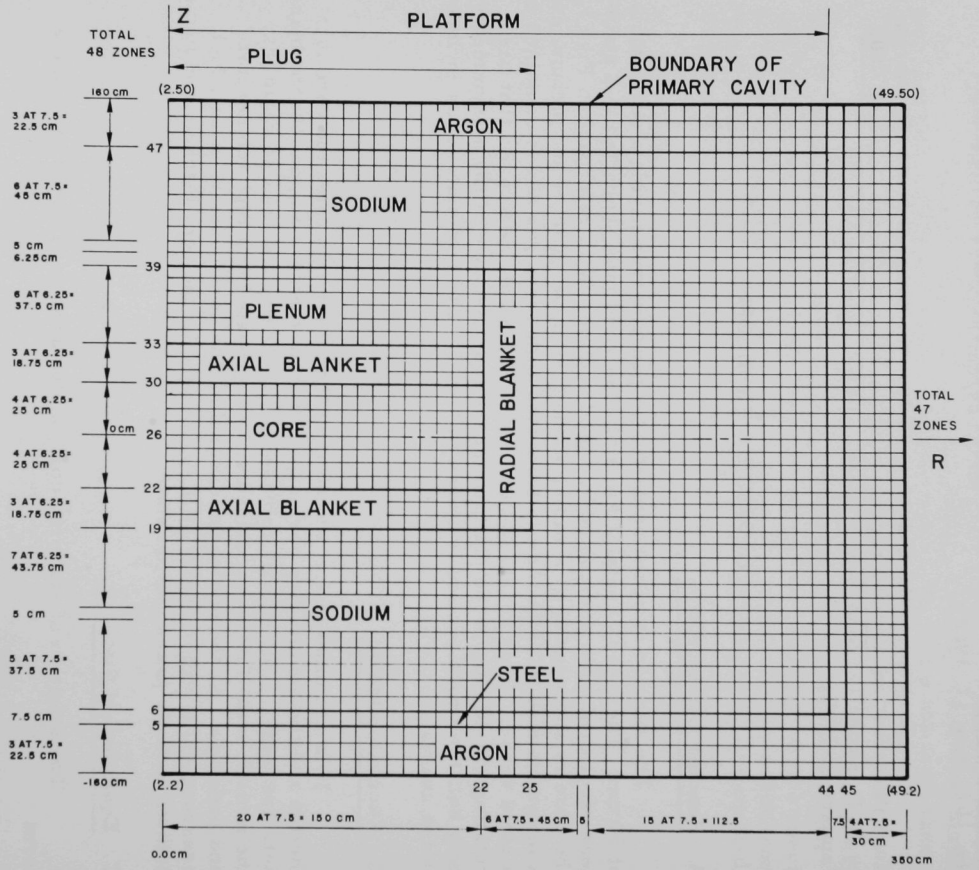


Fig. 3. Dimensions and Lagrangian Meshes of the Reactor Configuration Selected as Sample Problem. ANL Neg. No. 113-1931

p = pressure,

E = internal energy,

and the constants B and C, for p (in dynes/cm^2) and E (in dyne-cm/g) are

$$B = 7.6 \times 10^{11}$$

and

$$C = 5.88 \times 10^{10}.$$

The constant D is taken to be zero.

The equations of state of the core blankets, plenum, steel, and sodium are given by Eq. 18; values of p_H for sodium and steel are taken directly from Ref. 6.* The core blankets and plenum are made up of several different materials. The Hugoniot of these composite materials can be constructed from the experimental Hugoniot of the individual elements, using the method outlined in Appendix B.

The equation of state for argon is given by

$$p = \frac{(\eta - 1)\, E}{V}, \tag{49}$$

where η is the isentropic exponent. For the sake of simplification, this value is taken to be 1.4.

D. Boundary Conditions

The boundary conditions are as follows:

1. At the inner circumferential surface of the primary cavity, the radial velocity of particle is zero at all times; i.e., $\dot{r}(49,J) = 0$, where $J = 2, 3, \ldots, 50$;

2. At the top and bottom of the primary cavity, the axial velocity of particle is zero at all times; i.e., $\dot{z}(I,2) = \dot{z}(I,50) = 0$, where $I = 2, 3, \ldots, 49$.

E. Initial Conditions

Numerical solution of finite-difference equations requires that initial values for all parameters be known at time $t = 0$. In the sample

*The values for sodium were the shock adiabats of the solid-phase sodium. However, the deviations of p_H-V values between the solid-liquid phases are believed to be small.

problem, this is the time when the neutronics have ceased. Therefore, the pressure, internal energy, etc., obtained from the power-excursion calculations are the initial values of this problem. For purposes of checking the 2-D hydrodynamics code, initial values obtained from MARS calculation[9] have been modified. The assumed pressures and specific internal energies in the vaporized core are as tabulated below, where the upper values are pressures, in kilobars, and the lower values are specific internal energies, in joules.

	R(I)			
Z(J)	2-18	19	20	21
22,29	8.8 1318	8.8 1318	8.8 1318	8.8 1318
23,28	26.4 1750	26.4 1750	26.4 1750	8.8 1318
24,27	44.0 2063	44.0 2063	26.4 1750	8.8 1318
25,26	52.8 2205	44.0 2063	26.4 1750	8.8 1318

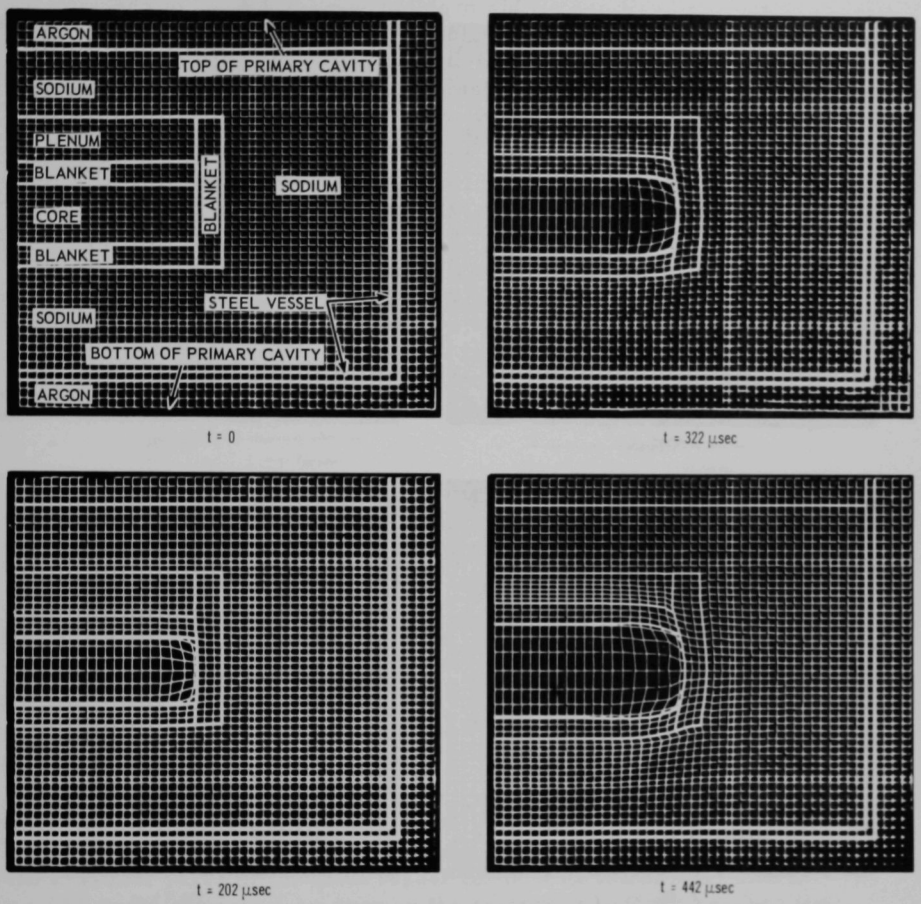
F. Results and Discussion

The outputs of the computer code are displacements, velocities, pressures, specific internal energies, densities, and strains. Here, the displacements and pressures are given in the form of pictures. Figures 4 and 5 show the time sequence of deformations of the Lagrangian meshes. The pressure profiles along the core vertical centerline [R(I) = 2] and the core horizontal axis [Z(J) = 26] at various times are shown in Figs. 6 and 7, respectively.

There are two destructive elements in a nuclear excursion: the initial shock wave, and the vapor pressure. The shock wave travels faster than the speed of sound, compresses the media it encounters, and imparts a velocity to the medium behind the front. After passage of the shock wave, the fluid particles move outward at very high speed in the direction of the shock wave. As a result, an impulse (often called the fluid momentum) is produced on the primary reactor-containment structure.

The second destructive element is a quasi-static pressure, as opposed to the short-duration pressure produced by the shock wave and the fluid momentum. The time needed for this destructive element to become significant would be of the order of 0.1 sec. The numerical method

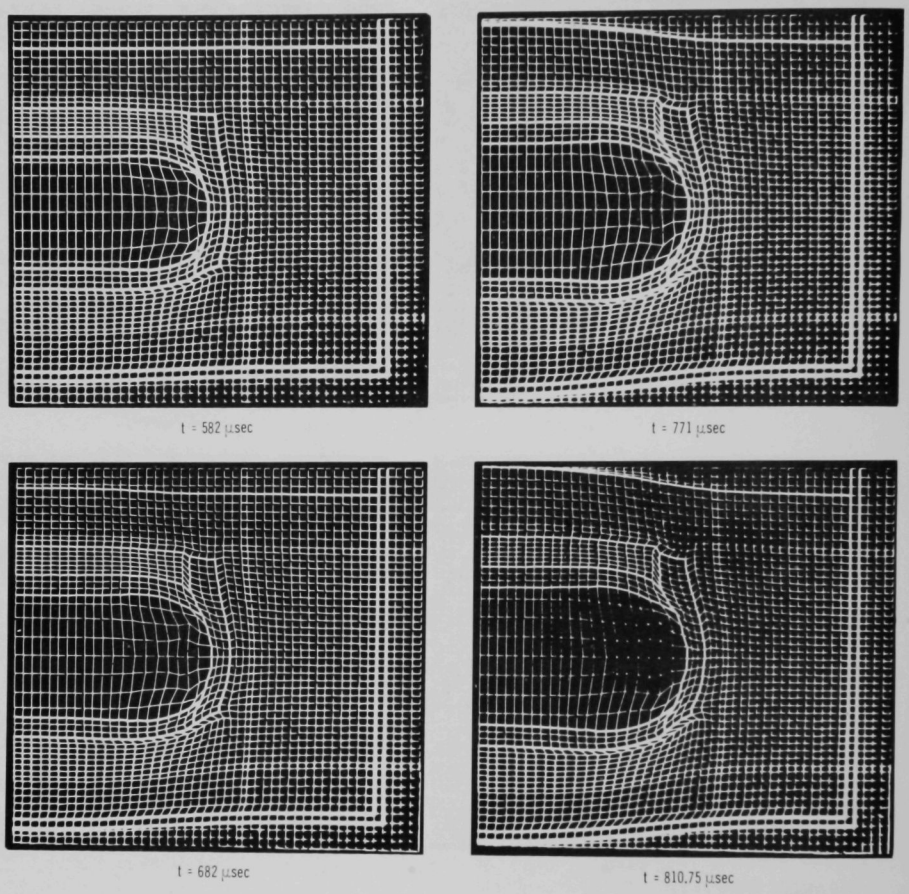
described in this report is not suited for this type of calculation.* Therefore, the balance of the discussion is limited to shock waves.



113-1943

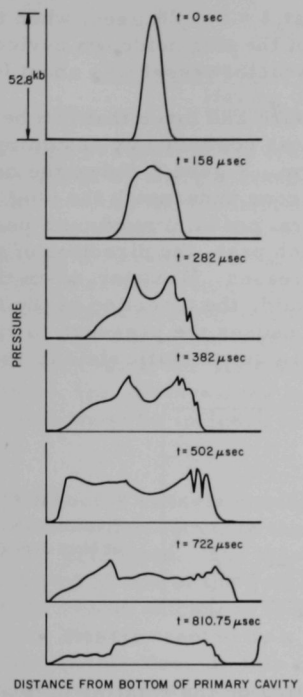
Fig. 4. Deformation of Lagrangian Grids from 0 to 442 μsec after Start of a Power Excursion in a "Pancake" Core Configuration

*At Argonne National Laboratory, a new computer code is being developed which will extend the calculation to the quasi-static pressure range.



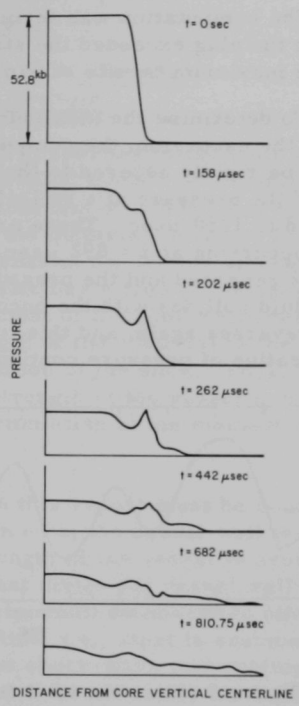
113-1938

Fig. 5. Deformation of Lagrangian Grids from 582 to 810.75 μsec after Start of a Power Excursion in a "Pancake" Core Configuration



113-1945

Fig. 6. Pressure Profiles along Core Vertical Centerline at Various Times



113-1946

Fig. 7. Pressure Profiles along Core Horizontal Axis at Various Times

Figures 6 and 7 show that, as expected, the magnitude of the peak pressure decreases as the shock wave propagates through the core into the surrounding media. The compressive effect of the shock wave is evident from the increase of the pressure pulse in the steel vessel. In Fig. 6, the pressure pulse in the steel vessel reaches its maximum value at t = 502 μsec. However, as soon as the compressive wave reaches the steel-argon interface (t = 722 μsec), a rarefaction wave is reflected back into the steel, with a consequent large reduction of the peak pressure in the shock wave. Figures 6 and 7 also show that a small compressive wave is transmitted to the argon-gas blanket.

The effect of fluid momentum is shown in Fig. 5. Motion of the fluid particles due to the passage of the shock wave causes the reactor vessel to deform and the argon gas to decrease in volume. This action continues until the gas pressure has increased to a point such that the direction of the fluid particles is reversed. Figure 6 shows that the pressure in the argon gas has increased considerably at t = 810.75 μsec.

The computation was terminated at t = 811.28 μsec, when the force acting on the plug exceeded the strength of the plug holddown device. At that time, the maximum tensile strain in the reactor vessel was about 0.0084.

To determine the maximum pressure and force that can be produced in the excursion, the computation was continued by assuming that the plug can be rigidly secured to the platform. Figure 8 shows the oscillating nature of the pressure at a typical mesh zone underneath the plug from t = 811.28 to 1350 μsec. There are several peaks, a maximum peak of 33.3 kb occurring at t = 892 μsec. At each peak, the direction of the fluid motion is reversed and the pressure decreases. However, when the reversed fluid collides with the oncoming fluid, the direction of the fluid motion reverses again, and this, in turn, causes the pressure to rise again. This pulsation of pressure continues until a quasi-static state is reached.

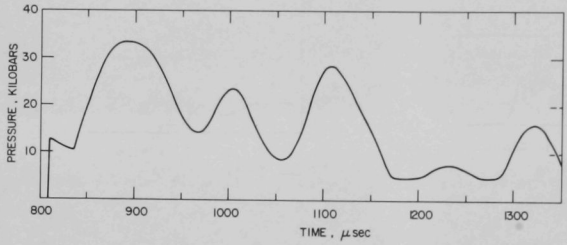


Fig. 8
Pressure-Time Curve at Mesh Zone (2.49)

113-1947

Figure 9 shows the total force acting on the rotating plug as a function of time. This force is also oscillating, the maximum being about 6.44×10^{14} dynes. The impulse acting on the plug is given by

$$I = \int_0^t F(t)\, dt.$$

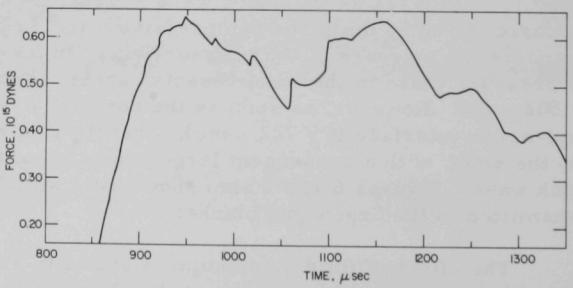


Fig. 9
Total Force Acting on the Rotating Plug as a Function of Time

113-1941

If the plug holddown device is designed to restrict plug movement during the excursion, it must be strong enough to resist this impulse.

APPENDIX A

Momentum Equations for Solid Materials Having Tensile Strength

At high pressures, the mechanical properties of the solid material can be described by using a compressible-fluid model. Thus, during the shock compression, the momentum equation that was developed for fluids can be used for solids too. However, as the material recovers from the shock loading, or the material is under tension, the tensile strength of the material becomes important. This is particularly true in the case of a cylindrical shell, where the radial movement of the shell produces membrane forces and where the tensile strength of the material is the most important mechanism for stopping the motion of the shell. Thus, for reactor vessels, the effects of the tensile strength of the vessel material on vessel motion must be included in the formulation of the momentum equations.

The reactor vessels considered in this report must be concentric right circular cylinders. Like the other media, the vessel wall is divided into zones by grid lines. The tensile strength of the vessel is assumed to be concentrated at one of the grid lines that divide the vessel wall into zones. If the vessel thickness is relatively small compared to other dimensions, it may be represented by a single line; i.e., steel is assumed to be concentrated at this line. In this case, the shock-wave propagation in the vessel is ignored; therefore deformations of the vessel in the direction of the wall thickness will not be calculated. For both cases, the momentum equations in finite-difference form are as follows:

1. Cylindrical Shell

Let the I line in Fig. 2 be the middle plane line of the cylindrical shell. Also, assume that the tensile strength of the vessel is concentrated there. The acceleration in z direction is given by Eq. 35; i.e.,

$$\ddot{z}_{I,J} = \frac{1}{(A\rho)_{I,J}}[(P_{I,J} - P_{I-1,J-1})(r_{I,J+1} - r_{I,J-1} + r_{I-1,J} - r_{I+1,J}) - (P_{I-1,J} - P_{I,J-1})(r_{I,J+1} - r_{I,J-1} + r_{I+1,J} - r_{I-1,J})], \qquad (A.1)$$

where

$$(A\rho)_{I,J} = A_{I,J}\rho_{I,J} + A_{I-1,J}\rho_{I-1,J} + A_{I-1,J-1}\rho_{I-1,J-1} + A_{I,J-1}\rho_{I,J-1}.$$

If a single-line vessel is used, the mass of the vessel must be included in the term $(A\rho)_{I,J}$. The acceleration in the r direction is given by

$$\ddot{r}_{I,J} = \frac{1}{(A\rho)_{I,J}} \left\{ [(P_{I-1,J-1} - P_{I,J})(z_{I,J+1} - z_{I,J-1} + z_{I-1,J} - z_{I+1,J}) \right.$$

$$- (P_{I,J-1} - P_{I-1,J})(z_{I,J+1} - z_{I,J-1} + z_{I+1,J} - z_{I-1,J})]$$

$$\left. - 4\ell_1 \left[\frac{h_1}{r^0_{I,J}} \sigma(\epsilon) \right] \right\}, \qquad (A.2)$$

where

$$\ell_1 = \tfrac{1}{2} \left\{ [(z_{I,J+1} - z_{I,J})^2 + (r_{I,J+1} - r_{I,J})^2]^{\frac{1}{2}} + [(z_{I,J} - z_{I,J-1})^2 \right.$$

$$\left. + (r_{I,J} - r_{I,J-1})^2]^{\frac{1}{2}} \right\},$$

h_1 = thickness of the cylindrical shell,

$$\epsilon = \text{strain} = \frac{r_{I,J} - r^0_{I,J}}{r^0_{I,J}},$$

and

$\sigma(\epsilon)$ = stress obtained from either the actual stress-strain curve or the power- or linear-strain hardening approximations.

In Eq. A.2, the first bracketed term is the acceleration due to pressure gradient, and the second bracketed term is the deceleration due to the membrane strength of the vessel. Again, if a single-line vessel is used, the mass of the vessel must be included in the term $(A\rho)_{I,J}$.

2. Bottom Plate

Let the J line in Fig. 2 be the middle plane line of the bottom plate. Also, assume that the membrane strength of the plate is concentrated there. The acceleration in the r direction is given by Eq. 35; i.e.,

$$\ddot{r}_{I,J} = \frac{-1}{(A\rho)_{I,J}} [(P_{I,J} - P_{I-1,J-1})(z_{I,J+1} - z_{I,J-1} + z_{I-1,J} - z_{I+1,J})$$

$$- (P_{I-1,J} - P_{I,J-1})(z_{I,J+1} - z_{I,J-1} + z_{I+1,J} - z_{I-1,J})] . \qquad (A.3)$$

The acceleration in the z direction is given by

$$\ddot{z}_{I,J} = \frac{1}{(A\rho)_{I,J}}[(P_{I,J} - P_{I-1,J-1})(r_{I,J+1} - r_{I,J-1} + r_{I-1,J} - r_{I+1,J})$$

$$- (P_{I-1,J} - P_{I,J-1})(r_{I,J+1} - r_{I,J-1} + r_{I+1,J} - r_{I-1,J})]$$

$$- \frac{4\,\ell_2 h_2}{(A\rho)_{I,J}}\left\{\frac{1}{(\Delta\bar{R})^2}[(z^0_{I+1,J} - z_{I+1,J}) - 2(z^0_{I,J} - z_{I,J})\right.$$

$$+ (z^0_{I-1,J} - z_{I-1,J})]\ \sigma_{rr} + \frac{1}{2(\Delta\bar{R})r_{I,J}}[(z^0_{I+1,J} - z_{I+1,J})$$

$$\left. - (z^0_{I-1,J} - z_{I-1,J})]\,\sigma_{\theta\theta}\right\},^{*} \qquad (A.4)$$

where

$$\ell_2 = \tfrac{1}{2}\left\{[(z_{I+1,J} - z_{I,J})^2 + (r_{I+1,J} - r_{I,J})^2]^{\frac{1}{2}}\right.$$

$$\left. + [(z_{I,J} - z_{I-1,J})^2 + (r_{I,J} - r_{I-1,J})^2]^{\frac{1}{2}}\right\},$$

h_2 = thickness of the plate,

$\Delta\bar{R} = \tfrac{1}{2}|r_{I+1,J} - r_{I-1,J}|$,

σ_{rr} = radial stress,

and

$\sigma_{\theta\theta}$ = tangential stress.

Again, the mass of the bottom plate must be included in the term $(A\rho)_{I,J}$, if the vessel is represented by a single line.

The stresses σ_{rr} and $\sigma_{\theta\theta}$ are computed from

$$\left.\begin{aligned} \sigma_{rr} &= \frac{E_s}{1 - \nu^2}(\epsilon_{rr} + \nu\epsilon_{\theta\theta}) \\ \text{and} \qquad & \\ \sigma_{\theta\theta} &= \frac{E_s}{1 - \nu^2}(\epsilon_{\theta\theta} + \nu\epsilon_{rr}) \end{aligned}\right\}, \qquad (A.5)$$

---

*The two terms in the bracket are $\frac{d^2\bar{w}}{dr^2}\,\sigma_{rr}$ and $\frac{1}{r}\frac{d\bar{w}}{dr}\sigma_{\theta\theta}$.

where

E_s = modulus of elasticity,

ν = Poisson's ratio,

ϵ_{rr} = radial strain = $\frac{d\bar{u}}{dr} + \frac{1}{2}\left[\left(\frac{d\bar{u}}{dr}\right)^2 + \left(\frac{d\bar{w}}{dr}\right)^2\right]$,

$\epsilon_{\theta\theta}$ = tangential strain = $\frac{\bar{u}}{r}$,

$\bar{u}$ = radial displacement,

and

$\bar{w}$ = axial displacement = $z^0_{I,J} - z_{I,J}$.

The radial displacement of the plate is assumed to have the form[10]

$$\bar{u} = r(b - r)\left(1.185\,\frac{\bar{w}_0^2}{b^3} - 1.75r\,\frac{\bar{w}_0^4}{b^4}\right), \tag{A.6}$$

where

b = radius of the plate,

and

$\bar{w}_0$ = axial displacement at the center of the plate.

Equation A.5 assumes that the material behaves elastically. If the plate is stressed beyond the elastic limit, we assume that the Mises yield criterion applies. Therefore,

$$(\sigma_{rr}^2 - \sigma_{rr}\sigma_{\theta\theta} + \sigma_{\theta\theta}^2)^{\frac{1}{2}} = \sigma(\epsilon), \tag{A.7}$$

where $\sigma(\epsilon)$ is the yield stress in tension, a function of the effective strain ϵ. The value of $\sigma(\epsilon)$ can be obtained from the actual stress-strain curve or from the power- or linear-strain hardening approximations. The effective strain ϵ for a Mises material is given by

$$\epsilon = \frac{2}{(3)^{\frac{1}{2}}}(\epsilon_{rr}^2 + \epsilon_{rr}\epsilon_{\theta\theta} + \epsilon_{\theta\theta}^2)^{\frac{1}{2}} \tag{A.8}$$

If the value of

$$(\sigma_{rr}^2 - \sigma_{rr}\sigma_{\theta\theta} + \sigma_{\theta\theta}^2)^{\frac{1}{2}}$$

is greater than $\sigma(\epsilon)$, the stresses computed from Eq. A.5 must be adjusted according to

$$\left.\begin{aligned} \sigma_{rr}(\text{adj}) &= \sigma_{rr}\frac{\sigma(\epsilon)}{(\sigma_{rr}^2 - \sigma_{rr}\sigma_{\theta\theta} + \sigma_{\theta\theta}^2)^{\frac{1}{2}}} \\ \text{and} \qquad & \\ \sigma_{\theta\theta}(\text{adj}) &= \sigma_{\theta\theta}\frac{\sigma(\epsilon)}{(\sigma_{rr}^2 - \sigma_{rr}\sigma_{\theta\theta} + \sigma_{\theta\theta}^2)^{\frac{1}{2}}} \end{aligned}\right\} . \tag{A.9}$$

APPENDIX B

Equation of State for a Group of Mixed Materials

As stated on p. 11, solution of the four hydrodynamic equations requires an equation of state of the form given by Eq. 18. Thus, analysis of the hydrodynamic response of a primary reactor-containment structure to high-energy excursions requires knowledge of the equations of state of the reactor materials.

In the numerical solution, the media under consideration are divided into meshes. For media like sodium, steel, and argon, this presents no problem, because there is only one material. Also, when divided, each mesh contains only one material. Experimental Hugoniot data for individual materials are available in the literature. They can be found, for example, in Ref. 6. (Note: Although the referenced experiments were performed under conditions different from a reactor environment, the results can be readily modified for use in excursion calculations.)

In contrast, division of media like the core blanket and plenum into meshes presents a problem because these media comprise several materials. Also, the meshes usually are not sufficiently fine to confine one material to each mesh zone. Consequently, a Hugoniot curve is needed that will describe the pressure-volume relations for a mixture of materials. Unfortunately, none of the existing experimental data are available for such mixtures; therefore it is necessary to develop a method of constructing a Hugoniot curve from available Hugoniot data for individual materials.

The method used in the present analysis is similar to that employed by Goranson et al.[11] in determining the dynamic compressibility of metals. The basic assumptions are: (1) All components of the mixture within a mesh zone are under the same pressure p_H and at the same temperature; and (2) all components remain intact under pressure.

1. Mixtures Containing Liquids and Solids

It is relatively easy to construct an equation of state for a mixture of several liquids and solids, because the Mie-Grüneisen type of equation of state is applicable to both liquids and solids.

For example, consider a mixture that has n components. The initial volume is

$$\mathcal{V}_0 = \sum_{i=1}^{n} \mathcal{V}_{0i}, \tag{B.1}$$

where $\forall_{0i}$ is the initial volume of the individual component. For a specific value of p_H, the total volume of the mixture becomes

$$\forall = \sum_{i=1}^{n} \forall_i = \sum_{i=1}^{n} \frac{\forall_i}{\forall_{0i}} \forall_{0i} = \sum_{i=1}^{n} \left(\frac{V_i}{V_{0i}}\right)_{p_H} \forall_{0i}, \tag{B.2}$$

where $\forall_i$ is the volume, and $(V_i/V_{0i})_{p_H}$ is the relative specific volume of the individual component. The value of V_i/V_{0i} is determined directly from the Hugoniot curve for the individual element.

Next, the relative compression of mixture is obtained from Eqs. B.1 and B.2:

$$\frac{\forall}{\forall_0} = \sum_{i=1}^{n} \left(\frac{V_i}{V_{0i}}\right)_{p_H} \frac{\forall_{0i}}{\forall_0}, \tag{B.3}$$

where $\forall/\forall_0$ is the relative volume of the mixture at pressure p_H, and $\forall_{0i}/\forall_0$ is the volume fraction of the ith component. A plot of p_H versus $(\forall/\forall_0)_{p_H}$ as determined by Eq. B.3 yields the Hugoniot curve of the mixture of n components.

2. Mixtures Containing Gases, Liquids, and Solids

In mixtures containing gases, liquids, and solids, the process of shock compression is more complicated than in mixtures containing liquids and solids. As a first approximation, the gases can be treated as internal voids; i.e., the mixture is considered a porous material. Therefore, at the beginning of the shock compression, the work of the external pressure is used in closing up the voids, in packing the material and reducing it to a standard volume, i.e., a volume that contains no internal voids. The work of the external pressure in compressing the gases is neglected. (For practical purposes, this work is small and can be taken equal to zero.)

For example, let V_0 be the standard specific volume of the mixture, and V_{00} the specific volume of the porous mixture. Then construct a p_H (V, V_0) curve for the continuous mixture (i.e., a mixture with no internal voids), using the method outlined in Section 1 above. Next, with the assumption that the Mie-Grüneisen's coefficient γ is a function of volume, the Hugoniot curve of the porous mixture can be approximated by

$$p_H(V, V_{00}) = p_H(V, V_0) + \frac{\gamma}{V} [E(V, V_{00}) - E(V, V_0)], \tag{B.4}$$

where

$$\left.\begin{aligned} E(V,V_{00}) &= \tfrac{1}{2}\, p_H(V,V_{00})(V_{00} - V) \\ \text{and} \qquad & \\ E(V,V_0) &= \tfrac{1}{2}\, p_H(V,V_0)(V_0 - V) \end{aligned}\right\} \tag{B.5}$$

are the internal energy of the porous and continuous mixture, respectively.

Finally, substitute Eqs. B.5 into Eq. B.4 and simplify to obtain

$$p_H(V,V_{00}) = \left[1 + \frac{1 + \dfrac{V_0}{V_{00}}}{\dfrac{V_0}{V_{00}} K - 1} \right] p_H(V,V_0), \tag{B.6}$$

where

$$K = \frac{2}{\gamma} + 1.$$

Thus, the Hugoniot plot of a mixture is represented by a straight line on the axis V from V_{00} to V_0, and then a curve from V_0 to V according to Eq. B.6.

APPENDIX C

FORTRAN Program Listing

```
C       MAIN PROGRAM                                                    0001
C       J.GVILDYS-2/27/68                                               0002
C       TWO-DIMENSIONAL REACTOR ACCIDENT ANALYSIS                       0003
        IMPLICIT REAL*8(A-H,O-Z)                                        0004
        COMMON /O/ R(5500),Z(5500),RDOT(5500),ZDOT(5500),RO(5500),ZO(5500)  0005
       1,MZERO(5500),E(5500),P(5500),RHO(5500),VP(5500),SC(5500),KTX(5500)  0006
       2,KTY(5500),KMX(5500),SSR(5500),SSZ(5500),SST(5500)              0007
        COMMON IMAX,JMAX,IMAX1,JMAX1,IMAX2,JMAX2,IMAX3,JMAX3,NCYCL,     0008
       1IW,JW,ISTOP,IQQ,JQQ,IDM,JDM,KB1,KB2,KB3,KPP,KPP1,KPP2,KPPX,KPPC,  0009
       2KPL1,KPL2,KB11,KB12,KB31,KB32,NPP,NCL,KXP(20),KYP(20),KXYP(1000)  0010
       3,KB21,KB22,M11,M12,M21,M22,M31,M32,KNX(10),IOUT                 0011
       4,MI(10),MJ(10),MTH(10),MTV(10),MTC(10),MTX(1000),MTP(1000),KNP(10)  0012
        COMMON I11(10),I12(10),I21(10),I22(10),J11(10),J12(10),J21(10),  0013
       1J22(10),I1X,I2X,J1X,J2X,KK(120)                                 0014
        COMMON /A/ DELT,DELTO,TIME,DIST,WMAX,TITLE(20),PP(20,50),VV(20,50)  0015
       1,PO(20),EO(20),GO(20),CO(20),AA(20),BB(20),CC(20),VO(20),CCP(20),  0016
       2CCK(20),CPLG(20),CRHO(20),CE(20),CP(20),CWB(20),CWN(20),CX(50),  0017
       3CXD(50),CV(50),CVD(50),CX1,CX2,CX3,PMASS,EZERO,EB,PLUG,PRS(6,1000)  0018
       4,TME(1000),PLGF(1000),YMX,YMS,YMN,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,  0019
       5RHO1,RHO2,RHO3,ZERM1,ZERM2,ZERM3,XYP(10),XEU(10),TKE,TIE,TM,YMXO  0020
       6,THH(10),THV(10),XME(10),XRHO(10),XPR(10),XSS(10,50),XSR(10,50)  0021
       7,XDBT                                                           0022
        REAL*4 SSR,SSZ,SST,VP,RO,ZO,SC,RDOT,ZDOT,MZERO                  0023
        REAL*4 TME,PLGF,PRS,TITLE,TIN                                   0024
        INTEGER*2 KTX,KTY,KMX                                           0025
C       FORMATS FOR MAIN PROGRAM                                        0026
  500   FORMAT(18A4)                                                    0027
  502   FORMAT(5I6,3F12.0)                                              0028
  503   FORMAT(6F12.0)                                                  0029
  504   FORMAT(12I6)                                                    0030
  506   FORMAT(1H1,20X,18A4)                                            0031
  508   FORMAT(1H0,' NO OF R ZONES =',I3,' NO OF Z ZONES =',I3,         0032
       1' INITIAL D-TIME =',E15.7)                                      0033
  510   FORMAT(1H0,' LIMITING CONSTANTS'/1H0,' MAX CYCLES =',I5,        0034
       1' MAX TIME =',E15.7,' MAX DISTORTION =',E15.7,                  0035
       2' MAX D-TIME =',E15.7)                                          0036
  512   FORMAT(1H0,' OUTPUT PARAMETERS')                                0037
  514   FORMAT(1H0,' DETAILED FULL ACCURACY PRINTOUT EVERY ',I3,        0038
       1' CYCLE UNTIL',I6,' PRINTOUT')                                  0039
  516   FORMAT(1H0,' LIMITED ACCURACY DISPLAY PRINTOUT OF 2D RESULTS EVERY  0040
       1',I3,' CYCLE')                                                  0041
  518   FORMAT(1H0,' PICTURE DISPLAY EVERY ',I3,' CYCLE')               0042
  520   FORMAT(1H0,' INITIAL WMAX=',E15.7,' D-TIME=',E15.7)             0043
  522   FORMAT(1H0,' CHANGED WMAX=',E15.7,' D-TIME=',E15.7)             0044
  524   FORMAT(1H0,' AT CYCLE',I5,' TIME=',E15.7,' D-TIME=',E15.7,      0045
       1' DISTORT=',E15.7,' AT ZONE ',2I4,/'  WMAX=',E15.7,' AT ZONE',2I4)  0046
  526   FORMAT(////,'   HYDRODYNAMICS ERROR STOP')                      0047
  528   FORMAT(1H0,' CALCULATED WMAX=',E15.7,' D-TIME=',E15.7)          0048
  530   FORMAT(1H0,' ADJUSTED   WMAX=',E15.7,' D-TIME=',E15.7)          0049
  532   FORMAT(1H0,' STOP  WMAX GREATER THAN 0.14')                     0050
  534   FORMAT(1H1,5HCYCLE,5X,4HTIME,7X,10HPLUG FORCE,2X,12HPRESSURE AT ,  0051
       12I3,5(8X,2I3))                                                  0052
  536   FORMAT(I6,2E14.6,4X,6E14.6)                                     0053
        DO 20 I=1,5500                                                  0054
        R(I)=0.                                                         0055
        Z(I)=0.                                                         0056
        RDOT(I)=0.                                                      0057
        ZDOT(I)=0.                                                      0058
        RO(I)=0.                                                        0059
        ZO(I)=0.                                                        0060
        MZERO(I)=0.                                                     0061
        E(I)=0.                                                         0062
```

```
      P(I)=0.
      RHO(I)=0.
      VP(I)=0.
      SC(I)=0.
      SSR(I)=0.
      SSZ(I)=0.
      SST(I)=0.
      KTX(I)=0
      KTY(I)=0
      KMX(I)=0
   20 CONTINUE
      DO 24 I=1,10
      MI(I)=0
      MJ(I)=0
      MTH(I)=0
      MTV(I)=0
      MTC(I)=0
      KNX(I)=0
      KNP(I)=0
      THH(I)=0.
      THV(I)=0.
      XME(I)=0.
      XPP(I)=0.
      XRHO(I)=0.
      DO 24 J=1,50
      XSS(I,J)=0.
      XSR(I,J)=0.
   24 CONTINUE
      DO 25 I=1,20
      PO(I)=0.
      EO(I)=0.
      GO(I)=0.
      CO(I)=0.
      AA(I)=0.
      BB(I)=0.
      CC(I)=0.
      VO(I)=0.
      CCP(I)=0.
      CCK(I)=0.
      CPLG(I)=0.0
      CWB(I)=0.
      CWN(I)=0.
      KXP(I)=0
      KYP(I)=0
      DO 25 J=1,50
      PP(I,J)=0.
      VV(I,J)=0.
   25 CONTINUE
      DO 26 J=1,50
      CX(J)=0.
      CV(J)=0.
      CXD(J)=0.
      CVD(J)=0.
   26 CONTINUE
      DO 27 I=1,1000
      MTX(I)=0
      MTP(I)=0
      KXYP(I)=0
      TME(I)=0.
      PLGF(I)=0.
      DO 27 J=1,6
      PRS(J,I)=0.
   27 CONTINUE
      READ 500,(TITLE(I),I=1,18)
C     INITIAL CONFIGURATION
      READ 502,IMAX,JMAX,KB1,KB2,KB3,TIME,DELT,DELTM
C     LIMITING CRITERIA CONSTANTS
```

```
      READ 503,CYCLM,TMAX,DISTM,DE1,DE2
      TIN=TIME
      IF(CYCLM)60,40,60
   40 CYCLM=10000.
   60 IF(TMAX)100,80,100
   80 TMAX=10000.
  100 IF(DISTM)120,110,120
  110 DISTM=10000.
C     OUTPUT CONSTANTS
  120 READ 504,IOUA,INUMBA,IOUB,IOUC,IOUT
      JQQ=IOUB
      MCYCL=CYCLM
      CX1=1.
      CX2=1.
      CX3=1.
      IF(KB1.EQ.0)GO TO 122
      CX1=-1.
  122 IF(KB2.EQ.0)GO TO 124
      CX2=-1.
  124 IF(KB3.EQ.0)GO TO 126
      CX3=-1.
  126 CONTINUE
      IF(DE1.NE.0.0)GO TO 128
      DE1=0.001
      DE2=0.005
  128 CONTINUE
      PRINT 506,(TITLE(I),I=1,18)
      PRINT 508,IMAX,JMAX,DELT
      PRINT 510,MCYCL,TMAX,DISTM,DELTM
      PRINT 512
      IF(IOUA)140,140,130
  130 PRINT 514,IOUA,INUMBA
      INUMBA=INUMBA-1
  140 IF(IOUB)160,160,150
  150 PRINT 516,IOUB
  160 IF(IOUC)180,180,170
  170 PRINT 518,IOUC
  180 CONTINUE
      IMAX1=IMAX+1
      IMAX2=IMAX+2
      IMAX3=IMAX+3
      JMAX1=JMAX+1
      JMAX2=JMAX+2
      JMAX3=JMAX+3
      IP1=2
      IP2=26
      IX=IMAX /25
      IF(IX)186,186,182
  182 DO 184 I=1,IX
      I11(I)=IP1
      I12(I)=IP2
      I21(I)=IP1
      I22(I)=IP2
      IP1=IP1+25
      IP2=IP2+25
  184 CONTINUE
  186 IXX=IMAX -IX*25
      IF(IXX)190,190,188
  188 I1X=IX+1
      I2X=I1X
      I11(I1X)=IP1
      I12(I1X)=IMAX1
      I21(I2X)=IP1
      I22(I2X)=IMAX2
      GO TO 192
  190 I1X=IX
      I2X=IX+1
```

```
      I21(I2X)=IP1
      I22(I2X)=IMAX2
  192 JP1=2
      JP2=51
      JX=JMAX /50
      IF(JX)198,198,194
  194 DO 196 I=1,JX
      J11(I)=JP1
      J12(I)=JP2
      J21(I)=JP1
      J22(I)=JP2
      JP1=JP1+50
      JP2=JP2+50
  196 CONTINUE
  198 JXX=JMAX -JX*50
      IF(JXX)202,202,200
  200 J1X=JX+1
      J2X=J1X
      J11(J1X)=JP1
      J12(J1X)=JMAX1
      J21(J2X)=JP1
      J22(J2X)=JMAX2
      GO TO 204
  202 J1X=JX
      J2X=JX+1
      J21(J2X)=JP1
      J22(J2X)=JMAX2
  204 CONTINUE
      DO 208 I=1,120
      KK(I)=I
  208 CONTINUE
      NCL=0
      ITIMA=0
      ISTOP=0
      NCYCL=0
      IQO=0
      LAST=-1
      IF(IOUT-1)215,215,210
  210 CALL OTAPE (R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,SST,
     1KTX,KTY,KMX)
      TIN=TIME
      GO TO 470
  215 CONTINUE
      CALL HYDRO1(R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,SST,
     1KTX,KTY,KMX)
      IF(IOUC.EQ.0)GO TO 220
      CALL PICT(R,Z,P,LAST)
  220 INDEXA=1
      IF(IOUA)240,240,650
  240 IF(IOUB)260,260,700
  260 PRINT 520,WMAX,DELT
  280 IF(WMAX.LT.0.14)GO TO 300
      DELT=0.5*DELT
      WMAX=0.25*WMAX
      PRINT 522,WMAX,DELT
      GO TO 280
  300 IF(WMAX.GT.0.035)GO TO 350
      IF(DELT.GT.0.5*DELTM)GO TO 350
      DELT=2.0*DELT
      WMAX=4.0*WMAX
      PRINT 522,WMAX,DELT
      GO TO 300
  350 CALL HYDRO1(R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,SST,
     1KTX,KTY,KMX)
      CALL HYDRO2(R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,SST,
     1KTX,KTY,KMX)
      IF(DABS(EZERO-EB)/EZERO.LT.DE1)GO TO 360
      IOUA=1
      INUMBA=MCYCL
```

```
      IF(DABS(EZERO-EB)/EZERO.GT.DE2)ISTOP=1
  360 CONTINUE
      NCYCL=NCYCL+1
      PRINT 524,NCYCL,TIME,DELT,DIST,IDM,JDM,WMAX,IW,JW
      IF(ISTOP.EQ.0)GO TO 370
      GO TO 450
  370 INDEXA=2
      IF(IOUA.EQ.0)GO TO 375
      IF(MOD(NCYCL,IOUA).NE.0)GO TO 375
      IF(ITIMA.LE.INUMBA)GO TO 380
  375 CONTINUE
      INDEXB=0
      GO TO 390
  380 ITIMA=ITIMA+1
      INDEXB=1
      GO TO 650
  390 IF(IOUB.EQ.0)GO TO 400
      IF(MOD(NCYCL,IOUB).EQ.0)GO TO 700
  400 IF(IOUC.EQ.0)GO TO 405
      IF(MOD(NCYCL,IOUC).EQ.0)
     1CALL PICT(R,Z,P,LAST)
  405 CONTINUE
  410 IF(NCYCL-MCYCL)420,450,450
  420 IF(TIME  - TMAX)430,450,450
  430 IF(DIST  - DISTM)470,450,450
  450 LAST=1
      IF(IOUB.EQ.0)GO TO 455
      INDEXA=3
      GO TO 700
  455 CONTINUE
      IF(IOUC.NE.0)
     1CALL PICT(R,Z,P,LAST)
      IF(INDEXB.EQ.1)GO TO 750
      IF(IOUA.EQ.0)GO TO 750
      INDEXA=3
      GO TO 650
  470 CONTINUE
  475 IF(WMAX.GT.20.0)GO TO 490
      IF(WMAX.LT.0.14)GO TO 480
      DELT=0.5*DELT
      WMAX=0.25*WMAX
      PRINT 530,WMAX,DELT
      GO TO 475
  480 IF(WMAX.GT.0.035)GO TO 350
      IF(DELT.GT.(0.5*DELTM))GO TO 350
      DELT=2.0*DELT
      WMAX=4.0*WMAX
      PRINT 530,WMAX,DELT
      GO TO 480
  490 PRINT 532
      GO TO 450
  650 CONTINUE
      CALL PRINTF(R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,SST,
     1KTX,KTY,KMX)
      GO TO (240,390,750),INDEXA
  700 CONTINUE
      CALL PRINTL(R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,SST,
     1KTX,KTY,KMX)
      GO TO (260,400,455),INDEXA
  750 CONTINUE
      IF(NPP.LE.0)GO TO 777
      PRINT 534,(KXP(L),KYP(L),L=1,NPP)
      DO 776 I=1,NCL
      PRINT 536,KXYP(I),TME(I),PLGF(I),(PRS(L,I),L=1,NPP)
  776 CONTINUE
      GO TO 779
  777 CONTINUE
      IF(KPL1.LE.0)GO TO 779
      PRINT 535
```

```
535 FORMAT(1H1,5HCYCLE,5X,4HTIME,7X,10HPLUG FORCE)
    DO 778 I=1,NCL
    PRINT 536,KXYP(I),TME(I),PLGF(I)
778 CONTINUE
779 CONTINUE
    CALL CALC(NCL,NPP,KPL1,KXP,KYP,TME,PLGF,PRS,TITLE,TIN)
    IF(IOUT-1)770,760,770
760 CALL OTAPE (R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,SST,
   1KTX,KTY,KMX)
770 CONTINUE
    STOP
    END
```

```
    SUBROUTINE CALC(NCL,NPP,KPL1,KXP,KYP,TME,PLGF,PRS,TITLE,TIN)
    DIMENSION KXP(20),KYP(20),TME(1000),PLGF(1000),PRS(6,1000),
   1DATA(1000),TITLE(18),CXY(1000)
    DIMENSION U1(2),U2(3),U3(5),U4(5),U6(4),U7(2),U5(2)
    DIMENSION SCTK(10)
    DATA U1/'  TIME  '/,U2/'PLUG FORCE'/,U3/'AXIAL DISPLACEMENT'/,
   1U4/'RADIAL DISPLACEMENT'/,U5/'AT ZONE'/,U6/'AT MESH POINT'/,
   2U7/'PRESSURE'/
501 FORMAT(I6,(6F12.0))
502 FORMAT(I6,7E15.7)
    IF(NPP.NE.0)GO TO 16
    READ 501,KCAL,SCT,SCT1
    PRINT 502,KCAL,SCT,SCT1
    GO TO 18
 16 CONTINUE
    READ 501,KCAL,SCT,SCT1,(SCTK(K),K=1,NPP)
    PRINT 502,KCAL,SCT,SCT1,(SCTK(K),K=1,NPP)
 18 CONTINUE
    IF(KCAL)200,200,20
 20 CALL PLOTS(DATA(1),4000,49)
    CALL PLOT(5.0,0,-3)
    CALL SYMBOL(0,5.0,0.14,TITLE(1),0,72)
    CALL PLOT(11.0,0,-3)
    IF(SCT.LE.0.0)SCT=.0001
    XMAX=(TME(NCL)-TIN)/SCT
    IF(KPL1)60,60,30
 30 CALL AXIS(0,0,U1,-8,XMAX,0,TIN,SCT,10.0)
    CALL SCALE(PLGF,10.0,NCL,1,10.0)
    PMIN=PLGF(NCL+1)
    PDEL=PLGF(NCL+2)
    IF(SCT1.NE.0.0)PDEL=SCT1
    CALL AXIS(0,0,U2,10,10.0,90.0,PMIN,PDEL,10.0)
    X=0.
    Y=(PLGF(1)-PMIN)/PDEL
    IF(Y.GE.10.0)Y=10.0
    CALL PLOT(X,Y,3)
    DO 50 I=1,NCL
    X=(TME(I) -TIN)/SCT
    Y=(PLGF(I)-PMIN)/PDEL
    IF(Y.GE.10.0)Y=10.0
    CALL PLOT(X,Y,2)
 50 CONTINUE
    XNEW=XMAX+2.
    XNEW=XNEW
    CALL PLOT(XNEW,0,-3)
    IF(NPP)180,180,60
 60 DO 150 L=1,NPP
    DO 70 I=1,NCL
    CXY(I)=PRS(L,I)
```

```
 70 CONTINUE
    CALL SCALE(CXY,10.0,NCL,1,10.0)
    FMIN=CXY(NCL+1)
    FDEL=CXY(NCL+2)
    IF(SCTK(L).NE.0.0)FDEL=SCTK(L)
    IF(KXP(L).GT.0)GO TO 80
    CALL SYMBOL(0,5.0,0.14,U4,0.0,19)
 75 CALL SYMBOL(0,4.5,0.14,U6,0.0,13)
    KXPL=IABS(KXP(L))
    KYPL=IABS(KYP(L))
    CALL NUMBER(1.8,4.5,0.14,FLOAT(KXPL),0.0,-1)
    CALL NUMBER(2.4,4.5,0.14,FLOAT(KYPL),0.0,-1)
    GO TO 100
 80 IF(KYP(L).GT.0)GO TO 90
    CALL SYMBOL(0,5.0,0.14,U3,0.0,18)
    GO TO 75
 90 CALL SYMBOL(0,5.0,0.14,U7,0.0,8)
    CALL SYMBOL(0,4.5,0.14,U5,0.0,7)
    CALL NUMBER(1.0,4.5,0.14,FLOAT(KXP(L)),0.0,-1)
    CALL NUMBER(1.5,4.5,0.14,FLOAT(KYP(L)),0.0,-1)
100 CALL PLOT(3.0,0.0,-3)
    CALL AXIS(0,0,U1,-8,XMAX,0,TIN,SCT,10.0)
    CALL AXIS(0,0,2H  ,2,10.0,90.0,FMIN,FDEL,10.0)
    X=0.
    Y=(CXY(1)-FMIN)/FDEL
    IF(Y.GE.10.0)Y=10.0
    CALL PLOT(X,Y,3)
    DO 120 I=1,NCL
    X=(TME(I)-TIN)/SCT
    Y=(CXY(I)-FMIN)/FDEL
    IF(Y.GE.10.0)Y=10.0
    CALL PLOT(X,Y,2)
120 CONTINUE
    XNEW=XMAX+2.
    XNEW=XNEW
    CALL PLOT(XNEW,0,-3)
150 CONTINUE
180 CALL PLOT(0,0,999)
200 CONTINUE
    RETURN
    END
```

```
    SUBROUTINE HYDROI(R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,
   1SST,KTX,KTY,KMX)
    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION RDD(20),ZDD(20)
    DIMENSION KR1(50),KR2(50),KZ1(50),KZ2(50),KTM(50),KT1(20),KT2(20)
    DIMENSION R(IMAX3,JMAX3),Z(IMAX3,JMAX3),RDOT(IMAX3,JMAX3),
   1ZDOT(IMAX3,JMAX3),RO(IMAX3,JMAX3),ZO(IMAX3,JMAX3),MZERO(IMAX3,JMAX
   23),E(IMAX3,JMAX3),P(IMAX3,JMAX3),RHO(IMAX3,JMAX3),VP(IMAX3,JMAX3),
   3SC(IMAX3,JMAX3),KTX(IMAX3,JMAX3),KTY(IMAX3,JMAX3)
   4,KMX(IMAX3,JMAX3)
    DIMENSION SSR(IMAX3,JMAX3),SSZ(IMAX3,JMAX3),SST(IMAX3,JMAX3)
    COMMON IMAX,JMAX,IMAX1,JMAX1,IMAX2,JMAX2,IMAX3,JMAX3,NCYCL,
   1IW,JW,ISTOP,IQQ,JQQ,IDM,JDM,KB1,KB2,KB3,KPP,KPP1,KPP2,KPPX,KPPC,
   2KPL1,KPL2,KB11,KB12,KB31,KB32,NPP,NCL,KXP(20),KYP(20),KXYP(1000)
   3,KR21,KR22,M11,M12,M21,M22,M31,M32,KNX(10),IOUT
   4,MI(10),MJ(10),MTH(10),MTV(10),MTC(10),MTX(1000),MTP(1000),KNP(10)
    COMMON I11(10),I12(10),I21(10),I22(10),J11(10),J12(10),J21(10),
   1J22(10),I1X,I2X,J1X,J2X,KQ(120)
    COMMON /A/ DELT,DELTO,TIME,DIST,WMAX,TITLE(20),PP(20,50),VV(20,50)
   1,PO(20),EO(20),GO(20),CO(20),AA(20),BB(20),CC(20),VO(20),CCP(20),
```

```
      2CCK(20),CPLG(20),CRHO(20),CE(20),CP(20),CWB(20),CWN(20),CX(50),
      3CXD(50),CV(50),CVD(50),CX1,CX2,CX3,PMASS,EZERO,EB,PLUG,PRS(6,1000)
      4,TME(1000),PLGE(1000),YMX,YMS,YMN,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,
      5RHO1,RHO2,RHO3,ZERM1,ZERM2,ZERM3,XYP(10),XEU(10),TKE,TIE,TM,YMXO
      6,THH(10),THV(10),XME(10),XRHO(10),XPR(10),XSS(10,50),XSR(10,50)
      7,XDRT
      REAL*4 SSR,SSZ,SST,VP,RO,ZO,SC,RDOT,ZDOT,MZERO
      REAL*4 TME,PLGE,PRS,TITLE,TIN
      INTEGER*2 KTX,KTY,KMX
      DATA PI/3.1415926536/
  502 FORMAT(8F9.0)
  504 FORMAT(I6)
  506 FORMAT(12I6)
  508 FORMAT(6F9.0,I3)
  510 FORMAT(2I3,7F9.0,2I3)
  512 FORMAT(1H0,' ERROR IN THE INPUT CARDS  ',/'  ZONE('I3,','I3,
     1')-R AND Z ON CARDS ARE ',2F9.3,' V.S. EXPECTED '2F9.3)
  514 FORMAT(1H0,' TOTAL ENERGY AT START=',E15.7)
  516 FORMAT(8F9.0)
  520 FORMAT('0 MATERIAL CONSTANTS')
  522 FORMAT(1H0,I3,6E15.7,I4)
  524 FORMAT(4X,8E15.7)
  526 FORMAT(1H0,3X,5E15.7)
  528 FORMAT(5I6,F12.0,2I6,F12.0)
  529 FORMAT(2I5,7E15.7,2I3)
  534 FORMAT(1H0,' PLUG CONSTANTS ',/5I6,E15.7,/18X,2I6,E15.7)
      PI2=2.*PI
C     READ R AND Z COORDINATES
      READ 502,(R(I,2),I=2,IMAX2)
      READ 502,(Z(2,J),J=2,JMAX2)
      DO 100 J=2,JMAX2
      DO 100 I=2,IMAX2
      R(I,J)=R(I,2)
      Z(I,J)=Z(2,J)
  100 CONTINUE
      READ 504,NSEC
      PRINT 504,NSEC
C     NSEC-NO OF RECTANGULAR SECTIONS IN THE SYSTEM
      DO 120 K=1,NSEC
      READ 507,KR1(K),KR2(K),KZ1(K),KZ2(K),KT1(K),KT2(K),KTM(K),
     1RDD(K),ZDD(K)
  507 FORMAT(7I6,2F9.0)
      PRINT 509,KR1(K),KR2(K),KZ1(K),KZ2(K),KT1(K),KT2(K),KTM(K),
     1RDD(K),ZDD(K)
  509 FORMAT(7I6,2E15.7)
  120 CONTINUE
C     KR1 AND KZ1 ARE INITIAL ZONE NUMBERS,KR2 AND KZ2 ARE FINAL
C     ZONE NUMBERS IN R AND Z DIRECTIONS-KTM IS MATERIAL CARD INDICATOR
      READ 504,NMAT
C     NMAT-NO OF DIFFERENT MATERIAL CARDS
      PRINT 520
      DO 140 K=1,NMAT
      READ 508,AA(K),BB(K),CC(K),CRHO(K),CE(K),CP(K),KKK
      PRINT 522,K,AA(K),BB(K),CC(K),CRHO(K),CE(K),CP(K),KKK
      READ  502,CWN(K),CWB(K),CPLG(K),CCP(K),CCK(K)
      PRINT 526,CWN(K),CWB(K),CPLG(K),CCP(K),CCK(K)
      IF(KKK)140,140,130
  130 READ 516, (PP(K,I),VV(K,I),I=1,KKK)
      PRINT 524,(PP(K,I),VV(K,I),I=1,KKK)
      READ  502,PO(K),ROK,EO(K),GO(K),CO(K)
      PRINT 526,PO(K),ROK,EO(K),GO(K),CO(K)
      VO(K)=1./ROK
  140 CONTINUE
C     CRHO,CE,CP,KT1,KT2 ARH THE INITIAL RHO,E,P,KTX AND KTY VALLIES
      DO 250 K=1,NSEC
      L=KT1(K)
      I1=KR1(K)
```

```
      I2=KR2(K)
      J1=KZ1(K)
      J2=KZ2(K)
      IF(KTM(K))200,200,160
  160 DO 180 J=J1,J2
      DO 180 I=I1,I2
      RHO(I,J)=CRHO(L)
      E(I,J)=CE(L)
      P(I,J)=CP(L)
      RDOT(I,J)=RDD(K)
      ZDOT(I,J)=ZDD(K)
      KTX(I,J)=KT1(K)
      KTY(I,J)=KT2(K)
  180 CONTINUE
      GO TO 250
  200 CONTINUE
C     CARD INPUT FOR THE CORE SECTION
      DO 210 J=J1,J2
      DO 210 I=I1,I2
      READ 510,II,JJ,RC1,ZC1,RDOT(I,J),ZDOT(I,J),RHO(I,J),E(I,J),P(I,J),
     1KTX(I,J),KTY(I,J)
      PRINT 529,II,JJ,RC1,ZC1,RDOT(I,J),ZDOT(I,J),RHO(I,J),E(I,J),P(I,J)
     1,KTX(I,J),KTY(I,J)
      IF(RC1.NE.R(I,J))GO TO 202
      IF(ZC1.EQ.Z(I,J))GO TO 210
  202 CONTINUE
      PRINT 512,II,JJ,RC1,ZC1,R(I,J),Z(I,J)
      STOP
  210 CONTINUE
  250 CONTINUE
      WMAX=0.
      DO 260 I=2,IMAX2
      R(I,1)=R(I,3)
      Z(I,1)=Z(I,2)-(Z(I,3)-Z(I,2))
      R(I,JMAX3)=R(I,JMAX1)
      Z(I,JMAX3)=Z(I,JMAX2)+(Z(I,JMAX2)-Z(I,JMAX1))
  260 CONTINUE
      DO 270 J=1,JMAX3
      R(1,J)=-R(3,J)
      Z(1,J)=Z(3,J)
      R(IMAX3,J)=R(IMAX2,J)+(R(IMAX2,J)-R(IMAX1,J))
      Z(IMAX3,J)=Z(IMAX1,J)
  270 CONTINUE
      DO 275 J=1,JMAX2
      RHO(IMAX2,J)=RHO(IMAX1,J)
  275 CONTINUE
      DO 300 J=2,JMAX1
      DO 300 I=2,IMAX2
      A1=0.5*((Z(I+1,J+1)-Z(I,J))*(R(I+1,J)-R(I,J+1))
     1-(R(I+1,J+1)-R(I,J))*(Z(I+1,J)-Z(I,J+1)))
      IF(I.EQ.IMAX2)GO TO 280
      LX=KTX(I,J)
      W=(CWN(LX)*(P(I,J)+CWB(LX))/(RHO(I,J)*A1))*(DELT/1.2)**2
      IF(W.LE.WMAX)GO TO 280
      WMAX=W
      IW=I
      JW=J
  280 CONTINUE
      RBAR1=0.25*(R(I,J+1)+R(I+1,J+1)+R(I+1,J)+R(I,J))
  300 MZERO(I,J)=A1*RBAR1*RHO(I,J)
      DO 340 I=2,IMAX1
      KTX(I,1)=KTX(I,2)
      KTY(I,1)=KTY(I,2)
      KTX(I,JMAX2)=KTX(I,JMAX1)
      KTY(I,JMAX2)=KTY(I,JMAX1)
      MZERO(I,1)=MZERO(I,2)
      MZERO(I,JMAX2)=MZERO(I,JMAX1)
```

```
340 CONTINUE
    DO 360 J=1,JMAX2
    KTX(1,J)=KTX(2,J)
    KTY(1,J)=KTY(2,J)
    KTX(IMAX2,J)=KTX(IMAX1,J)
    KTY(IMAX2,J)=KTY(IMAX1,J)
    MZERO(1,J)=MZERO(2,J)
360 CONTINUE
    MZERO(IMAX2,JMAX2)=MZERO(IMAX2,JMAX1)
    MZERO(IMAX2,1)=MZERO(IMAX2,2)
    EZERO=0.
    DO 380 J=1,JMAX3
    DO 380 I=1,IMAX3
    IF(I.EQ.1)GO TO 370
    IF(J.EQ.1)GO TO 370
    IF(J.GT.JMAX1)GO TO 370
    IF(I.GT.IMAX1)GO TO 370
    EZERO=EZERO+PI2*MZERO(I,J)*E(I,J)
370 CONTINUE
    RO(I,J)=R(I,J)
    ZO(I,J)=Z(I,J)
380 CONTINUE
    READ 528,KPP,KPP1,KPP2,KPPX,KPPC,PMASS,KPL1,KPL2,PLUG
    PRINT 534,KPP,KPP1,KPP2,KPPX,KPPC,PMASS,KPL1,KPL2,PLUG
    XDBT=0.
    IF(KPPX.LE.0)GO TO 80
    READ  516,(CX(I),CXD(I),I=1,KPPX)
    PRINT 524,(CX(I),CXD(I),I=1,KPPX)
 80 IF(KPPC.LE.0)GO TO 82
    READ 516,(CV(I),CVD(I),I=1,KPPC)
    PRINT 524,(CV(I),CVD(I),I=1,KPPC)
 82 CONTINUE
    READ 538,KB11,KB12,YMX,M11,M12,KB21,KB22,YMS,M21,M22,KB31,KB32,
   1YMN,M31,M32
    PRINT 539,KB11,KB12,YMX,M11,M12,KB21,KB22,YMS,M21,M22,KB31,KB32,
   1YMN,M31,M32
    YMXO=YMX
538 FORMAT(3(2I3,F12.0,2I3))
539 FORMAT(1H0,' MOVING SURFACE CONSTANTS',/3(2I6,E15.7,2I6))
    IF(KB1.NE.2)GO TO 634
    CRP1=CP(M11)
    CRE1=CE(M11)
    RHO1=CRHO(M11)
    IF(KB12.GE.IMAX2)GO TO 630
    RI2=R(KB12+1,JMAX2)
626 A1=(YMX-Z(KB11,JMAX2))*(RI2-R(KB11,JMAX2))
    RBAR1=0.5*(RI2+R(KB11,JMAX2))
    ZERM1=A1*RBAR1*CRHO(M11)
    EZERO=EZERO+PI2*ZERM1*CRE1
    GO TO 634
630 IF(KB12.EQ.IMAX2)GO TO 632
    RI2=YMS
    GO TO 626
632 RI2=R(IMAX2,JMAX2)
    GO TO 626
634 CONTINUE
    IF(KB3.NE.2)GO TO 644
    CRP3=CP(M31)
    CRE3=CE(M31)
    RHO3=CRHO(M31)
    IF(KB32.GE.IMAX2)GO TO 640
    RI2=R(KB32+1,2)
636 A1=(Z(KB31,2)-YMN)*(RI2-R(KB31,2))
    RBAR1=0.5*(RI2+R(KB31,2))
    ZERM3=A1*RBAR1*CRHO(M31)
    EZERO=EZERO+PI2*ZERM3*CRE3
    GO TO 644
640 IF(KB32.EQ.IMAX2)GO TO 642
```

```
      RI2=YMS
      GO TO 636
  642 RI2=R(IMAX2,2)
      GO TO 636
  644 CONTINUE
      IF(KB2.NE.2)GO TO 664
      RHO2=CRHO(M21)
      CRP2=CP(M21)
      CRE2=CE(M21)
      IF(KB21.LT.2)GO TO 652
      ZI1=Z(IMAX2,KB21)
      GO TO 656
  652 ZI1=YMN
  656 IF(KB22.GE.JMAX2)GO TO 660
      ZI2=Z(IMAX2,KB22+1)
  658 A1=(ZI2-ZI1)*(YMS-R(IMAX2,2))
      RBAR1=0.5*(YMS+R(IMAX2,2))
      ZERM2=A1*RBAR1*CRHO(M21)
      EZERO=EZERO+PI2*ZERM2*CRE2
      GO TO 664
  660 IF(KB22.EQ.JMAX2)GO TO 662
      ZI2=YMX
      GO TO 658
  662 ZI2=Z(IMAX2,JMAX2)
      GO TO 658
  664 CONTINUE
   86 CONTINUE
      READ 504,NPP
      PRINT 504,NPP
      IF(NPP.LE.0)GO TO 88
      READ  506,(KXP(L),KYP(L),L=1,NPP)
      PRINT 506,(KXP(L),KYP(L),L=1,NPP)
   88 CONTINUE
      PRINT 514,EZERO
      DELTO=0
      DIST=0.
      IQQ=0
  420 CONTINUE
      READ 506,NTW,NTM
      IF(NTW.EQ.0)GO TO 480
      IX=0
      PRINT 541
  541 FORMAT(' THIN VESSEL CONSTANTS')
      DO 460 L=1,NTW
      READ 542,MI(L),MJ(L),MTH(L),MTV(L),MTC(L),THH(L),THV(L)
      PRINT 543,L,MI(L),MJ(L),MTH(L),MTV(L),MTC(L),THH(L),THV(L)
  542 FORMAT(5I6,2F12.0)
  543 FORMAT(I3,5I6,2E14.6)
      I2=MJ(L)
      I1=MI(L)-1
      DO 430 I=2,I1
      IX=IX+1
      MTX(IX)=L
      KMX(I,I2)=IX
      MTP(IX)=1
  430 CONTINUE
      K=IX
      J1=MJ(L)
      J2=MI(L)
      DO 440 J=J1,JMAX2
      IX=IX+1
      MTX(IX)=L
      KMX(J2,J)=IX
      MTP(IX)=3
  440 CONTINUE
      MTP(K+1)=2
      MTP(IX)=MTC(L)
```

```
460 CONTINUE
    DO 470 K=1,NTM
    READ 544 ,XME(K),XYP(K),XEU(K),XRHO(K),XPR(K),KNP(K)
    PRINT 545,XME(K),XYP(K),XEU(K),XRHO(K),XPR(K),KNP(K)
544 FORMAT(5E12.0,2I6)
545 FORMAT(5E14.6,2I6)
    IF(KNP(K).EQ.0)GO TO 470
    KK=KNP(K)
    READ 516,(XSS(K,I),XSR(K,I),I=1,KK)
    PRINT 524,(XSS(K,I),XSR(K,I),I=1,KK)
470 CONTINUE
480 CONTINUE
    RETURN
    END
```

```
    SUBROUTINE THINV(LT,I,J,RDDOT,ZDDOT,R,Z,RO,ZO,P,KMX,KTX,KK,CONST)
    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSIONR(IMAX3,JMAX3),Z(IMAX3,JMAX3),RO(IMAX3,JMAX3),
   1ZO(IMAX3,JMAX3),P(IMAX3,JMAX3),KMX(IMAX3,JMAX3),KTX(IMAX3,JMAX3)
    COMMON IMAX,JMAX,IMAX1,JMAX1,IMAX2,JMAX2,IMAX3,JMAX3,NCYCL,
   1IW,JW,ISTOP,IQQ,JQQ,IDM,JDM,KB1,KB2,KB3,KPP,KPP1,KPP2,KPPX,KPPC,
   2KPL1,KPL2,KB11,KB12,KB31,KB32,NPP,NCL,KXP(20),KYP(20),KXYP(1000)
   3,KB21,KB22,M11,M12,M21,M22,M31,M32,KNX(10),IOUT
   4,MI(10),MJ(10),MTH(10),MTV(10),MTC(10),MTX(1000),MTP(1000),KNP(10)
    COMMON I11(10),I12(10),I21(10),I22(10),J11(10),J12(10),J21(10),
   1J22(10),I1X,I2X,J1X,J2X,KQ(120)
    COMMON /A/ DELT,DELTO,TIME,DIST,WMAX,TITLE(20),PP(20,50),VV(20,50)
   1,PO(20),EO(20),GO(20),CO(20),AA(20),BB(20),CC(20),VO(20),CCP(20),
   2CCK(20),CPLG(20),CRHO(20),CE(20),CP(20),CWB(20),CWN(20),CX(50),
   3CXD(50),CV(50),CVD(50),CX1,CX2,CX3,PMASS,EZERO,FB,PLUG,PRS(6,1000)
   4,TME(1000),PLGF(1000),YMX,YMS,YMN,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,
   5RHO1,RHO2,RHO3,ZERM1,ZERM2,ZERM3,XYP(10),XEU(10),TKE,TIE,TM,YMXO
   6,THH(10),THV(10),XME(10),XRHO(10),XPR(10),XSS(10,50),XSR(10,50)
   7,XDBT
    REAL*4 SSR,SSZ,SST,VP,RO,ZO,SC,RDOT,ZDOT,MZERO
    INTEGER*2 KTX,KTY,KMX
    REAL*4 TME,PLGF,PRS,TITLE,TIN
502 FORMAT(' STRAIN EXCEEDED ULTIMATE VALUE AT',2I4,' ER=',E14.6,
   1' ET=',E14.6,' EFF. E=',E14.6)
504 FORMAT(' STRAIN EXCEEDED ULTIMATE VALUE AT ',2I4,' STRAIN=',E14.6)
    CM1=0.0
    N=MTX(LT)
    M1=MI(N)
    M2=MJ(N)
    P1=P(I,J)
    P2=P(I-1,J)
    P3=P(I-1,J-1)
    P4=P(I,J-1)
    GO TO (100,200),KK
100 M=MTH(N)
    TH=THH(N)
    DR=(R(I+1,J)-R(I-1,J))/2.
    LX1=KTX(I,J)
    LX2=KTX(I,J-1)
    AR=R(M1,M2)-R(2,M2)
    WC=ZO(I,J)-Z(I,J)
    IF(I.NE.M1)GO TO 102
    IF(LX1.EQ.3)GO TO 101
    IF(LX2.EQ.3)GO TO 101
    MM=MTV(N)
    CL1=0.5*DSQRT((Z(I,J+1)-Z(I,J))**2+(R(I,J+1)-R(I,J))**2)
    CL2=0.5*DSQRT((Z(I,J)-Z(I-1,J))**2+(R(I,J)-R(I-1,J))**2)
```

```
      CM1=CL1*THV(N)*XRHO(MM)+CL2*TH*XRHO(M)
  101 CM=CM1+1./(4.*CONST)
      STRN=DABS(WC/(ZO(I,JMAX2)-ZO(I,J)))
      IF(DABS(STRN).GT.XEU(M))GO TO 103
      KNM=KNX(M)
      II=KNP(M)
      CALL        STRESS(STRN,STSN,XSS,XSR,II,KNM,M,I,J)
      IF(WC.LT.0.0)STSN=-STSN
      ZDDOT2=THV(N)*STSN/CM
      GO TO 180
  103 ZDDOT2=0.0
      PRINT 504,I,J,STRN
      GO TO 180
  102 CONTINUE
      CL1=0.5*DSQRT((Z(I+1,J)-Z(I,J))**2+(R(I+1,J)-R(I,J))**2)
      CL2=0.5*DSQRT((Z(I,J)-Z(I-1,J))**2+(R(I,J)-R(I-1,J))**2)
      IF(LX1.EQ.3)GO TO 104
      IF(LX2.EQ.3)GO TO 104
      CM1=(CL1+CL2)*TH*XRHO(M)
  104 AR=R(M1,M2)-R(2,M2)
      CM=CM1+1./(4.*CONST)
      WO=Z(M1,M2)-Z(2,J)
      U1=R(M1,M2)-RO(M1,M2)
      EE1=U1/(RO(M1,M2)-RO(2,M2))
      CEH=XME(M)/(1.-XPR(M)*XPR(M))
      W2A3=(WO*WO)/(AR**3)
      UC=R(I,J)*(AR-R(I,J))*W2A3*(1.185-1.75*R(I,J)/AR)
      UP1=R(I+1,J)*(AR-R(I+1,J))*W2A3*(1.185-1.75*R(I+1,J)/AR)
      WP1=ZO(I+1,J)-Z(I+1,J)
      IF(I.NE.2)GO TO 108
      UN1=-UP1
      WN1=WP1
      ET=(AR-R(I,J))*W2A3*(1.185-1.75*R(I,J)/AR)
      ER=((UP1-UN1)**2)/(8.*DR*DR)+(UP1-UN1)/(2.*DR)
      ER=ER+EE1
      CSR=CEH*(ER+XPR(M)*ET)
      IF(DABS(CSR).GT.XYP(M))GO TO 106
  105 XNR=CSR*TH
      XNT=0.
      GO TO 109
  106 ERR=1.1547*DABS(ER)
      IF(ERR.GT.XEU(M))GO TO 107
      CALL STRESS(ERR,STTN,XSS,XSR,II,KNM,M,I,J)
      IF(DABS(CSR).LT.STTN)GO TO 105
      XNR=CSR*TH*STTN/CCSS
      XNT=0.
      GO TO 109
  107 XNR=0.
      XNT=0.
      PRINT 502,I,J,ER,ET,ERR
  109 CONTINUE
      D2WR=(WP1-2.*WC+WN1)/(DR*DR)
C     CONTINUE
      ZDDOT2=-D2WR*XNR*(CL1+CL2)/CM
      GO TO 180
  108 UN1=R(I-1,J)*(AR-R(I-1,J))*W2A3*(1.185-1.75*R(I+1,J)/AR)
      WN1=ZO(I-1,J)-Z(I-1,J)
      ET=(UC+EE1)/R(I,J)
  118 ER=(((UP1-UN1)**2)+((WP1-WN1)**2))/(8.*DR*DR)+(UP1-UN1)/(2.*DR)
      ER=ER+EE1
      CSR=CEH*(ER+XPR(M)*ET)
      CST=CEH*(ET+XPR(M)*ER)
      CCSS=DSQRT(CSR*CSR-CSR*CST+CST*CST)
      IF(CCSS.GT.XYP(M))GO TO 121
  120 XNR=CSR*TH
      XNT=CST*TH
      GO TO 122
```

```
121 ERR=1.1547*DSQRT(ER*ER+ER*ET+ET*ET)
    IF(ERR.GT.XEU(M))GO TO 123
    CALL STRESS(ERR,STTN,XSS,XSR,II,KNM,M,I,J)
    IF(CCSS.LT.STTN)GO TO 120
    XNR=CSR*TH*STTN/CCSS
    XNT=CST*TH*STTN/CCSS
    GO TO 122
123 XNR=0.
    XNT=0.
    PRINT 502,I,J,ER,ET,ERR
122 CONTINUE
    DWR=(WP1-WN1)/(2.*DR)
    D2WR=(WP1-2.*WC+WN1)/(DR*DR)
124 ZDDOT2=-(D2WR*XNR+DWR*XNT/R(I,J))*(CL1+CL2)/CM
    GO TO 180
180 ZDDOT1= ((P1-P3)*(R(I,J+1)-R(I,J-1)+R(I-1,J)-R(I+1,J))-(P2-P4)*
   1(R(I,J+1)-R(I,J-1)-R(I-1,J)+R(I+1,J)))/(4.*CM)
    ZDDOT=ZDDOT1+ZDDOT2
    GO TO 300
200 M=MTV(N)
    TH=THV(N)
    LX1=KTX(I,J)
    LX2=KTX(I-1,J)
    AR=RO(M1,M2)-RO(2,M2)
    IF(J.NE.M2)GO TO 202
    IF(LX1.EQ.3)GO TO 201
    IF(LX2.EQ.3)GO TO 201
    MM=MTH(N)
    CL1=0.5*DSQRT((Z(I,J+1)-Z(I,J))**2+(R(I,J+1)-R(I,J))**2)
    CL2=0.5*DSQRT((Z(I,J)-Z(I-1,J))**2+(R(I,J)-R(I-1,J))**2)
    CM1=CL1*TH*XRHO(M)+CL2*THH(N)*XRHO(MM)
201 CM=CM1+1./(4.*CONST)
    U1=R(I,J)-RO(I,J)
    IF(THH(N).EQ.0.0)GO TO 204
    RDDOT2=-(XME(MM )*THH(N)*U1)/(AR*(1.-XPR(MM))*CM)
    GO TO 280
204 II=KNP(M)
    KNM=KNX(M)
    STRN=DABS(U1/RO(I,J))
    IF(STRN.GT.XEU(M))GO TO 226
    CALL STRESS(STRN,STSN,XSS,XSR,II,KNM,M,I,J)
    IF(U1.LT.0.0)STSN=-STSN
    RDDOT2=-TH*STSN*CL1*2./(RO(I,J)*CM)
    GO TO 280
202 CONTINUE
    CL1=0.5*DSQRT((Z(I,J+1)-Z(I,J))**2+(R(I,J+1)-R(I,J))**2)
    CL2=0.5*DSQRT((Z(I,J)-Z(I,J-1))**2+(R(I,J)-R(I,J-1))**2)
    IF(LX1.EQ.3)GO TO 203
    IF(LX2.EQ.3)GO TO 203
    CM1=(CL1+CL2)*TH*XRHO(M)
203 CM=CM1+1./(4.*CONST)
    IF(J.NE.JMAX2)GO TO 208
    IF(MTP(LT).EQ.6)GO TO 208
    RDDOT=0.
    GO TO 300
208 WC=R(I,J)-RO(I,J)
    IF(J.NE.JMAX2)GO TO 212
    WN1=R(I,J-1)-RO(I,J-1)
    WP1=2.*WC-WN1
    GO TO 224
212 WP1=R(I,J+1)-RO(I,J+1)
    WN1=R(I,J-1)-RO(I,J-1)
224 DZ=(Z(I,J+1)-Z(I,J-1))/2.
    STRN=DABS(WC/RO(I,J))
    II=KNP(M)
    KNM=KNX(M)
    IF(DABS(STRN).GT.XEU(M))GO TO 226
```

```
      CALL STRESS(STRN,STSN,XSS,XSR,II,KNM,M,I,J)
      IF(WC.LT.0.0)STSN=-STSN
      RDDOT2=(-TH*STSN/RO(I,J)*(CL1+CL2))/CM
      GO TO 280
  226 RDDOT2=0.0
      PRINT 504,I,J,STRN
      GO TO 280
  280 RDDOT1=-((P1-P3)*(Z(I,J+1)-Z(I,J-1)+Z(I-1,J)-Z(I+1,J))-(P2-P4)*
     1(Z(I,J+1)-Z(I,J-1)+Z(I+1,J)-Z(I-1,J)))/(4.*CM)
      RDDOT=RDDOT1+RDDOT2
  300 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STRESS(STRN,STSN,XSS,XSR,II,KNM,M,I,J)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XSS(10,50),XSR(10,50)
      IF(STRN.GE.XSR(M,1))GO TO 244
      DO 240 L=1,II
      IF(STRN-XSR(M,L))240,248,252
  240 CONTINUE
      STSN=0.
      GO TO 255
  244 STSN=0.0
      GO TO 255
  248 STSN=XSS(M,L)
      GO TO 255
  252 STSN=XSS(M,L-1)+(STRN-XSR(M,L-1))*(XSS(M,L)-XSS(M,L-1))/
     1(XSR(M,L)-XSR(M,L-1))
  255 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HYDRO1(R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,
     1SST,KTX,KTY,KMX)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(IMAX3,JMAX3),Z(IMAX3,JMAX3),RDOT(IMAX3,JMAX3),
     1ZDOT(IMAX3,JMAX3),RO(IMAX3,JMAX3),ZO(IMAX3,JMAX3),MZERO(IMAX3,JMAX
     23),E(IMAX3,JMAX3),P(IMAX3,JMAX3),RHO(IMAX3,JMAX3),VP(IMAX3,JMAX3),
     3SC(IMAX3,JMAX3),KTX(IMAX3,JMAX3),KTY(IMAX3,JMAX3)
     4,KMX(IMAX3,JMAX3)
      DIMENSION SSR(IMAX3,JMAX3),SSZ(IMAX3,JMAX3),SST(IMAX3,JMAX3)
      COMMON IMAX,JMAX,IMAX1,JMAX1,IMAX2,JMAX2,IMAX3,JMAX3,NCYCL,
     1IW,JW,ISTOP,IOQ,JOQ,IDM,JDM,KB1,KB2,KB3,KPP,KPP1,KPP2,KPPX,KPPC,
     2KPL1,KPL2,KB11,KB12,KB31,KB32,NPP,NCL,KXP(20),KYP(20),KXYP(1000)
     3,KB21,KB22,M11,M12,M21,M22,M31,M32,KNX(10),IOUT
     4,MI(10),MJ(10),MTH(10),MTV(10),MTC(10),MTX(1000),MTP(1000),KNP(10)
      COMMON /A/ DELT,DELTO,TIME,DIST,WMAX,TITLE(20),PP(20,50),VV(20,50)
     1,PO(20),EO(20),GO(20),CO(20),AA(20),BB(20),CC(20),VO(20),CCP(20),
     2CCK(20),CPLG(20),CRHO(20),CE(20),CP(20),CWB(20),CWN(20),CX(50),
     3CXO(50),CV(50),CVO(50),CX1,CX2,CX3,PMASS,EZERO,EB,PLUG,PRS(6,1000)
     4,TME(1000),PLGF(1000),YMX,YMS,YMN,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,
     5RHO1,RHO2,RHO3,ZERM1,ZERM2,ZERM3,XYP(10),XEU(10),TKE,TIE,TM,YMXO
     6,THH(10),THV(10),XME(10),XRHO(10),XPR(10),XSS(10,50),XSR(10,50)
     7,XDRT
      REAL*4 SSR,SSZ,SST,VP,RO,ZO,SC,RDOT,ZDOT,MZERO
      REAL*4 TME,PLGF,PRS,TITLE,TIN
```

```
      INTEGER*2 KTX,KTY,KMX
      DATA PI/3.1415926536/
      PIO4=PI/4.
      PI2=PI*2.
      PIO2=PI/2.
      UEP=0.
      TKE=0.
      TIE=0.
      WMAX=0.
      DIST=0.
      DELTR=0.5*(DELTO+DELT)
      TM=TIME+0.5*DELT
      TIME=TIME+DELT
      DELTO=DELT
      DO 40 I=2,IMAX2
      R(I,1)=R(I,3)
      Z(I,1)=Z(I,2)-(Z(I,3)-Z(I,2))
      R(I,JMAX3)=R(I,JMAX1)
      Z(I,JMAX3)=Z(I,JMAX2)+(Z(I,JMAX2)-Z(I,JMAX1))
   40 CONTINUE
      DO 60 J=1,JMAX3
      R(1,J)=-R(3,J)
      Z(1,J)=Z(3,J)
      R(IMAX3,J)=R(IMAX2,J)+(R(IMAX2,J)-R(IMAX1,J))
      Z(IMAX3,J)=Z(IMAX1,J)
   60 CONTINUE
      DO 80 I=2,IMAX1
      RHO(I,1)=RHO(I,2)
      P(I,1)=CX3*P(I,2)
      RHO(I,JMAX2)=RHO(I,JMAX1)
      P(I,JMAX2)=CX1*P(I,JMAX1)
   80 CONTINUE
      IF(KB1.LT.2)GO TO 88
      I11=KB11
      I12=KB12
      IF(KB12.GE.IMAX2)I12=IMAX1
      DO 84 I=I11,I12
      IF(Z(I,JMAX2).LT.YMX)GO TO 81
      IF(Z(I+1,JMAX2).GE.YMX)GO TO 82
   81 CONTINUE
      P(I,JMAX2)=-P(I,JMAX1)+2.*CRP1
      GO TO 84
   82 P(I,JMAX2)=P(I,JMAX1)
   84 CONTINUE
      IF(KB12.GE.IMAX1)GO TO 85
      IL=KB12+1
      DO 86 I=IL,IMAX1
      P(I,JMAX2)=P(I,JMAX1)
   86 CONTINUE
   85 CONTINUE
      IF(KB11.LE.2)GO TO 88
      IL=KB11-1
      DO 87 I=2,IL
      P(I,JMAX2)=P(I,JMAX1)
   87 CONTINUE
   88 CONTINUE
      IF(KB3.LT.2)GO TO 98
      I11=KB31
      I12=KB32
      IF(KB32.GE.IMAX2)I12=IMAX1
      DO 94 I=I11,I12
      IF(Z(I,2).GT.YMN)GO TO 91
      IF(Z(I+1,2).LE.YMN)GO TO 92
   91 CONTINUE
      P(I,1)=-P(I,2)+2.*CRP3
      GO TO 94
   92 P(I,1)=P(I,2)
```

```
 94 CONTINUE
    IF(KB32.GE.IMAX1)GO TO 95
    IL=KB32+1
    DO 96 I=IL,IMAX1
    P(I,1)=P(I,2)
 96 CONTINUE
 95 CONTINUE
    IF(KB31.LE.2)GO TO 98
    IL=KB31-1
    DO 97 I=2,IL
    P(I,1)=P(I,2)
 97 CONTINUE
 98 CONTINUE
    DO 100 J=2,JMAX1
    RHO(1,J)=RHO(2,J)
    P(1,J)=P(2,J)
    RHO(IMAX2,J)=RHO(IMAX1,J)
    P(IMAX2,J)=CX2*P(IMAX1,J)
100 CONTINUE
    P(IMAX2,1)=P(IMAX1,1)
    P(IMAX2,JMAX2)=P(IMAX1,JMAX2)
    P(1,1)=P(2,1)
    P(1,JMAX2)=P(2,JMAX2)
    RHO(IMAX2,1)=RHO(IMAX1,1)
    RHO(IMAX2,JMAX2)=RHO(IMAX1,JMAX2)
    RHO(1,1)=RHO(2,1)
    RHO(1,JMAX2)=RHO(2,JMAX2)
    IF(KB1.NE.0)GO TO 41
    IF(KB2.EQ.0)GO TO 54
    IF(KB2.EQ.2)GO TO 52
 41 IF(KB1.NE.2)GO TO 42
    IF(KB2.EQ.0)GO TO 54
 42 CONTINUE
    IF(KB12.LT.IMAX2)GO TO 52
    IF(KB22.LT.JMAX2)GO TO 54
    IF(R(IMAX2,JMAX2).GE.YMS)GO TO 52
    IF(Z(IMAX2,JMAX2).GE.YMX)GO TO 54
    IF(KB12.GT.IMAX2)GO TO 54
 52 P(IMAX2,JMAX2)=P(IMAX2,JMAX1)
 54 CONTINUE
    IF(KB3.NE.0)GO TO 43
    IF(KB2.EQ.0)GO TO 58
    IF(KB2.EQ.2)GO TO 56
 43 IF(KB3.NE.2)GO TO 44
    IF(KB2.EQ.0)GO TO 58
 44 CONTINUE
    IF(KB32.LT.IMAX2)GO TO 56
    IF(KB21.GT.2)GO TO 58
    IF(R(IMAX2,2     ).GE.YMS)GO TO 56
    IF(Z(IMAX2,2     ).LE.YMN)GO TO 58
    IF(KB32.GT.IMAX2)GO TO 58
 56 P(IMAX2,1     )=P(IMAX2,2)
 58 CONTINUE
    IF(KB2.LT.2)GO TO 78
    I11=KB21
    I12=KB22
    IF(KB21.LE.2)I11=2
    IF(KB22.GE.JMAX2)I12=JMAX1
    DO 64 J=I11,I12
    IF(R(IMAX2,J).LT.YMS)GO TO 61
    IF(R(IMAX2,J+1).GE.YMS)GO TO 62
 61 CONTINUE
    P(IMAX2,J)=-P(IMAX1,J)+2.*CRP2
    GO TO 64
 62 P(IMAX2,J)=P(IMAX1,J)
 64 CONTINUE
    IF(KB22.GE.JMAX1)GO TO 68
```

```
      JL=KB22+1
      DO 66 J=JL,JMAX1
      P(IMAX2,J)=P(IMAX1,J)
   66 CONTINUE
   68 CONTINUE
      IF(KB21.LE.2)GO TO 72
      JL=KB21-1
      DO 70 J=2,JL
      P(IMAX2,J)=P(IMAX1,J)
   70 CONTINUE
   72 CONTINUE
      IF(KB21.GE.2)GO TO 74
      P(IMAX2,1)=P(IMAX2,2)
      GO TO 76
   74 P(IMAX2,1)=P(IMAX1,1)
   76 CONTINUE
      IF(KB22.LE.IMAX2)GO TO 77
      P(IMAX2,JMAX2)=P(IMAX2,JMAX1)
      GO TO 78
   77 P(IMAX2,JMAX2)=P(IMAX1,JMAX2)
   78 CONTINUE
      DO 140 J=2,JMAX2
      DO 140 I=2,IMAX2
      RDDOT=0.
      ZDDOT=0.
  119 CONTINUE
      CTD1=.25*DABS(R(I,J+1)+R(I+1,J+1)+R(I+1,J)+R(I,J))
      CTD2=.25*DABS(R(I-1,J+1)+R(I,J+1)+R(I,J)+R(I-1,J))
      CTD3=.25*DABS(R(I-1,J)+R(I,J)+R(I,J-1)+R(I-1,J-1))
      CTD4=.25*DABS(R(I,J)+R(I+1,J)+R(I+1,J-1)+R(I,J-1))
      CONST=1./(MZERO(I,J)/CTD1+MZERO(I-1,J)/CTD2+MZERO(I-1,J-1)/CTD3
     1+MZERO(I,J-1)/CTD4)
      P1=P(I,J)-P(I-1,J-1)
      P2=P(I-1,J)-P(I,J-1)
      IF(I.EQ.2)GO TO 110
      IF(I.NE.IMAX2)GO TO 108
      IF(KB2.EQ.0)GO TO 110
      IF(KB2.NE.2)GO TO 108
      IF(J.LT.KB21)GO TO 110
      IF(J.GT.(KB22+1))GO TO 110
      IF(R(I,J).LT.YMS)GO TO 108
      RDOT(I,J)=0.0
      GO TO 110
  108 CONTINUE
      IF(KMX(I,J).LE.0)GO TO 109
      LT=KMX(I,J)
      IF(MTP(LT).LT.2)GO TO 106
      CALL        THINV(LT,I,J,RDDOT,ZDDOT,R,Z,RO,ZO,P,KMX,KTX,2 ,CONST)
      GO TO 111
  106 IF(J.EQ.2)GO TO 107
      IF(J.NE.JMAX2)GO TO 109
      P1=P(I,J-1)-P(I-1,J-1)
      P2=-P1
      GO TO 109
  107 CONTINUE
      P1=P(I,J)-P(I-1,J)
      P2=-P1
  109 CONTINUE
      RDDOT=-CONST*(P1*(Z(I,J+1)-Z(I,J-1)+Z(I-1,J)-Z(I+1,J))
     1-P2*(Z(I,J+1)-Z(I,J-1)+Z(I+1,J)-Z(I-1,J)))
  111 CONTINUE
      RDB=RDDOT*DELTB
      IF(DABS(RDB).LT.UEP)GO TO 120
      RDOT(I,J)=RDOT(I,J)+RDB
      GO TO 120
  110 CONTINUE
  120 CONTINUE
```

```
      P1=P(I,J)-P(I-1,J-1)
      P2=P(I-1,J)-P(I,J-1)
      IF(J.NE.JMAX2)GO TO 122
      IF(KB1.EQ.0)GO TO 140
      IF(KB1.NE.2)GO TO 124
      IF(I.LT.KB11)GO TO 140
      IF(I.GT.(KB12+1))GO TO 140
      IF(KPP.NE.0)GO TO 121
      IF(Z(I,J).LT.YMX)GO TO 124
      ZDOT(I,J)=0.
      GO TO 140
  121 IF(I.LT.KPP1)GO TO 126
      IF(I.GT.(KPP2+1))GO TO 126
      IF(Z(I,J).LT.YMX)GO TO 124
      GO TO 140
  126 IF(Z(I,J).LT.YMXO)GO TO 124
      ZDOT(I,J)=0.
      GO TO 140
  122 IF(J.NE.2)GO TO 124
      IF(KB3.EQ.0)GO TO 140
      IF(KB3.NE.2)GO TO 124
      IF(I.LT.KB31)GO TO 140
      IF(I.GT.(KB32+1))GO TO 140
      IF(Z(I,J).GT.YMN)GO TO 124
      ZDOT(I,J)=0.
      GO TO 140
  124 CONTINUE
      IF(KMX(I,J).LE.0)GO TO 125
      LT=KMX(I,J)
      IF(MTP(LT).GT.2)GO TO 123
      CALL        THINV(LT,I,J,RDDOT,ZDDOT,R,Z,RO,ZO,P,KMX,KTX,1 ,CONST)
      GO TO 127
  123 IF(I.NE.IMAX2)GO TO 125
      P1=P(I-1,J)-P(I-1,J-1)
      P2=P1
  125 CONTINUE
      ZDDOT=CONST*(P1*(R(I,J+1)-R(I,J-1)+R(I-1,J)-R(I+1,J))
     2-P2*(R(I,J+1)-R(I,J-1)+R(I+1,J)-R(I-1,J)))
  127 CONTINUE
      ZDB=ZDDOT*DELTB
      IF(DABS(ZDB).LT.UEP)GO TO 140
      ZDOT(I,J)=ZDOT(I,J)+ZDB
  140 CONTINUE
      IF(KPP.EQ.0)GO TO 496
      KPP22=KPP2+1
      TOTP=0.0
      DO 420 I=KPP1,KPP2
      R21=R(I+1,JMAX2)**2-R(I,JMAX2)**2
      IF(KB1.NE.2)GO TO 418
      IF(Z(I,JMAX2).LT.YMX)GO TO 411
      IF(Z(I+1,JMAX2).GE.YMX)GO TO 418
  411 CONTINUE
      TOTP=TOTP+PIO2*CRP1*R21
      GO TO 420
  418 TOTP=TOTP+PIO2*P(I,JMAX1)*R21
  420 CONTINUE
      IF(KB1.EQ.2)GO TO 422
      XZ=Z(KPP1,JMAX2)-ZO(KPP1,JMAX2)
      GO TO 423
  422 XZ=YMX-YMXO
  423 IF(XZ)424,424,426
  424 CXXP=0.
      CXVP=0.
      GO TO 460
  426 XZZ=XDRT
  415 IF(KPPX.LE.0)GO TO 432
      IF(XZ.GE.CXD(1))GO TO 434
```

```
      DO 430 L=1,KPPX
      IF(XZ-CXD(L))430,436,438
  430 CONTINUE
  432 CXXP=0.
      GO TO 440
  434 CXXP=CX(1)
      GO TO 440
  436 CXXP=CX(L)
      GO TO 440
  438 CXXP=CX(L-1)+(XZ -CXD(L-1))*(CX(L)-CX(L-1))/(CXD(L)-CXD(L-1))
  440 CONTINUE
      IF(KPPC.LE.0)GO TO 452
      IF(XZZ.GE.CVD(1))GO TO 454
      DO 450 L=1,KPPC
      IF(XZZ-CVD(L))450,456,458
  450 CONTINUE
  452 CXVP=0.
      GO TO 460
  454 CXVP=CV(1)
      GO TO 460
  456 CXVP=CV(L)
      GO TO 460
  458 CXVP=CV(L-1)+(XZZ-CVD(L-1))*(CV(L)-CV(L-1))/(CVD(L)-CVD(L-1))
  460 XDDOT=(TOTP-CXXP-CXVP)/PMASS-980.7
      XDB=XDDOT*DELTB
      IF(DABS(XDB).LT.UEP)GO TO 496
      XDBT=XDBT+XDB
      IF(KB1.EQ.2)GO TO 485
      ZZZ=Z(KPP1,JMAX2)+XDBT*DELT
      IF(ZZZ-ZO(KPP1,JMAX2))480,480,482
  480 XDBT=0.
      ZZZ=ZO(KPP1,JMAX2)
  482 CONTINUE
      DO 484 I=KPP1,KPP22
      Z(I,JMAX2)=ZZZ
  484 ZDOT(I,JMAX2)=XDBT
      GO TO 496
  485 ZZZ=YMX+XDBT*DELT
      IF(ZZZ-YMXO)486,486,487
  486 XDBT=0.
      ZZZ=YMXO
  487 CONTINUE
      DO 490 I=KPP1,KPP22
      IF(Z(I,JMAX2)-YMX)490,489,489
  489 ZDOT(I,JMAX2)=XDBT
      Z(I,JMAX2)=ZZZ
  490 CONTINUE
      YMX=ZZZ
  496 CONTINUE
      DO 150 J=2,JMAX2
      DO 150 I=2,IMAX2
      R(I,J)=R(I,J)+RDOT(I,J)*DELT
      IF(J.NE.JMAX2)GO TO 146
      IF(KPP.NE.0)GO TO 141
      Z(I,J)=Z(I,J)+ZDOT(I,J)*DELT
      IF(KB1.NE.2)GO TO 148
      IF(Z(I,J).GE.YMX)Z(I,J)=YMX
      GO TO 148
  141 IF(KB1.NE.2)GO TO 148
      IF(I.LT.KB11)GO TO 148
      IF(I.GT.(KB12+1))GO TO 148
      IF(I.LT.KPP1)GO TO 142
      IF(I.GT.(KPP2+1))GO TO 142
      IF(Z(I,J).LT.YMX)GO TO 146
      Z(I,J)=YMX
      GO TO 148
  142 IF(Z(I,J).LT.YMXO)GO TO 146
```

```
      Z(I,J)=YMXO
      GO TO 148
146   Z(I,J)=Z(I,J)+ZDOT(I,J)*DELT
148   CONTINUE
      IF(I.NE.IMAX2)GO TO 147
      IF(KB2.NE.2)GO TO 147
      IF(R(I,J).GE.YMS)R(I,J)=YMS
147   CONTINUE
      IF(J.NE.2)GO TO 149
      IF(KB3.NE.2)GO TO 149
      IF(Z(I,J).LE.YMN)Z(I,J)=YMN
149   CONTINUE
      IF(J.EQ.JMAX2)GO TO 150
      IF(I.EQ.IMAX2)GO TO 150
      TKE=TKE+PIO4*MZERO(I,J)     *(RDOT(I,J)*RDOT(I,J)+ZDOT(I,J)*
     1ZDOT(I,J)+RDOT(I,J+1)*RDOT(I,J+1)+ZDOT(I,J+1)*ZDOT(I,J+1)+
     2RDOT(I+1,J+1)*RDOT(I+1,J+1)+ZDOT(I+1,J+1)*ZDOT(I+1,J+1)+
     3RDOT(I+1,J)*RDOT(I+1,J)+ZDOT(I+1,J)*ZDOT(I+1,J))
150   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE HYDRO2(R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,
     1SST,KTX,KTY,KMX)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(IMAX3,JMAX3),Z(IMAX3,JMAX3),RDOT(IMAX3,JMAX3),
     1ZDOT(IMAX3,JMAX3),RO(IMAX3,JMAX3),ZO(IMAX3,JMAX3),MZERO(IMAX3,JMAX
     23),E(IMAX3,JMAX3),P(IMAX3,JMAX3),RHO(IMAX3,JMAX3),VP(IMAX3,JMAX3),
     3SC(IMAX3,JMAX3),KTX(IMAX3,JMAX3),KTY(IMAX3,JMAX3)
     4,KMX(IMAX3,JMAX3)
      DIMENSION SSR(IMAX3,JMAX3),SSZ(IMAX3,JMAX3),SST(IMAX3,JMAX3)
      COMMON IMAX,JMAX,IMAX1,JMAX1,IMAX2,JMAX2,IMAX3,JMAX3,NCYCL,
     1IW,JW,ISTOP,IQQ,JQQ,IDM,JDM,KB1,KB2,KB3,KPP,KPP1,KPP2,KPPX,KPPC,
     2KPL1,KPL2,KB11,KB12,KB31,KB32,NPP,NCL,KXP(20),KYP(20),KXYP(1000)
     3,KB21,KB22,M11,M12,M21,M22,M31,M32,KNX(10),IOUT
     4,MI(10),MJ(10),MTH(10),MTV(10),MTC(10),MTX(1000),MTP(1000),KNP(10)
      COMMON /A/ DELT,DELTO,TIME,DIST,WMAX,TITLE(20),PP(20,50),VV(20,50)
     1,PO(20),EO(20),GO(20),CO(20),AA(20),BB(20),CC(20),VO(20),CCP(20),
     2CCK(20),CPLG(20),CRHO(20),CE(20),CP(20),CWB(20),CWN(20),CX(50),
     3CXD(50),CV(50),CVD(50),CX1,CX2,CX3,PMASS,EZERO,EB,PLUG,PRS(6,1000)
     4,TME(1000),PLGF(1000),YMX,YMS,YMN,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,
     5RHO1,RHO2,RHO3,ZERM1,ZERM2,ZERM3,XYP(10),XEU(10),TKE,TIE,TM,YMXO
     6,THH(10),THV(10),XME(10),XRHO(10),XPR(10),XSS(10,50),XSR(10,50)
     7,XOBT
      REAL*4 SSR,SSZ,SST,VP,RO,ZO,SC,RDOT,ZDOT,MZERO
      REAL*4 TME,PLGF,PRS,TITLE,TIN
      INTEGER*2 KTX,KTY,KMX
      DATA PI/3.1415926536/
504   FORMAT(' PRESSURE-ENERGY ITERATION HAS NOT CONVERGED FOR POINT',
     12I5)
506   FORMAT(1H0,7X,'TIME   INTERNAL ENERGY KINETIC ENERGY',3X,
     1'TOTAL ENERGY',/4E15.7)
512   FORMAT(5X,' FORCE ON PLUG EXCEEDS ALLOWABLE VALUE ',2E15.7)
513   FORMAT(5X,' ALLOWABLE STRAIN IS EXCEEDED')
514   FORMAT(I6,5E14.6)
536   FORMAT(I5,2E14.6,4X,6E14.6)
541   FORMAT(' PRESSURE ENERGY ITERATION HAS NOT CONVERGED-TOP')
543   FORMAT(' PRESSURE ENERGY ITERATION HAS NOT CONVERGED-BOTTOM')
542   FORMAT(' PRESSURE ENERGY ITERATION HAS NOT CONVERGED-SIDE')
      PIO4=PI/4.
      PI2=PI*2.
      PIO2=PI/2.
```

```
      UEP=0.
      DO 220 J=2,JMAX1
      DO 220 I=2,IMAX2
      DDTK=0.
      LY=KTY(I,J)
      LX=KTX(I,J)
      IF(I.EQ.IMAX2)GO TO 202
      I1=I+1
      J1=J+1
      R1=R(I1,J)-R(I,J)
      Z1=Z(I1,J)-Z(I,J)
      D1=R1*R1+Z1*Z1
      V1=(RDOT(I1,J)-RDOT(I,J))*R1+(ZDOT(I1,J)-ZDOT(I,J))*Z1
      R2=R(I1,J1)-R(I1,J)
      Z2=Z(I1,J1)-Z(I1,J)
      D2=R2*R2+Z2*Z2
      V2=(RDOT(I1,J1)-RDOT(I1,J))*R2+(ZDOT(I1,J1)-ZDOT(I1,J))*Z2
      R3=R(I,J1)-R(I1,J1)
      Z3=Z(I,J1)-Z(I1,J1)
      D3=R3*R3+Z3*Z3
      V3=(RDOT(I,J1)-RDOT(I1,J1))*R3+(ZDOT(I,J1)-ZDOT(I1,J1))*Z3
      R4=R(I,J)-R(I,J1)
      Z4=Z(I,J)-Z(I,J1)
      D4=R4*R4+Z4*Z4
      V4=(RDOT(I,J)-RDOT(I,J1))*R4+(ZDOT(I,J)-ZDOT(I,J1))*Z4
      R5=R(I1,J1)-R(I,J)
      Z5=Z(I1,J1)-Z(I,J)
      D5=R5*R5+Z5*Z5
      X5=RDOT(I1,J1)-RDOT(I,J)
      Y5=ZDOT(I1,J1)-ZDOT(I,J)
      V5=X5*R5+Y5*Z5
      R6=R(I1,J)-R(I,J1)
      Z6=Z(I1,J)-Z(I,J1)
      D6=R6*R6+Z6*Z6
      X6=RDOT(I1,J)-RDOT(I,J1)
      Y6=ZDOT(I1,J)-ZDOT(I,J1)
      V6=X6*R6+Y6*Z6
      DMST=DMAX1(     DMAX1(D1,D3)/DMIN1(D1,D3),DMAX1(D2,D4)/DMIN1(D2,
     1D4),DMAX1(D5,D6)/DMIN1(D5,D6))
      IF(DMST.LE.DIST)GO TO 162
      DIST=DMST
      IDM=I
      JDM=J
  162 CONTINUE
      AREA=0.5*(Z5*R6-R5*Z6)
      CTRD=0.25*(R(I,J)+R(I,J1)+R(I1,J)+R(I1,J1))
      VOL=AREA*CTRD
      IF(RDOT(I ,J ).NE.0.0)GO TO 165
      IF(RDOT(I1,J ).NE.0.0)GO TO 165
      IF(RDOT(I ,J1).NE.0.0)GO TO 165
      IF(RDOT(I1,J1).NE.0.0)GO TO 165
      IF(ZDOT(I ,J ).NE.0.0)GO TO 165
      IF(ZDOT(I1,J ).NE.0.0)GO TO 165
      IF(ZDOT(I ,J1).NE.0.0)GO TO 165
      IF(ZDOT(I1,J1).NE.0.0)GO TO 165
      DELV=0.0
      RHOT=RHO(I,J)
      PTEMP=P(I,J)
      ETEMP=E(I,J)
      VP1=0.
      GO TO 200
  165 CONTINUE
      RHOT=MZERO(I,J)/VOL
      DELV=1.0/RHOT-1.0/RHO(I,J)
      IF(DELV.GE.0.0)GO TO 170
      VP1=1.44*AREA*CRHO(LX)*RHOT*RHOT*(DELV/DELT)**2
      GO TO 180
  170 VP1=0.
```

```
  180 CONTINUE
      PSTAR=P(I,J)
      IF(CPLG(LX).EQ.0.0)GO TO 186
  182 A2A=AREA+AREA
      DRDR=(X6*Z5-X5*Z6)/A2A
      DZDZ=(Y5*R6-Y6*R5)/A2A
      RDR=DELV*RHOT/DELT-(DRDR+DZDZ)
      SSR(I,J)=SSR(I,J)+DRDR*DELT
      SSZ(I,J)=SSZ(I,J)+DZDZ*DELT
      SST(I,J)=SST(I,J)+ RDR*DELT
      IF(SSR(I,J).GE.CPLG(LX))GO TO 184
      IF(SSZ(I,J).GE.CPLG(LX))GO TO 184
      IF(SST(I,J).GE.CPLG(LX))GO TO 184
      GO TO 186
  184 ISTOP=2
  186 CONTINUE
      GO TO (260,300),LY
  260 RATIO=1./(RHOT*VO(LX))
      IF(RATIO.LE.VV(LX,1))GO TO 278
      DO 270 L=1,50
      IF(VV(LX,L).EQ.0.)GO TO 279
      IF(RATIO-VV(LX,L))290,280,270
  270 CONTINUE
      PH=0.
      GO TO 295
  278 PH=PP(LX,1)
      GO TO 295
  279 L=L-1
  280 PH=PP(LX,L)
      GO TO 295
  290 PH=PP(LX,L-1)+(RATIO-VV(LX,L-1))*(PP(LX,L)-PP(LX,L-1))/(VV(LX,L)-
     1VV(LX,L-1))
  295 CONTINUE
      HP=PH
      PH=PH*1.0E9+PO(LX)
      IF(LX.NE.8)GO TO 296
      IF(HP.LE.136.)GO TO 297
      IF(HP.GE.217.9)GO TO 298
      G=GO(LX)+(CO(LX)-GO(LX))*(HP-136)/81.9
      GO TO 299
  296 CONTINUE
      IF(LX.NE.3)GO TO 297
      IF(HP.LE.131.)GO TO 297
      IF(HP.GE.321.)GO TO 298
      G=GO(LX)+(CO(LX)-GO(LX))*(HP-131.)/190.
      GO TO 299
  297 G=GO(LX)
      GO TO 299
  298 G=CO(LX)
  299 CONTINUE
C     EQUATION OF STATE - SOLIDS AND LIQUIDS
      EH=EO(LX)+0.5*(PH+PO(LX))*(VO(LX)-1./RHOT)
      ETEMP=(E(I,J)+DDTK-0.5*(PH-G*RHOT*EH+VP1+P(I,J))*DELV)/
     1(1.+0.5*G*RHOT*DELV)
      PTEMP=PH+G*RHOT*(ETEMP-EH)+VP1
      GO TO 200
  300 CONTINUE
      DO 190 L=1,20
      ETEMP=E(I,J)+DDTK-0.5*(PSTAR+P(I,J))*DELV
      IF(LX.GT.1)GO TO 344
C     EQUATION OF STATE - CORE
      PTEMP=AA(LX)*DEXP(-BB(LX)/(ETEMP+CC(LX)))
      GO TO 350
  344 CONTINUE
C     EQUATION OF STATE - ARGON
      PTEMP=AA(LX)*ETEMP*RHOT
  350 CONTINUE
```

```
      PTEMP=PTEMP+VP1
      IF(DABS(PTEMP-PSTAR)/DABS(PTEMP+.001).LT.0.001)GO TO 200
  190 PSTAR=PTEMP
      PRINT 504,I,J
      ISTOP=1
      RETURN
  200 TIE=TIE+PI*MZERO(I,J)*(ETEMP+E(I,J))
      W=(CWN(LX)*(PTEMP+CWB(LX))/(RHOT*AREA))*(DELT/1.2)**2+
     14.0*DABS(DELV*RHOT)
      IF(W.LE.WMAX)GO TO 210
      WMAX=W
      IW=I
      JW=J
  210 RHO(I,J)=RHOT
      P(I,J)=PTEMP
      E(I,J)=ETEMP
      VP(I,J)=VP1
  202 IF(KMX(I,J).EQ.0)GO TO 220
      LT=KMX(I,J)
      L=MTX(LT)
      IF(MTP(LT).GT.1)GO TO 218
      IF(THH(L).LE.0.0)GO TO 220
      M1=MI(L)
      M2=MJ(L)
      AR=R(M1,M2)-R(2,M2)
      WO=Z(M1,M2)-Z(2,M2)
      U1=R(M1,M2)-RO(M1,M2)
      CU1=U1/(AR-U1)
      W3=WO*WO/(AR**3)
      IF(I.NE.2)GO TO 204
      UP1=R(I+1,J)*(AR-R(I+1,J))*W3*(1.185-1.75*R(I+1,J)/AR)
      UT1=(AR-R(I,J))*W3*(1.185-1.75*R(I,J)/AR)
      DUP1=UP1/R(I+1,J)
      UP2=R(I+2,J)*(AR-R(I+2,J))*W3*(1.185-1.75*R(I+2,J)/AR)
      UPN=0.0
      UR1=DUR1+DUR1*DUR1/2.
      GO TO 215
  204 IF(I.NE.3)GO TO 206
      UN1=0.0
      GO TO 211
  206 IF(I.NE.(M1-2))GO TO 208
      UP2=CU1
      UN1=R(I-1,J)*(AR-R(I-1,J))*W3*(1.185-1.75*R(I-1,J)/AR)
      GO TO 213
  208 IF(I.NE.(M1-1))GO TO 209
      UP1=CU1
      UN1=R(I-1,J)*(AR-R(I-1,J))*W3*(1.185-1.75*R(I-1,J)/AR)
      UPN=R(I  ,J)*(AR-R(I  ,J))*W3*(1.185-1.75*R(I  ,J)/AR)
      UT2=UP1/R(I+1,J)
      DR2=(R(I+1,J)-R(I,J))/2.
      DUR2=(UP1-UPN)/(2.*DR2)
      UR2=DUR2+DUR2*DUR2/2.
      GO TO 214
  209 UN1=R(I-1,J)*(AR-R(I-1,J))*W3*(1.185-1.75*R(I-1,J)/AR)
  211 UP2=R(I+2,J)*(AR-R(I+2,J))*W3*(1.185-1.75*R(I+2,J)/AR)
  213 UPN=R(I  ,J)*(AR-R(I  ,J))*W3*(1.185-1.75*R(I  ,J)/AR)
      UP1=R(I+1,J)*(AR-R(I+1,J))*W3*(1.185-1.75*R(I+1,J)/AR)
  214 DR1=(R(I+1,J)-R(I-1,J))/2.
      DUR1=(UP1-UN1)/(2.*DR1)
      UT1=(UPN+CU1)/R(I,J)
      DWR1=((ZO(I+1,J)-Z(I+1,J))-(ZO(I-1,J)-Z(I-1,J)))/(2.*DR1)
      UR1=DUR1+(DUR1*DUR1+DWR1*DWR1)/2.
      IF(I.EQ.(M1-1))GO TO 216
  215 DR2=(R(I+2,J)-R(I,J))/2.
      DUR2=(UP2-UPN)/(2.*DR2)
      DWR2=((ZO(I+2,J)-Z(I+2,J))-(ZO(I  ,J)-Z(I  ,J)))/(2.*DR2)
      UT2=(UP1+CU1)/R(I+1,J)
```

```
      UR2=DUR2+(DUR2*DUR2+DWR2*DWR2)/2.
  216 SST(I,J)=(UT1+UT2)/2.
      SSR(I,J)=(UR1+UR2)/2.+CU1
      GO TO 220
  218 CONTINUE
      IF(THV(L).LE.0.0)GO TO 220
      SST(I,J)=0.5*((R(I,J)-RO(I,J))/RO(I,J)+(R(I,J+1)-RO(I,J+1))/RO(I,J
     1+1))
  220 CONTINUE
      TPL=0.0
      IF(KPL1.EQ.0)GO TO 240
      DO 232 I=KPL1,KPL2
      IF(KR1.NE.2)GO TO 230
      IF(Z(I,JMAX2).LT.YMX)GO TO 221
      IF(Z(I+1,JMAX2).GE.YMX)GO TO 230
  221 CONTINUE
      GO TO 232
  230 TPL=TPL+PI02*P(I,JMAX1)*(R(I+1,JMAX2)**2-R(I,JMAX2)**2)
  232 CONTINUE
      IF(PLUG.LE.0.0)GO TO 240
      IF(TPL.LE.PLUG)GO TO 240
      PRINT 512,TPL,PLUG
      ISTOP=1
  240 CONTINUE
      IF(ISTOP.NE.2)GO TO 245
      PRINT 513
      ISTOP=1
  245 CONTINUE
      IF(KR1.NE.2)GO TO 618
      IF(ZERM1.EQ.0)GO TO 618
      IF(KR12.GE.IMAX2)GO TO 612
      I1=KR11
      I2=KR12
  608 VLM1=0.0
      AR1=0.0
      DO 609 J=I1,I2
      IF(ZDOT(J,JMAX2).NE.0.0)GO TO 611
  609 CONTINUE
      GO TO 618
  611 CONTINUE
      DO 610 I=I1,I2
      D1=R(I+1,JMAX2)-R(I,JMAX2)
      Z1=YMX-0.5*(Z(I+1,JMAX2)+Z(I,JMAX2))
      A1=Z1*D1
      AR1=AR1+A1
      CTRD=0.5*(R(I+1,JMAX2)+R(I,JMAX2))
      VLM1=VLM1+A1*CTRD
  610 CONTINUE
      GO TO 614
  612 I1=KR11
      I2=IMAX1
      GO TO 608
  614 IF(KR12.LE.IMAX2)GO TO 616
      D1=YMS-R(IMAX2,JMAX2)
      Z1=YMX-Z(IMAX2,JMAX2)
      A1=Z1*D1
      AR1=AR1+A1
      CTRD=0.5*(YMS+R(IMAX2,JMAX2))
      VLM1=VLM1+A1*CTRD
  616 CONTINUE
      KJJ=1
      RHOT=ZERM1/VLM1
      DELV=1./RHOT-1./RHO1
      VP1=0.
      IF(DELV.GE.0.0)GO TO 620
      VP1=1.44*AR1*CRHO(M11)*RHOT*RHOT*(DELV/DELT)**2
  620 PSTAR=CRP1
```

```
      ACH=AA(M11)
      DDTK=0.0
      CALL EQSR(PTEMP,ETEMP,CRP1,CRE1,DDTK,DELV,VP1,ACH,RHOT,ISTP)
      IF(ISTP.EQ.0)GO TO 622
      PRINT 541
  622 TIE=TIE+PI*ZERM1*(ETEMP+CRE1)
      RHO1=RHOT
      CRP1=PTEMP
      CRE1=ETEMP
      PRINT 514,KJJ,PTEMP,ETEMP,RHOT,VP1,DELV
  618 CONTINUE
      IF(KB3.NE.2)GO TO 638
      IF(ZERM3.EQ.0)GO TO 638
      IF(KB32.GE.IMAX2)GO TO 632
      I1=KB31
      I2=KB32
  628 VLM3=0.0
      DO 629 J=I1,I2
      IF(ZDOT(J,2).NE.0.0)GO TO 631
  629 CONTINUE
      GO TO 638
  631 CONTINUE
      AR3=0.0
      DO 630 I=I1,I2
      D1=R(I+1,2)-R(I,2)
      Z1=0.5*(Z(I+1,2)+Z(I,2))-YMN
      A1=Z1*D1
      AR3=AR3+A1
      CTRD=0.5*(R(I+1,2)+R(I,2))
      VLM3=VLM3+A1*CTRD
  630 CONTINUE
      GO TO 634
  632 I1=KB31
      I2=IMAX1
      GO TO 628
  634 IF(KB32.LE.IMAX2)GO TO 636
      D1=YMS-R(IMAX2,2)
      Z1=Z(IMAX2,2)-YMN
      A1=Z1*D1
      AR3=AR3+A1
      CTRD=0.5*(YMS+R(IMAX2,2))
      VLM3=VLM3+A1*CTRD
  636 CONTINUE
      KJJ=3
      RHOT=ZERM3/VLM3
      DELV=1./RHOT-1./RHO3
      VP1=0.
      IF(DELV.GE.0.0)GO TO 640
      VP1=1.44*AR3*CRHO(M31)*RHOT*RHOT*(DELV/DELT)**2
  640 PSTAR=CRP3
      ACH=AA(M31)
      CALL EQSR(PTEMP,ETEMP,CRP3,CRE3,DDTK,DELV,VP1,ACH,RHOT,ISTP)
      DDTK=0.0
      IF(ISTP.EQ.0)GO TO 642
      PRINT 543
  642 TIE=TIE+PI*ZERM3*(ETEMP+CRE3)
      RHO3=RHOT
      CRP3=PTEMP
      CRE3=ETEMP
      PRINT 514,KJJ,PTEMP,ETEMP,RHOT,VP1,DELV
  638 CONTINUE
      IF(KB2.NE.2)GO TO 658
      IF(ZERM2.EQ.0)GO TO 658
      VLM2=0.0
      AR2=0.0
      IF(KB21.GE.2)GO TO 644
      D1=YMS-R(IMAX2,2)
```

```
      Z1=Z(IMAX2,2)-YMN
      A1=Z1*D1
      AR2=AR2+A1
      CTRD=0.5*(YMS+R(IMAX2,2))
      VLM2=VLM2+A1*CTRD
      I1=2
      GO TO 646
  644 I1=KR21
  646 IF(KR22.GE.JMAX2)GO TO 652
      I2=KR22
  648 CONTINUE
      DO 649 J=I1,I2
      IF(RDOT(IMAX2,J).NE.0.0)GO TO 651
  649 CONTINUE
      GO TO 658
  651 CONTINUE
      DO 650 I=I1,I2
      D1=YMS-0.5*(R(IMAX2,I)+R(IMAX2,I+1))
      Z1=Z(IMAX2,I+1)-Z(IMAX2,I)
      A1=Z1*D1
      AR2=AR2+A1
      CTRD=0.5*(YMS+0.5*(R(IMAX2,I)+R(IMAX2,I+1)))
      VLM2=VLM2+A1*CTRD
  650 CONTINUE
      GO TO 654
  652 I2=JMAX1
      GO TO 648
  654 IF(KR22.LE.JMAX2)GO TO 656
      D1=YMS-R(IMAX2,JMAX2)
      Z1=YMX-Z(IMAX2,JMAX2)
      A1=Z1*D1
      AR2=AR2+A1
      CTRD=0.5*(YMS+R(IMAX2,JMAX2))
      VLM2=VLM2+A1*CTRD
  656 CONTINUE
      KJJ=2
      RHOT=ZERM2/VLM2
      DELV=1./RHOT-1./RHO2
      VP1=0.
      IF(DELV.GE.0.0)GO TO 660
      VP1=1.44*AR2*CRHO(M21)*RHOT*RHOT*(DELV/DELT)**2
  660 PSTAR=CRP2
      ACH=AA(M21)
      CALL EQSB(PTEMP,ETEMP,CRP2,CRE2,DDTK,DELV,VP1,ACH,RHOT,ISTP)
      DDTK=0.0
      IF(ISTP.EQ.0)GO TO 662
      PRINT 542
  662 TIE=TIE+PI*ZERM2*(ETEMP+CRE2)
      RHO2=RHOT
      CRP2=PTEMP
      CRE2=ETEMP
      PRINT 514,KJJ,PTEMP,ETEMP,RHOT,VP1,DELV
  658 CONTINUE
      EB=TIE+TKE
      PRINT 506,TIME,TIE,TKE,EB
      NCL=NCL+1
      NPN=1
      IF(NPP.GT.0)GO TO 243
      IF(KPL1.LE.0)GO TO 249
      GO TO 248
  243 CONTINUE
      IF(NPP.LE.0)GO TO 248
      NPN=NPP
      DO 247 L=1,NPP
      II=IABS(KXP(L))
      JJ=IABS(KYP(L))
      IF(KXP(L).GT.0)GO TO 244
```

```
      PRS(L,NCL)=R(II,JJ)-RO(II,JJ)
      GO TO 247
  244 IF(KYP(L).GT.0)GO TO 246
      PRS(L,NCL)=Z(II,JJ)-ZO(II,JJ)
      GO TO 247
  246 PRS(L,NCL)=P(II,JJ)
  247 CONTINUE
  248 CONTINUE
      PLGF(NCL)=TPL
      TME(NCL)=TIME
      KXYP(NCL)=NCYCL+1
      PRINT 536,KXYP(NCL),TME(NCL),PLGF(NCL),(PRS(L,NCL),L=1,NPN)
  249 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EQS8(PTEMP,ETEMP,PTP,ETP,DDTK,DELV,VP1,ACH,RHOT,ISTP)
      IMPLICIT REAL*8(A-H,O-Z)
      ISTP=0
      PSTAR=PTP
      DO 192 L=1,20
      ETEMP=ETP+DDTK-0.5*(PSTAR+PTP)*DELV
      PTEMP=ACH*ETEMP*RHOT
      PTEMP=PTEMP+VP1
      IF(DABS(PTEMP-PSTAR)/DABS(PTEMP+0.001).LT.0.001)GO TO 195
  192 PSTAR=PTEMP
      ISTP=1
  195 RETURN
      END
```

```
      SUBROUTINE PRINTF(R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,
     1SST,KTX,KTY,KMX)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(IMAX3,JMAX3),Z(IMAX3,JMAX3),RDOT(IMAX3,JMAX3),
     1ZDOT(IMAX3,JMAX3),RO(IMAX3,JMAX3),ZO(IMAX3,JMAX3),MZERO(IMAX3,JMAX
     23),E(IMAX3,JMAX3),P(IMAX3,JMAX3),RHO(IMAX3,JMAX3),VP(IMAX3,JMAX3),
     3SC(IMAX3,JMAX3),KTX(IMAX3,JMAX3),KTY(IMAX3,JMAX3)
     4,KMX(IMAX3,JMAX3)
      DIMENSION SSR(IMAX3,JMAX3),SSZ(IMAX3,JMAX3),SST(IMAX3,JMAX3)
      COMMON IMAX,JMAX,IMAX1,JMAX1,IMAX2,JMAX2,IMAX3,JMAX3,NCYCL,
     1IW,JW,ISTOP,IQQ,JQQ,IDM,JDM,KB1,KB2,KB3,KPP,KPP1,KPP2,KPPX,KPPC,
     2KPL1,KPL2,KB11,KB12,KB31,KB32,NPP,NCL,KXP(20),KYP(20),KXYP(1000)
     3,KB21,KB22,M11,M12,M21,M22,M31,M32,KNX(10),IOUT
     4,MI(10),MJ(10),MTH(10),MTV(10),MTC(10),MTX(1000),MTP(1000),KNP(10)
      COMMON /A/ DELT,DELTO,TIME,DIST,WMAX,TITLE(20),PP(20,50),VV(20,50)
     1,PO(20),FO(20),GO(20),CO(20),AA(20),BB(20),CC(20),VO(20),CCP(20),
     2CCK(20),CPLG(20),CRHO(20),CE(20),CP(20),CWR(20),CWN(20),CX(50),
     3CXD(50),CV(50),CVD(50),CX1,CX2,CX3,PMASS,EZERO,EB,PLUG,PRS(6,1000)
     4,TME(1000),PLGF(1000),YMX,YMS,YMN,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,
     5RHO1,RHO2,RHO3,ZERM1,ZERM2,ZERM3,XYP(10),XEU(10),TKE,TIF,TM,YMXO
     6,THH(10),THV(10),XME(10),XRHO(10),XPR(10),XSS(10,50),XSR(10,50)
     7,XDRT
      REAL*4 SSR,SSZ,SST,VP,RO,ZO,SC,RDOT,ZDOT,MZERO
      REAL*4 TME,PLGF,PRS,TITLE,TIN
      INTEGER*2 KTX,KTY,KMX
  510 FORMAT(1H ,30X,18A4//)
  512 FORMAT(1H0,' CYCLE NO=',I5,' AT TIME ',E15.7)
```

```
514 FORMAT(1H0,'  ZONE',4X,'R',14X,'Z',12X,'RADIAL VEL.',4X,
   1'AXIAL VEL.',5X,'PRESSURE',7X,'ENERGY',9X,'DENSITY',8X,'MASS',
   2  9X,'MT PH',/'   R  Z',/)
516 FORMAT(1H ,2I3,8E15.7,I3,I2)
518 FORMAT(1H0,'  ZONE',4X,'R',14X,'Z',12X,'RADIAL VEL.',4X,
   1'AXIAL VEL.',5X,'PRESSURE',6X,'VISCOUS PRESS.  ENERGY',9X,'DENSITY
   2',6X,'MT PH',/'   R  Z',/)
520 FORMAT(1H1)
522 FORMAT(1H )
    PRINT 520
    PRINT 510,(TITLE(I),I=1,18)
    PRINT 512,NCYCL,TIME
    IF(NCYCL.NE.0)GO TO 30
    PRINT 514
    DO 25 J=2,JMAX2
    DO 20 I=2,IMAX2
    PRINT 516,I,J,R(I,J),Z(I,J),RDOT(I,J),ZDOT(I,J),P(I,J),E(I,J),
   1RHO(I,J),MZERO(I,J),KTX(I,J),KTY(I,J)
 20 CONTINUE
    PRINT 522
 25 CONTINUE
    GO TO 50
 30 PRINT 518
    DO 45 J=2,JMAX2
    DO 40 I=2,IMAX2
    PRINT 516,I,J,R(I,J),Z(I,J),RDOT(I,J),ZDOT(I,J),P(I,J),VP(I,J),
   1E(I,J),RHO(I,J),KTX(I,J),KTY(I,J)
 40 CONTINUE
    PRINT 522
 45 CONTINUE
 50 CONTINUE
    RETURN
    END
```

```
    SUBROUTINE PRINTL(R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,
   1SST,KTX,KTY,KMX)
    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION R(IMAX3,JMAX3),Z(IMAX3,JMAX3),RDOT(IMAX3,JMAX3),
   1ZDOT(IMAX3,JMAX3),RO(IMAX3,JMAX3),ZO(IMAX3,JMAX3),MZERO(IMAX3,JMAX
   23),E(IMAX3,JMAX3),P(IMAX3,JMAX3),RHO(IMAX3,JMAX3),VP(IMAX3,JMAX3),
   3SC(IMAX3,JMAX3),KTX(IMAX3,JMAX3),KTY(IMAX3,JMAX3)
   4,KMX(IMAX3,JMAX3)
    DIMENSION SSR(IMAX3,JMAX3),SSZ(IMAX3,JMAX3),SST(IMAX3,JMAX3)
    DIMENSION U1(12),U2(12),U3(12),U4(12),U5(12),U6(12),U7(12),
   1U8(12),UA(12),UB(12),UC(12),UD(12),UE(12),U(12)
    COMMON IMAX,JMAX,IMAX1,JMAX1,IMAX2,JMAX2,IMAX3,JMAX3,NCYCL,
   1IW,JW,ISTOP,IQQ,JQQ,IDM,JDM,KB1,KB2,KB3,KPP,KPP1,KPP2,KPPX,KPPC,
   2KPL1,KPL2,KB11,KB12,KB31,KB32,NPP,NCL,KXP(20),KYP(20),KXYP(1000)
   3,KB21,KB22,M11,M12,M21,M22,M31,M32,KNX(10),IOUT
   4,MI(10),MJ(10),MTH(10),MTV(10),MTC(10),MTX(1000),MTP(1000),KNP(10)
    COMMON I11(10),I12(10),I21(10),I22(10),J11(10),J12(10),J21(10),
   1J22(10),I1X,I2X,J1X,J2X,KK(120)
    COMMON /A/ DELT,DELTO,TIME,DIST,WMAX,TITLE(20),PP(20,50),VV(20,50)
   1,PO(20),EO(20),GO(20),CO(20),AA(20),BB(20),CC(20),VO(20),CCP(20),
   2CCK(20),CPLG(20),CRHO(20),CE(20),CP(20),CWB(20),CWN(20),CX(50),
   3CXD(50),CV(50),CVD(50),CX1,CX2,CX3,PMASS,EZERO,EB,PLUG,PRS(6,1000)
   4,TME(1000),PLGF(1000),YMX,YMS,YMN,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,
   5RHO1,RHO2,RHO3,ZERM1,ZERM2,ZERM3,XYP(10),XEU(10),TKE,TIE,TM,YMXO
   6,THH(10),THV(10),XME(10),XRHO(10),XPR(10),XSS(10,50),XSR(10,50)
   7,XDBT
    REAL*4 SSR,SSZ,SST,VP,RO,ZO,SC,RDOT,ZDOT,MZERO
    REAL*4 U1,U2,U3,U4,U5,U6,U7,U8,UA,UB,UC,UD,UE,U
    REAL*4 TME,PLGF,PRS,TITLE,TIN
```

```
      REAL*4 RM,ZM,RDOTM,ZDOTM,SMR,SMZ,SMT,PM,EM,VPM,RHOM
      INTEGER*2 KTX,KTY,KMX
      DATA U1 /'     RADIAL POSITION OF GRID POINTS AT TIME ='/,
     1     U2 /'      AXIAL POSITION OF GRID POINTS AT TIME ='/,
     2     U3 /'     RADIAL VELOCITY OF GRID POINTS AT TIME ='/,
     3     U4 /'      AXIAL VELOCITY OF GRID POINTS AT TIME ='/,
     4     U5 /'                  PRESSURE OF ZONES AT TIME ='/,
     5     U6 /'          VISCOUS PRESSURE OF ZONES AT TIME ='/,
     6     U7 /' SPECIFIC INTERNAL ENERGY OF ZONES AT TIME ='/,
     7     U8 /'                   DENSITY OF ZONES AT TIME ='/,
     8     UA /'RADIAL DISPLACEMENT OF GRID POINTS AT TIME ='/,
     9     UB /' AXIAL DISPLACEMENT OF GRID POINTS AT TIME ='/,
     A     UC /'             RADIAL STRAIN OF ZONES AT TIME ='/,
     A     UD /'              AXIAL STRAIN OF ZONES AT TIME ='/,
     A     UE /'            ANGULAR STRAIN OF ZONES AT TIME ='/
      SMR=0.
      SMZ=0.
      SMT=0.
      RM=0.
      ZM=0.
      RDOTM=0
      ZDOTM=0
      PM=0
      VPM=0
      EM=0
      RHOM=0
      DO 80 J=2,JMAX2
      DO 80 I=2,IMAX2
      RM=AMAX1(RM,SNGL(DABS(R(I,J)-RO(I,J))))
      ZM=AMAX1(ZM,SNGL(DABS(Z(I,J)-ZO(I,J))))
      RDOTM=AMAX1(RDOTM,ABS(RDOT(I,J)))
      ZDOTM=AMAX1(ZDOTM,ABS(ZDOT(I,J)))
      SMR=AMAX1(SMR,ABS(SSR(I,J)))
      SMZ=AMAX1(SMZ,ABS(SSZ(I,J)))
      SMT=AMAX1(SMT,ABS(SST(I,J)))
      IF(I.EQ.IMAX2)GO TO 80
      IF(J.EQ.JMAX2)GO TO 80
      PM=AMAX1(PM,SNGL(DABS(P(I,J))))
      EM=AMAX1(EM,SNGL(DABS(E(I,J))))
      VPM=AMAX1(VPM,ABS(VP(I,J)))
      RHOM=AMAX1(RHOM,SNGL(DABS(RHO(I,J))))
   80 CONTINUE
      IF(NCYCL)85,85,88
   85 CONTINUE
      RM=R(IMAX2,2)
      ZM=Z(2,JMAX2)
      DO 101 K=1,12
  101 U(K)=U1(K)
      DO 86 J=2,JMAX2
      DO 86 I=2,IMAX2
      SC(I,J)=R(I,J)
   86 CONTINUE
      CALL DISP(SC,SC,RM,U ,I2X,J2X,I21,I22,J21,J22,KK)
      DO 102 K=1,12
  102 U(K)=U2(K)
      DO 87 J=2,JMAX2
      DO 87 I=2,IMAX2
      SC(I,J)=Z(I,J)
   87 CONTINUE
      CALL DISP(SC,SC,ZM,U ,I2X,J2X,I21,I22,J21,J22,KK)
      GO TO 100
   88 CONTINUE
      DO 90 J=2,JMAX2
      DO 90 I=2,IMAX2
      SC(I,J)=R(I,J)-RO(I,J)
   90 CONTINUE
      DO 109 K=1,12
```

```
109 U(K)=UA(K)
    CALL DISP(SC,SC,RM,U ,I2X,J2X,I21,I22,J21,J22,KK)
    DO 95 J=2,JMAX2
    DO 95 I=2,IMAX2
    SC(I,J)=Z(I,J)-ZO(I,J)
 95 CONTINUE
    DO 110 K=1,12
110 U(K)=UB(K)
    CALL DISP(SC,SC,ZM,U ,I2X,J2X,I21,I22,J21,J22,KK)
100 CONTINUE
    DO 103 K=1,12
103 U(K)=U3(K)
    CALL DISP(SC,RDOT,RDOTM,U ,I2X,J2X,I21,I22,J21,J22,KK)
    DO 104 K=1,12
104 U(K)=U4(K)
    CALL DISP(SC,ZDOT,ZDOTM,U ,I2X,J2X,I21,I22,J21,J22,KK)
    DO 105 K=1,12
105 U(K)=U5(K)
    DO 96 J=2,JMAX2
    DO 96 I=2,IMAX2
    SC(I,J)=P(I,J)
 96 CONTINUE
    CALL DISP(SC,SC,PM,U ,I1X,J1X,I11,I12,J11,J12,KK)
    DO 106 K=1,12
106 U(K)=U6(K)
    CALL DISP(SC,VP,VPM,U ,I1X,J1X,I11,I12,J11,J12,KK)
    DO 107 K=1,12
107 U(K)=U7(K)
    DO 97 J=2,JMAX2
    DO 97 I=2,IMAX2
    SC(I,J)=E(I,J)
 97 CONTINUE
    CALL DISP(SC,SC,EM,U ,I1X,J1X,I11,I12,J11,J12,KK)
    DO 108 K=1,12
108 U(K)=U8(K)
    DO 98 J=2,JMAX2
    DO 98 I=2,IMAX2
    SC(I,J)=RHO(I,J)
 98 CONTINUE
    CALL DISP(SC,SC,RHOM,U ,I1X,J1X,I11,I12,J11,J12,KK)
    DO 111 K=1,12
111 U(K)=UC(K)
    CALL DISP(SC,SSR,SMR,U ,I2X,J2X,I21,I22,J21,J22,KK)
    DO 112 K=1,12
112 U(K)=UD(K)
    CALL DISP(SC,SSZ,SMZ,U ,I2X,J2X,I21,I22,J21,J22,KK)
    DO 113 K=1,12
113 U(K)=UE(K)
    CALL DISP(SC,SST,SMT,U ,I2X,J2X,I21,I22,J21,J22,KK)
    PRINT 530
530 FORMAT(1H1)
    RETURN
    END
```

```
    SUBROUTINE DISP(KC,A,AMAX,U,IX,JX,I1,I2,J1,J2,KK)
    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION KC(IMAX3,JMAX3),A(IMAX3,JMAX3),U(12),I1(10),I2(10),
   1J1(10),J2(10),KK(120)
    COMMON IMAX,JMAX,IMAX1,JMAX1,IMAX2,JMAX2,IMAX3,JMAX3,NCYCL,
   1IW,JW,ISTOP,IQQ,JQQ,IDM,JDM,KB1,KB2,KB3,KPP,KPP1,KPP2,KPPX,KPPC,
   2KPL1,KPL2,KB11,KB12,KB31,KB32,NPP,NCL,KXP(20),KYP(20),KXYP(1000)
   3,KB21,KB22,M11,M12,M21,M22,M31,M32,KNX(10),IOUT
```

```
     4,MI(10),MJ(10),MTH(10),MTV(10),MTC(10),MTX(1000),MTP(1000),KNP(10)
      COMMON /A/ DELT,DELTO,TIME,DIST,WMAX,TITLE(20),PP(20,50),VV(20,50)
     1,PO(20),EO(20),GO(20),CO(20),AA(20),BB(20),CC(20),VO(20),CCP(20),
     2CCK(20),CPLG(20),CRHO(20),CE(20),CP(20),CWB(20),CWN(20),CX(50),
     3CXO(50),CV(50),CVO(50),CX1,CX2,CX3,PMASS,EZERO,EB,PLUG,PRS(6,1000)
     4,TME(1000),PLGE(1000),YMX,YMS,YMN,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,
     5RHO1,RHO2,RHO3,ZERM1,ZERM2,ZERM3,XYP(10),XEU(10),TKE,TIE,TM,YMXO
     6,THH(10),THV(10),XME(10),XRHO(10),XPR(10),XSS(10,50),XSR(10,50)
     7,XDRT
      REAL*4 U
      REAL*4 A,AMAX
      REAL*4 TME,PLGE,PRS,TITLE,TIN
  500 FORMAT(1H1)
  502 FORMAT(25X,18A4)
  504 FORMAT(1H0,5X,11A4,E15.7,' AND D-TIME=',E15.7,' AT CYCLE-',I4)
  508 FORMAT(1H ,5X,'MAXIMUM ABS. VALUE =',E15.7,' TO SMALL-NO PRINTOUT'
     1)
  510 FORMAT(1H ,5X,'MAXIMUM ABS. VALUE =',E15.7,' SCALE FACTOR =',E10.2
     1)
  512 FORMAT(1H0,2X,' R',25I5)
  514 FORMAT(I4,1X,25I5)
  516 FORMAT(3X,'Z')
      CMAX=AMAX
      IF(AMAX.EQ.0)GO TO 240
      I=0
      DO 30 J=1,100
      KAM=AMAX
      IF(KAM.EQ.0)GO TO 40
      AMAX=0.1*AMAX
      I=I+1
   30 CONTINUE
   40 DO 50 J=1,100
      AMAX=10.*AMAX
      KAM=AMAX
      IF(KAM.GT.0)GO TO 60
      I=I-1
   50 CONTINUE
      GO TO 240
   60 IF(AMAX.GT.9.999)I=I+1
      C=10.**(4-I)
      DO 80 J=2,JMAX2
      DO 80 I=2,IMAX2
      KC(I,J)=A(I,J)*C
   80 CONTINUE
      DO 200 I=1,IX
      DO 200 J=1,JX
      PRINT 500
      PRINT 502,(TITLE(K),K=1,18)
      PRINT 504,(U(K),K=1,11),TIME,DELT,NCYCL
      PRINT 510,CMAX,C
      K1=I1(I)
      K2=I2(I)
      PRINT 512,(KK(K),K=K1,K2)
      PRINT 516
      L1=J1(J)
      L2=J2(J)
      DO 140 L=L1,L2
      PRINT 514,L,(KC(K,L),K=K1,K2)
  140 CONTINUE
  200 CONTINUE
      GO TO 300
  240 PRINT 500
      PRINT 502,(TITLE(K),K=1,18)
      PRINT 504,(U(K),K=1,11),TIME,DELT,NCYCL
      PRINT 508,CMAX
  300 RETURN
      END
```

```
      SUBROUTINE PICT(R,Z,P,LAST)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IX1(50),IX2(50),JX1(50),JX2(50)
      DIMENSION R(IMAX3,JMAX3),Z(IMAX3,JMAX3),P(IMAX3,JMAX3)
      DIMENSION LAME(50),LLAME(1),NX1(50)
      DIMENSION AX(20)
      COMMON IMAX,JMAX,IMAX1,JMAX1,IMAX2,JMAX2,IMAX3,JMAX3,NCYCL,
     1IW,JW,ISTOP,IQQ,JQQ,IDM,JDM,KB1,KB2,KB3,KPP,KPP1,KPP2,KPPX,KPPC,
     2KPL1,KPL2,KB11,KB12,KB31,KB32,NPP,NCL,KXP(20),KYP(20),KXYP(1000)
     3,KB21,KB22,M11,M12,M21,M22,M31,M32,KNX(10),IOUT
     4,MI(10),MJ(10),MTH(10),MTV(10),MTC(10),MTX(1000),MTP(1000),KNP(10)
      COMMON I11(10),I12(10),I21(10),I22(10),J11(10),J12(10),J21(10),
     1J22(10),I1X,I2X,J1X,J2X,KK(120)
      COMMON /A/ DELT,DELTO,TIME,DIST,WMAX,TITLE(20),PP(20,50),VV(20,50)
     1,PO(20),EO(20),GO(20),CO(20),AA(20),BB(20),CC(20),VO(20),CCP(20),
     2CCK(20),CPLG(20),CRHO(20),CE(20),CP(20),CWB(20),CWN(20),CX(50),
     3CXD(50),CV(50),CVD(50),CX1,CX2,CX3,PMASS,EZERO,EB,PLUG,PRS(6,1000)
     4,TME(1000),PLGF(1000),YMX,YMS,YMN,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,
     5RHO1,RHO2,RHO3,ZERM1,ZERM2,ZERM3,XYP(10),XEU(10),TKE,TIE,TM,YMXO
     6,THH(10),THV(10),XME(10),XRHO(10),XPR(10),XSS(10,50),XSR(10,50)
     7,XDBT
      REAL*4 SSR,SSZ,SST,VP,RO,ZO,SC,RDOT,ZDOT,MZERO
      REAL*4 TME,PLGF,PRS,TITLE,TIN
      REAL*4 DEL,R1,R2,RR,RR1,RR2,ZZ,ZZ1,ZZ2
      REAL*4 PMAX,PZ,PP1
      REAL*4 AX,TMM
      INTEGER*2 KTX,KTY,KMX
      IF(LAST)20,40,40
   20 CONTINUE
      CALL FINIT(1,2)
      READ 520,N
      PRINT 520,N
      IF(N)22,22,21
   21 READ 520,(IX1(L),IX2(L),JX1(L),JX2(L),L=1,N)
      PRINT 520,(IX1(L),IX2(L),JX1(L),JX2(L),L=1,N)
   22 CONTINUE
  520 FORMAT(12I6)
  530 FORMAT(18A4)
      READ 522,NNM,PMAX
      PRINT 523,NNM,PMAX
  522 FORMAT(I6,F12.0)
      IF(NNM)32,32,30
   30 READ 520,(NX1(L),L=1,NNM)
      PRINT 520,(NX1(L),L=1,NNM)
      DO 31 L=1,NNM
      J=IABS(NX1(L))
      IF(NX1(L))27,31,29
   27 CALL CONVO('("Z=",I2)',LAME(L),0,KRR,J)
      GO TO 31
   29 CALL CONVO('("R=",I2)',LAME(L),0,KRR,J)
   31 CONTINUE
  523 FORMAT(I6,E15.7)
  531 FORMAT(1H0,18A4)
   32 CONTINUE
      NNN=0
      CALL FLIND(1)
      CALL FLINW(2)
      CALL FCHSZ(3)
      CALL FTEXT(TITLE,72,1,100,2000)
      CALL FCHS7(2)
      CALL FMAREA(2200,2200)
      CALL FXYTRN(1000,1000)
      CALL FADV(4)
      R5=R(2,2)
      IF(KB3.EQ.2)GO TO 34
      Z5=Z(2,2)
      GO TO 36
```

```
34 Z5=YMN
36 CONTINUE
   IF(KR2.EQ.2)GO TO 35
   R6=R(IMAX2,JMAX2)
   GO TO 38
35 R6=YMS
38 CONTINUE
   IF(KR1.EQ.2)GO TO 37
   Z6=Z(IMAX2,JMAX2)
   GO TO 39
37 Z6=YMX
39 CONTINUE
   R3=R6-R5
   Z3=Z6-Z5
   IF(R3-Z3)24,24,26
24 R1=0.05*Z3
   R2=Z3+2.*R1
   GO TO 28
26 R1=0.05*R3
   R2=R3+2.*R1
28 CONTINUE
   LAST=0
   R1=-R1
40 CONTINUE
   TMM=TIME
   CALL CONVO('("CYCLE=",I4," TIME=",F11.8)',AX,0,KRR,NCYCL,TMM)
   CALL FDATM(2)
   CALL FXYLIM(R1,R1,R2,R2)
   DO 60 I=2,IMAX2
   I1=I+1
   DO 60 J=2,JMAX2
   J1=J+1
   RR=R(I,J)-R5
   ZZ=Z(I,J)-Z5
   IF(I.EQ.IMAX2)GO TO 50
   RR1=R(I1,J)-R5
   ZZ1=Z(I1,J)-Z5
   CALL FLNSG(RR,ZZ,RR1,ZZ1)
50 IF(J.EQ.JMAX2)GO TO 60
   RR1=R(I,J1)-R5
   ZZ1=Z(I,J1)-Z5
   CALL FLNSG(RR,ZZ,RR1,ZZ1)
60 CONTINUE
   IF(KR1.NE.2)GO TO 61
   RR=R(2,JMAX2)-R5
   ZZ=Z(2,JMAX2)-Z5
   RR1=RR
   ZZ1=YMX-Z5
   CALL FLNSG(RR,ZZ,RR1,ZZ1)
   IF(KR2.EQ.2)GO TO 57
   RR=R(IMAX2,JMAX2)-R5
   ZZ=Z(IMAX2,JMAX2)-Z5
   RR1=RR
   CALL FLNSG(RR,ZZ,RR1,ZZ1)
   RR1=RR
   ZZ=ZZ1
   RR=R(2,JMAX2)-R5
   CALL FLNSG(RR,ZZ,RR1,ZZ1)
   GO TO 61
57 CONTINUE
   RR1=YMS-R5
   ZZ=ZZ1
   CALL FLNSG(RR,ZZ,RR1,ZZ1)
61 IF(KR3.NE.2)GO TO 62
   RR=R(2,2)-R5
   ZZ=Z(2,2)-Z5
   RR1=RR
```

```
      ZZ1=YMN-Z5
      CALL FLNSG(RR,ZZ,RR1,ZZ1)
      IF(KB2.EQ.2)GO TO 58
      RR=R(IMAX2,2)-R5
      ZZ=Z(IMAX2,2)-Z5
      RR1=RR
      CALL FLNSG(RR,ZZ,RR1,ZZ1)
      RR1=RR
      ZZ=ZZ1
      RR=R(2,2)-R5
      CALL FLNSG(RR,ZZ,RR1,ZZ1)
      GO TO 62
   58 CONTINUE
      RR1=YMS-R5
      ZZ=ZZ1
      CALL FLNSG(RR,ZZ,RR1,ZZ1)
   62 IF(KB2.NE.2)GO TO 63
      RR=YMS-R5
      RR1=RR
      ZZ=Z(IMAX2,2)-Z5
      ZZ1=Z(IMAX2,JMAX2)-Z5
      IF(KB1.EQ.2)ZZ1=YMX-Z5
      IF(KB3.EQ.2)ZZ =YMN-Z5
      CALL FLNSG(RR,ZZ,RR1,ZZ1)
      IF(KB1.EQ.2)GO TO 65
      RR=R(IMAX2,JMAX2)-R5
      RR1=YMS-R5
      ZZ =Z(IMAX2,JMAX2)-Z5
      ZZ1=ZZ
      CALL FLNSG(RR,ZZ,RR1,ZZ1)
   65 CONTINUE
      IF(KB3.EQ.2)GO TO 63
      RR=R(IMAX2,2)-R5
      RR1=YMS-R5
      ZZ=Z(IMAX2,2)-Z5
      ZZ1=ZZ
      CALL FLNSG(RR,ZZ,RR1,ZZ1)
   63 CONTINUE
      IF(N)79,79,64
   64 CONTINUE
      DO 78 L=1,N
      IP1=IX1(L)
      IP2=IX2(L)
      JP1=JX1(L)
      JP2=JX2(L)
      IF(IP2-IP1)66,66,72
   66 I=IP1
      KP2=JP2-1
      DO 70 J=JP1,KP2
      J1=J+1
      RR=R(I,J)-R5
      ZZ=Z(I,J)-Z5
      RR1=R(I,J1)-R5
      ZZ1=Z(I,J1)-Z5
      DO 68 K=1,3
      CALL FLNSG(RR,ZZ,RR1,ZZ1)
   68 CONTINUE
   70 CONTINUE
      GO TO 78
   72 J=JP1
      KP2=IP2-1
      DO 76 I=IP1,KP2
      I1=I+1
      RR=R(I,J)-R5
      ZZ=Z(I,J)-Z5
      RR1=R(I1,J)-R5
      ZZ1=Z(I1,J)-Z5
```

```
      DO 74 K=1,3
      CALL FLNSG(RR,ZZ,RR1,ZZ1)
   74 CONTINUE
   76 CONTINUE
   78 CONTINUE
   79 CONTINUE
      CALL FTEXT(AX,27,1,R2*.05,R2*.95)
      CALL FADV(4)
      IF(NNM)220,220,110
  110 RR1=PMAX*1.1
      CALL FXYLIM(R1,R1,R2,RR1)
      DO 200 K=1,NNM
      LLAME(1)=LAME(K)
      CALL FTEXT(AX,27,1,R2*.05,PMAX*1.05)
      CALL FTEXT(LLAME,4,1,R2*.05,PMAX*1.00)
      IF(NX1(K))120,160,160
  120 J=IABS(NX1(K))
      RR=R(2,2)-R5
      ZZ=0.
      RR1=R(IMAX2,2)-R5
      IF(KB2.EQ.2)RR1=YMS-R5
      ZZ1=PMAX
      DO 122 L=1,3
      CALL FLNSG(RR,ZZ,RR1,ZZ)
      CALL FLNSG(RR,ZZ,RR,ZZ1)
  122 CONTINUE
      I1=2
      I2=IMAX
      DO 140 I=I1,I2
      RR=R(I,J)-R5
      PZ=P(I,J)
      IF(PZ.GT.PMAX)PZ=PMAX
      RR1=R(I+1,J)-R5
      PP1=P(I+1,J)
      IF(PP1.GT.PMAX)PP1=PMAX
      CALL FLNSG(RR,PZ,RR1,PP1)
  140 CONTINUE
      GO TO 190
  160 I=NX1(K)
      RR=Z(2,2)-Z5
      IF(KB3.EQ.2)RR =YMN-Z5
      ZZ=0.
      RR1=Z(2,JMAX2)-Z5
      IF(KB1.EQ.2)RR1=YMX-Z5
      ZZ1=PMAX
      DO 162 L=1,3
      CALL FLNSG(RR,ZZ,RR1,ZZ)
      CALL FLNSG(RR,ZZ,RR,ZZ1)
  162 CONTINUE
      J1=2
      J2=JMAX
      DO 180 J=J1,J2
      RR=Z(I,J)-Z5
      PZ=P(I,J)
      IF(PZ.GT.PMAX)PZ=PMAX
      RR1=Z(I,J+1)-Z5
      PP1=P(I,J+1)
      IF(PP1.GT.PMAX)PP1=PMAX
      CALL FLNSG(RR,PZ,RR1,PP1)
  180 CONTINUE
  190 CONTINUE
      CALL FADV(4)
  200 CONTINUE
  220 CONTINUE
      IF(LAST)100,100,80
   80 CONTINUE
      CALL FADV(4)
```

```
      CALL FDATM(3)
      CALL FMAREA(4000,4000)
      CALL FXYTRN(0,0)
      CALL FCHSZ(3)
      CALL FTEXT(TITLE,72,1,100,2000)
      CALL FCHSZ(2)
      CALL FADV(4)
      CALL FEOF
  100 RETURN
      END
```

```
      SUBROUTINE OTAPE (R,Z,RDOT,ZDOT,MZERO,P,VP,E,RHO,RO,ZO,SC,SSR,SSZ,
     1SST,KTX,KTY,KMX)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R(IMAX3,JMAX3),Z(IMAX3,JMAX3),RDOT(IMAX3,JMAX3),
     1ZDOT(IMAX3,JMAX3),RO(IMAX3,JMAX3),ZO(IMAX3,JMAX3),MZERO(IMAX3,JMAX
     23),E(IMAX3,JMAX3),P(IMAX3,JMAX3),RHO(IMAX3,JMAX3),VP(IMAX3,JMAX3),
     3SC(IMAX3,JMAX3),KTX(IMAX3,JMAX3),KTY(IMAX3,JMAX3)
     4,KMX(IMAX3,JMAX3)
      DIMENSION SSR(IMAX3,JMAX3),SSZ(IMAX3,JMAX3),SST(IMAX3,JMAX3)
      COMMON IMAX,JMAX,IMAX1,JMAX1,IMAX2,JMAX2,IMAX3,JMAX3,NCYCL,
     1IW,JW,ISTOP,IQQ,JQQ,IDM,JDM,KB1,KB2,KB3,KPP,KPP1,KPP2,KPPX,KPPC,
     2KPL1,KPL2,KB11,KB12,KB31,KB32,NPP,NCL,KXP(20),KYP(20),KXYP(1000)
     3,KB21,KB22,M11,M12,M21,M22,M31,M32,KNX(10),IOUT
     4,MI(10),MJ(10),MTH(10),MTV(10),MTC(10),MTX(1000),MTP(1000),KNP(10)
      COMMON I11(10),I12(10),I21(10),I22(10),J11(10),J12(10),J21(10),
     1J22(10),I1X,I2X,J1X,J2X,KQ(120)
      COMMON /A/ DELT,DELTO,TIME,DIST,WMAX,TITLE(20),PP(20,50),VV(20,50)
     1,PO(20),EO(20),GO(20),CO(20),AA(20),BB(20),CC(20),VO(20),CCP(20),
     2CCK(20),CPLG(20),CRHO(20),CE(20),CP(20),CWB(20),CWN(20),CX(50),
     3CXD(50),CV(50),CVD(50),CX1,CX2,CX3,PMASS,EZERO,EB,PLUG,PRS(6,1000)
     4,TME(1000),PLGF(1000),YMX,YMS,YMN,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,
     5RHO1,RHO2,RHO3,ZERM1,ZERM2,ZERM3,XYP(10),XEU(10),TKE,TIE,TM,YMXO
     6,THH(10),THV(10),XME(10),XRHO(10),XPR(10),XSS(10,50),XSR(10,50)
     7,XDBT
      REAL*4 SSR,SSZ,SST,VP,RO,ZO,SC,RDOT,ZDOT,MZERO
      REAL*4 TME,PLGF,PRS,TITLE,TIN
      INTEGER*2 KTX,KTY,KMX
      IOUT=IOUT
      IF(IOUT-2)100,200,300
  100 CONTINUE
      REWIND 8
      WRITE(8)NCYCL,IW,JW,KB1,KB2,KB3,IQQ,JQQ,KPP,KPP1,KPP2,KPPX,KPPC
      WRITE(8)KPL1,KPL2,NPP,KB11,KB12,KB21,KB22,KB31,KB32,IDM,JDM
      WRITE (8)M11,M12,M21,M22,M31,M32
      WRITE(8)KXP,KYP,KNX,KNP,MI,MJ,MTH,MTV,MTC
      WRITE(8)MTX,MTP
      WRITE (8)KTX,KTY,KMX
      WRITE(8)TIME,DELT,DIST,DELTO,WMAX,PMASS,YMX,YMS,YMN,YMXO,EZERO
      WRITE(8)PLUG,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,RHO1,RHO2,RHO3,XDBT
      WRITE(8)ZERM1,ZERM2,ZERM3
      WRITE(8)PO,EO,GO,CO,VO,AA,BB,CC
      WRITE(8)CCP,CCK,CRHO,CE,CP,CWB,CWN,CPLG
      WRITE(8)CX,CXD,CV,CVD
      WRITE(8)XSS,XSR
      WRITE(8)PP,VV
      WRITE(8)XYP,XEU,THH,THV,XME,XRHO,XPR
      WRITE(8)SSR,SSZ,SST,VP,SC
      WRITE(8)RO,ZO,RDOT,ZDOT,MZERO
      WRITE(8)R,Z,P,E,RHO
      END FILE 8
      GO TO 450
```

```
200 CONTINUE
    REWIND 9
    READ (9)NCYCL,IW,JW,KB1,KB2,KB3,IQQ,JQQ,KPP,KPP1,KPP2,KPPX,KPPC
    READ (9)KPL1,KPL2,NPP,KB11,KB12,KB21,KB22,KB31,KB32,IDM,JDM
    READ  (9)M11,M12,M21,M22,M31,M32
    READ (9)KXP,KYP,KNX,KNP,MI,MJ,MTH,MTV,MTC
    READ (9)MTX,MTP
    READ  (9)KTX,KTY,KMX
    READ (9)TIME,DELT,DIST,DELTO,WMAX,PMASS,YMX,YMS,YMN,YMXO,EZERO
    READ (9)PLUG,CRP1,CRP2,CRP3,CRE1,CRE2,CRE3,RHO1,RHO2,RHO3,XDBT
    READ (9)ZERM1,ZERM2,ZERM3
    READ (9)PO,EO,GO,CO,VO,AA,BB,CC
    READ (9)CCP,CCK,CRHO,CE,CP,CWB,CWN,CPLG
    READ (9)CX,CXD,CV,CVD
    READ (9)XSS,XSR
    READ (9)PP,VV
    READ (9)XYP,XEU,THH,THV,XME,XRHO,XPR
    READ (9)SSR,SSZ,SST,VP,SC
    READ (9)RO,ZO,RDOT,ZDOT,MZERO
    READ (9)P,Z,P,E,RHO
    GO TO 450
300 IOUT=1
    GO TO 200
450 RETURN
    END
```

REFERENCES

1. H. C. Sorenson and S. H. Fistedis, *Hydrodynamics of a New Concept of Primary Containment by Energy Absorption*, ANL-7214 (Dec 1966).

2. J. von Neumann and R. D. Richtmyer, *A Method for the Numerical Calculation of Hydrodynamic Shocks*, J. Appl. Phys. *21*, No. 3, 232-237 (March 1950).

3. W. B. Goad, *WAT: A Numerical Method for Two-dimensional Unsteady Fluid Flow*, LAMS-2365 (Nov 1960).

4. W. D. Schulz, "Two-dimensional Lagrangian Hydrodynamic Difference Equations," Vol. *3*, Chap. I, *Methods in Computational Physics*, eds. B. Alder, S. Fernback, and M. Rotenberg, Academic Press, Inc., New York (1964).

5. G. T. Richards, *Derivation of a Generalized von Neumann Pseudo-Viscosity with Directional Properties*, UCRL-14244 (July 1965).

6. M. van Thiel, *Compendium of Shock Wave Data*, UCRL-50108 (June 1966).

7. R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial-value Problems*, p. 350, 2nd ed., Interscience Publishers, Inc., New York (1957).

8. L. Amurud and R. S. Orr, *A Note on Inverted Centers of Pressure and Crossed Mass Points in a Two-dimensional Hydrodynamic Calculation*, LASL unpublished note (May 1963).

9. N. Hirakawa, *MARS: A Two-dimensional Excursion Code*, APDA-198 (July 1967).

10. S. Timoshenko and S. Woinowsky-Krieger, *Theory of Plates and Shells*, 2nd ed., pp. 400-401, McGraw-Hill Book Co., Inc., New York (1959).

11. R. W. Goranson, D. Bancroft, B. L. Burton, T. Blechar, E. F. Houston, E. F. Gittings, and S. A. Landeen, *Dynamic Determination of the Compressibility of Metals*, J. Appl. Phys. *26*, 1472-1479 (1955).

ARGONNE NATIONAL LAB WEST
3 4444 00011445 4