LA-8642-M

Manual

2

University of California

# HYDROX:

## A One-Dimensional Lagrangian Hydrodynamics Code

**DO NOT CIRCULATE**

**PERMANENT RETENTION REQUIRED BY CONTRACT**



LOS ALAMOS SCIENTIFIC LABORATORY
Post Office Box 1663 Los Alamos. New Mexico 87545

An Affirmative Action/Equal Opportunity Employer



**UNITED STATES**
**DEPARTMENT OF ENERGY**
**CONTRACT W-7405-ENG. 36**

LA-8642-M
Manual

UC-32
Issued: March 1981

# HYDROX:

## A One-Dimensional Lagrangian

## Hydrodynamics Code

Milton Samuel Shaw
Galen K. Straub

CONTENTS

# HYDROX: A ONE-DIMENSIONAL LAGRANGIAN HYDRODYNAMICS CODE*

by

Milton Samuel Shaw and Galen K. Straub

ABSTRACT

HYDROX is a one-dimensional Lagrangian hydrodynamics computer code written in FORTRAN for the solution of problems with plane, cylindrical, or spherical symmetry. A user may request automatic problem zoning, rezoning, and automatic time step controls. Equation-of-state libraries for HOM and SESAME are available. Input to HYDROX is by way of NAMELIST and output may be sent to several different disk files, including a file that is directly readable by the interactive graphics code GAS. A restart capability is also provided. This document is intended to serve as more than just a manual for problem setup; information has been included on the derivation of and differencing schemes for the equation of motion, detailed notes on each subroutine, sample problems, and HMLB and SESAME equation-of-state libraries.

*Los Alamos National Laboratory Identification No. LP-3052.

## I. INTRODUCTION

HYDROX is a one-dimensional Lagrangian hydrodynamics computer code written in FORTRAN for the solution of problems with plane, cylindrical, or spherical symmetry. The code may be compiled with up to 20,000 spatial cells on the CDC 7600 series computers and a potentially higher number on the CRAY-1. Versions of the code are available on both the above-mentioned machines as well as the VAX-11.*

HYDROX draws heavily upon the features incorporated in the SIN hydrodynamics code,[1] but also includes several automatic features that simplify user interaction. The user may request the following options: automatic problem zoning, rezoning, and automatic time step controls. HYDROX has been written to reference equation-of-state (EOS) libraries for certain EOS types: HOM,[1] the Barnes EOS form,[2] SESAME tables,[3] and reactive equations of state using HOM. Eight-parameter polynomial EOS's are also available to the user.

The features of SIN for treating explosives were directly adapted into HYDROX. These include Arrhenius reaction kinetics, C-J volume burn with buildup, and Forest Fire.[4-7] Material descriptions for plasticity and spallation are also included.

Input to HYDROX is by way of NAMELIST and output is sent to several different disk files. In addition to printer listable files, HYDROX writes a random access data dump file that is directly readable by the interactive computer graphics code GAS.[8] This file may also be read for cell quantity data and additional information processing. Other dump files may also be written for problem restarting.

HYDROX was written to serve the dual purpose of being the core of a production code for engineering design problems and also a research code

*Execution times for the VAX-11 are about seven times slower than the CDC 7600 for single precision (32-bit) arithmetic and ten times slower for double-precision (64-bit) arithmetic.

for the study and modeling of dynamic flow problems. By using the same code for both types of problems, improved physical descriptions that are being developed are most readily available for design studies. For this reason, we have allowed the user to select options such as extremely small spatial zones or time steps to minimize any numerical error in describing the physics of the dynamic flow. When a highly accurate numerical solution is not needed for a particular portion of a calculation, the user may choose a faster option.

The remainder of the introduction contains a table of consistent sets of units for HYDROX and some useful conversion factors. Section II of this manual discusses the equations of motion in plane, cylindrical, and spherical geometries as well as the accuracy of the finite difference equations. Section III contains input and output information. Section IV is composed of a subroutine-by-subroutine description of the physical models represented in the code and a listing of each subroutine annotated for ease of understanding. Section V contains a short selection of sample problems that illustrate procedures for problem setup and output. Section VI discusses the use of the equation-of-state library HMLB for use with the HOM EOS and the SESAME tabular EOS library.

CONSISTENT SETS OF UNITS FOR HYDROX

| | | | |
|---|---|---|---|
| Time | $\mu$s | s | s |
| Length | cm | cm | meter |
| Mass | g | g | kg |
| Density | $g/cm^3$ | $g/cm^3$ | $kg/m^3$ |
| Energy | $10^{12}$ ergs | erg | joule |
| Energy density | Mbar $cm^3/g$ | erg/g | joule/kg |
| Pressure | Mbar | $dyne/cm^2$ | $Newton/m^2$ |

The preferred set of units is in the first column. EOS libraries contain dimensional constants that are consistent only with this set of units. In using any other set of units the user must make sure they are consistent with internal subroutines that also contain dimensional constants.

<u>Useful Conversion Factors</u>

1 kilobar = $10^9$ dynes/cm$^2$

1 megabar = $10^3$ kilobars = $10^{12}$ dynes/cm$^2$ = g/cm $\mu s^2$

Mbar cm$^3$/g = $10^{12}$ erg/g

1 gigapascal = 10 kilobars

1 cm/$\mu$s = 10 km/s

1 electron volt = 11604.7 K

Avogadro's number = 6.02252 x $10^{23}$/mole

Boltzmann's constant = 1.38054 x $10^{-16}$ erg/K

Planck's constant = $\hbar$ = 1.054494 x $10^{-27}$ erg s

Atomic mass unit = 1.66043 x $10^{-24}$ g/amu

Bohr radius $a_0$ = 0.529167 x $10^{-8}$ cm

Rydberg = 2.17971 x $10^{-11}$ erg

1 Rydberg/$a_0^3$ = 147.103 Mbar

ACKNOWLEDGMENTS

We thank those who have contributed suggestions, support, and criticism that helped mold the final product of this work: Joe Abdallah, Alice Baker, Jack Barnes, Judith Binstock, Art Dana, Nancy Freed, Britt Girard, Frank Guy, Brad Holian, Jim Johnson, Al Juveland, Gerald Kerley, Jim Kershner, Richard Krajcik, Charles Mader, Bob Osborne, John Richter, Harold Rogers, Thurman Talley, Mel Thieme, and Duane Wallace.

REFERENCES (Section I)

1. Charles L. Mader and Milton Samuel Shaw, "User's Manual for SIN," Los Alamos Scientific Laboratory report LA-7264-M (September 1978).

2. John F. Barnes, "Statistical Atom Theory and the Equation of State of Solids," Phys. Rev. 153, 269 (1967).

3. B. I. Bennett, J. D. Johnson, G. I. Kerley, and G. T. Rood, "Recent Developments in the SESAME Equation-of-State Library," Los Alamos Scientific Laboratory report LA-7130 (February 1978).

4. Charles L. Mader and B. G. Craig, "Nonsteady-State Detonations in One-Dimensional Plane, Diverging, and Converging Geometries," Los Alamos Scientific Laboratory report LA-5865 (April 1975).

5. Charles L. Mader and Charles A. Forest, "Two-Dimensional Homogeneous and Heterogeneous Detonation Wave Propagation," Los Alamos Scientific Laboratory report LA-6259 (June 1976).

6. Charles A. Forest, "Burning and Detonation," Los Alamos Scientific Laboratory report LA-7245 (July 1978).

7. Charles L. Mader, Numerical Modeling of Detonations, (University of California Press, Berkeley, California, 1979).

8. Cliff Plopper, Lynn D. Maas, and Eldon C. Pequette, "GAS (Graphical Analysis System)," LASL Utility Routine LTSS-523 (May 1977).

## II. HYDRODYNAMIC EQUATIONS OF MOTION (In collaboration with B. L. Holian)

In this section we try to give an intuitive derivation of the hydrodynamic equations of motion by considering the appropriate volume element and applying the conservation laws of mass, momentum, and energy. The only completely rigorous manner to derive the equations of motion is to consider the full tensor properties of the stress and strain, and then make the appropriate coordinate transformations corresponding to the symmetry of the problem.

To obtain the appropriate partial differential equations we must consider both Eulerian and Lagrangian coordinates. Eulerian coordinates are a spatially fixed coordinate system sometimes called a laboratory frame of reference. Lagrangian coordinates move through space with the body that they describe and may be thought of as labels for mass points. One may easily transform all quantities from Eulerian to Lagrangian coordinates and we usually visualize a mass element in an Eulerian system, calculate the desired quantities, and transform the results to a Lagrangian system. For a hydrodynamics computer code, the most useful form of the equations is a hybrid of Eulerian positions and velocities used to describe Lagrangian mass points.

### A. Conservation of Mass

In order to satisfy the law of conservation of mass, we merely require that the mass of a volume element remain constant even though its shape may change. That is

$$\text{Mass} = \rho V = \text{constant}, \tag{1}$$

where $\rho$ and $V$ are the density and volume of the mass element respectively. In the following,

$$\begin{aligned} r &= \text{Lagrangian position,} \\ R &= \text{Eulerian position.} \end{aligned} \tag{2}$$

At time $t = t_0$ we pick the Lagrangian and Eulerian coordinates to be equal, as well as the length of a mass element:

$$r(t_0) = R(t_0) \quad ; \quad \delta r(t_0) = \delta R(t_0) \quad . \tag{3}$$

At some later time t, the density is $\rho(t)$ and

$$r(t) = r(t_0) \quad \text{and} \quad r(t) + \delta r(t) = r(t_0) + \delta r(t_0) \tag{4}$$

are unchanged with time. The Eulerian coordinates become

$$R(t) \neq R(t_0) \quad \text{and} \quad \delta R(t) \neq \delta R(t_0) \quad . \tag{5}$$

1. Planar Geometry

In Fig. 1 we illustrate a mass element with planar symmetry. The volume of this mass is $dV = \delta R \, \Delta y \, \Delta z$ * and its incremental mass is

$$\begin{aligned} \Delta m' &= \rho(t) \, \delta R(t) \, \Delta y \, \Delta z \quad \text{(Eulerian)}, \\ &= \rho_0 \, \delta r \, \Delta y \, \Delta z = \text{constant} \quad \text{(Lagrangian)}. \end{aligned} \tag{6}$$

Defining the mass per unit area as $\Delta m = \frac{\Delta m'}{\Delta y \, \Delta z}$ and going to the infinitesimal limit, we have

$$dm = \rho \, dR = \rho_0 \, dr \quad , \tag{7}$$

where dm is also a mass per unit area.

2. Cylindrical Geometry

In cylindrical coordinates the mass elements may be written:

$$\begin{aligned} dm' &= \rho_0 \, r \, dr \, d\theta \, dz \quad \text{(Lagrangian)}, \\ &= \rho(t) \, R \, dR \, d\theta \, dz \quad \text{(Eulerian)}. \end{aligned} \tag{8}$$

---

* $\delta R$ and $\delta r$ are used to denote finite distances for the volume elements, while dR and dr denote infinitesimals. Although not strictly correct, $\delta R$ and $\delta r$ may be interchanged with dR and dr by taking a limit where $\delta R$ and $\delta r$ go to zero.

We can define a mass per unit length $dm = dm'/dz$ and, assuming cylindrical symmetry for the problem, integrate over $\theta$ to get

$$dm = 2\pi\rho_0 \, r \, dr = 2\pi\rho(t) \, R \, dR \quad . \tag{9}$$

### 3. Spherical Geometry

In spherical coordinates the mass elements may be written

$$\begin{aligned} dm = dm' &= \rho_0 \, r^2 \, dr \sin\theta \, d\theta \, d\phi \quad \text{(Lagrangian)}, \\ &= \rho(t) \, R^2 \, dR \sin\theta \, d\theta \, d\phi \quad \text{(Eulerian)}. \end{aligned} \tag{10}$$

Integration over $\theta$ and $\phi$ gives

$$dm = 4\pi\rho_0 \, r^2 \, dr = 4\pi\rho(t) \, R^2 \, dR \quad . \tag{11}$$

## B. Conservation of Momentum

To determine the net momentum flux through a mass element at any time, we use Newton's equation of motion in the form:

$$\frac{\text{force in the R direction}}{\text{unit mass}} = \frac{F_R}{dm} = \frac{\partial u_R}{\partial t} \equiv \dot{u}_R \quad , \tag{12}$$

where $u_R$ = velocity in the R direction.

### 1. Planar Geometry

Figure 1 shows a planar mass element subjected to a stress $\sigma_r$ in the positive direction and a stress $\sigma_r + \Delta\sigma_r$ in the negative direction. Since stress is defined as the force per unit area, then

$$F_R = \sigma_r \, \Delta y \, \Delta z \quad . \tag{13}$$

The net force acting on the volume element is $F_R - F_{R+\delta R}$, giving

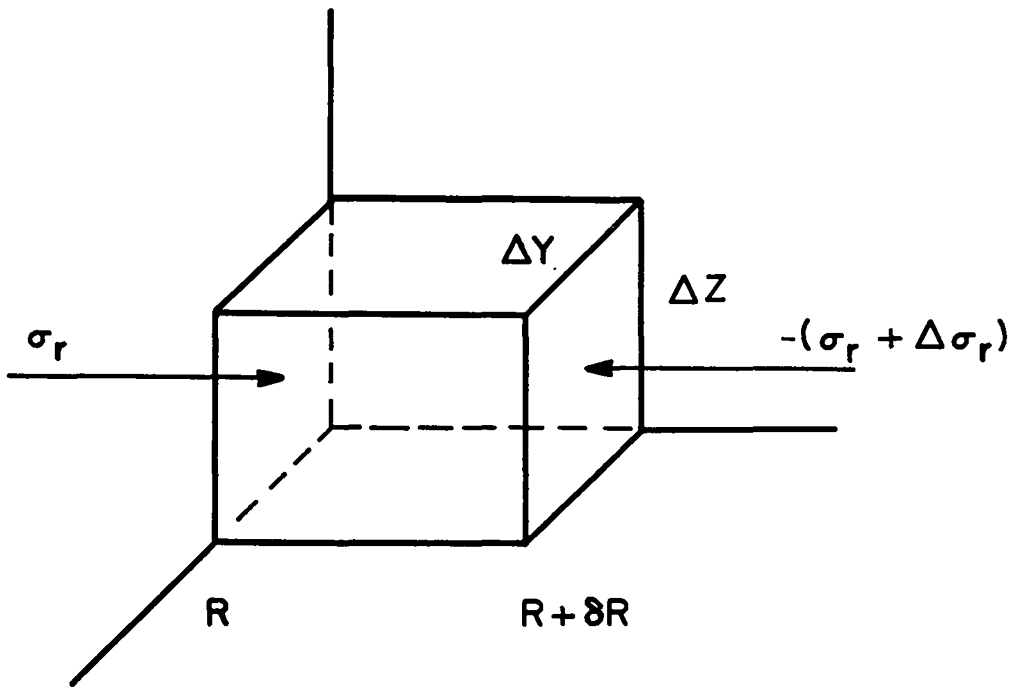



Fig. 1.
A planar mass element subjected to a stress $\sigma_r$ in the positive direction and a stress $\sigma_r + \Delta\sigma_r$ in the negative direction. The cross-sectional area that these stresses act upon is $\Delta y\ \Delta z$.

$$\frac{\partial u_R}{\partial t} = \frac{F_R - F_{R+\delta R}}{dm'} = \frac{\sigma_r \, \Delta_y \, \Delta_z}{\rho \; \delta R \; \Delta y \; \Delta z} - \frac{(\sigma_r + \Delta\sigma_r)\Delta y \; \Delta z}{\rho \; \delta R \; \Delta y \; \Delta z}$$

$$= - \frac{1}{\rho} \frac{\Delta\sigma_r}{\delta R} \xrightarrow[\lim \; \delta R \to 0]{} - \frac{1}{\rho} \frac{\partial \sigma_r}{\partial R} \quad , \tag{14}$$

or

$$\frac{\partial u_R}{\partial t} = - \frac{1}{\rho} \frac{\partial \sigma_R}{\partial R} \quad . \tag{15}$$

Transforming to the Lagrangian variables $dm = \rho_0 \; dr$ by using the relation $\rho_0 \; dr = \rho \; dR$, we have

$$\left(\frac{\partial u}{\partial t}\right)_r = - \frac{1}{\rho_0}\left(\frac{\partial \sigma}{\partial r}\right)_t = -\left(\frac{\partial \sigma}{\partial m}\right)_t \quad , \tag{16}$$

where $(\partial u/\partial t)_r$ denotes the acceleration of a single Lagrangian mass point with coordinate r and the stress gradient is evaluated at constant time t.

## 2. Cylindrical Geometry

For a system with cylindrical symmetry, we must include the contributions to the stress from the θ-direction as well as the r-direction. Figure 2 shows a mass element in cylindrical coordinates subjected to the stresses $\sigma_{rr}$ and $\sigma_{rr} + \Delta\sigma_{rr}$ in the radial direction, and stresses $+\sigma_{\theta\theta}$ and $-\sigma_{\theta\theta}$ in the angular direction. We need only worry about the components of $\sigma_{\theta\theta}$ in the radial direction because the net force in the θ-direction at the center of the mass point is

$$F_\theta = +\sigma_{\theta\theta} \cos \frac{\theta}{2} \; \delta R \; \Delta z - \sigma_{\theta\theta} \cos \frac{\theta}{2} \; \delta R \; \Delta z = 0 \quad . \tag{17}$$

In the radial direction we have

$$\begin{aligned} F_R = m\dot{u}_R &= \rho R \; \delta R \; \Delta\theta \; \Delta z \; \dot{u}_R \\ &= \sigma_{rr} \; R \sin \Delta\theta \; \Delta z - (\sigma_{rr} + \Delta\sigma_{rr})(R + \delta R) \sin \Delta\theta \; \Delta z + 2\sigma_{\theta\theta} \sin \frac{\Delta\theta}{2} \; \delta R \; \Delta z \\ &= -\sigma_{rr} \; \delta R \sin \Delta\theta \; \Delta z - \Delta\sigma_{rr}(R + \delta R) \sin \Delta\theta \; \Delta z + 2\sigma_{\theta\theta} \sin \frac{\Delta\theta}{2} \; \delta R \Delta z \quad . \end{aligned} \tag{18}$$

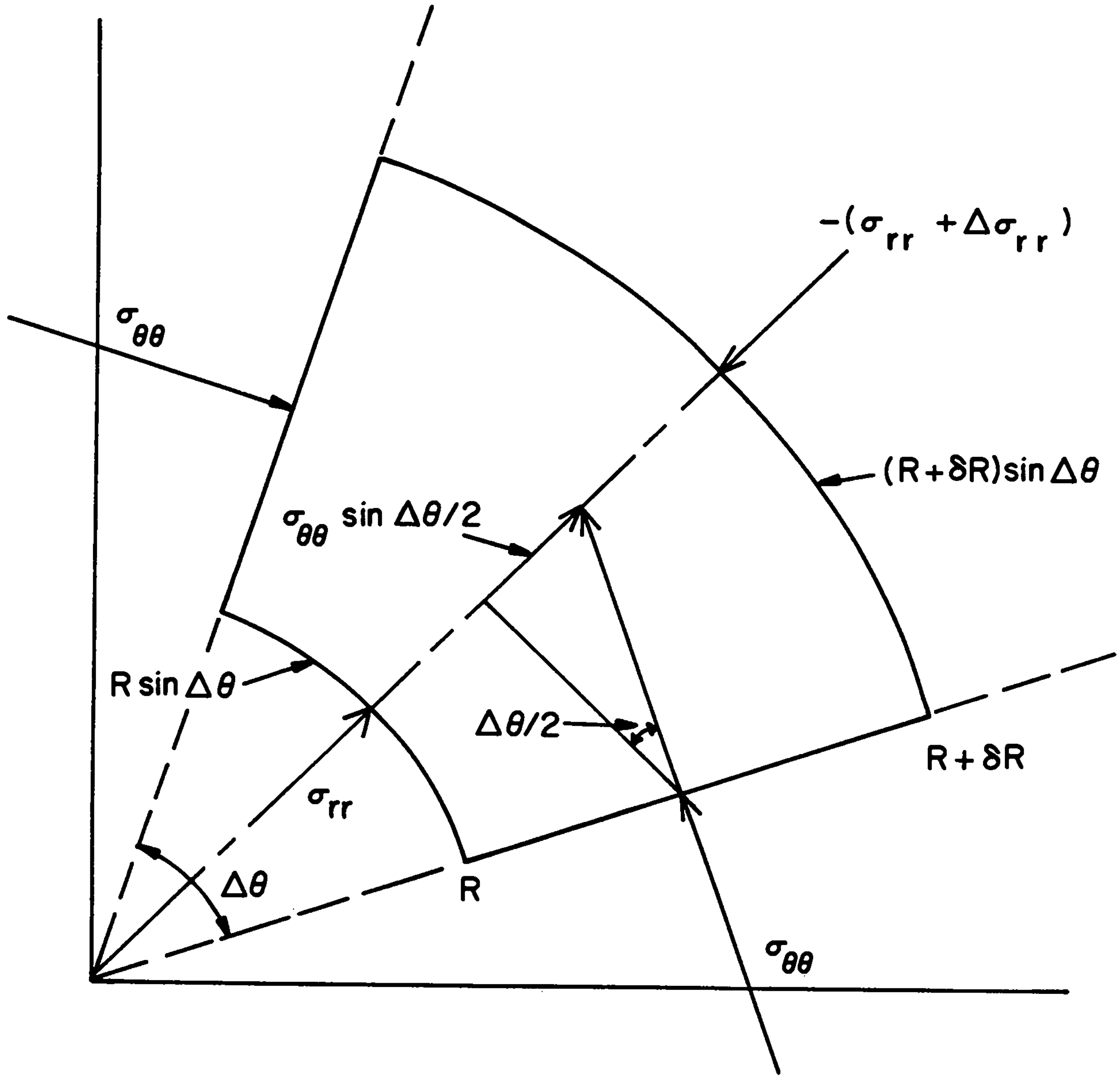



Fig. 2.
A mass element in cylindrical coordinates subjected to the stresses $\sigma_{rr}$ and $-(\sigma_{rr} + \Delta\sigma_{rr})$ in the radial direction and to $+\sigma_{\theta\theta}$ and $-\sigma_{\theta\theta}$ in the angular direction.

Because mass = $\rho R \; \delta R \; \Delta\theta \; \Delta z$, we have

$$\rho \dot{u}_R = - \frac{\sigma_{rr}}{R} \frac{\sin \Delta\theta}{\Delta\theta} - \frac{\Delta\sigma_{rr}}{\delta R}\left(1 + \frac{\delta R}{R}\right) \frac{\sin \Delta\theta}{\Delta\theta} + \frac{2\sigma_{\theta\theta}}{R} \frac{\sin \Delta\theta/2}{\Delta\theta} \quad .$$

In the limit as $\delta R$, $\Delta\theta$ goes to zero, $\frac{\sin \Delta\theta}{\Delta\theta} \rightarrow 1$ giving

$$\rho \dot{u}_R = - \frac{\sigma_{rr}}{R} - \frac{\partial\sigma_{rr}}{\partial R} + \frac{\sigma_{\theta\theta}}{R} + O(\delta R) + O(\Delta\theta) \quad ,$$

where $O(x) \equiv$ order of x.

Thus,

$$\rho \dot{u}_R = - \frac{\partial\sigma_{rr}}{\partial R} + \frac{\sigma_{\theta\theta} - \sigma_{rr}}{R} \quad . \tag{19}$$

To change to Lagrangian coordinates, we use $dm = \rho_0 \; r \; dr = \rho R \; dR \Rightarrow dr = dm/\rho R$, giving

$$\left(\frac{\partial u_R}{\partial t}\right)_r = -R\left(\frac{\partial\sigma_{rr}}{\partial m}\right)_t + \frac{(\sigma_{\theta\theta} - \sigma_{rr})}{\rho R} \quad , \tag{20}$$

where $(\;)_r$ denotes that we are considering a single Lagrangian mass element (constant r or dm), and the $u_R$ is the velocity measured in the Eulerian reference frame.

### 3. Spherical Geometry

A spherical volume element is shown in Fig. 3. The spherical case is slightly more difficult to visualize because $\Delta\phi$ is measured in the x-y plane and the arc length swept by a rotation in $\phi$ must be projected up to the volume element. We have introduced the quantity $\Delta\phi'$ (not equal to $\Delta\phi$) to help avoid confusion. As in the cylindrical case, there are components in the radial directions from $\sigma_{\theta\theta}$ and $\sigma_{\phi\phi}$, but no net force in either the $\theta$ or $\phi$ direction.

(a)

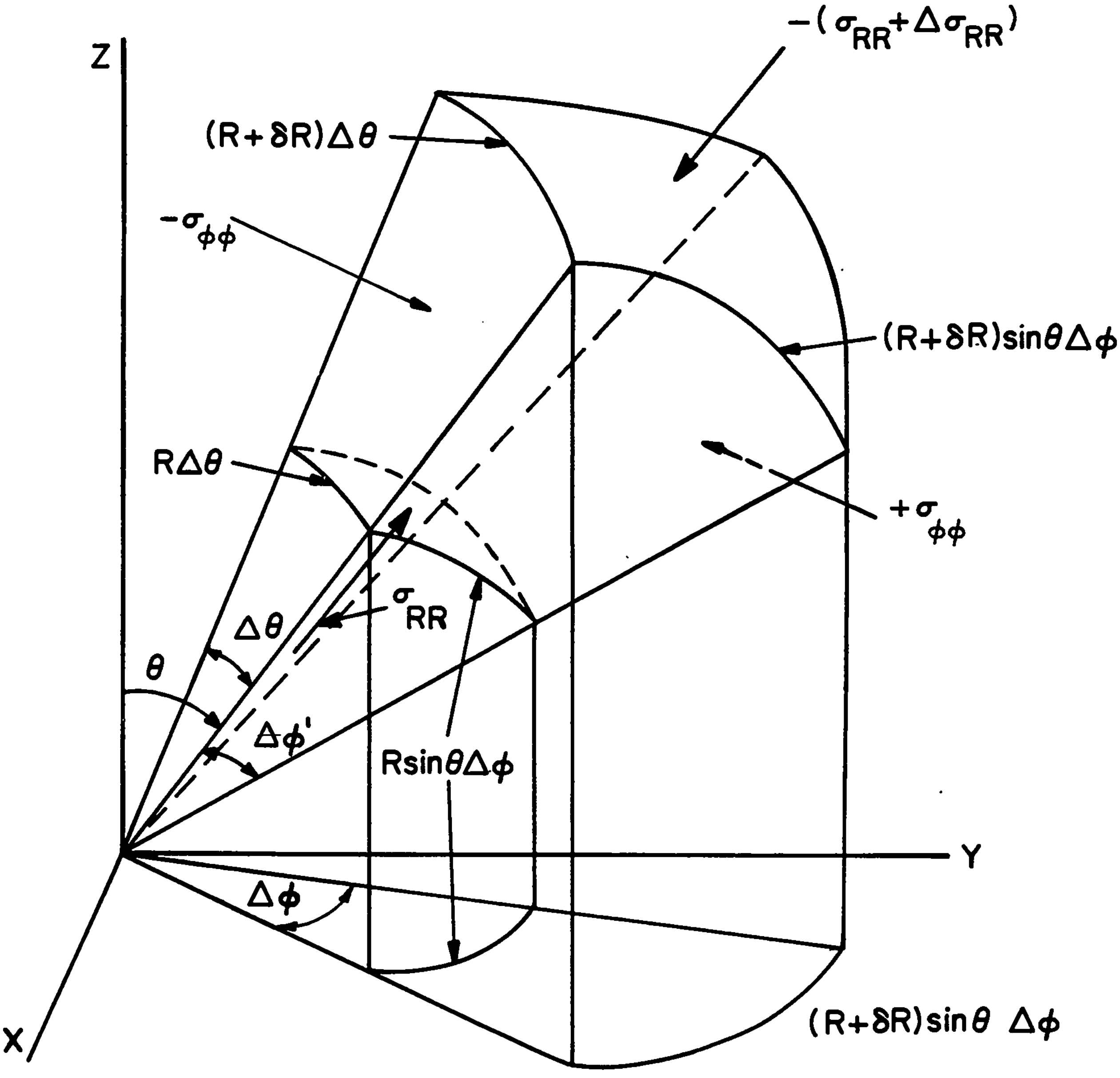



Fig. 3a.
A spherical volume element subjected to the stresses $\sigma_{RR}$ and $-(\sigma_{RR} + \Delta\sigma_{RR})$ in the radial direction, $+\sigma_{\theta\theta}$ and $-\sigma_{\theta\theta}$ in one angular direction, and $+\sigma_{\phi\phi}$ and $-\sigma_{\phi\phi}$ in the other angular direction. Note that $\Delta\phi \neq \Delta\phi'$.

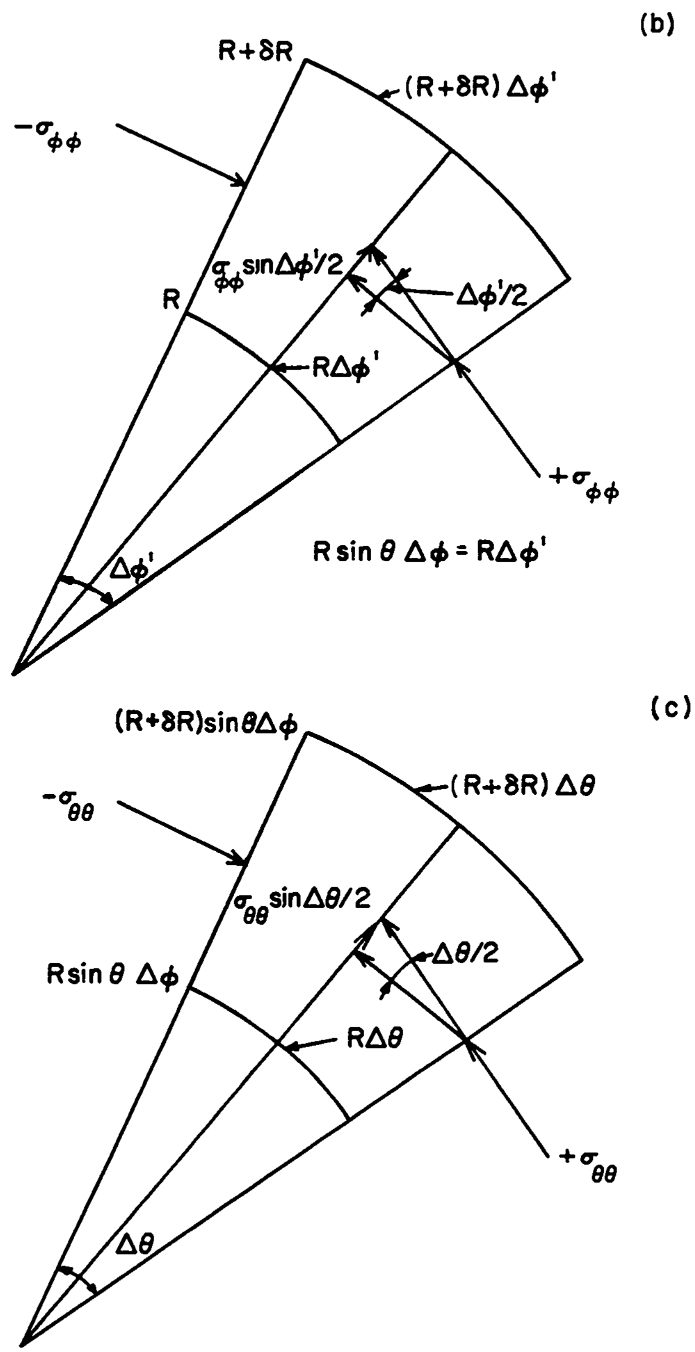



Fig. 3b,c.

(b) The stress $+\sigma_{\phi\phi}$ and $-\sigma_{\phi\phi}$ acting on the spherical volume element depend on the angle $\Delta\phi'$ and not on its projection in the x-y plane, $\Delta\phi$.

(c) For the $\theta$-direction, the stresses $+\sigma_{\theta\theta}$ and $-\sigma_{\theta\theta}$ are dependent only upon $\Delta\theta$.

Consider first the stresses $\sigma_{\phi\phi}$ acting on the volume element as shown in Fig. 3b. The net force in the radial direction from the stresses acting on each side of the volume element is

$$\left(\sigma_{\phi\phi} \sin \frac{\Delta\phi'}{2} + \sigma_{\phi\phi} \sin \frac{\Delta\phi'}{2}\right) R \, \delta R \, \Delta\theta = 2\sigma_{\phi\phi} \sin \frac{\Delta\phi'}{2} R \, \delta R \, \Delta\theta \quad , \tag{21}$$

with $\sigma_{\phi\phi}$ acting on an area $R \, \delta R \, \Delta\theta$. By requiring the arc length swept by the $\Delta\phi$ rotation to be the same as the $\Delta\phi'$ rotation, we have

$$\begin{aligned} R\Delta\phi' &= R \sin \theta \, \Delta\phi \quad , \\ \Delta\phi' &= \sin \theta \, \Delta\phi \quad , \end{aligned} \tag{22}$$

or making Eq. (21)

$$2\sigma_{\phi\phi} \sin[\tfrac{1}{2} \sin \theta \, \Delta\phi] \, R \, \delta R \, \Delta\theta \quad . \tag{23}$$

The net force in the radial direction from the $\sigma_{\theta\theta}$'s is

$$2\sigma_{\theta\theta} \sin \frac{\Delta\theta}{2} R \, \delta R \sin \theta \, \Delta\phi \quad . \tag{24}$$

The net total force in the radial direction is

$$\begin{aligned} F_R = {} & \sigma_{rr} R^2 \sin \theta \, \Delta\theta \, \Delta\phi - (\sigma_{rr} + \Delta\sigma_{rr})(R + \delta R)^2 \sin \theta \, \Delta\theta \, \Delta\phi \\ & + 2\sigma_{\phi\phi} \sin [\tfrac{1}{2} \sin \theta \, \Delta\phi] \, R \, \delta R \, \Delta\theta + 2\sigma_{\theta\theta} \sin \frac{\Delta\theta}{2} R \, \delta R \sin \theta \, \Delta\phi \quad . \end{aligned} \tag{25}$$

$$\frac{F_R}{\text{unit mass}} = \frac{F_R}{\rho R^2 \sin \theta \;\; \Delta\theta \, \Delta\phi} \Rightarrow \rho \frac{\partial U_R}{\partial t} = \frac{F_R}{R^2 \, \delta R \sin \theta \, \Delta\theta \, \Delta\phi} \quad .$$

In Eq. (25) we again take the limit of small $\Delta\theta$ and $\Delta\phi$ :

$$\begin{aligned} \sin \frac{\Delta\theta}{2} &\to \frac{\Delta\theta}{2} \quad , \\ \sin[\tfrac{1}{2} \sin \theta \, \Delta\phi] &\to \tfrac{1}{2} \sin \theta \, \Delta\phi \quad . \end{aligned}$$

Thus,

$$\rho \frac{\partial u_R}{\partial t} = -\sigma_{rr}\left(\frac{2}{R} + \frac{\delta R}{R^2}\right) - \frac{\Delta\sigma_{rr}}{\delta R}\left(1 + \frac{2\delta R}{R} + \frac{(\delta R)^2}{R^2}\right) + \frac{\sigma_{\phi\phi}}{R} + \frac{\sigma_{\theta\theta}}{R} \quad ,$$

or, taking the limit as $\delta R$ goes to zero,

$$\rho \frac{\partial u_R}{\partial t} = -\frac{\partial \sigma_{rr}}{\partial R} + \frac{1}{R}(\sigma_{\phi\phi} + \sigma_{\theta\theta} - 2\sigma_{rr}) + O(\delta R) + O(\delta R^2) \quad .$$

Because of the spherical symmetry, $\sigma_{\phi\phi} = \sigma_{\theta\theta}$, and if we neglect terms of order $\delta R$ and higher, our result is

$$\rho \frac{\partial u_R}{\partial t} = -\frac{\partial \sigma_{rr}}{\partial R} - \frac{2}{R}(\sigma_{rr} - \sigma_{\theta\theta}) \quad . \tag{26}$$

To go to Lagrangian coordinates we use $dm = \rho R^2\, dR$, and obtain the acceleration equation for the mass point labeled with the coordinate r:

$$\left(\frac{\partial u_R}{\partial t}\right)_r = -R^2\left(\frac{\partial \sigma_{rr}}{\partial m}\right)_t + \frac{2(\sigma_{\theta\theta} - \sigma_{rr})}{\rho R} \quad . \tag{27}$$

## C. Conservation of Energy

We wish to calculate the increase in the <u>total</u> energy of a mass element during some time $\Delta t$ due to the work done by the stresses in the radial direction. The stresses in the angular direction do <u>no</u> work on the mass element because there is no motion in the angular directions.

### 1. Planar Geometry

A planar mass element is depicted in Fig. 4 at times t and t + $\Delta$t. Letting $\Delta E$ be the change in energy per unit mass = $\rho\delta R\ \Delta y\ \Delta z$, we can write (energy = force x distance):

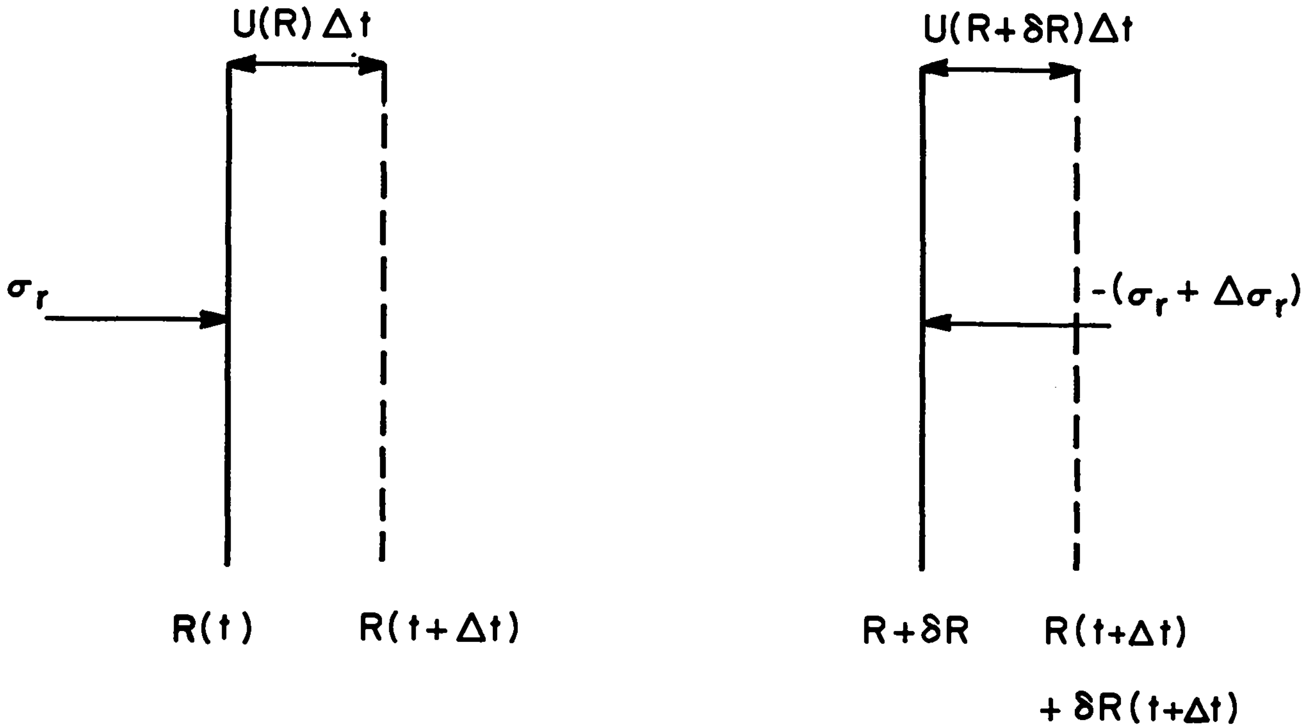



Fig. 4.
A planar mass element at time t (solid line) and $t + \Delta t$ (dashed line).

$$\Delta E = \frac{\sigma_r \; \Delta y \; \Delta z \; u(R) \; \Delta t - (\sigma_r + \Delta\sigma_r)\Delta y \; \Delta z \; u(R + \delta R)\Delta t}{\rho\delta R \; \Delta y \; \Delta z} \quad ,$$

$$\rho \frac{\partial E}{\partial t} = \frac{\sigma_r u(R) - (\sigma_r + \Delta\sigma_r)u(R + \delta R)}{\delta R} \quad . \tag{28}$$

We may expand $u(R + \delta R)$ about R:

$$u(R + \delta R) = u(R) + \left(\frac{\partial u}{\partial R}\right)\delta R + \cdots \; .$$

Substituting this result to first order in $\delta R$ into Eq. (28), we get

$$\rho\left(\frac{\partial E}{\partial t}\right) = - \frac{\Delta\sigma_r u}{\delta R} - \sigma_r\left(\frac{\partial u}{\partial R}\right) - \Delta\sigma_r\left(\frac{\partial u}{\partial R}\right) \; . \tag{29}$$

Taking the limit as $\delta R \to 0$,

$$\rho\left(\frac{\partial E}{\partial t}\right) = - u\left(\frac{\partial\sigma_r}{\partial R}\right) - \sigma_r\left(\frac{\partial u}{\partial R}\right) \quad .$$

In the limit $\delta R \to 0$, then $\Delta\sigma_r \to 0$.

Using $dm = \rho \; dR$, we can write

$$\left(\frac{\partial E}{\partial t}\right)_r = - \frac{\partial}{\partial m}(\sigma_r u) \quad , \tag{30}$$

where the $(\;)_r$ expresses the fact that we are considering a single Lagrangian mass element.

### 2. Cylindrical Geometry

The calculation for cylindrical geometry proceeds in the same manner as the planar case except that the unit mass element = $\rho R\delta R \; \Delta\theta \; \Delta z$. The angular stresses $\sigma_{\theta\theta}$ and the axial stresses $\sigma_{zz}$ do no work because motion is permitted in the R direction only. Figure 5 shows the appropriate mass element at times t and $t + \Delta t$.

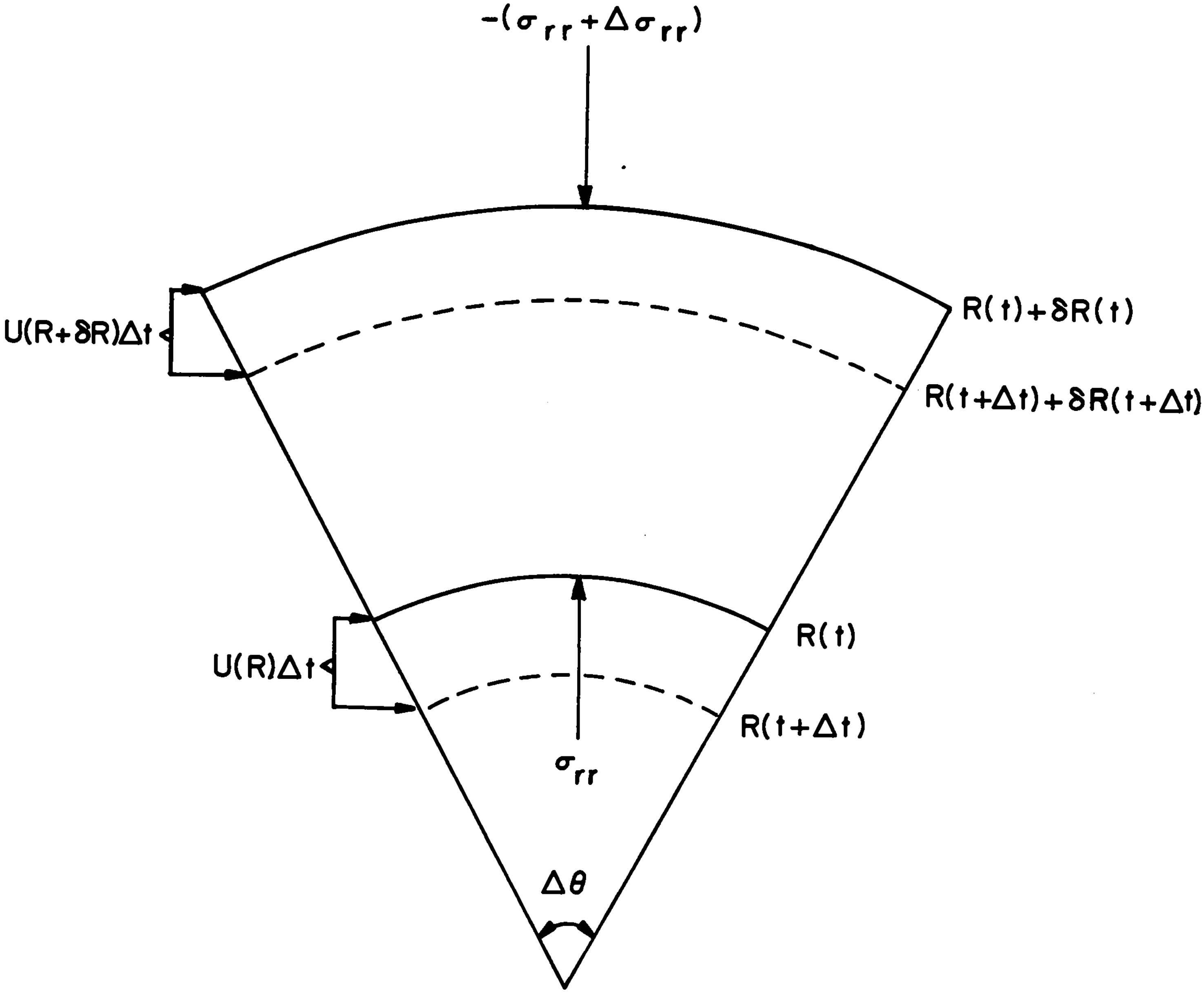



Fig. 5.
A cylindrical mass element at times t (solid lines) and t + Δt (dashed lines). The angular stresses, $+\sigma_{\theta\theta}$ and $-\sigma_{\theta\theta}$, do <u>no</u> work because motion is permitted in the radial direction only.

The change in energy per unit mass is again calculated by considering the distance the forces acting on the mass element move.

$$dE = \frac{+\sigma_{rr}R(t)\Delta\theta\ \Delta z\,[u(R)\Delta t] - (\sigma_{rr} + \Delta\sigma_{rr})(R + \delta R)\Delta\theta\ \Delta z[u(R + \delta R)\Delta t]}{\rho R\ \delta R\ \Delta\theta\ \Delta z} \quad .$$

$$\frac{\partial E}{\partial t} = + \frac{\sigma_{rr}u(R)}{\rho\delta R} - \frac{(\sigma_{rr} + \Delta\sigma_{rr})(R + \delta R)u(R + \delta R)}{\rho R\ \delta R} \quad .$$

Using $u(R + \delta R) = u(R) + (\partial u/\partial R)\delta R + \cdots$, we obtain

$$\frac{\partial E}{\partial t} = -\left\{\frac{\sigma_{rr}u(R)}{R} + \sigma_{rr}\frac{\partial u}{\partial R} + \sigma_{rr}\left(\frac{\partial u}{\partial R}\right)\frac{\delta R}{R} + \frac{\Delta\sigma_{rr}u(R)}{\delta R} + \frac{\Delta\sigma_{rr}u(R)}{\rho R}\right.$$

$$\left. + \Delta\sigma_{rr}\left(\frac{\partial u}{\partial r}\right) + \Delta\sigma_{rr}\left(\frac{\partial u}{\partial R}\right)\frac{\delta R}{R}\right\} \quad , \tag{31}$$

$$= O(1) + O(1) + O(\delta R) + O(1) + O(\Delta\sigma_{rr}) + O(\Delta\sigma_{rr}) + O(\Delta\sigma_{rr}\delta R) \quad .$$

In the last equation we have written the relative order of the terms for Eq. (31). Keeping only the leading order terms we have, after taking the appropriate limits,

$$\rho\,\frac{\partial E}{\partial t} = -\left\{\sigma_{rr}\frac{u}{R} + \sigma_{rr}\left(\frac{\partial u}{\partial R}\right) + \left(\frac{\partial\sigma_{rr}}{\partial R}\right)u\right\} \quad .$$

Using $dm = \rho R\ dR$ or $\partial R/\partial m = 1/\rho R$, we write

$$\left(\frac{\partial E}{\partial t}\right)_r = -\frac{\partial}{\partial m}\,(\sigma_{rr}uR) \quad , \tag{32}$$

for the Lagrangian mass element labeled by r.

### 3. Spherical Geometry

A spherical mass element is shown in Fig. 6 where a unit mass is $\rho R^2\ \delta R \sin\theta\ \Delta\theta\ \Delta\phi$. In the same manner as for the planar and cylindrical cases, the change in energy per unit mass due to the work done by the radial stresses is

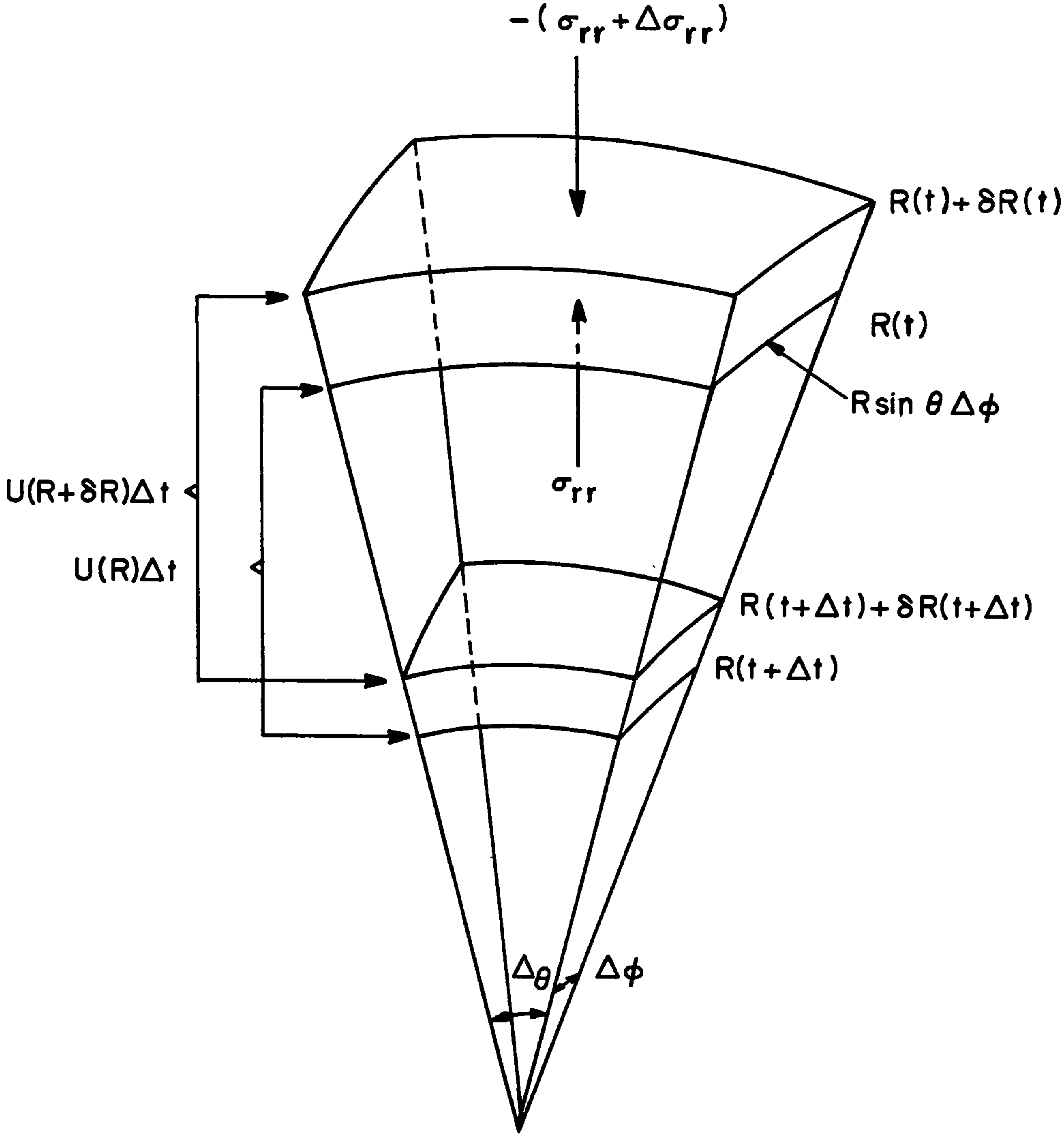



Fig. 6.
A spherical mass element at times t and t + Δt. The two sets of stresses in the angular direction do no net work and are not shown here.

$$dE = \frac{\sigma_{rr}(R\Delta\theta)(R\sin\theta\ \Delta\phi)[u(R)\Delta t] - (\sigma_{rr} + \Delta\sigma_{rr})(R+\delta R)\Delta\theta(R+\delta R)\sin\theta\ \Delta\phi[u(R+\delta R)\Delta t]}{\rho R^2\ \delta R\ \sin\theta\ \Delta\theta\ \Delta\phi}$$

or, rewriting, the rate of change of energy is given by

$$\rho\frac{\partial E}{\partial t} = \frac{\sigma_{rr}R^2 u(R) - (\sigma_{rr} + \Delta\sigma_{rr})(R + \delta R)^2 u(R + \delta R)}{R^2\ \delta R} \quad .$$

Using $u(R + \delta R) = u(R) + (\partial u/\partial R)\delta R + \cdots$ , we may obtain

$$\rho\frac{\partial E}{\partial t} = -\left\{\frac{2\sigma_{rr}u}{R} + \frac{\sigma_{rr}\delta R u}{R^2} + \sigma_{rr}\left[1 + \frac{2\delta R}{R} + \left(\frac{\delta R}{R}\right)^2\right]\left(\frac{\partial u}{\partial R}\right) + \frac{\Delta\sigma_{rr}}{\delta R}\left[1 + \frac{2\delta R}{R} + \left(\frac{\delta R}{R}\right)^2\right]u\right.$$
$$\left. + \Delta\sigma_{rr}\left[1 + \frac{2\delta R}{R} + \left(\frac{\delta R}{R}\right)^2\right]\left(\frac{\partial u}{\partial R}\right)\right\} \quad .$$

Keeping only the leading order terms, our result becomes

$$\rho\frac{\partial E}{\partial t} = -\left\{\frac{2\sigma_{rr}u}{R} + u\left(\frac{\partial\sigma_{rr}}{\partial R}\right) + \sigma_{rr}\left(\frac{\partial u}{\partial R}\right)\right\} \quad .$$

Again, $\sigma_{\theta\theta}$ and $\sigma_{\phi\phi}$ do no work because of the requirement that there be no motion in the $\theta$ and $\phi$ directions. To write in a Lagrangian form, use $dm = \rho R^2\ dR$ to get $1/\rho R = R\ \partial R/\partial m$, and

$$\left(\frac{\partial E}{\partial t}\right)_r = -\frac{\partial}{\partial m}\left(R^2 u\sigma_{rr}\right) \quad . \tag{33}$$

## 4. The Internal Energy

For some calculations, it is more convenient to work directly with the internal energy than with the total energy. The internal energy, I, may be determined from

$$E = \frac{1}{2} u^2 + I \quad ,$$

where the first term in this expression gives the kinetic energy. Equations (30), (32), and (33) may be written in the combined form

$$\left(\frac{\partial E}{\partial t}\right)_r = - \frac{\partial}{\partial m}(\sigma_{rr} u R^{d-1}) \quad ,$$

where d = 1, 2, or 3 for planar, cylindrical, or spherical geometry, respectively. We now wish to obtain the time rate of change of the internal energy. Thus,

$$\left(\frac{\partial E}{\partial t}\right) = u\left(\frac{\partial u}{\partial t}\right) + \left(\frac{\partial I}{\partial t}\right)_r = -uR^{d-1}\left(\frac{\partial \sigma_{rr}}{\partial m}\right)_t - \sigma_{rr}\left(\frac{\partial}{\partial m} uR^{d-1}\right)_t \quad .$$

In analogy with the above expression for the conservation of total energy, we may write the conservation of momentum relation as

$$\left(\frac{\partial u}{\partial t}\right)_r = -R^{d-1}\left(\frac{\partial \sigma_{rr}}{\partial m}\right)_t + (d - 1) \frac{(\sigma_{\theta\theta} - \sigma_{rr})}{\rho R} \quad .$$

Using this expression allows one to write

$$\left(\frac{\partial I}{\partial t}\right)_r = -\sigma_{rr}\left(\frac{\partial u R^{d-1}}{\partial m}\right)_t + \frac{u(d - 1)}{\rho R} (\sigma_{rr} - \sigma_{\theta\theta}) \quad .$$

## D. Summary of Equations

In this section we shall summarize the equations obtained by the application of the conservation laws of mass, momentum, and energy on a one-dimensional Lagrangian mass element under stress.

Conservation of mass requires

$$\begin{aligned} dm &= \rho dR = \rho_0 dr \quad \text{(planar)}, \\ &= 2\pi\rho R dR = 2\pi\rho_0 r dr \quad \text{(cylindrical)}, \\ &= 4\pi\rho R^2 dR = 4\pi\rho_0 r^2 dr \quad \text{(spherical)}. \end{aligned} \tag{34}$$

Conservation of momentum gives

$$\begin{aligned} \left(\frac{\partial u}{\partial t}\right)_r &= -\frac{1}{\rho_0}\left(\frac{\partial \sigma_r}{\partial r}\right)_t = -\left(\frac{\partial \sigma_r}{\partial m}\right)_t \quad \text{(planar)}, \\ &= -R\left(\frac{\partial \sigma_{rr}}{\partial m}\right)_t + \frac{(\sigma_{\theta\theta} - \sigma_{rr})}{\rho R} \quad \text{(cylindrical)}, \\ &= -R^2\left(\frac{\partial \sigma_{rr}}{\partial m}\right)_t + 2\,\frac{(\sigma_{\theta\theta} - \sigma_{rr})}{\rho R} \quad \text{(spherical)} . \end{aligned} \tag{35}$$

Conservation of energy gives

$$\begin{aligned} \left(\frac{\partial E}{\partial t}\right)_r &= -\frac{\partial}{\partial m}(\sigma_r u) \quad \text{(planar)}, \\ &= -\frac{\partial}{\partial m}(\sigma_r uR) \quad \text{(cylindrical)}, \\ &= -\frac{\partial}{\partial m}(\sigma_r uR^2) \quad \text{(spherical)}. \end{aligned} \tag{36}$$

## E. Finite Difference Equations

In the present section we will present the procedure for evaluating the differential equations for fluid motion on a finite Lagrangian space-time grid. We begin with a general discussion of Taylor series expansions of known functions and then use these results for the fluid equations. At the end of this section the conservation properties of the difference equations are discussed. For simplicity we give only the results for planar geometry.

### 1. Expansions of a Function on a Lagrangian Lattice

Consider some function $f(r)$ of the Lagrangian coordinate $r$ on a grid of Lagrangian points as shown in Fig. 7. The distance between the $j-\frac{1}{2}$ and $j+\frac{1}{2}$ boundaries is called $r_j$ for cell $j$. The function $f(r)$ evaluated at the center of the cell is written $f_j$ and the same function evaluated at the $j+\frac{1}{2}$ boundary is written $f_{j+\frac{1}{2}}$.

Assume that the values of $f(r)$ are known at the cell center and we want to calculate spatial derivatives of $f$ at cell boundaries; that is, we know $f_j$, $f_{j\pm1}$, $f_{j\pm2}$, $\cdots$ and we want to calculate $(\partial f/\partial r)_{j+\frac{1}{2}}$. Because $j+\frac{1}{2}$ is the boundary between cell $j$ and cell $j+1$, we will need to use $f_j$ and $f_{j+1}$ to determine the derivative. To do this, we make a Taylor series expansion of our function about the point $j+\frac{1}{2}$.

$$f_j = f_{j+\frac{1}{2}} - \left(\frac{\partial f}{\partial r}\right)_{j+\frac{1}{2}} \frac{r_j}{2} + \frac{1}{2}\left(\frac{\partial^2 f}{\partial r^2}\right)_{j+\frac{1}{2}} \left(\frac{r_j}{2}\right)^2 + O(r_j^3) \quad , \tag{37}$$

where $\frac{1}{2} r_j$ is the distance from the cell center to the boundary and $O(r_j^3)$ indicates that the next term in the expansion is of order $r_j^3$. Similarly, we may write

$$f_{j+1} = f_{j+\frac{1}{2}} + \left(\frac{\partial f}{\partial r}\right)_{j+\frac{1}{2}} \frac{r_{j+1}}{2} + \frac{1}{2}\left(\frac{\partial^2 f}{\partial r^2}\right)_{j+\frac{1}{2}} \left(\frac{r_{j+1}}{2}\right)^2 + O(r_{j+1}^3) \quad . \tag{38}$$

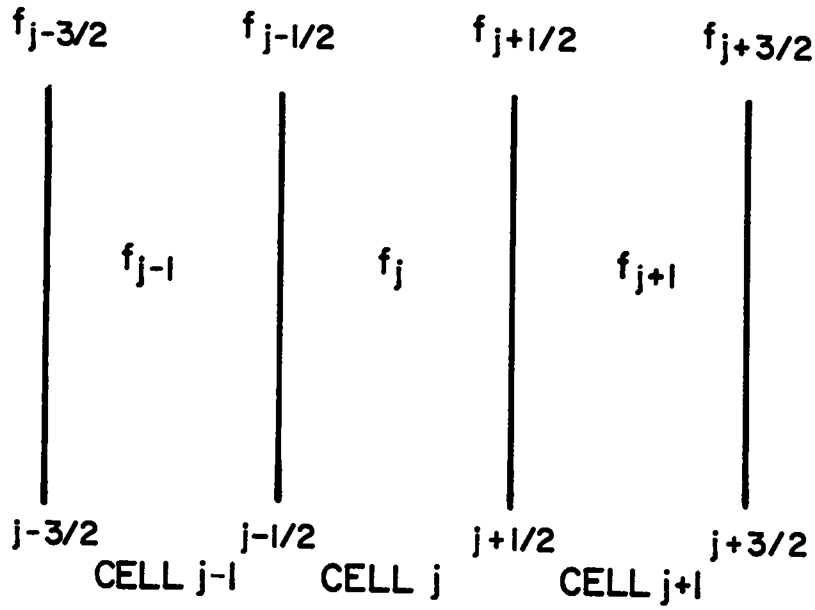



Fig. 7.
A function f evaluated on a Lagrangian coordinate system in one dimension. Integer values of the subscript denote cell centers and half-integer values denote cell boundaries.

Subtracting Eq. (37) from Eq. (38), we obtain the desired result:

$$f_{j+1} - f_j = \frac{1}{2}\left(\frac{\partial f}{\partial r}\right)_{j+\frac{1}{2}}(r_j + r_{j+1}) + O(r^2) \quad ,$$

$$\left(\frac{\partial f}{\partial r}\right)_{j+\frac{1}{2}} \cong \frac{(f_{j+1} - f_j)}{\frac{1}{2}(r_{j+1} + r_j)} + O(r) \quad . \tag{39}$$

If we multiply Eq. (37) by $r_{j+1}$, Eq. (38) by $r_j$, and add, we get

$$r_{j+1}f_j + r_jf_{j+1} = r_{j+1}f_{j+\frac{1}{2}} + r_jf_{j+\frac{1}{2}} + O(r^3) \quad ,$$

$$f_{j+\frac{1}{2}} \cong \frac{r_{j+1}f_j + r_jf_{j+1}}{r_j + r_{j+1}} + O(r^2) \quad . \tag{40}$$

This result gives us the value of a function at the boundary between two cells knowing only the values at the adjacent cell centers.

To obtain the derivative of a function at the cell center, we make a different expansion:

$$f_{j+\frac{1}{2}} = f_j + \left(\frac{\partial f}{\partial r}\right)_j \frac{r_j}{2} + \frac{1}{2}\left(\frac{\partial^2 f}{\partial r^2}\right)_j\left(\frac{r_j}{2}\right)^2 + O(r_j^3) \quad ,$$

$$f_{j-\frac{1}{2}} = f_j - \left(\frac{\partial f}{\partial r}\right)_j \frac{r_j}{2} + \frac{1}{2}\left(\frac{\partial^2 f}{\partial r^2}\right)_j\left(\frac{r_j}{2}\right)^2 + O(r_j^3) \quad .$$

Subtracting,

$$f_{j+\frac{1}{2}} - f_{j-\frac{1}{2}} = \left(\frac{\partial f}{\partial r}\right)_j r_j + O(r_j^3)$$

or

$$\left(\frac{\partial f}{\partial r}\right)_j = \frac{f_{j+\frac{1}{2}} - f_{j-\frac{1}{2}}}{r_j} + O(r_j^2) \quad . \tag{41}$$

The results of this section (Eqs. (39)-(41)) will be used to obtain the finite difference form of the equations of motion. As will be seen, we will have information about different quantities at different Lagrangian positions (cell centers or boundaries) and must use the results of this section to evaluate the function or its derivatives where they are needed.

2. Space and Time Grid

In writing down the finite difference solutions for the differential equations of motion, we must make some decisions about where in the space-time grid the various dynamic and thermodynamic quantities should be evaluated. Any choice we make will not be unique. Figure 8 illustrates the space-time grid we will be using. Cell j has boundaries at $j \pm \frac{1}{2}$ and we associate with the center of the cell the density (or volume), mass, stress, total energy, and internal energy.

$V_j$ = volume of cell j $(V = \rho^{-1})$

$M_j$ = mass of cell j (Lagrangian variable)*

$\rho_j$ = density of cell j

$\sigma_j$ = stress of cell j

$E_j$ = total energy of cell j

$I_j$ = internal energy of cell j

We locate the positions, R, of the cell boundaries at $j \pm \frac{1}{2}$ as measured in the laboratory coordinate frame. This also locates the velocities at the boundaries.

$R_{j+\frac{1}{2}}$ = position of one boundary of cell j

$$u_{j+\frac{1}{2}} = \frac{\partial R_{j+\frac{1}{2}}}{\partial t} = \text{velocity of one boundary of cell j}$$

---

*$M_j$ is equivalent to the differential mass element dm introduced in Sec. III.A; that is, $dm = M_j$.

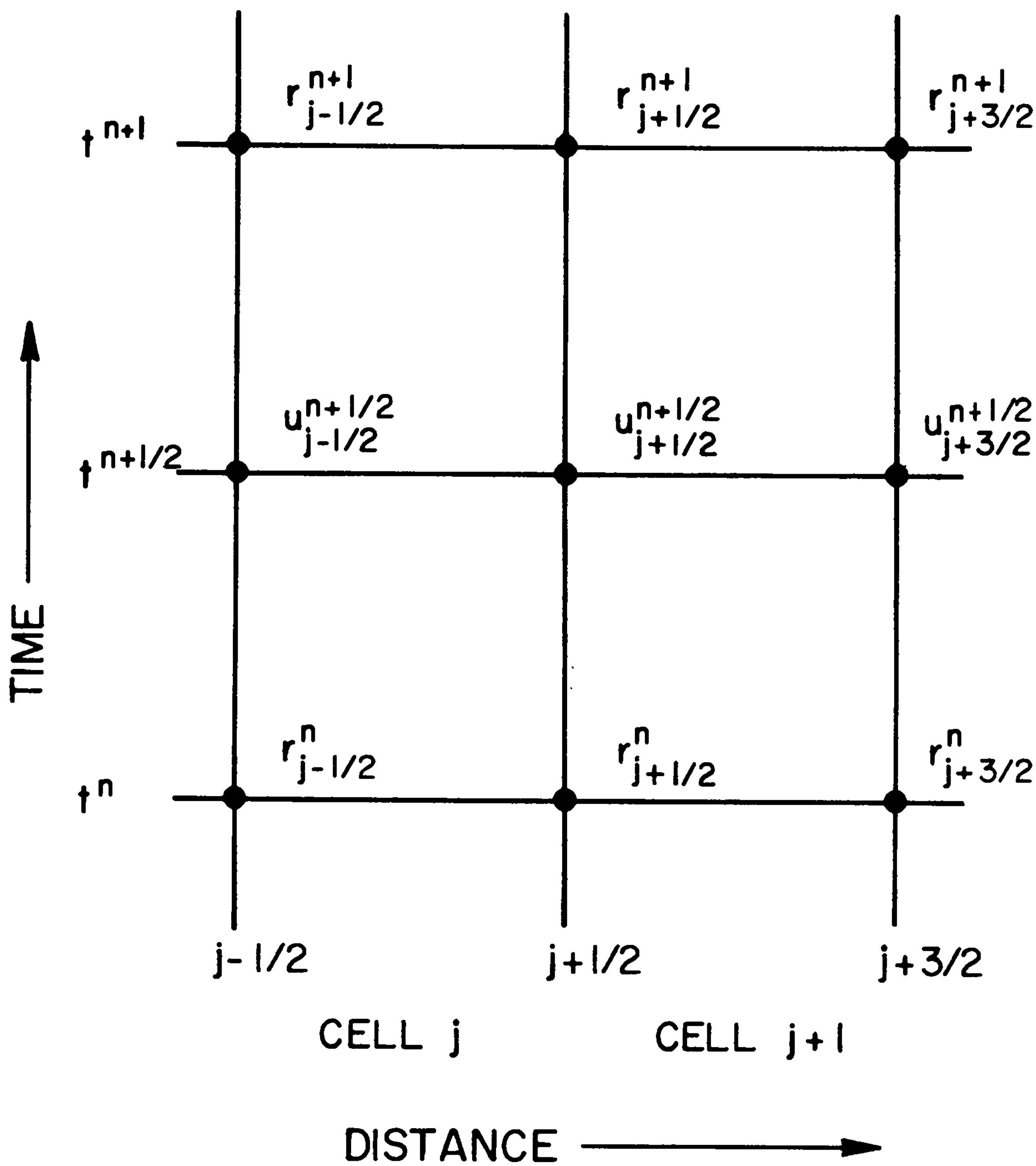



Fig. 8.
Space-time grid for the evaluation of cell quantities as used in HYDROX.

The choice of the time grid is slightly more difficult and we introduce the notion of a "half time step" to avoid confusion. By inspection of Eq. (39) of the previous section, assuming that we replace the spatial variable j by the time variable n, we see that knowing a quantity at time t = n and t = n+1 allows a straightforward determination of derivatives at t = n+½. The same argument applies for quantities known at t = n+½ and t = n+3/2 when we need to evaluate derivatives at t = n+1. Thus, we make the following choices at integer values of n (indicated by a superscript):

$$R^n_{j+\frac{1}{2}},\ V^n_j,\ \rho^n_j,\ \sigma^n_j,\ E^n_j,\ I^n_j \quad .$$

At half-integer values of n we choose

$$u^{n+\frac{1}{2}}_{j+\frac{1}{2}} \quad .$$

The mass of cell j, $M_j$, is a constant in time and does not need the time superscript.

The velocity is given by

$$u^{n+\frac{1}{2}}_{j+\frac{1}{2}} = \left(\frac{\partial R}{\partial t}\right)^{n+\frac{1}{2}}_{j+\frac{1}{2}} = \frac{R^{n+1}_{j+\frac{1}{2}} - R^n_{j+\frac{1}{2}}}{\Delta t^n} + O(\Delta t^2) \quad ,$$

($O(\Delta t^2)$ means order of $\Delta t$ squared; not to be confused with $\Delta t^n$ meaning the value of the time step for the $n^{th}$ cycle) which can be rewritten to give the position of the cell boundary at t = n+1, knowing the position at t = n and the velocity at t = n+½:

$$R^{n+1}_{j+\frac{1}{2}} = R^n_{j+\frac{1}{2}} + u^{n+1}_{j+\frac{1}{2}}\, \Delta t^n + O(\Delta t^3) \quad . \tag{42}$$

The volume of a cell can be calculated from a knowledge of the boundaries and the relation

$$dm = \rho\, r^{(d-1)}\, dr \quad ,$$

where d is the dimension (1, 2, or 3). Integrating both sides over a single cell of mass $M_j$, density $\rho_j$, from $R_{j-\frac{1}{2}}$ to $R_{j+\frac{1}{2}}$, we get

$$M_j = \rho_j \int_{R_{j-\frac{1}{2}}}^{R_{j+\frac{1}{2}}} r^{(d-1)}\, dr = \frac{\rho_j}{d}\left(R^d_{j+\frac{1}{2}} - R^d_{j-\frac{1}{2}}\right) \quad .$$

Using $V_j = \rho_j^{-1}$, we obtain for t = n

$$V_j^n = \frac{1}{M_j}\,\frac{1}{d}\left[\left(R^n_{j+\frac{1}{2}}\right)^d - \left(R^n_{j-\frac{1}{2}}\right)^d\right] \quad .$$

3. Momentum Equation

We want to write

$$\left(\frac{\partial u}{\partial t}\right)_r = -\left(\frac{\partial \sigma}{\partial m}\right)_t$$

in finite difference form, or more specifically, evaluate

$$\left(\frac{\partial u}{\partial t}\right)^n_{j+\frac{1}{2}} = -\left(\frac{\partial \sigma}{\partial m}\right)^n_{j+\frac{1}{2}} \quad .$$

For the left-hand side, we make Taylor series expansions of the velocity in time about t = n:

$$u^{n+\frac{1}{2}}_{j+\frac{1}{2}} = u^n_{j+\frac{1}{2}} + \left(\frac{\partial u}{\partial t}\right)^n_{j+\frac{1}{2}} \frac{\Delta t^n}{2} + \left(\frac{\partial^2 u}{\partial_t{}^2}\right)^n_{j+\frac{1}{2}} \frac{(\Delta t^n)^2}{8} + O(\Delta t^3) \quad ,$$

$$u^{n-\frac{1}{2}}_{j+\frac{1}{2}} = u^n_{j+\frac{1}{2}} - \left(\frac{\partial u}{\partial t}\right)^n_{j+\frac{1}{2}} \frac{\Delta t^n}{2} + \left(\frac{\partial^2 u}{\partial t^2}\right)^n_{j+\frac{1}{2}} \frac{(\Delta t^n)^2}{8} - O(\Delta t^3) \quad .$$

Subtracting and solving for $(\partial u/\partial t)^n_{j+\frac{1}{2}}$, we get

$$\left(\frac{\partial u}{\partial t}\right)^n_{j+\frac{1}{2}} = \frac{u^{n+\frac{1}{2}}_{j+\frac{1}{2}} - u^{n-\frac{1}{2}}_{j+\frac{1}{2}}}{\Delta t^n} + O(\Delta t^2) \quad . \tag{43}$$

For the stress derivatives, expand $\sigma^n_j$ and $\sigma^n_{j+1}$ about the point $j+\frac{1}{2}$ and denote the Lagrangian incremental spatial variable dm by $M_j$:

$$\sigma^n_{j+1} = \sigma^n_{j+\frac{1}{2}} + \left(\frac{\partial\sigma}{\partial m}\right)^n_{j+\frac{1}{2}} \frac{M_{j+1}}{2} + O(M^2_{j+1}) \quad ,$$

$$\sigma^n_j = \sigma^n_{j+\frac{1}{2}} - \left(\frac{\partial\sigma}{\partial m}\right)^n_{j+\frac{1}{2}} \frac{M_j}{2} + O(M^2_j) \quad .$$

Subtracting and solving for $(\partial\sigma/\partial m)^n_{j+\frac{1}{2}}$, we get

$$-\left(\frac{\partial\sigma}{\partial m}\right)^n_{j+\frac{1}{2}} = -\frac{\sigma^n_{j+1} - \sigma^n_j}{\frac{1}{2}(M_{j+1} + M_j)} + O(\Delta M) \quad . \tag{44}$$

where $\Delta M = M_{j+1} - M_j$. Our final result obtained from Eqs. (43) and (44) is

$$\frac{u^{n+\frac{1}{2}}_{j+\frac{1}{2}} - u^{n-\frac{1}{2}}_{j+\frac{1}{2}}}{\Delta t^n} = \frac{\sigma^n_j - \sigma^n_{j+1}}{\frac{1}{2}(M_{j+1} + M_j)} + O(\Delta M) + O(\Delta t^2) \quad ,$$

which can be used to obtain the velocity at $t = n+\frac{1}{2}$ by knowing the stresses at $t = n$ to $O(\Delta M \Delta t)$ in the cell size and $O(\Delta t^3)$ in the time step.

$$u^{n+\frac{1}{2}}_{j+\frac{1}{2}} = u^{n-\frac{1}{2}}_{j+\frac{1}{2}} + \frac{\sigma^n_j - \sigma^n_{j+1}}{\frac{1}{2}(M_{j+1} + M_j)} \Delta t^n + O(\Delta M \Delta t) + O(\Delta t^3) \quad . \tag{45}$$

## 4. Energy Equation

The energy equation is

$$\left(\frac{\partial E}{\partial t}\right)_r = -\frac{\partial}{\partial m}(\sigma u) \quad .$$

In finite difference form, we will want to evaluate this equation at $t = n+\frac{1}{2}$ for mass element $M_j$:

$$\left(\frac{\partial E}{\partial t}\right)_j^{n+\frac{1}{2}} = -\left(\frac{\partial \sigma u}{\partial m}\right)_j^{n+\frac{1}{2}} \quad .$$

For the left-hand side, we make a Taylor series expansion about $t = n+\frac{1}{2}$:

$$E_j^{n+1} = E_j^{n+\frac{1}{2}} + \left(\frac{\partial E}{\partial t}\right)_j^{n+\frac{1}{2}} \frac{\Delta t^n}{2} + \left(\frac{\partial^2 E}{\partial t^2}\right)_j^{n+\frac{1}{2}} \frac{(\Delta t^n)^2}{8} + O(\Delta t^3) \quad ,$$

$$E_j^{n} = E_j^{n+\frac{1}{2}} - \left(\frac{\partial E}{\partial t}\right)_j^{n+\frac{1}{2}} \frac{\Delta t^n}{2} + \left(\frac{\partial^2 E}{\partial t^2}\right)_j^{n+\frac{1}{2}} \frac{(\Delta t^n)^2}{8} - O(\Delta t^3) \quad .$$

Subtracting and solving for $(\partial E/\partial t)_j^{n+\frac{1}{2}}$, we get

$$\left(\frac{\partial E}{\partial t}\right)_j^{n+\frac{1}{2}} = \frac{E_j^{n+1} - E_j^n}{\Delta t^n} + O(\Delta t^2) \quad . \tag{46}$$

For the right-hand side, we note that the mass element $dm = M_j$, the $j^{th}$ Lagrangian coordinate. Expanding $\sigma u$, we get

$$(\sigma u)_{j+\frac{1}{2}}^{n+\frac{1}{2}} = (\sigma u)_j^{n+\frac{1}{2}} + \left(\frac{\partial \sigma u}{\partial m}\right)_j^{n+\frac{1}{2}} \frac{M_j}{2} + \left(\frac{\partial^2 \sigma u}{\partial m^2}\right)_j^{n+\frac{1}{2}} \frac{M_j^2}{8} + O(M_j^3) \quad ,$$

$$(\sigma u)_{j-\frac{1}{2}}^{n+\frac{1}{2}} = (\sigma u)_j^{n+\frac{1}{2}} - \left(\frac{\partial \sigma u}{\partial m}\right)_j^{n+\frac{1}{2}} \frac{M_j}{2} + \left(\frac{\partial^2 \sigma u}{\partial m^2}\right)_j^{n+\frac{1}{2}} \frac{M_j^2}{8} + O(M_j^3) \quad .$$

Subtracting and solving for the quantity of interest, we get

$$\left(\frac{\partial \sigma u}{\partial m}\right)_j^{n+\frac{1}{2}} = \frac{(\sigma u)_{j+\frac{1}{2}}^{n+\frac{1}{2}} - (\sigma u)_{j-\frac{1}{2}}^{n+\frac{1}{2}}}{M_j} + O(M_j^2) \quad . \tag{47}$$

We still have the task of evaluating $(\sigma u)^{n+\frac{1}{2}}_{j+\frac{1}{2}}$. This can be written as

$$(\sigma u)^{n+\frac{1}{2}}_{j+\frac{1}{2}} = \sigma^{n+\frac{1}{2}}_{j+\frac{1}{2}}\, u^{n+\frac{1}{2}}_{j+\frac{1}{2}} \quad ,$$

and $u^{n+\frac{1}{2}}_{j+\frac{1}{2}}$ is calculated from Eq. (45). For the stress, we shall approximate the time evaluation by

$$\sigma^{n+\frac{1}{2}}_{j+\frac{1}{2}} = \frac{\sigma^{n}_{j+\frac{1}{2}} + \sigma^{n+1}_{j+\frac{1}{2}}}{2} + O(\Delta t^2)$$

$$= \sigma^{n}_{j+\frac{1}{2}} + \frac{1}{2}\left(\sigma^{n+1}_{j+\frac{1}{2}} - \sigma^{n}_{j+\frac{1}{2}}\right) \cong \sigma^{n}_{j+\frac{1}{2}} + O(\Delta t) \quad .$$

$\sigma^{n+1}_{j+\frac{1}{2}}$ will be evaluated later from the equation of state using the results for $V^{n+1}_{j}$ and $I^{n+1}_{j}$.

To evaluate the stress at the cell interface, we use Eq. (40):

$$\sigma^{n}_{j+\frac{1}{2}} = \frac{M_{j+1}\sigma^{n}_{j} + M_{j}\sigma^{n}_{j+1}}{M_{j} + M_{j+1}} + O(M_j^2) \quad . \tag{48}$$

We can now write the result using Eqs. (46)-(48):

$$\frac{E^{n+1}_{j} - E^{n}_{j}}{\Delta t^{n}} = -\frac{1}{M_j}\left\{\left[\frac{M_{j+1}\sigma^{n}_{j} + M_{j}\sigma^{n}_{j+1}}{M_{j} + M_{j+1}}\right]u^{n+\frac{1}{2}}_{j+\frac{1}{2}} - \left[\frac{M_{j}\sigma^{n}_{j-1} + M_{j-1}\sigma^{n}_{j}}{M_{j-1} + M_{j}}\right]u^{n+\frac{1}{2}}_{j-\frac{1}{2}}\right\}$$
$$+ O(M_j^2) + O(\Delta t) \quad . \tag{49}$$

This equation can then be used to solve for $E^{n+1}_{j}$.

5. Kinetic and Internal Energies

We shall define the internal energy $I_j$ by the relations

$$E_j^{n+1} = I_j^{n+1} + \frac{1}{2}\left(u_j^{n+\frac{1}{2}}\right)^2 \quad , \tag{50}$$

$$E_j^{n} = I_j^{n} + \frac{1}{2}\left(u_j^{n-\frac{1}{2}}\right)^2 \quad . \tag{51}$$

To calculate $u_j^{n+\frac{1}{2}}$, we expand about j to get

$$u_{j+\frac{1}{2}}^{n+\frac{1}{2}} = u_j^{n+\frac{1}{2}} + \left(\frac{\partial u}{\partial m}\right)_j^{n+\frac{1}{2}} \frac{M_j}{2} + O(M_j^2) \quad ,$$

$$u_{j-\frac{1}{2}}^{n+\frac{1}{2}} = u_j^{n+\frac{1}{2}} - \left(\frac{\partial u}{\partial m}\right)_j^{n+\frac{1}{2}} \frac{M_j}{2} + O(M_j^2) \quad .$$

Adding gives

$$u_j^{n+\frac{1}{2}} = \frac{1}{2}\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right) + O(M_j^2) \quad .$$

Substituting the result in Eqs. (50) and (51), combining with Eq. (49), and solving for $I_j^{n+1}$, we get

$$\begin{aligned} I_j^{n+1} = I_j^n &+ \frac{1}{8}\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right)^2 - \frac{1}{8}\left(u_{j+\frac{1}{2}}^{n-\frac{1}{2}} + u_{j-\frac{1}{2}}^{n-\frac{1}{2}}\right)^2 \\ &+ \frac{\Delta t^n}{M_j}\left\{\left[\frac{M_j\sigma_{j-1}^n + M_{j-1}\sigma_j^n}{M_{j-1} + M_j}\right]u_{j-\frac{1}{2}}^{n+\frac{1}{2}} - \left[\frac{M_{j+1}\sigma_j^n + M_j\sigma_{j+1}^n}{M_j + M_{j+1}}\right]u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right\} \\ &+ O(M_j^2\Delta t) + O(\Delta t^2) \quad . \end{aligned} \tag{52}$$

This result is the expression used for the SIN difference equations.

For the HYDROX difference equation, the change in internal energy is calculated from the results of Sec. 4 above. In planar geometry, the time rate of change for the internal energy I is given by

$$\left(\frac{\partial I}{\partial t}\right)_r = -\sigma\left(\frac{\partial u}{\partial m}\right)_t \quad . \tag{53}$$

We can evaluate $I_j^{n+1}$ by using a Taylor series expansion at t = n:

$$I_j^{n+1} = I_j^n + \Delta t^n\left(\frac{\partial I}{\partial t}\right)_j^n + O(\Delta t^2) \quad ,$$

which can be written, using Eq. (53), as

$$I_j^{n+1} = I_j^n - \Delta t^n \sigma_j^n\left(\frac{\partial u}{\partial m}\right)_j^n + O(\Delta t^2) \quad . \tag{54}$$

The derivative in Eq. (54) can be readily evaluated from Taylor series expansions about t = n and j:

$$u_{j+\frac{1}{2}}^{n+\frac{1}{2}} = u_{j+\frac{1}{2}}^n + \frac{\Delta t^n}{2}\left(\frac{\partial u}{\partial t}\right)_{j+\frac{1}{2}}^n + O(\Delta t^2) \quad , \tag{55}$$

$$u_{j+\frac{1}{2}}^{n-\frac{1}{2}} = u_{j+\frac{1}{2}}^n - \frac{\Delta t^n}{2}\left(\frac{\partial u}{\partial t}\right)_{j+\frac{1}{2}}^n + O(\Delta t^2) \quad , \tag{56}$$

$$u_{j+\frac{1}{2}}^n = u_j^n + \frac{M_j}{2}\left(\frac{\partial u}{\partial m}\right)_j^n + \frac{M_j^2}{8}\left(\frac{\partial^2 u}{\partial m^2}\right)_j^n + O(M_j^3) \quad , \tag{57}$$

$$u_{j-\frac{1}{2}}^n = u_j^n - \frac{M_j}{2}\left(\frac{\partial u}{\partial m}\right)_j^n + \frac{M_j^2}{8}\left(\frac{\partial^2 u}{\partial m^2}\right)_j^n + O(M_j^3) \quad . \tag{58}$$

Adding Eq. (55) and Eq. (56), we get

$$u_{j+\frac{1}{2}}^n = \frac{1}{2}\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + u_{j+\frac{1}{2}}^{n-\frac{1}{2}}\right) + O(\Delta t^2) \quad . \tag{59}$$

Subtracting Eq. (58) from Eq. (57) leads to

$$\left(\frac{\partial u}{\partial m}\right)_j^n = \frac{u_{j+\frac{1}{2}}^n - u_{j-\frac{1}{2}}^n}{M_j} + O(M_j^2) \quad . \tag{60}$$

Combining Eq. (59) and Eq. (60), we get

$$\left(\frac{\partial u}{\partial m}\right)_j^n = \frac{1}{2M_j}\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + u_{j+\frac{1}{2}}^{n-\frac{1}{2}} - u_{j-\frac{1}{2}}^{n+\frac{1}{2}} - u_{j-\frac{1}{2}}^{n-\frac{1}{2}}\right) + O(M_j^2) + O(\Delta t^2) \quad . \tag{61}$$

Inserting Eq. (61) in Eq. (54), we have the HYDROX difference equation for internal energy:

$$I_j^{n+1} = I_j^n - \frac{\Delta t^n \sigma_j^n}{2M_j}\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + u_{j+\frac{1}{2}}^{n-\frac{1}{2}} - u_{j-\frac{1}{2}}^{n+\frac{1}{2}} - u_{j-\frac{1}{2}}^{n-\frac{1}{2}}\right) + O(\Delta t^2) + O(M_j^2) \quad .$$

One can, at the expense of iterating on the equation of state, get a result for the internal energy with error $O(\Delta t^3)$ provided the velocities are calculated to $O(\Delta t^3)$. The error in Eq. (45) reduces to $O(\Delta t^3)$ for the special case of all $M_j$'s equal. Future versions of HYDROX will include this iterative difference equation as an option.

## 6. Conservation Properties for the Difference Equations

In this section we shall investigate to what degree our difference equations conserve momentum and energy. To do this we sum the total momentum and energy of the system at two different times and compare the results. At $t = n+\frac{1}{2}$, the total momentum is

$$(Mu)^{n+\frac{1}{2}} = \sum_{j=0}^{N} \frac{1}{2}(M_j + M_{j+1})u_{j+\frac{1}{2}}^{n+\frac{1}{2}} \quad ,$$

where N is the number of cells. Using Eq. (45), we have

$$(Mu)^{n+\frac{1}{2}} = \sum_{j=0}^{N} \frac{1}{2}(M_j + M_{j+1})u_{j+\frac{1}{2}}^{n-\frac{1}{2}} + \sum_{j=0}^{N} \Delta t^n(\sigma_j^n - \sigma_{j+1}^n) + O(\Delta t^3) + O(\Delta M \Delta t)$$

$$= (Mu)^{n-\frac{1}{2}} + \Delta t^n \sum_{j=0}^{N} (\sigma_j^n - \sigma_{j+1}^n) + O(\Delta t^3) + O(\Delta M \Delta t)$$

$$= (Mu)^{n-\frac{1}{2}} + \Delta t^n \left[(\sigma_0^n - \sigma_1^n) + (\sigma_1^n - \sigma_2^n) + \cdots + (\sigma_{N-1}^n - \sigma_N^n) + (\sigma_N^n - \sigma_{N+1}^n)\right]$$
$$+ O(\Delta t^3) + O(\Delta M \Delta t)$$

$$= (Mu)^{n-\frac{1}{2}} + \Delta t^n(\sigma_0^n - \sigma_{N+1}^n) + O(\Delta t^3) + O(\Delta M \Delta t)$$

But the j = 0 and j = N+1 are effectively boundary cells that give free surface boundary conditions such that $M_0$ and $M_{n+1} = 0$ and $\sigma_0^n = \sigma_{N+1}^n = 0$ giving our conservation of momentum result,

$$(Mu)^{n+\frac{1}{2}} = (Mu)^{n-\frac{1}{2}} + O(\Delta t^3) + O(\Delta M \Delta t) \quad .$$

The total energy at t = n+1 is

$$(ME)^{n+1} = \sum_{j=1}^{N} M_j E_j^{n+1} \quad .$$

Using Eq. (49), we have for the SIN difference equations:

$$(ME)^{n+1} = \sum_{j=1}^{N} \left\{ M_j E_j^n + \Delta t^n \left( \left[ \frac{M_j \sigma_{j-1}^n + M_{j-1}\sigma_j^n}{M_j + M_{j-1}} \right] u_{j-\frac{1}{2}}^{n+\frac{1}{2}} - \left[ \frac{M_{j+1}\sigma_j^n + M_j \sigma_{j+1}^n}{M_j + M_{j+1}} \right] u_{j+\frac{1}{2}}^{n+\frac{1}{2}} \right) + O(M_j \Delta t^2) \right\} \quad .$$

When the summation over j is performed, the first term is just the total energy at t = n. In the second term, let k = j-1 such that

$$\sum_{j=1}^{N} \rightarrow \sum_{k=0}^{N-1} \quad \text{and} \quad \begin{matrix} j \rightarrow k+1 \\ j-\frac{1}{2} \rightarrow k+\frac{1}{2} \end{matrix} \quad .$$

In a similar manner to the conservation of momentum calculation above, all terms cancel between the two summations except the k=0 term in the first summation and the j=N term in the second. Again, free surface boundary conditions give effective values of $M_0 = M_{N+1} = 0$ and $\sigma_0^n = \sigma_{N+1}^n = 0$ to obtain

$$(ME)^{n+1} = (ME)^n + O(M_j \Delta t^2) \quad .$$

Similarly, for the HYDROX difference equations, the conservation of total energy can be evaluated. The change in internal energy is given by

$$\Delta(MI) = \sum_{j=0}^{N} M_j (I_j^{n+1} - I_j^n)$$

$$= \sum_{j=0}^{N} M_j \sigma_j \frac{\left(u_{j-\frac{1}{2}}^{n+\frac{1}{2}} + u_{j-\frac{1}{2}}^{n-\frac{1}{2}} - u_{j+\frac{1}{2}}^{n+\frac{1}{2}} - u_{j+\frac{1}{2}}^{n-\frac{1}{2}}\right)\Delta t^n}{2M_j} + O(M_j \Delta t^2) + O(M_j^3) \quad ,$$

which can be rewritten as

$$\Delta(MI) = -\sum_{j=0}^{N} (\sigma_j^n - \sigma_{j+1}^n) \frac{1}{2}\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + u_{j+\frac{1}{2}}^{n-\frac{1}{2}}\right)\Delta t + O(M_j \Delta t^2) + O(M_j^3) \quad .$$

The change in kinetic energy can be written as

$$\Delta \frac{1}{2} Mu^2 = \frac{1}{2} \sum_{j=0}^{N} \frac{1}{2} (M_j + M_{j+1}) \left[ \left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right)^2 - \left(u_{j+\frac{1}{2}}^{n-\frac{1}{2}}\right)^2 \right] + O(M_j \Delta M \Delta t) \quad ,$$

which can be rewritten, using Eq. (45), to get

$$\Delta\left(\frac{1}{2}\, Mu^2\right) = \sum_{j=0}^{N} \left(\sigma_j^n - \sigma_{j+1}^n\right) \frac{1}{2}\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + u_{j+\frac{1}{2}}^{n-\frac{1}{2}}\right)\Delta t + O(M_j \Delta M \Delta t) \quad .$$

The total energy is then conserved to $O(M_j \Delta t^2) + O(M_j^3) + O(M_j \Delta M \Delta t)$.

## III. INPUT AND OUTPUT

The input file for HYDROX is a namelist read file called DATA. HYDROX creates the following output files.

DOUT - a summary of all material and EOS constants for the entire problem

XOUT - the cycle print file

OUTPUT - a summary of material energies plus records of zoning, spalling, EOS errors, void closures, and restart dumps

GASSIN - random access graphics file ready to be processed by the LTSS utility GAS (LTSS-523)

DUMPO - a dump file for restarting a problem

Section A describes the variables for the input file DATA. Section B describes the output files. Section C tells how to process the GASSIN file for graphical output.

### Defaults

The default value for all parameters listed in the namelist statements is 0 unless otherwise specified.

## A. Namelist Input for the Input File DATA

Problem input to HYDROX is handled through the file DATA. The structure of the file is given by:

| | | | |
|---|---|---|---|
| P$INP | Parameters for problem control | $ | Required |
| - - - | - - - | - - - | - - - |
| P$SU | Parameters for Material 1 | $ | Required |
| P$ESC | EOS constants for Material 1 | $ | Required only if ME $\neq$ 0 in the SU namelist |
| P$BURN | Reactive EOS constants for Material 1 | $ | Required only if IBRN $\neq$ 0 in the ESC namelist or in data read from a library |
| - - - | - - - | - - - | - - - |
| P$SU | Parameters for Material 2 | $ | Required |
| P$ESC | EOS constants for Material 2 | $ | Required for ME $\neq$ 0 |
| P$BURN | Reactive EOS constants for Material 2 | $ | Required for IBRN $\neq$ 0 |
| - - - | - - - | - - - | - - - |

Repeat SU, ESC, BURN for each material

1. Namelist INP

NM = number of material regions.

IALPH = 1,2,3, for plane, cylindrical, spherical geometry (default = 3).

LABEL = up to 80 characters of Hollerith data to be used as a label on the printout. Also, the first 30 characters will be used as a label on the GAS plots.

TEND = ending time (if NI is large enough). For TEND = 0, no check for TEND is made.

NI = maximum # of cycles the problem may run (default = 10000).

NDF = type of difference equations used. 1 = HYDROX, 2 = SIN (default = 1).

ND = approximate # of cells in the problem if the automatic zoner is used (default = 180).

MSFF = flag to use Multiple-Shock Forest Fire (MSFF = 1) instead of the usual Forest Fire (default).

PRINT and GASSIN Dump Controls

NP = print every NP cycles. For NP $\leq$ 0, no check for cycle print is made.

NG = GAS dump every NG cycles. For NG $\leq$ 0, no check for cycle GAS dump is made.

TP = $t_1,\Delta t_1,t_2,\Delta t_2,\cdots,\Delta t_{n-1},t_n$; print every $\Delta t_1$ μs from $t_1$ to $t_2$, every $\Delta t_2$ from $t_2$ to $t_3$, etc. (must end with $t_n$, not $\Delta t_n$). For $\Delta t \leq 0$, no check for time prints is made.

TG = same as TP except for GAS dump instead of print.

Automatic Time Step Parameters

NDELT = 0 for automatic time step control; 1 for $\Delta t$ = DTO of the last active material region.

DTCF = automatic time step control parameter; $\Delta t$ = DTCF* $\Delta X/C$ for all materials (see SU namelist).

## Active Cell Control Parameters

NADD = add NADD new cells when the last cell becomes active.

NMAX = # of cells used initially (for NADD $\leqslant$ 0, all cells are used).

## Piston Boundary Conditions

UI = initial piston velocity for HE initiation (no piston if UI = 0).

UF = final piston velocity for HE initiation.

RO = initial radius for piston.

UII = same as UI except for inside piston rather than outside piston.

UFI = same as UF except for inside piston rather than outside piston.

## Restart Control Parameters

NM1 = minimum region # for which data will be read in (default = 1). Used primarily for a restart in which regions NM1 to NM are changed or added. If NM1 = 1 for a restart, no new data is read in except for that in the INP namelist.

IDMP = restart the problem at the IDMPth dump. If IDMP = 0, initialize problem from the data set.

IV = see SU namelist. Used in INP namelist only for restart with NM1 > 1 where IV(NM1-1) is set. (Default = -1.)

NDUMP = make a restart dump every NDUMP cycles. After MXDUMP (set in parameter statement, usually = 30) dumps, the code will stop. (Default = 10,000.)

TD = same as TP except for restart dumps.

## SESAME Interpolation Option

IFN = 0 for rational function algorithm (default), 1 for bilinear algorithm.

## 2. Namelist SU

### EOS Specifications

IEOS = type of EOS: 1 = HOM, 2 = buildup, 3 = 8-parameter fit, 4 = SESAME (default = 1).

MAT = EOS number in library for that type of EOS. A library is not searched if MAT = 0.

ME = 0 for no changes in library values, 1 for library values changed by the ESC (and sometimes BURN) namelist(s). ME must be 1 if MAT = 0.

### Initial Positions and Velocity

R1 = outside radius for this material region (default = R0 for the first region; default = R2 of the previous region for other regions).

R2 = inside radius for this material region (required).

U0 = initial velocity for each cell in this region.

### Zoning

NCI = number of cells in this region (must be at least 2 if used).

DR1, DR2 } for NCI = 0 (default) and DR1 > 0, NCI and DR (for each cell) are computed using DR1 and DR2. DR1 is the cell size for the outside cell of the region and DR2 is approximately the cell size for the inside cell of the region. The cell size varies linearly with cell number and DR2 is adjusted so that an integer number of cells is required. For NCI = 0 and DR1 $\leqslant$ 0, an automatic zoning scheme (described in SETUP) is used.

NOSPLT = 1 calculates and allows rezoning; $\leqslant$0 doesn't even check for rezoning.

### Voids

IV = void index for the interface between this material and the next: -1 = no void (the two materials are "glued together" so that under tension they remain in contact), 0 = open void, +1 = closed void (which becomes an open void under tension)(default = -1).

Time-Step Controls

DTO = maximum allowed time step when this material region is the last active region. If DTO $\leq$ 0, it is replaced by DTO for the first region (which should not be 0). (Default = 1.)

DTCF = Automatic time-step control parameter for this material only; $\Delta t$ = DTCF* $\Delta X/C$ (see DELT for details, default = 0.5).

Active Cell Control Parameters

UT = absolute value of the velocity that must be exceeded before the cell becomes active (default = $10^{-10}$).

GAS Dump

IJK = GAS dumps. Include every IJK'th cell (default = 1).

3. Namelist ESC - Read Only if ME ≠ in the Immediately Preceding SU Namelist

ROW = initial density.

VO = initial volume.

PO = initial pressure.

XISP = pressure at which a closed void opens.

IE = initial region # (default, IE(I) = I).

QO = pressure at which viscosity is turned on (otherwise noise can make the problem unstable in regions where nothing should be happening) (default = $10^{-10}$).

TO = initial temperature (for IEOS = 4 and TO ≠ 0, ZI (see below) is calculated using density and temperature as given quantities).

ZI = initial specific internal energy for all cells in a region.

HOM Parameters (IEOS = 1)
Library = HMLB

$$\left.\begin{array}{l}\text{C1}\\ \text{S1}\end{array}\right\} \begin{array}{l}C_0\\ S\end{array} \text{ of } U_S = C_0 + SU_p$$

$$\left.\begin{array}{l}\text{C2}\\ \text{S2}\\ \text{SWV}\end{array}\right\} \text{second set of constants that are switched to when the volume is} \leq \text{SWV.}$$

VMN = VMN < volume < SWV the volume is set to VMN in the EOS calculation.

GAMMA = Grüneisen $\gamma$ (constant for HOM).

ALP = thermal expansion coefficient $\alpha$ used for HOM EOS in tension.

FS,GS,HS,SI,SJ = HOM parameters for the solid temperature fit (F,G,H,I,J).

CV = specific heat of the solid.

GC = array GC containing the HOM GAS constants (A,B,C,D,E,K,L,M,N,O,Q, R,S,T,U,$C_V'$, and Z).

XL = thermal heat conductivity coefficient (not currently used).

XMU = $\mu$ = shear modulus.

YO = 2/3 $Y_0$.

PLAP = PLAP (see EPP).

TMLT = melt temperature at normal density.

TMC = melt constant for linear function of specific volume.

## HOM Parameters for Reactive Materials

IBRN = type of burn: 0 = no burn, 1 = Arrhenius, 2 = CJ, 3 = sharp shock, 4 = Forest Fire. 5 = FF (temperature), 6 = FF (internal energy), 7 = gamma-law Taylor wave. If IBRN $\neq$ 0, the BURN namelist is read.

WO = initial burn fraction. Default = 1 - all solid, no products.

## HE Buildup EOS (IEOS = 2)
## Library = HMLB

BUA = A

BUB = B

BUMAX = $\gamma_{max}$

BUDV = D

where, for the detonation products, $\gamma_g$ = A + B/X and X = distance from the detonation point, with the constraint $\gamma \leq \gamma_{max}$.

BUR = shift in effective distance of run.

BUD = region over which the "break" in $\gamma$ is smoother (default = 0.2).

## 8-Parameter Polynomial Fits (IEOS = 3)

CF = 8-parameter EOS constants (see POLY).

## SESAME EOS (IEOS = 4)
## Library = SES2L

SR = density scale factor (default = 1).

ES = energy shift (Mbar - $cm^3/g$).

A1,A2,A3 = ramp parameters with a ramp pressure given by P = $MIN\{A1*(\rho/\rho_0 - 1), A2*(\rho/\rho_0 - A_3)\}$.

IRV = reversible (0)/irreversible flag (1).

EM = "melt" energy. For XI(J) > EM, a flag is set to turn off the ramp for that cell for the remainder of the problem.

<u>Barnes EOS - Used with HOM</u>

A,BR,BA,VBO,VBSW = constants used in Barnes EOS.

<u>Spall Parameters - Used with HOM</u>

SP = SPA in SIN (coefficient for the gradient spall pressure).

USP = ultimate spall pressure.

<u>Viscosity Parameters</u>

NV = viscosity type: 0 = "real," 1 = PIC (default), 2 = Landshoff.

XV = viscosity coefficient (default = 2.0).

## 4. Namelist BURN - Read Only if IBRN $\neq$ 0 in the Immediately Preceding ESC Namelist

Z = frequency for Arrhenius burn.

E = activation energy for Arrhenius burn.

VCJ = CJ volume for CJ burn.

PCJ = pressure at which W is set to 0 for Forest Fire burn.

DWDT = Forest Fire constants (up to 20).

PM = pressure below which dW/dt is assumed to be 0 for Forest Fire burn.

ND = # of Forest Fire constants.

SAMPLE DATA DECK

```
P$INP LABEL=50HSYMMETRIC PLATE IMPACT CU/CU
      NM=2,
      IALPH=1,
      TEND=4.,
      TP=0., 1., 4.0,
      TG=0., 0.5, 2., 0.1, 4.0,
      $
P$SU IEOS=1, MAT=4,
      R1=10., R2=9.,
      U0=-0.030,
      NCI=100
      $
P$SU IEOS=1, MAT=4,
      R2=8.,
      NCI=100,
      $
```

## B. Output Files

In this section we present sample listings of the output files DOUT, XOUT, and OUTPUT. Two additional output files are also created during the execution of HYDROX: GASSIN for use by GAS, and DUMPO, a restart dump file.

1. DOUT. The file DOUT is written after the input namelists have been read and the problem has been set up. DOUT provides a means of checking and verifying the problem input. A sample DOUT listing is given in the following pages. The variables in the namelists INP, SU, and ESC are printed in the order they appear. See Section VI.A for a list of variables. Array variables are printed for all values of the array indices. Since many of the variables are dimensioned for the number of materials allowed, there can be many zeros. In the sample there were only 2 materials in the calculation, but HYDROX was compiled to allow 20.

## SAMPLE DOUT LISTING

### Begin Namelist INP

```
$INP    --

NM------------------------------         2
UI------------------------------   0.
UF------------------------------   0.
RO------------------------------   0.
IALPH---------------------------         1
NDF-----------------------------         1
NI------------------------------     10000
NP------------------------------         0
NG------------------------------         0
NAOO----------------------------         0
NMAX----------------------------       201
TENO----------------------------   4.0000000000000E+00
TP(1)---------------------------   0.                     1.0000000000000E+00    4.0000000000000E+00
TP(4)---------------------------   0.                     0.                     0.
TP(7)---------------------------   0.                     0.                     0.
TP(10)--------------------------   0.                     0.                     0.
TP(13)--------------------------   0.                     0.                     0.
TP(16)--------------------------   0.                     0.                     0.
TP(19)--------------------------   0.                     0.                     0.
TG(1)---------------------------   0.                     5.0000000000000E-01    2.0000000000000E+00
TG(4)---------------------------   1.0000000000000E-01    4.0000000000000E+00    0.
TG(7)---------------------------   0.                     0.                     0.
TG(10)--------------------------   0.                     0.                     0.
TG(13)--------------------------   0.                     0.                     0.
TG(16)--------------------------   0.                     0.                     0.
TG(19)--------------------------   0.                     0.                     0.
NOELT---------------------------         0
UII-----------------------------   0.
UFI-----------------------------   0.
LABEL(1)------------------------ **********  **********  **********          0          0          0          0
LABEL(8)------------------------         0
NM1-----------------------------         1
IOMP----------------------------         0
IV(1)---------------------------        -1        -1        -1        -1        -1        -1        -1
IV(8)---------------------------        -1        -1        -1        -1        -1        -1        -1
IV(15)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(22)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(29)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(36)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(43)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(50)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(57)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(64)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(71)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(78)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(85)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(92)--------------------------        -1        -1        -1        -1        -1        -1        -1
IV(99)--------------------------        -1        -1
NOUMP---------------------------     10000
TO(1)---------------------------   0.                     0.                     0.
TO(4)---------------------------   0.                     0.                     0.
TO(7)---------------------------   0.                     0.                     0.
TO(10)--------------------------   0.                     0.                     0.
TO(13)--------------------------   0.                     0.                     0.
TO(16)--------------------------   0.                     0.                     0.
TO(19)--------------------------   0.                     0.                     0.
IFN-----------------------------         0
NO------------------------------       180
OTCF(1)-------------------------   5.0000000000000E-01    5.0000000000000E-01    5.0000000000000E-01
OTCF(4)-------------------------   5.0000000000000E-01    5.0000000000000E-01    5.0000000000000E-01
OTCF(7)-------------------------   5.0000000000000E-01    5.0000000000000E-01    5.0000000000000E-01
```

```
OTCF(10)------------------------   5.0000000000000E-01   5.0000000000000E-01   5.0000000000000E-01
OTCF(13)------------------------   5.0000000000000E-01   5.0000000000000E-01   5.0000000000000E-01
OTCF(16)------------------------   5.0000000000000E-01   5.0000000000000E-01   5.0000000000000E-01
OTCF(19)------------------------   5.0000000000000E-01   5.0000000000000E-01   5.0000000000000E-01
$END
```

End Namelist INP

```
$SU     --
```

Begin Namelist SU

```
OTO(1)--------------------------   1.0000000000000E+00   1.0000000000000E+00   1.0000000000000E+00
OTO(4)--------------------------   1.0000000000000E+00   1.0000000000000E+00   1.0000000000000E+00
OTO(7)--------------------------   1.0000000000000E+00   1.0000000000000E+00   1.0000000000000E+00
OTO(10)-------------------------   1.0000000000000E+00   1.0000000000000E+00   1.0000000000000E+00
OTO(13)-------------------------   1.0000000000000E+00   1.0000000000000E+00   1.0000000000000E+00
OTO(16)-------------------------   1.0000000000000E+00   1.0000000000000E+00   1.0000000000000E+00
OTO(19)-------------------------   1.0000000000000E+00   1.0000000000000E+00   1.0000000000000E+00
NOSPLT(1)-----------------------          0        0        0        0        0        0        0
NOSPLT(8)-----------------------          0        0        0        0        0        0        0
NOSPLT(15)----------------------          0        0        0        0        0        0        0
NOSPLT(22)----------------------          0        0        0        0        0        0        0
NOSPLT(29)----------------------          0        0        0        0        0        0        0
NOSPLT(36)----------------------          0        0        0        0        0        0        0
NOSPLT(43)----------------------          0        0        0        0        0        0        0
NOSPLT(50)----------------------          0        0        0        0        0        0        0
NOSPLT(57)----------------------          0        0        0        0        0        0        0
NOSPLT(64)----------------------          0        0        0        0        0        0        0
NOSPLT(71)----------------------          0        0        0        0        0        0        0
NOSPLT(78)----------------------          0        0        0        0        0        0        0
NOSPLT(85)----------------------          0        0        0        0        0        0        0
NOSPLT(92)----------------------          0        0        0        0        0        0        0
NOSPLT(99)----------------------          0        0
IV(1)---------------------------         -1       -1       -1       -1       -1       -1       -1
IV(8)---------------------------         -1       -1       -1       -1       -1       -1       -1
IV(15)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(22)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(29)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(36)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(43)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(50)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(57)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(64)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(71)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(78)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(85)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(92)--------------------------         -1       -1       -1       -1       -1       -1       -1
IV(99)--------------------------         -1       -1
IEOS(1)-------------------------          1        1        1        1        1        1        1
IEOS(8)-------------------------          1        1        1        1        1        1        1
IEOS(15)------------------------          1        1        1        1        1        1        1
MAT(1)--------------------------          4        4        4        0        0        0        0
MAT(8)--------------------------          0        0        0        0        0        0        0
MAT(15)-------------------------          0        0        0        0        0        0        0
ME(1)---------------------------          1        1        1        0        0        0        0
ME(8)---------------------------          0        0        0        0        0        0        0
ME(15)--------------------------          0        0        0        0        0        0        0
R1------------------------------   8.0000000000000018E+00
R2------------------------------   8.0000000000000018E+00
U0(1)---------------------------  -3.0000000000000E-02   0.                    0.
U0(4)---------------------------   0.                    0.                    0.
U0(7)---------------------------   0.                    0.                    0.
U0(10)--------------------------   0.                    0.                    0.
U0(13)--------------------------   0.                    0.                    0.
U0(16)--------------------------   0.                    0.                    0.
U0(19)--------------------------   0.                    0.                    0.
```

```
NCI----------------------------      100
OR1----------------------------    0.
OR2----------------------------    0.
UT(1)--------------------------    1.0000000000000E-10    1.0000000000000E-10    1.0000000000000E-10
UT(4)--------------------------    1.0000000000000E-10    1.0000000000000E-10    1.0000000000000E-10
UT(7)--------------------------    1.0000000000000E-10    1.0000000000000E-10    1.0000000000000E-10
UT(10)-------------------------    1.0000000000000E-10    1.0000000000000E-10    1.0000000000000E-10
UT(13)-------------------------    1.0000000000000E-10    1.0000000000000E-10    1.0000000000000E-10
UT(16)-------------------------    1.0000000000000E-10    1.0000000000000E-10    1.0000000000000E-10
UT(19)-------------------------    1.0000000000000E-10    1.0000000000000E-10    1.0000000000000E-10
OTCF(1)------------------------    5.0000000000000E-01    5.0000000000000E-01    5.0000000000000E-01
OTCF(4)------------------------    5.0000000000000E-01    5.0000000000000E-01    5.0000000000000E-01
OTCF(7)------------------------    5.0000000000000E-01    5.0000000000000E-01    5.0000000000000E-01
OTCF(10)-----------------------    5.0000000000000E-01    5.0000000000000E-01    5.0000000000000E-01
OTCF(13)-----------------------    5.0000000000000E-01    5.0000000000000E-01    5.0000000000000E-01
OTCF(16)-----------------------    5.0000000000000E-01    5.0000000000000E-01    5.0000000000000E-01
OTCF(19)-----------------------    5.0000000000000E-01    5.0000000000000E-01    5.0000000000000E-01
$END
```

End Namelist SU

```
$ESC   --
```

Begin Namelist ESC*

```
C1(1)--------------------------    3.9580000000000E-01    3.9580000000000E-01    3.9580000000000E-01
C1(4)--------------------------    0.                     0.                     0.
C1(7)--------------------------    0.                     0.                     0.
C1(10)-------------------------    0.                     0.                     0.
C1(13)-------------------------    0.                     0.                     0.
C1(16)-------------------------    0.                     0.                     0.
C1(19)-------------------------    0.                     0.                     0.
S1(1)--------------------------    1.4970000000000E+00    1.4970000000000E+00    1.4970000000000E+00
S1(4)--------------------------    0.                     0.                     0.
S1(7)--------------------------    0.                     0.                     0.
S1(10)-------------------------    0.                     0.                     0.
S1(13)-------------------------    0.                     0.                     0.
S1(16)-------------------------    0.                     0.                     0.
S1(19)-------------------------    0.                     0.                     0.
C2(1)--------------------------    0.                     0.                     0.
C2(4)--------------------------    0.                     0.                     0.
C2(7)--------------------------    0.                     0.                     0.
C2(10)-------------------------    0.                     0.                     0.
C2(13)-------------------------    0.                     0.                     0.
C2(16)-------------------------    0.                     0.                     0.
C2(19)-------------------------    0.                     0.                     0.
S2(1)--------------------------    0.                     0.                     0.
S2(4)--------------------------    0.                     0.                     0.
S2(7)--------------------------    0.                     0.                     0.
S2(10)-------------------------    0.                     0.                     0.
S2(13)-------------------------    0.                     0.                     0.
S2(16)-------------------------    0.                     0.                     0.
S2(19)-------------------------    0.                     0.                     0.
SWV(1)-------------------------    1.0000000000000E-03    1.0000000000000E-03    1.0000000000000E-03
SWV(4)-------------------------    0.                     0.                     0.
SWV(7)-------------------------    0.                     0.                     0.
SWV(10)------------------------    0.                     0.                     0.
SWV(13)------------------------    0.                     0.                     0.
SWV(16)------------------------    0.                     0.                     0.
SWV(19)------------------------    0.                     0.                     0.
VMN(1)-------------------------    0.                     0.                     0.
VMN(4)-------------------------    0.                     0.                     0.
VMN(7)-------------------------    0.                     0.                     0.
VMN(10)------------------------    0.                     0.                     0.
VMN(13)------------------------    0.                     0.                     0.
VMN(16)------------------------    0.                     0.                     0.
VMN(19)------------------------    0.                     0.                     0.
```

---

*Not all variables in the ESC namelist are printed here since each variable is printed for all materials (20 allowed in this case) before going to the next variable.

2. XOUT. XOUT is the print file containing the cell dumps as controlled by either the NP or TP parameters in the INP namelist. Besides giving the cell quantities, the kinetic and internal energies by material and the problem totals are given at the beginning of each dump. This is the same information on material energies and problem totals that is contained on the OUTPUT file. A partial listing from a sample problem is given on the following page.

## SAMPLE XOUT LISTING

SYMMETRIC PLATE IMPACT CU/CU

TIME= 4.00455E+00 DT= 1.26326E-02 CYCLE= 317

MATERIAL 1 ENERGY= 6.45511E-04 INTERNAL ENERGY= 4.10028E-04 KINETIC ENERGY= 2.35482E-04
MATERIAL 2 ENERGY= 3.34059E-03 INTERNAL ENERGY= 4.13702E-04 KINETIC ENERGY= 2.92689E-03
TOTAL ENERGY= 3.98610E-03 TOTAL INTERNAL ENERGY= 8.23730E-04 TOTAL KINETIC ENERGY= 3.16237E-03

| J | IEOS | MAT | M | R | V | U | I | T | P | Q | SX | SZ | W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 4 | 8.903E-02 | 9.928E+00 | 1.123E-01 | -4.943E-04 | 6.664E-07 | 3.079E+02 | -1.063E-07 | 1.887E-07 | -5.470E-04 | 2.735E-04 | 1.000E+00 |
| 3 | 1 | 4 | 8.903E-02 | 9.918E+00 | 1.121E-01 | -4.942E-04 | 9.704E-06 | 3.107E+02 | -6.085E-07 | 2.860E-07 | -2.998E-03 | 1.499E-03 | 1.000E+00 |
| 4 | 1 | 4 | 8.903E-02 | 9.908E+00 | 1.121E-01 | -4.940E-04 | 1.487E-05 | 3.121E+02 | -1.342E-06 | 1.243E-07 | -2.995E-03 | 1.498E-03 | 1.000E+00 |
| 5 | 1 | 4 | 8.903E-02 | 9.898E+00 | 1.121E-01 | -4.939E-04 | 1.762E-05 | 3.128E+02 | -1.777E-06 | 5.713E-08 | -2.999E-03 | 1.499E-03 | 1.000E+00 |
| 6 | 1 | 4 | 8.903E-02 | 9.888E+00 | 1.121E-01 | -4.938E-04 | 1.872E-05 | 3.130E+02 | -1.936E-06 | 0. | -2.999E-03 | 1.500E-03 | 1.000E+00 |
| 7 | 1 | 4 | 8.903E-02 | 9.878E+00 | 1.121E-01 | -4.940E-04 | 1.871E-05 | 3.130E+02 | -1.574E-06 | 0. | -2.999E-03 | 1.500E-03 | 1.000E+00 |
| 8 | 1 | 4 | 8.903E-02 | 9.868E+00 | 1.121E-01 | -4.942E-04 | 1.788E-05 | 3.128E+02 | -7.946E-07 | 0. | -2.999E-03 | 1.500E-03 | 1.000E+00 |
| 9 | 1 | 4 | 8.903E-02 | 9.858E+00 | 1.121E-01 | -4.944E-04 | 1.830E-05 | 3.129E+02 | 7.702E-07 | 0. | -2.997E-03 | 1.499E-03 | 1.000E+00 |
| 10 | 1 | 4 | 8.903E-02 | 9.848E+00 | 1.121E-01 | -4.945E-04 | 2.011E-05 | 3.134E+02 | 2.127E-06 | 1.536E-07 | -2.997E-03 | 1.499E-03 | 1.000E+00 |
| 11 | 1 | 4 | 8.903E-02 | 9.838E+00 | 1.121E-01 | -4.943E-04 | 2.124E-05 | 3.137E+02 | 3.157E-06 | 2.849E-07 | -2.997E-03 | 1.499E-03 | 1.000E+00 |
| 12 | 1 | 4 | 8.903E-02 | 9.828E+00 | 1.121E-01 | -4.941E-04 | 2.177E-05 | 3.138E+02 | 3.832E-06 | 2.972E-07 | -2.997E-03 | 1.498E-03 | 1.000E+00 |
| 13 | 1 | 4 | 8.903E-02 | 9.818E+00 | 1.121E-01 | -4.939E-04 | 2.199E-05 | 3.139E+02 | 4.130E-06 | 2.350E-07 | -2.997E-03 | 1.498E-03 | 1.000E+00 |
| 14 | 1 | 4 | 8.903E-02 | 9.808E+00 | 1.121E-01 | -4.937E-04 | 2.206E-05 | 3.139E+02 | 4.227E-06 | 1.727E-07 | -2.997E-03 | 1.499E-03 | 1.000E+00 |
| 15 | 1 | 4 | 8.903E-02 | 9.798E+00 | 1.121E-01 | -4.936E-04 | 2.211E-05 | 3.139E+02 | 4.415E-06 | 1.623E-07 | -2.997E-03 | 1.498E-03 | 1.000E+00 |
| 16 | 1 | 4 | 8.903E-02 | 9.788E+00 | 1.121E-01 | -4.935E-04 | 2.213E-05 | 3.139E+02 | 4.771E-06 | 1.621E-07 | -2.998E-03 | 1.499E-03 | 1.000E+00 |
| 17 | 1 | 4 | 8.903E-02 | 9.778E+00 | 1.121E-01 | -4.933E-04 | 2.214E-05 | 3.139E+02 | 5.085E-06 | 1.211E-07 | -2.996E-03 | 1.498E-03 | 1.000E+00 |
| 18 | 1 | 4 | 8.903E-02 | 9.768E+00 | 1.121E-01 | -4.933E-04 | 2.216E-05 | 3.139E+02 | 5.154E-06 | 4.327E-08 | -2.997E-03 | 1.499E-03 | 1.000E+00 |
| 19 | 1 | 4 | 8.903E-02 | 9.758E+00 | 1.121E-01 | -4.932E-04 | 2.217E-05 | 3.139E+02 | 4.885E-06 | 0. | -2.998E-03 | 1.499E-03 | 1.000E+00 |
| 20 | 1 | 4 | 8.903E-02 | 9.748E+00 | 1.121E-01 | -4.933E-04 | 2.218E-05 | 3.139E+02 | 4.362E-06 | 0. | -2.998E-03 | 1.499E-03 | 1.000E+00 |
| 21 | 1 | 4 | 8.903E-02 | 9.738E+00 | 1.121E-01 | -4.934E-04 | 2.219E-05 | 3.139E+02 | 4.011E-06 | 0. | -2.998E-03 | 1.499E-03 | 1.000E+00 |
| 22 | 1 | 4 | 8.903E-02 | 9.728E+00 | 1.121E-01 | -4.935E-04 | 2.220E-05 | 3.139E+02 | 2.969E-06 | 0. | -2.998E-03 | 1.499E-03 | 1.000E+00 |
| 23 | 1 | 4 | 8.903E-02 | 9.718E+00 | 1.121E-01 | -4.938E-04 | 2.220E-05 | 3.139E+02 | 2.029E-06 | 0. | -2.999E-03 | 1.500E-03 | 1.000E+00 |
| 24 | 1 | 4 | 8.903E-02 | 9.708E+00 | 1.121E-01 | -4.942E-04 | 2.221E-05 | 3.139E+02 | 1.882E-06 | 0. | -2.999E-03 | 1.500E-03 | 1.000E+00 |
| 25 | 1 | 4 | 8.903E-02 | 9.698E+00 | 1.121E-01 | -4.945E-04 | 2.222E-05 | 3.139E+02 | 1.748E-06 | 0. | -2.999E-03 | 1.499E-03 | 1.000E+00 |
| 26 | 1 | 4 | 8.903E-02 | 9.688E+00 | 1.121E-01 | -4.948E-04 | 2.223E-05 | 3.139E+02 | 1.799E-06 | 0. | -3.000E-03 | 1.500E-03 | 1.000E+00 |
| 27 | 1 | 4 | 8.903E-02 | 9.678E+00 | 1.121E-01 | -4.948E-04 | 2.224E-05 | 3.139E+02 | 1.844E-06 | 6.353E-08 | -3.000E-03 | 1.500E-03 | 1.000E+00 |
| 28 | 1 | 4 | 8.903E-02 | 9.668E+00 | 1.121E-01 | -4.948E-04 | 2.224E-05 | 3.139E+02 | 1.451E-06 | 2.645E-07 | -2.999E-03 | 1.500E-03 | 1.000E+00 |
| 29 | 1 | 4 | 8.903E-02 | 9.658E+00 | 1.121E-01 | -4.946E-04 | 2.225E-05 | 3.139E+02 | 3.039E-07 | 0. | -2.999E-03 | 1.500E-03 | 1.000E+00 |
| 30 | 1 | 4 | 8.903E-02 | 9.648E+00 | 1.121E-01 | -4.946E-04 | 2.226E-05 | 3.139E+02 | -1.715E-06 | 0. | -3.000E-03 | 1.500E-03 | 1.000E+00 |
| 31 | 1 | 4 | 8.903E-02 | 9.638E+00 | 1.121E-01 | -4.947E-04 | 2.227E-05 | 3.139E+02 | -2.055E-06 | 4.162E-07 | -3.000E-03 | 1.500E-03 | 1.000E+00 |
| 32 | 1 | 4 | 8.903E-02 | 9.628E+00 | 1.121E-01 | -4.944E-04 | 2.227E-05 | 3.139E+02 | -4.445E-07 | 5.810E-07 | -2.999E-03 | 1.500E-03 | 1.000E+00 |
| 33 | 1 | 4 | 8.903E-02 | 9.618E+00 | 1.121E-01 | -4.940E-04 | 2.228E-05 | 3.139E+02 | 4.906E-07 | 0. | -2.999E-03 | 1.500E-03 | 1.000E+00 |
| 34 | 1 | 4 | 8.903E-02 | 9.608E+00 | 1.121E-01 | -4.941E-04 | 2.229E-05 | 3.139E+02 | 2.597E-06 | 3.884E-07 | -2.998E-03 | 1.499E-03 | 1.000E+00 |
| 35 | 1 | 4 | 8.903E-02 | 9.598E+00 | 1.121E-01 | -4.938E-04 | 2.229E-05 | 3.139E+02 | 2.452E-06 | 0. | -3.000E-03 | 1.500E-03 | 1.000E+00 |
| 36 | 1 | 4 | 8.903E-02 | 9.588E+00 | 1.121E-01 | -4.944E-04 | 2.230E-05 | 3.139E+02 | 3.972E-06 | 2.711E-07 | -3.000E-03 | 1.500E-03 | 1.000E+00 |
| 37 | 1 | 4 | 8.903E-02 | 9.578E+00 | 1.121E-01 | -4.942E-04 | 2.231E-05 | 3.139E+02 | 1.547E-06 | 0. | -3.000E-03 | 1.501E-03 | 1.000E+00 |
| 38 | 1 | 4 | 8.903E-02 | 9.568E+00 | 1.121E-01 | -4.960E-04 | 2.232E-05 | 3.139E+02 | 9.448E-06 | 0. | -3.000E-03 | 1.500E-03 | 1.000E+00 |
| 39 | 1 | 4 | 8.903E-02 | 9.558E+00 | 1.121E-01 | -4.964E-04 | 2.232E-05 | 3.139E+02 | 1.359E-05 | 0. | -2.999E-03 | 1.500E-03 | 1.000E+00 |
| | | | Mass | Radius | Volume | Velocity | Internal Energy | Tempera-ture | Pressure | Viscosity | Stress Deviators | | Mass Fraction |

3. OUTPUT. The file OUTPUT contains a summary of the problem energies plus various other information about what happened during the problem execution. The following list contains the information that may be written to OUTPUT.

Number of zones in each material, EOS type and number, $\sqrt{\rho_0}\Delta r$ for the and outside cell.

Error messages for materials not in an EOS library.

Time, $\Delta t$, cycle, and energy sums by material and problem total (same as in XOUT).

Record of any dump written or read for a restart.

Record of any spalling.

Record of any HOM iteration errors for a mixture of solid and gas product.

Record of void collapses or openings.

Record of any iteration failures for high-velocity void collapses.

A sample OUTPUT file is listed on the following page. Information about EOS errors, spalling, etc. is listed only if they occur.

## SAMPLE OUTPUT LISTING

| Material | # Zones | Type | EOS # | $\sqrt{\rho_0}\Delta r$ Outside Cell | $\sqrt{\rho_0}\Delta r$ Inside Cell |
|---|---|---|---|---|---|
| 1 | 100 | 1 | 4 | 2.984E-02 | 2.984E-02 |
| 2 | 100 | 1 | 4 | 2.984E-02 | 2.984E-02 |

```
TIME=     0.          DT=  1.00000E+00 CYCLE=          0

 MATERIAL  1 ENERGY= 3.97630E-03 INTERNAL ENERGY= 0.          KINETIC ENERGY= 3.97630E-03
 MATERIAL  2 ENERGY= 0.          INTERNAL ENERGY= 0.          KINETIC ENERGY= 0.
 TOTAL ENERGY= 3.97630E-03 TOTAL INTERNAL ENERGY= 0.          TOTAL KINETIC ENERGY= 3.97630E-03

TIME=     1.26326E-02 DT=  1.26326E-02 CYCLE=          1

 MATERIAL  1 ENERGY= 3.97630E-03 INTERNAL ENERGY= 0.          KINETIC ENERGY= 3.97630E-03
 MATERIAL  2 ENERGY= 0.          INTERNAL ENERGY= 0.          KINETIC ENERGY= 0.
 TOTAL ENERGY= 3.97630E-03 TOTAL INTERNAL ENERGY= 0.          TOTAL KINETIC ENERGY= 3.97630E-03

TIME=     2.52653E-02 DT=  1.26326E-02 CYCLE=          2

 MATERIAL  1 ENERGY= 3.98155E-03 INTERNAL ENERGY= 1.86438E-05 KINETIC ENERGY= 3.96290E-03
 MATERIAL  2 ENERGY= 1.35924E-06 INTERNAL ENERGY= 1.77453E-16 KINETIC ENERGY= 1.35924E-06
 TOTAL ENERGY= 3.98291E-03 TOTAL INTERNAL ENERGY= 1.86438E-05 TOTAL KINETIC ENERGY= 3.96426E-03

TIME=     1.01061E+00 DT=  1.26326E-02 CYCLE=         80

 MATERIAL  1 ENERGY= 3.15778E-03 INTERNAL ENERGY= 4.42471E-04 KINETIC ENERGY= 2.71531E-03
 MATERIAL  2 ENERGY= 8.27660E-04 INTERNAL ENERGY= 4.26922E-04 KINETIC ENERGY= 4.00738E-04
 TOTAL ENERGY= 3.98544E-03 TOTAL INTERNAL ENERGY= 8.69393E-04 TOTAL KINETIC ENERGY= 3.11604E-03

TIME=     2.00859E+00 DT=  1.26326E-02 CYCLE=        159

 MATERIAL  1 ENERGY= 2.32025E-03 INTERNAL ENERGY= 8.69209E-04 KINETIC ENERGY= 1.45104E-03
 MATERIAL  2 ENERGY= 1.66530E-03 INTERNAL ENERGY= 8.53659E-04 KINETIC ENERGY= 8.11641E-04
 TOTAL ENERGY= 3.98555E-03 TOTAL INTERNAL ENERGY= 1.72287E-03 TOTAL KINETIC ENERGY= 2.26268E-03

TIME=     3.00657E+00 DT=  1.26326E-02 CYCLE=        238

 MATERIAL  1 ENERGY= 1.48288E-03 INTERNAL ENERGY= 7.91417E-04 KINETIC ENERGY= 6.91463E-04
 MATERIAL  2 ENERGY= 2.50313E-03 INTERNAL ENERGY= 7.95195E-04 KINETIC ENERGY= 1.70793E-03
 TOTAL ENERGY= 3.98601E-03 TOTAL INTERNAL ENERGY= 1.58661E-03 TOTAL KINETIC ENERGY= 2.39940E-03
DUMP   1 AT CYCLE        317,TIME=  .40045E+01

TIME=     4.00455E+00 DT=  1.26326E-02 CYCLE=        317

 MATERIAL  1 ENERGY= 6.45511E-04 INTERNAL ENERGY= 4.10028E-04 KINETIC ENERGY= 2.35482E-04
 MATERIAL  2 ENERGY= 3.34059E-03 INTERNAL ENERGY= 4.13702E-04 KINETIC ENERGY= 2.92689E-03
 TOTAL ENERGY= 3.98610E-03 TOTAL INTERNAL ENERGY= 8.23730E-04 TOTAL KINETIC ENERGY= 3.16237E-03
```

4. GASSIN. GASSIN is a random access file written in the MAGEE Movie format for direct processing by the graphics utility GAS (LTSS-523). The structure of GASSIN consists of a file index that is $1003_{10}$ words long and dumps for each specified problem time.

File Index

| Disk Address | Word | Contents |
|---|---|---|
| 0 | 1 | Integer giving the number of words in the index |
| 1 | 2 | Integer giving the number of dumps in this file |
| 2 | 3 | Disk address of the last word in this file |
| 3 | 4 | Problem dump time for the first dump } 1 |
| 4 | 5 | Disk address for the first dump in this file } 1 |
| 5 | 6 | Problem dump time for the second dump } 2 |
| 6 | 7 | Disk address for the second dump in this file } 2 |
| | | . |
| | | . |
| | | . |
| | | Repeat dump time, disk address for each dump |

## Data Dumps

Each data dump consists of two parts. The first 100 words contain information about the data in the dump. The data begins at word 101 after the beginning of the data dump and is packed three HYDROX cell variables per word (see GAS writeup). The contents of the first 100 words are as follows.

(I = integer, F = floating point, H = Hollerith. Omitted numbers are not used.)

| Word | Contents |
|---|---|
| 1 | Dump Time (F) |
| 2 | The number of zones for the problem (I) |
| 3 | 1 (I) |
| 4 | Not used |
| 5 | Number of packed words per cell = 4 (I) |
| 6 | Number of cell variables per word = 3 (I) |
| ⋮ | |
| 10 | Number of fraction bits in the packing format = 14 (I) |
| 11 | Number of exponent bits in the packing format = 5 (I) |
| ⋮ | |
| 81 | Date (H) |
| ⋮ | |
| 90 | Problem label, first 10 characters (H) |
| 91 | Problem label, second 10 characters (H) |
| 92 | Problem label, third 10 characters (H) |
| 93 | First value of the cell number = 1 (I) |
| 94 | 1 (I) |

5. DUMPO. A restart capability is provided by the writing and reading of the dump file DUMPO. The frequency of dumps may be selected by either specifying the problem time or cycle number. The problem geometry may also be changed, adding or deleting zones, materials, equation-of-state parameters, or anything capable of being specified in the original problem setup. A description of the control variables is given in the "Restart Control Parameters" of Sec. III.A.1, Namelist INP. Further details are provided in the descriptions of subroutines WDUMP and RDUMP of Sec. IV, HYDROX Description by Subroutines.

## C. Graphics

The graphical output file GASSIN is written in a random access MAGEE movie format for processing by the LTSS utility GAS (LTSS-523). GAS allows the users to make plots of all cell quantities as a function of distance or any other cell variable. In addition, time plots of cell quantities and contour plots in position-time (X-t) space can be made. GAS can be run as an interactive utility or through a controller.

The variable numbers used by GAS and their corresponding HYDROX quantities are given by:

| Gas Variable Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cell Quantity | region index | t | r | u | V | I | P | $S_X$ | $S_Z$ (or W if $\mu = 0$) | EE* | T | q |

*The variable EE contains energy sums in cell quantities in the following order:

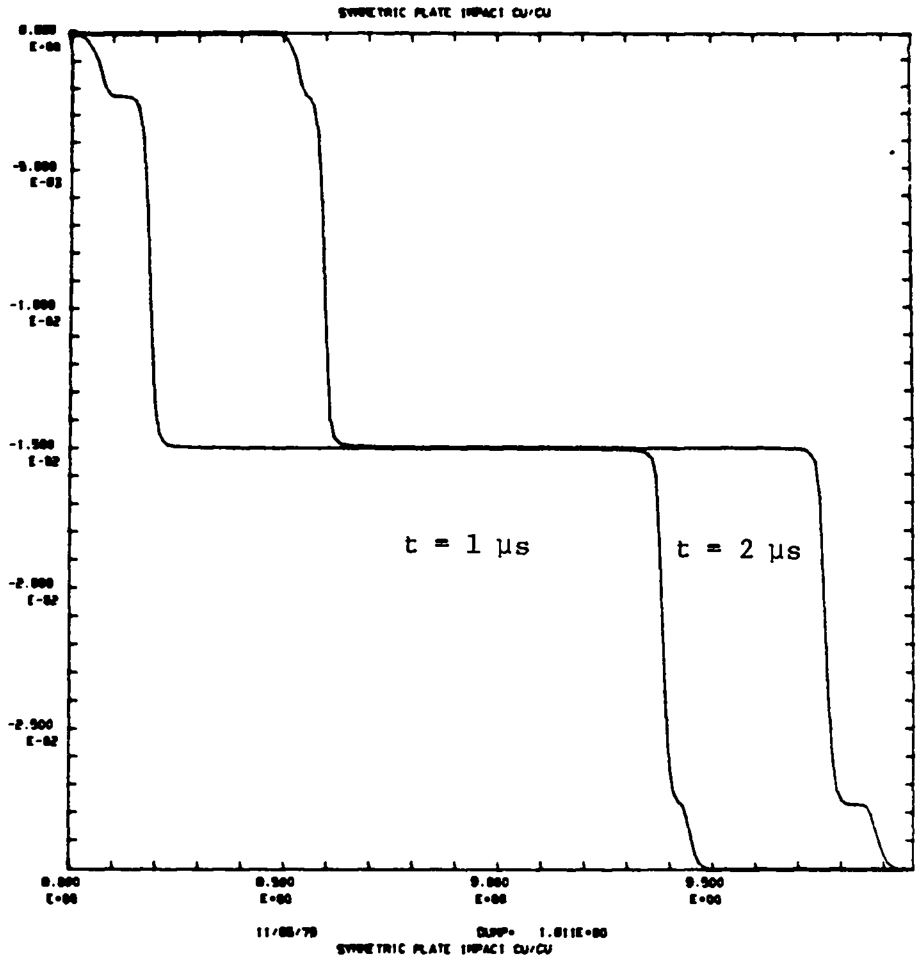
| | |
|---|---|
| 1 to ML-1 | total energy for region 1 to ML-1 |
| ML | total energy for the problem plus work done on pistons |
| ML+1 to 2*ML-1 | internal energy for region 1 to ML-1 |
| 2*ML | total internal energy for the problem |
| 2*ML+1 to 3*ML-1 | kinetic energy for region 1 to ML-1 |
| 3*ML | total kinetic energy for the problem |
| 4*ML+1 | work done on the outside piston |
| 4*ML+2 | work done on the inside piston |

ML is set in a parameter statement and is usually 21, the number of allowed materials plus one.

The sample GAS plots on the following page were generated by the commands listed below.

| | |
|---|---|
| GAS!GASSIN!YES!ME | Initialize GAS and enter the MESH plot mode |
| DC,3,4!SR!DU,1! | Plot the particle velocity vs radius and allow GAS to select a rectangle to plot the data; the dump at time = 1 μs was specified |
| DU,2!MP! | Plot the particle velocity vs radius at time = 2 μs and overlay it on the previous graph |
| CV,3,7!SR!DU,1! | Plot the pressure vs radius for time = 1 μs |
| DU,2!MP! | Plot the pressure vs radius at time = 2 μs and overlay it on the previous graph |
| END | Terminate the execution of GAS |

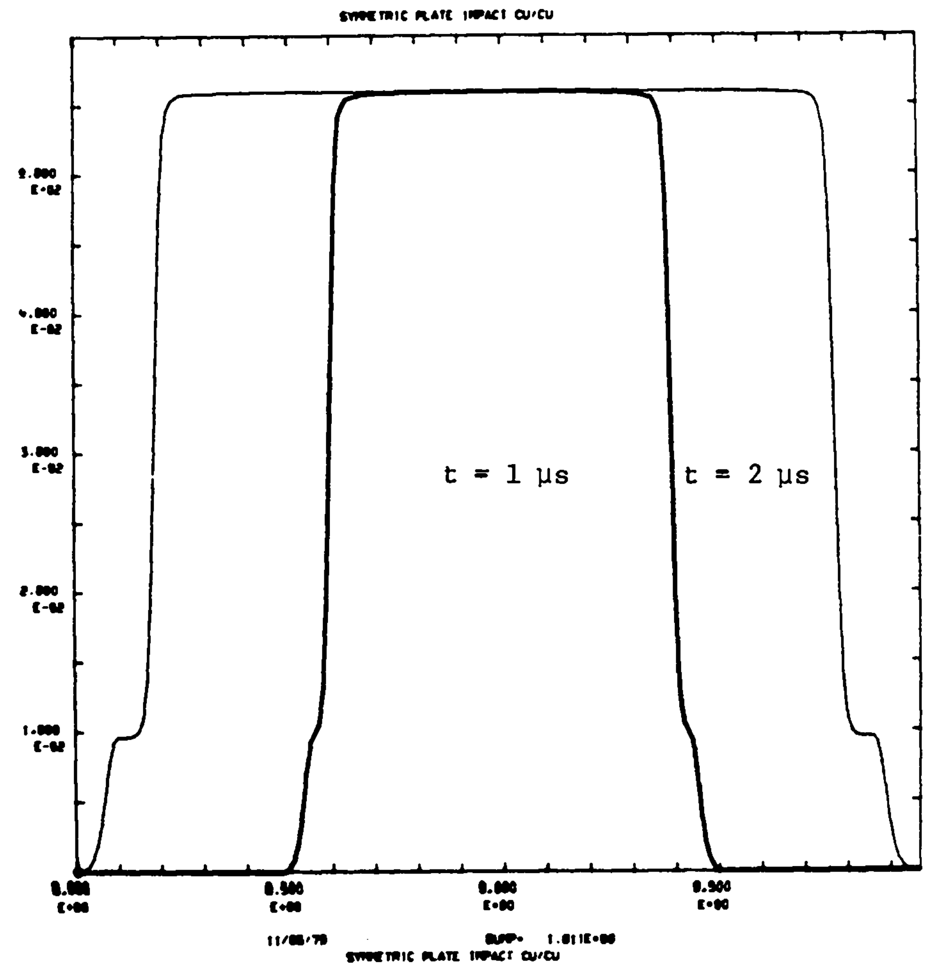
Velocity
vs
Radius
(CV,3,4)




Sample

Gas Plots

Pressure
vs
Radius
(DV,3,7)



## IV. HYDROX DESCRIPTION BY SUBROUTINES

This section contains the most detailed information about the inner workings of HYDROX. Part A contains a summary description of each subroutine and Part B contains a logical flow diagram. Part C contains further information about each subroutine by first giving an annotated FORTRAN listing and then giving detailed notes on local variables, relevant physical models, and numerical algorithms.

### A. Summary Description of the Subroutines

#### MAIN

Calls routines to set up the problem.

Contains the main cycle loop of the code which checks whether to add cells, print, make a GAS dump, make a restart dump, or end the problem; calls subroutines to rezone if necessary, determine the time step, and run one hydro cycle.

#### SETUP

Controls the setup of the problem, reads INP namelist, checks for a restart dump, calls other routines to read the rest of the namelists data from EOS files, initializes all of the cell quantities except pressure, writes out all variables in all namelists to the file DOUT.

#### SSU

Reads the SU namelist and copies material data to region I+1. In order to keep the namelist variable names the same as those used in the code and at the same time avoid requiring region number subscripts in the input, region number one (i.e., no subscript) is used in the namelist. The data is then copied to region number I+1 where I is the actual region number. After all data is read in, every variable associated with regions has all of its data shifted down by one to the proper region. See subroutine RSTORE.

ESSU

Reads the ESC namelist (equation-of-state constants) and copies material data to region I+1.

BRSU

Reads the BURN namelist (various burn constants) and copies material data to region I+1.

CLR

Resets region 1 data to the default values.

RSTORE

Shifts region I+1 data back to I (where it should be) for each region I.

EOSDSK

Switching routine that assigns units for EOS files and calls routines to read them. Data from EOS files can then be overridden by namelist reads.

RHOM

Reads EOS file HMLB to get HOM EOS data.

RBLDUP

Reads EOS file HMLB to get data for the buildup EOS and burn model.

RPOLY

Dummy routine because a library is not provided for the eight-parameter fit constants.

RSESAME

Reads data from disk for SESAME materials.

JMNMX

Sets indices to determine the minimum and maximum cell numbers for each material. Also sets indices for the last region with a cell turned on and the last cell turned on.

HEI

Calculates the total internal energy of a region of solid HE relative to the energy of its products at infinite expansion at T = 0.

GASLM

Finds limits for the region in which two of the analytic fits in GAS are reasonable.

BLDSM

Calculates the $\gamma$ for each cell using the buildup model. The transition from constant $\gamma_{max}$ to the $\gamma = A + B/R$ form is smoothed out with a parabola which joins both curves, leaving the first derivative continuous.

PRNT

Makes a cycle printout including time, $\Delta t$, cycle #, region and total energies, and cell quantities for active cells.

ESUM

Calculates kinetic, internal, and total energies for each region and for the whole problem.

WDUMP

Writes a restart dump (all of the necessary data to restart the problem at a given cycle). Inactive regions may be replaced with new setup information so that two different problems that start out the same may be restarted at a time before they differ without completely rerunning the problem.

RDUMP

Reads the restart dump and stores all of the data in the appropriate locations.

OUTGAS

Makes a GAS dump to file GASSIN which includes most cell quantities. GASSIN may be postprocessed to give on the Tektronix/film/fiche any cell variable as a function of any other cell variable (e.g., pressure vs radius) at a given time, time plot a cell variable for a given cell, r-t plots of interfaces, cell positions for each cell, contour plots of a cell variable in r-t space, etc.

ICONV

Takes a 60-bit floating point word and converts it to a 20-bit floating point word.

DIFEQ

Switching routine to determine the type of difference equation scheme to be used in the main hydro cycle. Default is HYDRO.

HYDRO

The main hydro cycle using the HYDROX difference equations. New values of radii, velocities, specific volumes, specific internal energies, and stress deviators are calculated. Subroutines are called to get new pressures, temperatures, and artificial viscosities.

SINX

The main hydro cycle using the SIN difference equations. New values of radii, velocities, specific volumes, specific internal energies, and stress deviators are calculated. Subroutines are called to get new pressures, temperatures, and artificial viscosities.

EOS

Switching routine to call the appropriate equation of state. The spalling and elastic-plastic treatments are also called if turned on.

PTEOS

Controls calls to EOS subroutines with energy and volume as input rather than region # and cell #.

HOM

Switching routine for deciding which type of EOS is used for a cell for the HOM EOS (e.g., determines whether a material is a solid, gas, or mixture).

USUP

USUP EOS allows for two USUP fits with a phase change. At high density the Barnes EOS is used. In tension, the Grüneisen EOS with the P=0 line as the standard curve is used.

## GAS

Calculates the EOS for gases using analytic fits to the results of the BKW code. By special choice of constants, a $\gamma$-law gas EOS may be calculated.

## SSBGAS

Calculates the pressure and specific internal energy for a cell that has just been burned using the sharp-shock burn method. The pressure and specific internal energy are calculated on the Hugoniot for the HE products at the given volume.

## MIX

Calculates pressure and temperature for a mixture of solid and gas where temperature and pressure are assumed to be in equilibrium. The equations of state for the solid and gas are described more fully in USUP and GAS, respectively.

## LFB

A two-point iteration scheme to find the zero of a function of one variable. The iteration is a slightly modified form of the secant method. This method is faster than Newton-Raphson iteration for the case where the time required to evaluate the derivative is longer than 0.44 of the time required to evaluate the function.

## BEQST

The Barnes EOS is used for the high-pressure region where the USUP fit becomes unphysical.

## BLDUP

Calculates the equation of state to be used with the buildup burn model. The EOS is that of a $\gamma$-law gas but the $\gamma$ is not necessarily the same for all cells in a given material.

## SPEOS

Determines whether a cell should spall by using the gradient spall model. As a special case, a constant spall pressure may be specified.

## POLY

An eight-parameter fit to the equation of state that is basically a polynomial in two variables divided by a linear function in one of the variables. The two variables are related to specific volume and specific internal energy.

## VISC

Computes the viscosity for all cells using either "real," PIC, or Landshoff-type viscosity.

## BURN

Switching routine to determine type of burn to be used.

## ARH

Calculates the decomposition due to an Arrhenius rate law for region I.

## CJ

Calculates the decomposition of a detonating HE using the CJ burn model.

## SSB

Calculates the decomposition of an HE using a sharp shock model. All of the HE is burned at the shock front.

## FOREST

Calculates the decomposition using the Forest Fire burn model. This model is appropriate for cases that require a non-negligible distance of run to detonation for the given input shock strength.

## FFT

The Forest Fire rate is calculated as a function of temperature.

FFI

The Forest Fire rate is calculated as a function of specific internal energy.

GLTW

An entire region of explosive is burned using the gamma-law Taylor-wave description.

BNDR1

Calculates several special boundary conditions such as an applied piston.

SL

Does all the bookkeeping required to create a spall.

SPLTCHK

Checks whether rezoning is required in a region and if so calls subroutines to do the rezoning.

SHFT

Shifts all cells with cell # $\geqslant$ J up by N. Used when new cells are created in the middle of the problem; e.g., for spall and rezoning.

SPLIT

Splits N cells starting at cell #J into two cells. All cell quantities are linearly interpolated and conservation of mass is explicitly required.

EPP

An elastic—perfectly plastic model with the von Mises yield model and an optional correction term to put shock data fit equations of state on the hydrostat.

DELT

Calculates the time step to be used. The time step may be input data or may be evaluated from several criteria in order to keep the problem numerically stable.

C

Switching function subroutine to pick the appropriate sound speed subroutine.

CUSUP

Calculates the sound speed for a USUP EOS with constant Grüneisen $\gamma$.

CBLDUP

Calculates the sound speed for a buildup EOS in cell J.

CPOLY

Calculates the sound speed at specific volume VC, pressure PC, and specific internal energy XC for the eight-parameter fit EOS in subroutine POLY.

CSES

Calculates the sound speed for a SESAME EOS.

RLEOS

The Rayleigh line in P-V space is used as an equation of state for the initial compression of the two cells touching an interface that has just become a closed void when the relative velocity of the two surfaces was large.

RL

Calculates parameters for the Rayleigh line EOS. This primarily consists of iteration to find the interface velocity which sends shock waves into both materials with the same final pressure.

G

Given a value for the interface particle velocity, UV, the difference in the corresponding Hugoniot pressures of the two bounding cells is calculated.

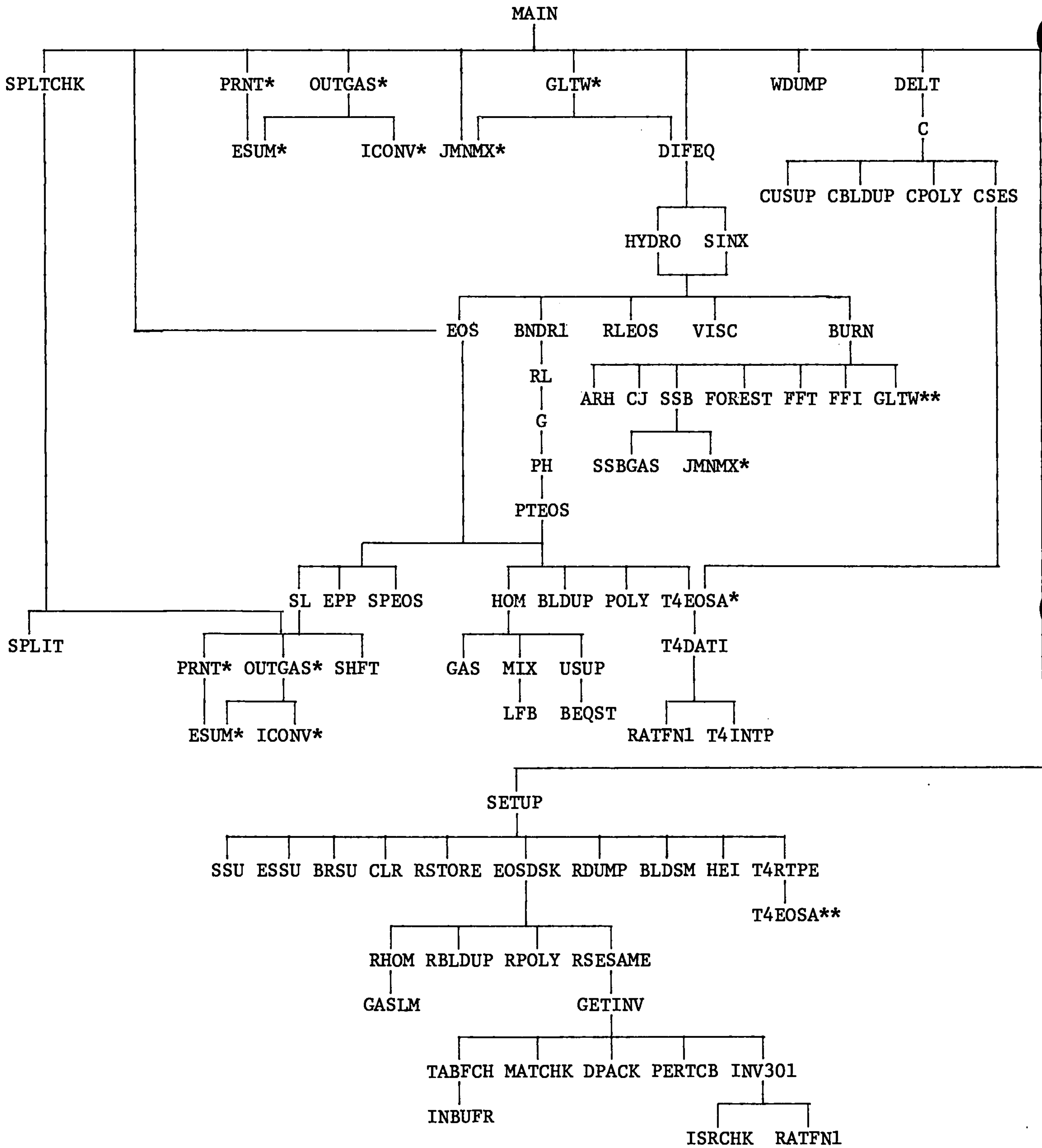
PH

For a given specific internal energy, the volume on the Hugoniot and the Hugoniot pressure are determined.

Subroutines Needed for the SESAME Tables

The following subroutines are used in conjunction with the SESAME EOS tables and are described in Sec. IV.E: MATCHK, TABFCH, INBUFR, DPACK, ISRCHK, T4INTP, GETINV, RATFN1, T4DATI, T4RTPE, INV301, T4EOSA, PERTCR.

B. Chart of the Relation Between Subroutines




*Subroutines that appear more than once.
**Subroutines that appear more than once and subroutines called by this subroutine are shown elsewhere on the chart.

## C. Variables in Common Blocks Not Already Described

/CELL/

R = outside radius of a cell (cm).

U = velocity (cm/μs).

V = specific volume = $1/\rho$ ($cm^3g$).

XI = specific internal energy (Mbar-$cm^3$/g).

P = pressure (Mbar).

SX = stress deviator in the X-direction (Mbar).

SZ = stress deviator in the Z-direction (Mbar).

EE = energy sums, see ESUM.

T = temperature (K) or $\gamma$ for Buildup EOS.

Q = artificial viscosity (Mbar).

XM = mass in grams per unit length or solid angle.

IFLAG = flags associated with a cell.

W = mass fraction of undecomposed explosive (i.e., W = 1 for all solid, W = 0 for all gas).

/OVL/ See INP namelist.

/MISC/

TIME = time (μs).

ICYCL = cycle #.

DT = time step (μs).

NCL = last cell # + 1.

IA = IALPH - 1.

BU = current outside piston velocity (cm/μs).

BUI = current inside piston velocity (cm/μs).

F2,F3 = geometry-dependent coefficients used in the calculation of the specific volume of a cell.

/BRNS/ See BEQST for details.

A = A.

BR = $b_r$.

BA = $b_a$.

VBO = $V_0$.

VBSW = volume below which BEQST is used instead of USUP.

/EOSN/ See SU namelist.

/NSPLT/ See SU namelist.

/SPC/ See ESC namelist.

/POLYC/ See ESC namelist.

/GAS/ See OUTGAS for details.

FI = index for GAS dumps.

DI = array for all cell variables (equivalenced to R).

/LEV/

DMPNO = dump # = time (see OUTGAS).

/BUX/ See ESC namelist.

/ES/ See ESC namelist.

/RLC/ See RLEOS, RL, G, and PH for details.

RC = $R_c$.

RP = $R_p$.

RLV = $R_\ell$.

PH1 = $P_H^{(1)}$

DV1 = $\Delta V_1$

DV2 = $\Delta V_2$

/PWORK/ See BNDR1, ESUM.

PW = work done by the outside piston (if used).

PWI = work done by the inside piston (if used).

JS = spall indicator. Whenever JS $\neq 0$, a new void is created at the outside radius of cell JS.

/INIT/ See SU, ESC namelists.

JMIN = minimum cell # for this region.

JMAX = maximum active cell # for this region.

DRO = initial $\Delta r$ for the innermost cell of the region.

/USUPC/ See ESC namelist.

/BRND/ See BURN namelist.

/GASC/ See ESC namelist.

/FGHIJC/ See ESC namelist.

/UCJC/

UCJ = CJ velocity.

JJ = cell # being burned in SSB.

NMAX = last cell currently active.

RCJ = radius of the cell being burned in SSB.

DCJ = CJ detonation velocity.

/VOID/

INTX = type of interface: $1(\mu_I = \mu_{I+1} = 0)$, $2(\mu_I = 0, \mu_{I+1} \neq 0)$, $3(\mu_I \neq 0, \mu_{I+1} = 0)$, $4(\mu_I \neq \mu_{I+1} \neq 0)$, $5(\mu_I = \mu_{I+1} \neq 0)$.

JV = cell # of the artificial cell used to describe a void between region I and I + 1.

IV = see SU namelist.

NNV = # of voids.

/MNMX/

KMAX = maximum cell # for a region.

KMIN = minimum cell # for a region.

NMC = # of regions currently active.

/XCOM/ See SU, ESC namelists.

/INTORD/ See INP namelist.

/EOSCOM/ See ESC namelist.

/XEOS/

IX = region #.

/SESDAT/

DC = array for SESAME tables.

/S2DIR/

LCMX = # of words in DC.

NREG = # of regions allowed.

LCFW = word # in DC that begins data for region I.

/SESIN/

II = region #.

IDT = data type.

RPT4 = density.

XIPT4 = specific internal energy.

IBR = 0 to output P and T; 1 to output P; 2 to output T.

IFL = 0 allows for a ramp; 1 requires use of tables.

/SESOUT/

PPT4 = P, $\partial P/\partial \rho$, $\partial P/\partial E$.

TPT4 = T, $\partial T/\partial \rho$, $\partial T/\partial E$.

```
      PROGRAM HYDROX(INPUT,OUTPUT,DATA,TAPE5=DATA,DOUT,TAPE6=DOUT,        MAIN
     +XOUT,TAPE8=XOUT,TTY,TAPE9=TTY)                                      MAIN
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,           PARAM
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                    PARAM
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                      PARAM
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                          PARAM
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                           MCELL
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)     MCELL
     +,W(MCL)                                                             MCELL
      LEVEL 2,R                                                           MCELL
      COMMON/DVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UTI,UFI,NADD,NM,   MCELL
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                      MCELL
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                    MCELL
      LEVEL 2,TIME                                                        MCELL
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO   INIT
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),  INIT
     +MAT(ML),UD(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)              INIT
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),           US
     +GAMMA(ML),ALP(ML)                                                   US
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML)  BRD
     +,MSFF                                                               BRD
      COMMON/GASC/GC(NGC,ML)                                              GC
      COMMON/FGHIJC/FS(ML),GS(ML),HS(ML),SI(ML),SJ(ML),CV(ML)             FG
      COMMON/UCJC/UCJ,JJ,NMAX,RCJ,DCJ                                     UC
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                           VD
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                                 MN
      COMMON/BRNS/A(ML),BR(ML),BA(ML),VBO(ML),VBSW(ML)                    BRN
      COMMON/EOSN/IEOS(ML),ME(ML)                                         EN
      COMMON/NSPLT/NOSPLT(ML2)                                            NSP
      COMMON/SPC/SP(ML),USP(ML)                                           SPLC
     +,XISP(ML)                                                           SPLC
      COMMON/POLYC/CF(NCF,ML),PS(ML)                                      PLC
      COMMON/GAS/FI(1003),DI(MQL)                                         GS
      LEVEL 2,FI                                                          GS
      COMMON/LEV/DMPNO                                                    GS
      LEVEL 2,DMPNO                                                       GS
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)                                   BUP
     +,BUR,BUD                                                            BUP
      COMMON/ES/TE(ML2),NME                                               ESM
      COMMON/RLC/RC(ML),RP(ML),RLV(ML),PH1,DV1,DV2                        RLC
      COMMON/PWORK/PW,PWI                                                 PWORK
      TIME=0.  Start at t = 0                                             MAIN
      CALL SETUP  Set up the problem                                      MAIN
      ISS=0                                                               MAIN
      DO 8 I=1,NM                                                         MAIN
      IF(IBRN(I).NE.3)GO TO 8                                             MAIN
      IF(ISS.NE.0)GO TO 8   Initialization for sharp-shock burn only      MAIN
      JJ=JMIN(I)                                                          MAIN
      ISS=I                                                               MAIN
      NMAX=JJ                                                             MAIN
    8 CONTINUE                                                            MAIN
      CALL JMNMX(NMAX) Initialize indices for min. and max. cell # in each region MAIN
      IF(IDMP.NE.0)GO TO 10  Skip to 10 if a restart                      MAIN
      NMCT=NMC                                                            MAIN
      NMC=NM                                                              MAIN
      JMT=JMAX(NMCT)                                                      MAIN
      JMAX(NMCT)=KMAX(NMCT)                                               MAIN
      RCJ=0.                                                              MAIN
      DO 9 I=1,NM                                                         MAIN
      IF(IBRN(I).EQ.7)CALL GLTW(I)  Do any gamma-law Taylor wave first    MAIN
```

```
      IF(RCJ.NE.0..OR.IBRN(I).NE.3)GO TO 9
      RCJ=R(JJ)                                   More setup for
      IF(I.EQ.1)GO TO 9                            sharp shock burn
      DCJ=VCJ(T)
      DT=(R(JJ)-R(JJ+1))/(DCJ*4)
      UCJ=-E(I)
    9 CONTINUE
      CALL EOS  Initialize P,T for all cells
      JMAX(NMCT)=JMT
      NMC=NMCT
   10 CONTINUE
      MG=0
      TMP=TP(1)
      TMG=TG(1)
      TPMX=TMP
      TGMX=TMG    Setup for print, gas dump, restart dump
      ITP=0       Keyed to certain cycle #'s and times
      ITG=0
      TMD=TD(1)
      TDMX=TMD
      ITD=0
      MP=0
      MD=0
      ICYCL=0
      CALL PRNT   Printout of initial conditions
      BU=UI
      BUI=UII
      PW=0.     Initial piston velocities and work
      PWI=0.
      CALL QASSIGN(3,6HGASSIN,0,0)  GASSIN = file for GAS dumps
      IF(NADD.LE.0)NADD=5
      CALL OUTGAS  GAS dump of initial conditions
      DO 20 II=1,NT   Main do loop of the code.  NI is the maximum # of cycles
      ICYCL=II                                                    allowed
      MP=MP+1
      MG=MG+1
      MD=MD+1
      IF(IBRN(NMC).EQ.3)GO TO 123   Except for sharp-shock burn
      IF(NMAX.EQ.NCL-1)GO TO 123     or for all cells active
      JMC=JMAX(NMC)                                  Check to see if the last
      IF(ABS(U(JMC)-UO(NMC)).LT.UT(NMC))GO TO 123     active cell is moving
      NMAX=NMAX+NADD
      IF(NMAX.GT.NCL-1)NMAX=NCL-1  If so, add NADD
      CALL JMNMX(NMAX)             New active cells
123   CONTINUE
      CALL SPLTCHK  Check for rezoning
      CALL DFLT   Check time step
      TIME=TIME+DT  Increment time step
      IF(W(3).LT.0.02) BU=UF        Use final piston velocity when the
      IF(W(NCL-3).LT.0.02)BUI=UFI    3rd cell in has burned
      CALL DIFEQ  Main hydro done here
      IF(NP.LE.0)GO TO 15
      IF(MP.LT.NP)GO TO 15   Check for print every NP cycles
      MP=0
      CALL PRNT
   15 CONTINUE
      IF(NG.LE.0)GO TO 16
      IF(MG.LT.NG)GO TO 16   Check for GAS dump every NG cycles
      MG=0
      CALL OUTGAS
```

```
   16 CONTINUE
      IF(TP(2).LE.0.)GO TO 17
      IF(TIME.LT.TMP)GO TO 17
      TMP=TMP+TP(2)
      IF(TMP.LT.TPMX)GO TO 27
      TMP=TPMX                          Check for print on time interval
      ITP=ITP+2
      TP(2)=TP(ITP)
      TPMX=TP(ITP+1)
   27 CALL PRNT
   17 CONTINUE
      IF(TG(2).LE.0.)GO TO 18
      IF(TIME.LT.TMG)GO TO 18
      TMG=TMG+TG(2)
      IF(TMG.LT.TGMX)GO TO 28
      TMG=TGMX                          Check for GAS dump on time interval
      ITG=ITG+2
      TG(2)=TG(ITG)
      TGMX=TG(ITG+1)
   28 CALL OUTGAS
   18 CONTINUE
      IF(NDUMP.EQ.0)GO TO 19            Check for restart dump every ND cycles
      IF(MD.LT.NDUMP)GO TO 19
      MD=0
      CALL WDUMP
   19 CONTINUE
      IF(TD(2).LE.0.)GO TO 29
      IF(TIME.LT.TMD)GO TO 29
      TMD=TMD+TD(2)
      IF(TMD.LT.TDMX)GO TO 39           Check for restart dump on time interval
      TMD=TDMX
      ITD=ITD+2
      TD(2)=TD(ITD)
      TDMX=TD(ITD+1)
   39 CALL WDUMP
   29 CONTINUE
      IF(TIME.GE.TEND.AND.TEND.GT.0.)GO TO 999   Stop for t > TEND ≠ 0
   20 CONTINUE
  999 CONTINUE
      CALL WDUMP    Make a last restart dump and print
      CALL PRNT
      STOP
      END
```

Calls routines to set up the problem.

Contains the main cycle loop of the code which checks whether to add cells, print, make a GAS dump, make a restart dump, or end the problem; calls subroutines to rezone if necessary, determine the time step, and run one hydro cycle.

Local Variables

MP,MG,MD = # of cycles since the last print, GAS dump, restart dump.

TMP,TMG,TMD = time at which the next print, GAS dump, restart dump will be made.

ITP,ITG,ITD = index for which t,$\Delta$t to use.

TPMX,TGMX,TDMX = time at which a switch is made to the next $\Delta$t.

II = cycle #, do loop count.

JMC = JMAX(NMC) is the last cell currently turned on.

JMT = temporary storage of JMAX(NMC) so that it can be changed for the call to EOS.

NMCT = temporary storage of NMC.

Notes

The sharp-shock burn uses its own method for adding a cell every four cycles as the shock goes through the material (see SSB).

The algorithm for printing, etc., every N dumps is: initialize an index M to 0, increment by 1 each cycle. When M = N print, etc. and reset M to 0.

The algorithm for printing, etc., on $t_1, \Delta t_1, \cdots$ is: initialize a parameter T to $t_1$. When the time is $\geqslant$ T print, etc. Reset T to T + $\Delta t_1$ unless T + $\Delta t_1 > t_2$. Then set T to $t_2$ and increment by $\Delta t_2$, etc.

The common blocks should be all kept in MAIN even though they are not all used. This is due to the fact that the restart dumps are keyed on the first location in one common block and the last location in a different common block. The order in which the common blocks are stored is, therefore, important. By including all the common blocks required for a restart dump in MAIN, their order in storage will be that required by WDUMP and RDUMP (q.v.).

```
      SUBROUTINE SETUP                                                  SETUP      2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTA8=1,NTA8=MTA8*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL      3
     +,W(MCL)                                                           MCELL      4
      LEVEL 2,R                                                         MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,8U,8UI,F2,F3,JS                  MCELL      8
      LEVEL 2,TIME                                                      MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),I8RN(ML),PLAP(ML),DRO(ML), INIT      3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT       4
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),         US         2
     +GAMMA(ML),ALP(ML)                                                 US         3
      COMMON/8RND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML) BRD       2
     +,MSFF                                                             BRD        3
      COMMON/GASC/GC(NGC,ML)                                            GC         2
      COMMON/FGHTJC/FS(ML),GS(ML),HS(ML),SI(ML),SJ(ML),CV(ML)           FG         2
      COMMON/UCJC/UCJ,JJ,NMAX,RCJ,DCJ                                   UC         2
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                         VD         2
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                               MN         2
      COMMON/8RNS/A(ML),8R(ML),BA(ML),VBO(ML),V8SW(ML)                  8RN        2
      COMMON/EOSN/IEOS(ML),ME(ML)                                       EN         2
      COMMON/NSPLT/NOSPLT(ML2)                                          NSP        2
      COMMON/SPC/SP(ML),USP(ML)                                         SPLC       2
     +,XISP(ML)                                                         SPLC       3
      COMMON/POLYC/CF(NCF,ML),PS(ML)                                    PLC        2
      COMMON/8UX/8UA,8UB,8UMAX,8UDV(ML)                                 8UP        2
     +,8UR,8UD                                                          8UP        3
      COMMON/ES/IE(ML2),NME                                             ESM        2
      LEVEL 2,DC                                                        LCMC       2
      COMMON/LCMC/DC(NSD)                                               LCMC       3
      COMMON/XCOM/R1,R2,DR1,DR2,WO,NCI,DR,ZI                            XC         2
      COMMON/INTORD/IFN                                                 SETUP     24
      COMMON/EOSCOM/SR(ML),ES(ML),A1(ML),A2(ML),A3(ML),EM(ML),IRV(ML)   EOSCOM     2
      NAMELIST/SU/DTO,NOSPLT,IV,IEOS,MAT,ME,R1,R2,UO,NCI                NMLST      2
     +,DR1,DR2,UT,DTCF,IJK                                              NMLST      3
      NAMELIST/ESC/C1,S1,C2,S2,SWV,VMN,GAMMA,ALP,A,8R,BA,VBO,VBSW,      NMLST      4
     +FS,GS,HS,SI,SJ,CV,GC,SP,USP,CF,PS,8UA,8U8,8UMAX,8UDV,TMLT,TMC,    NMLST      5
     +XMU,YO,XL,XV,VO,PO,TO,ROW,PLAP,I8RN,NV,WO,ZI                      NMLST      6
     +,SR,ES,A1,A2,A3,EM,IRV,8UR,8UD,XISP,QO,IE                         NMLST      7
      NAMELIST/BURN/Z,E,VCJ,PCJ,DWDT,PM,ND                              NMLST      8
      NAMELIST/INP/NM,UI,UF,RO,IALPH,NDF,NI,NP,NG,NADD,NMAX,            NMLST      9
     +TEND,TP,TG,NDELT,UII,UFI,LABEL,NM1,IDMP,IV,NDUMP,TD,IFN           NMLST     10
     +,NO,DTCF                                                          SETUP     29
     +,MSFF                                                             SETUP     30
      DATA NM1/1/,IDMP/0/,NNV/0/,NDUMP/10000/,TD/ML*0./                 SETUP     31
      DATA NM,UI,UF,RO,IALPH,NDF,NI,NP,NG,NADD,NMAX,TEND,TP,TG          SETUP     32
     +,UII,UFI,LABEL/0,3*0.,3,1,10000,4*0,0.,ML*0.,ML*0.,2*0.,          SETUP     33
     +8*10H          /                                                  SETUP     34
      DATA DTO,XMU,YO,XL,XV,VO,PO,TO,ROW,PLAP,I8RN,NV,NOSPLT,IV         SETUP     35
     +,UO,QO,TMLT,TMC                                                   SETUP     36
     +/ML*1.,ML*0.,ML*0.,ML*0.,ML*2.,ML*0.,ML*1.E-10,ML*0.,             SETUP     37
     +ML*0.,ML*0.,ML*0,ML*1,ML2*0,ML2*-1,ML*0.,ML*1.E-10,ML*0.,ML*0./   SETUP     38
      DATA R1,R2,WO,ZI,NCI/2*0.,1.,0.,0/                                SETUP     39
      DATA A,BR,BA,VBO,V8SW/ML*0.,ML*0.,ML*0.,ML*0.,ML*0./              SETUP     40
```

```
      DATA IEOS,MAT,ME/ML*1,ML*0,ML*0/
      DATA C1,S1,C2,S2,SWV,VMN,GAMMA,ALP/ML*0,ML*0,ML*0,ML*0,ML*0,
     +ML*0,ML*0,ML*0/
      DATA Z,E,VCJ,PCJ,PM,ND/ML*0,ML*0,ML*0,ML*0,ML*0,ML*0/
      DATA FS,GS,HS,SI,SJ,CV/ML*0.,ML*0.,ML*0.,ML*0.,ML*0.,ML*0./
      DATA SP,USP/ML*0.,ML*0./
      DATA GC/MLGC*0/
      DATA DWDT/MLDWDT*0/
      DATA IFN/0/
      DATA SR,ES,A1,A2,A3,EM,IRV/ML*1.,ML*0.,ML*0.,ML*0.,ML*0.,ML*1000.
     +,ML*0/
      DATA UT/ML*1.E-10/,NO/180/,DTCF/ML*0.5/
      DATA MSFF/0/
      DATA XISP/ML*0./
      DATA BUD/.2/
      READ(5,INP)    Read INP namelist
      IF(IDMP.EQ.0)GO TO 99    Check whether to use a restart dump
      CALL RDUMP(IDMP)    Read 1st dump after cycle IDMP
      READ(5,INP)   Make any necessary changes in INP variables
      IF(NM1.EQ.1)RETURN    If no new materials are added, setup is complete
   99 CONTINUE
      DO 1 I=NM1,NM
    1 IE(I)=I
      SRDR=RO/NO   Default value used in automatic zoning
      R1=RO   Default value for R1
      IA=IALPH-1
      J=1    J = cell #
      IF(IDMP.NE.0)J=KMAX(NM1-1)   For a restart with new materials added, do
      F2=0.5                         setup only for the new materials
      F3=0.
      IF(IA.NE.2)GO TO 15   Geometry factors
      F2=1./3.
      F3=1.
   15 CONTINUE
      IF(IDMP.EQ.0)R(1)=RO   Set piston radius to RO unless a restart
      DO 10 I=NM1,NM
      CALL CLR   Set default values for region 1
      CALL SSU(I)   Read SU namelist
      IF(MAT(I+1).NE.0) CALL EOSDSK(I)   For MAT ≠ 0—read EOS data from disk
      IF(ME(I+1).EQ.0.AND.MAT(I+1).NE.0) GO TO 30  More data?
      CALL ESSU(I)   Read ESC namelist
   20 IF(IBRN(I+1).EQ.0) GO TO 30
      CALL BRSU(I)   Read BURN namelist
   30 R(J+1)=R1   Outside radius for region I
      IF(ZI.NE.0.)GO TO 50
      IF(TO(I+1).EQ.0..OR.IEOS(I+1).NE.4)GO TO 50   Calculate I0 for
      II=I+1                                          input T0 in SESAME
      CALL T4RTPE(I,1,DC,ROW(II),TO(II),PP,ZI,IFL)
   50 CONTINUE
      IFL=64*I  Region # flag
      IF(DTO(I+1).LE.0..OR.DTO(I+1).EQ.1.)DTO(I+1)=DTO(2)
      IF(DTCF(I+1).LE.0..OR.DTCF(I+1).EQ.0.5)DTCF(I+1)=DTCF(2)
      U(J+1)=UO(I+1)  Initial velocity for the region
      JMIN(I)=J+1
      DS=0.
      IF(NCI.NE.0.)GO TO 12
      IF(DR1.LE.0.)GO TO 14
      NCI=2*(R1-R2)/(DR1+DR2)
      DS=2*(R1-R2-NCI*DR1)/(NCI*(NCI-1))   Variable zone size
      DR=DR1
```

```
      DRO(I+1)=DR2                                                      SETUP
      GO TO 13                                                          SETUP
   14 CONTINUE                                                          SETUP
      DR=SRDR/SQRT(ROW(I+1))                      ]                     SETUP
      NCI=(R1-R2)/DR+0.5                          ]                     SETUP
      IF(NCI.LT.4)NCI=4                           ] Automatic zoning    SETUP
      IF(NCI.GT.NO/4.AND.ROW(I+1).GT.10.)NCI=NO/4 ]                     SETUP
      IF(NCI.LT.8.AND.ROW(I+1).GT.5.)NCI=8        ]                     SETUP
   12 CONTINUE                                                          SETUP
      DR=(R1-R2)/NCI ] NCI equally sized zones                          SETUP
      DRO(I+1)=DR    ]                                                  SETUP
   13 CONTINUE                                                          SETUP
     L DO 11 K=1,NCI  Initialize cell quantities in this region         SETUP
      J=J+1                                                             SETUP
      W(J)=WO  Burn fraction                                            SETUP
      T(J)=TO(I+1)  Temperature                                         SETUP
      XI(J)=ZI  Specific internal energy                                SETUP
      R2=R1-DR                                                          SETUP
      XM(J)=F2*DR*(R1**IA+R2**IA+F3*R1*R2)*ROW(I+1) Mass/unit solid angle or area SETUP
      V(J)=F2*DR*(R1**IA+R2**IA+F3*R1*R2)/XM(J)  Specific volume        SETUP
      R1=R2                                                             SETUP
      R(J+1)=R1  Outside radius of the cell                             SETUP
      IFLAG(J)=IFL                                                      SETUP
      U(J+1)=UO(I+1)  Velocity                                          SETUP
      DR=DR+DS  For variable zone size                                  SETUP
11    CONTINUE                                                          SETUP
      JMAX(I)=J  Maximum cell # in the region                           SETUP
C BUILD UP EOS CONSTANTS                                                 SETUP
      IF(IEOS(I+1).NE.2) GO TO 40                                       SETUP
      JMN=JMIN(I)                     ]                                 SETUP
      JMX=JMAX(I)                     ]                                 SETUP
      BUDV(I+1)=BUDV(I+1)**2          ] For Buildup EOS calculate γ for each SETUP
      DO 41 K=JMN,JMX                 ]  cell and store in temperature  SETUP
      DR=R(JMN)-(R(K+1)+R(K))/2+BUR   ]                                 SETUP
      T(K)=BLDSM(DR,I)                ]                                 SETUP
   41 CONTINUE                                                          SETUP
   40 CONTINUE                                                          SETUP
      VO(I+1)=V(J)  Initial specific volume for the region              SETUP
      IF(IBRN(I+1).NE.0)CALL HEI(I+1)  Shift in energy zero for HE's    SETUP
      IF(IV(I+1).LT.0)GO TO 10                                          SETUP
      J=J+1     ]                                                       SETUP
      NNV=NNV+1 ] Set up the artificial cell used for voids             SETUP
      XM(J)=0.  ]                                                       SETUP
      JV(I)=J   ]                                                       SETUP
10    CONTINUE                                                          SETUP
      CALL RSTORE  All region quantities shifted down one to their proper places SETUP
      NMM=NM-1                                                          SETUP
      DO 180 I=1,NMM                              ]                     SETUP
      I1=1                                        ]                     SETUP
      I2=1                                        ]                     SETUP
      IF(XMU(I).EQ.0.) I2=0                       ] Set up flags for the type SETUP
      IF(XMU(I+1).EQ.0.)I1=0                      ]  of interface (see HYDRO) SETUP
      II=I2*2+I1+1                                ]                     SETUP
      IF(II.EQ.4.AND.XMU(I).EQ.XMU(I+1))II=5      ]                     SETUP
      INTX(I)=II                                  ]                     SETUP
  180 CONTINUE                                                          SETUP
      INTX(NM)=1                                                        SETUP
      IF(XMU(NM).NE. 0.) INTX(NM)=3  Interface flag for inside free surface SETUP
      NCL=J+1                                                           SETUP
      R(NCL+1)=R(NCL)                                                   SETUP
```

```
      IFLAG(NCL)=IFLAG(NCL-1)+64
      DT=DTO(1)
      IF(NADD.LE.0)NMAX=NCL-1    for NADD < 0, start with all cells active
      DO 200 I=1,NM
      SROW=SQRT(ROW(I))
      JMN=JMIN(I)
      JMX=JMAX(I)
      J=JMX-JMN+1
      SR1=(R(JMN)-R(JMN+1))*SROW
      SR2=(R(JMX)-R(JMX+1))*SROW
      PRINT 201,T,J,IEOS(I),MAT(I),SR1,SR2
  201 FORMAT(4I5,2(1PE10.3))
  200 CONTINUE
      IF(UI.NE.0..OR.IEOS(1).NE.2)GO TO 211
      UI=-SQRT(BUDV(1))/(T(2)+1)   Automatic setup of piston velocities for
      UF=-UI*.8                     Buildup EOS
  211 CONTINUE
      IF(IBRN(1).NE.3.OR.IDMP.NE.0)GO TO 210
      DCJ=VCJ(1)
      IF(IEOS(1).EQ.2)DCJ=SQRT(8UDV(1))
      DT=(R(2)-R(3))/(DCJ*4)
      UCJ=UI                                 Setup for sharp shock burn
      IF(E(1).LE.0.)GO TO 210
      UI=-E(1)
      UF=E(1)*0.8
      UCJ=UI
  210 CONTINUE
      WRITE(6,INP)
      WRITE(6,SU)
      WRITE(6,ESC)    Write out all of the namelist variables on DOUT
      WRITE(6,BURN)
      CALL CLOSE(6)
      RETURN
      END
```

## SETUP

Controls the setup of the problem, reads INP namelist, checks for a restart dump, calls other routines to read the rest of the namelists data from EOS files, initializes all of the cell quantities except pressure, writes out all variables in all namelists to the file DOUT.

Local Variables

SRDR = RO/NO = the value of $\sqrt{\rho_0}\ \Delta r$ to be used in automatic zoning. It is set such that approximately NO cells would be used in the problem if $\rho_0$ were 1.0 for each material and the innermost cell of the problem were at r = 0.

J = cell # index.

RD = outside radius of the problem.

I = region # index.

PP = pressure from SESAME for input $\rho_0, T_0$.

IFL = flag = 1 for success = 0 for failure to find P,I for input $\rho_0, T_0$; also the region number flag in IFLAG used in OUTGAS.

DS = the change in $\Delta r$ per cell if a linearly varying $\Delta r$ is used.

K = kth cell in a region or cell # index.

NMM = NM-1.

I1= 0 if $\mu_I = 0$, 1 if $\mu_I \neq 0$.

I2 = 0 if $\mu_{I+1} = 0$, 1 if $\mu_{I+1} \neq 0$.

II = temporary variable in which INTX is computed; also I + 1.

SROW = $\sqrt{\rho_0}$ .

SR1 = $\sqrt{\rho_0}\ \Delta r$ for the outermost cell of the region.

SR2 = $\sqrt{\rho_0}\ \Delta r$ for the innermost cell of the region.

JMN,JMX = JMIN(I),JMAX(I).

Notes

The zoning in a region may be set up such that the cell size varies linearly with cell number; that is,

$$\Delta r_n = \Delta r^{(1)} + S(n - 1) \quad , \tag{1}$$

where n is the number of the cell counting inward from the first cell in the region, $\Delta r^{(1)}$ is $\Delta r_1$, and S is a constant to be determined. The total distance spanned by N cells for given S is

$$r_1 - r_2 = \sum_{i=1}^{N} \Delta r_i = N\Delta r^{(1)} + S\,\frac{N(N-1)}{2} \quad , \tag{2}$$

where $r_1$ is the outside radius of the region and $r_2$ is the inside radius. The cell size of the innermost cell is

$$\Delta r^{(2)} = \Delta r_N = \Delta r^{(1)} + S(N - 1) \quad . \tag{3}$$

The usual input quantities are $r_1$, $r_2$, $\Delta r^{(1)}$, and $\Delta r^{(2)}$. Using this information we can express S and N as

$$S = \frac{2\left(r_1 - r_2 - N\,r^{(1)}\right)}{N(N-1)} \quad , \tag{4}$$

$$N = \frac{2(r_1 - r_2)}{\Delta r^{(1)} + \Delta r^{(2)}} \quad . \tag{5}$$

Note, however, that N will not be an integer for arbitrary input values. In order to avoid this problem, we take N as the integer part of the value given by Eq. (5). Then Eq. (4) is evaluated using the new integer value of N. The value of $\Delta r^{(2)}$ will then be slightly different from the input value.

The mass and volume calculations are the same as in HYDRO (q.v.). For Buildup EOS (see BLDUP), the value of $\gamma$ for each cell is stored in the temperature, T. BUDV is input as the detonation velocity D, but it is stored in the code as $D^2$ to avoid recalculating the same thing many times.

```
      SUBROUTINE  SSU(I)                                                    SSU          2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,            PARAM        2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                     PARAM        3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                       PARAM        4
     +,NSM=4,NWPM=3729,NSD=NSM*NWPM+132,ML2=100)                           PARAM        5
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO    INIT         2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),   INIT         3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)               INIT         4
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                            VD           2
      COMMON/EOSN/IEOS(ML),ME(ML)                                          EN           2
      COMMON/XCOM/R1,R2,DR1,DR2,WO,NCI,DR,ZI                               XC           2
      COMMON/NSPLT/NOSPLT(ML2)                                             NSP          2
      NAMELIST/SU/DTO,NOSPLT,IV,IEOS,MAT,ME,R1,R2,UO,NCI                   NMLST        2
     +,DR1,DR2,UT,DTCF,IJK                                                 NMLST        3
      NAMELIST/ESC/C1,S1,C2,S2,SWV,VMN,GAMMA,ALP,A,BR,BA,V80,VBSW,         NMLST        4
     +FS,GS,HS,ST,SJ,CV,GC,SP,USP,CF,PS,BUA,BUB,BUMAX,BUDV,TMLT,TMC,       NMLST        5
     +XMU,YO,XL,XV,VO,PO,TO,ROW,PLAP,IBRN,NV,WO,ZI                         NMLST        6
     +,SR,ES,A1,A2,A3,EM,IRV,BUR,BUD,XISP,QO,IE                            NMLST        7
      NAMELIST/BURN/Z,E,VCJ,PCJ,DWDT,PM,ND                                 NMLST        8
      NAMELIST/INP/NM,UI,UF,RO,IALPH,NDF,NI,NP,NG,NADD,NMAX,               NMLST        9
     +TEND,TP,TG,NDELT,UII,UFI,LABEL,NM1,IDMP,IV,NDUMP,TD,IFN              NMLST       10
      READ(5,SU)    Read SU namelist (data goes into region 1)             SSU         10
      J=I+1                                                                SSU         11
      DTO(J)=DTO(1)                                                        SSU         12
      NOSPLT(J)=NOSPLT(1)                                                  SSU         13
      IV(J)=IV(1)                                                          SSU         14
      IEOS(J)=IEOS(1)                                                      SSU         15
      UO(J)=UO(1)          Copy all of the data into region I + 1          SSU         16
      UT(J)=UT(1)                                                          SSU         17
      DTCF(J)=DTCF(1)                                                      SSU         18
      MAT(J)=MAT(1)                                                        SSU         19
      ME(J)=ME(1)                                                          SSU         20
      RETURN                                                               SSU         21
      END                                                                  SSU         22
```

## SSU(I)

Reads the SU namelist and copies material data to region I+1. (In order to keep the namelist variable names the same as those used in the code and at the same time avoid requiring region number subscripts in the input, region number one (i.e., no subscript) is used in the namelist. The data is then copied to region number I+1 where I is the actual region number. After all data is read in, every variable associated with regions has all of its data shifted down by one to the proper region. See subroutine RSTORE.)

Local Variables

J = I+1 is the region in which data is temporarily put (see above).

```
      SUBROUTINE ESSU(I)                                                ESSU       2
      COMMON/XCOM/R1,R2,DR1,DR2,WO,NCI,DR,ZI                            XC         2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT      3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT       4
      COMMON/POLYC/CF(NCF,ML),PS(ML)                                    PLC        2
      COMMON/SPC/SP(ML),USP(ML)                                         SPLC       2
     +,XISP(ML)                                                         SPLC       3
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),         US         2
     +GAMMA(ML),ALP(ML)                                                 US         3
      COMMON/BRNS/A(ML),BR(ML),BA(ML),VBO(ML),VBSW(ML)                  BRN        2
      COMMON/FGHTJC/FS(ML),GS(ML),HS(ML),SI(ML),SJ(ML),CV(ML)           FG         2
      COMMON/GASC/GC(NGC,ML)                                            GC         2
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)                                 BUP        2
     +,BUR,BUD                                                          BUP        3
      COMMON/EOSCOM/SR(ML),ES(ML),A1(ML),A2(ML),A3(ML),EM(ML),IRV(ML)   EOSCOM     2
      NAMELIST/SU/DTO,NOSPLT,IV,IEOS,MAT,ME,R1,R2,UO,NCI                NMLST      2
     +,DR1,DR2,UT,DTCF,IJK                                              NMLST      3
      NAMELIST/ESC/C1,S1,C2,S2,SWV,VMN,GAMMA,ALP,A,BR,BA,VBO,VBSW,      NMLST      4
     +FS,GS,HS,SI,SJ,CV,GC,SP,USP,CF,PS,BUA,BUB,BUMAX,BUDV,TMLT,TMC,    NMLST      5
     +XMU,YO,XL,XV,VO,PO,TO,ROW,PLAP,IBRN,NV,WO,ZI                      NMLST      6
     +,SR,ES,A1,A2,A3,EM,IRV,BUR,BUD,XISP,QO,IE                         NMLST      7
      NAMELIST/BURN/Z,E,VCJ,PCJ,DWDT,PM,ND                              NMLST      8
      NAMELIST/INP/NM,UI,UF,RO,IALPH,NDF,NI,NP,NG,NADD,NMAX,            NMLST      9
     +TEND,TP,TG,NDELT,UII,UFI,LABEL,NM1,IDMP,IV,NDUMP,TD,IFN           NMLST     10
      READ(5,ESC)    Read ESC namelist (data goes into region I)        ESSU      17
      J=I+1                                                             ESSU      18
      C1(J)=C1(1)                                                       ESSU      19
      S1(J)=S1(1)                                                       ESSU      20
      C2(J)=C2(1)                                                       ESSU      21
      S2(J)=S2(1)                                                       ESSU      22
      SWV(J)=SWV(1)                                                     ESSU      23
      VMN(J)=VMN(1)                                                     ESSU      24
      GAMMA(J)=GAMMA(1)                                                 ESSU      25
      ALP(J)=ALP(1)                                                     ESSU      26
      A(J)=A(1)                                                         ESSU      27
      BR(J)=BR(1)                                                       ESSU      28
      BA(J)=BA(1)                                                       ESSU      29
      VBO(J)=VBO(1)                                                     ESSU      30
      VBSW(J)=VBSW(1)                                                   ESSU      31
      FS(J)=FS(1)          Copy ESC data into region I + 1              ESSU      32
      GS(J)=GS(1)                                                       ESSU      33
      HS(J)=HS(1)                                                       ESSU      34
      SI(J)=SI(1)                                                       ESSU      35
      SJ(J)=SJ(1)                                                       ESSU      36
      CV(J)=CV(1)                                                       ESSU      37
      SP(J)=SP(1)                                                       ESSU      38
      USP(J)=USP(1)                                                     ESSU      39
      PS(J)=PS(1)                                                       ESSU      40
      XMU(J)=XMU(1)                                                     ESSU      41
      YO(J)=YO(1)                                                       ESSU      42
      TMLT(J)=TMLT(1)                                                   ESSU      43
      TMC(J)=TMC(1)                                                     ESSU      44
      XL(J)=XL(1)                                                       ESSU      45
      XV(J)=XV(1)                                                       ESSU      46
      VO(J)=VO(1)                                                       ESSU      47
```

```
      PO(J)=PO(1)                                                       ESSU      48
      QO(J)=QO(1)                                                       ESSU      49
      TO(J)=TO(1)                                                       ESSU      50
      ROW(J)=ROW(1)                                                     ESSU      51
      PLAP(J)=PLAP(1)                                                   ESSU      52
      IBRN(J)=IBRN(1)                                                   ESSU      53
      NV(J)=NV(1)                                                       ESSU      54
      BUDV(J)=BUDV(1)                                                   ESSU      55
      DO 10 K=1,NGC                                                     ESSU      56
   10 GC(K,J)=GC(K,1)                                                   ESSU      57
      DO 20 K=1,NCF                                                     ESSU      58
   20 CF(K,J)=CF(K,1)                                                   ESSU      59
      SR(J)=SR(1)                                                       ESSU      61
      ES(J)=ES(1)                                                       ESSU      62
      A1(J)=A1(1)                                                       ESSU      63
      A2(J)=A2(1)                                                       ESSU      64
      A3(J)=A3(1)                                                       ESSU      65
      EM(J)=EM(1)                                                       ESSU      66
      IRV(J)=IRV(1)                                                     ESSU      67
      RETURN                                                            ESSU      69
      END                                                               ESSU      70
```

## ESSU(I)

Reads the ESC namelist (equation-of-state constants) and copies material data to region I+1.

Local Variables

J = I+1 is the region in which data is temporarily put.

K = do loop index.

NGC = the first dimension in the GC array = the number of gas constants allowed. (NGC is set in the parameter statement.)

NCF = the first dimension of the CF array = the number of parameters allowed in the 8-parameter fit. (NCF is set in the parameter statement.)

```
      SUBROUTINE BRSU(I)                                                BRSU         2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM        2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM        3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM        4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM        5
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML) BRD         2
     +,MSFF                                                             BRD          3
      NAMELIST/SU/DTO,NOSPLT,IV,IEOS,MAT,ME,R1,R2,UO,NCI                NMLST        2
     +,DR1,DR2,UT,DTCF,IJK                                              NMLST        3
      NAMELIST/ESC/C1,S1,C2,S2,SWV,VMN,GAMMA,ALP,A,BR,BA,V80,VBSW,      NMLST        4
     +FS,GS,HS,SI,SJ,CV,GC,SP,USP,CF,PS,BUA,BUB,BUMAX,BUDV,TMLT,TMC,    NMLST        5
     +XMU,YO,XL,XV,VO,PO,TO,ROW,PLAP,IBRN,NV,WO,ZI                      NMLST        6
     +,SR,ES,A1,A2,A3,EM,IRV,BUR,BUD,XISP,QO,IE                         NMLST        7
      NAMELIST/BURN/Z,E,VCJ,PCJ,DWDT,PM,ND                              NMLST        8
      NAMELIST/INP/NM,UI,UF,RO,IALPH,NDF,NI,NP,NG,NADD,NMAX,            NMLST        9
     +TEND,TP,TG,NDELT,UII,UFI,LABEL,NM1,IDMP,IV,NDUMP,TP,IFN           NMLST       10
      READ(5,BURN)  Read BURN namelist (data goes into region 1)        BRSU         6
      J=I+1                                                             BRSU         7
      Z(J)=Z(1)                                                         BRSU         8
      E(J)=E(1)                                                         BRSU         9
      VCJ(J)=VCJ(1)                                                     BRSU        10
      PCJ(J)=PCJ(1)   Copy all BURN data into region I + 1              BRSU        11
      PM(J)=PM(1)                                                       BRSU        12
      ND(J)=ND(1)                                                       BRSU        13
      DO 10 K=1,NDW                                                     BRSU        14
   10 DWDT(K,J)=DWDT(K,1)                                               BRSU        15
      RETURN                                                            BRSU        16
      END                                                               BRSU        17
```

## BRSU(I)

Reads the BURN namelist (various burn constants) and copies material data to region I+1.

Local Variables

J = I+1 is the region in which data is temporarily put.

K = do loop index.

NDW = the first dimension of the DWDT array = the number of Forest Fire constants allowed for each material. (NDW is set by the parameter statement.)

```
      SUBROUTINE CLR
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NOX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),
     +GAMMA(ML),ALP(ML)
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML)
     +,MSFF
      COMMON/GASC/GC(NGC,ML)
      COMMON/FGHIJC/FS(ML),GS(ML),HS(ML),SI(ML),SJ(ML),CV(ML)
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV
      COMMON/BRNS/A(ML),BR(ML),BA(ML),VBO(ML),VBSW(ML)
      COMMON/EOSN/IEOS(ML),ME(ML)
      COMMON/NSPLT/NOSPLT(ML2)
      COMMON/SPC/SP(ML),USP(ML)
     +,XISP(ML)
      COMMON/POLYC/CF(NCF,ML),PS(ML)
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)
     +,BUR,BUD
      COMMON/XCOM/R1,R2,DR1,DR2,WO,NCI,DR,ZI
      COMMON/EOSCOM/SR(ML),ES(ML),A1(ML),A2(ML),A3(ML),EM(ML),IRV(ML)
      DTO(1)=1.
      NOSPLT(1)=0
      IV(1)=-1         Reinitialize all variables in namelists
      UO(1)=0.         SU, ESC, and BURN (for region 1) to the default values
      UT(1)=1.E-10
      DTCF(1)=0.5
      IEOS(1)=1
      MAT(1)=0
      ME(1)=0
      R2=0.
      NCI=0
      C1(1)=0.
      S1(1)=0.
      C2(1)=0.
      S2(1)=0.
      SWV(1)=0.
      VMN(1)=0.
      GAMMA(1)=0.
      ALP(1)=0.
      A(1)=0.
      BR(1)=0.
      BA(1)=0.
      VBO(1)=0.
      VBSW(1)=0.
      FS(1)=0.
      GS(1)=0.
      HS(1)=0.
      SI(1)=0.
      SJ(1)=0.
      CV(1)=0.
      SP(1)=0.
      USP(1)=0.
      PS(1)=0.
      BUDV(1)=0.
      BUA=0.
```

```
      BUB=0.
      WO=1.
      ZI=0.
      XMU(1)=0.
      YO(1)=0.
      TMLT(1)=0.
      TMC(1)=0.
      XL(1)=0.
      XV(1)=2.
      VO(1)=0.
      PO(1)=1.E-10
      QO(1)=1.E-10
      TO(1)=0.
      ROW(1)=0.
      PLAP(1)=0.
      IBRN(1)=0
      NV(1)=1
      BUMAX=0.
      DO 100 K=1,NGC
  100 GC(K,1)=0.
      DO 200 K=1,NCF
  200 CF(K,1)=0.
      Z(1)=0.
      E(1)=0.
      VCJ(1)=0.
      PCJ(1)=0.
      PM(1)=0.
      ND(1)=0
      DO 300 K=1,NDW
  300 DWDT(K,1)=0.
      SR(1)=1.
      ES(1)=0.
      A1(1)=0.
      A2(1)=0.
      A3(1)=0.
      EM(1)=0.
      IRV(1)=0
      RETURN
      END
```

## CLR

Resets region 1 data to the default values.

### Local Variables

K = do loop index.

NGC = the first dimension in the GC array = the number of gas constants allowed. (NGC is set in the parameter statement.)

NCF = the first dimension in the CF array = the number of parameters allowed in the 8-parameter fit. (NCF is set in the parameter statement.)

NDW = the first dimension in the DWDT array = the number of Forest Fire constants allowed for each material. (NDW is set in the parameter statement.)

### Notes

Default values must be included here as well as in the data statements in SETUP. (For data read in from disk, values are stored directly into region I+1. However, if more data is to be read from namelists, then the default values have to be reset in region 1 before the disk read is made.)

```
      SUBROUTINE RSTORE                                                  RSTORE     2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,          PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                   PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                     PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                         PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                          MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)    MCELL      3
     +,W(MCL)                                                            MCELL      4
      LEVEL 2,R                                                          MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,  MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                     MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                   MCELL      8
      LEVEL 2,TIME                                                       MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO  INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT       3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)             INIT       4
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),          US         2
     +GAMMA(ML),ALP(ML)                                                  US         3
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML) BRD        2
     +,MSFF                                                              BRD        3
      COMMON/GASC/GC(NGC,ML)                                             GC         2
      COMMON/FGHIJC/FS(ML),GS(ML),HS(ML),SI(ML),SJ(ML),CV(ML)            FG         2
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                          VD         2
      COMMON/BRNS/A(ML),BR(ML),BA(ML),VBO(ML),VBSW(ML)                   BRN        2
      COMMON/EOSN/IEOS(ML),ME(ML)                                        EN         2
      COMMON/NSPLT/NOSPLT(ML2)                                           NSP        2
      COMMON/SPC/SP(ML),USP(ML)                                          SPLC       2
     +,XISP(ML)                                                          SPLC       3
      COMMON/POLYC/CF(NCF,ML),PS(ML)                                     PLC        2
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)                                  BUP        2
     +,BUR,BUD                                                           BUP        3
      COMMON/TABG/ESC(NTAB),NLOC(ML)                                     TABG       2
      COMMON/EOSCOM/SR(ML),ES(ML),A1(ML),A2(ML),A3(ML),EM(ML),IRV(ML)    EOSCOM     2
      DO 10 I=1,NM                                                       RSTORE    21
      J=I+1                                                              RSTORE    22
      DTO(I)=DTO(J)        Shift all variables in namelists              RSTORE    23
      XMU(I)=XMU(J)         SU, ESC, and BURN down one region            RSTORE    24
      YO(I)=YO(J)           to their proper place                        RSTORE    25
      TMLT(I)=TMLT(J)                                                    RSTORE    26
      TMC(I)=TMC(J)                                                      RSTORE    27
      XL(I)=XL(J)                                                        RSTORE    28
      XV(I)=XV(J)                                                        RSTORE    29
      VO(I)=VO(J)                                                        RSTORE    30
      PO(I)=PO(J)                                                        RSTORE    31
      QO(I)=QO(J)                                                        RSTORE    32
      TO(I)=TO(J)                                                        RSTORE    33
      ROW(I)=ROW(J)                                                      RSTORE    34
      PLAP(I)=PLAP(J)                                                    RSTORE    35
      IBRN(I)=IBRN(J)                                                    RSTORE    36
      NV(I)=NV(J)                                                        RSTORE    37
      NOSPLT(I)=NOSPLT(J)                                                RSTORE    38
      IV(I)=IV(J)                                                        RSTORE    39
      UO(I)=UO(J)                                                        RSTORE    40
      UT(I)=UT(J)                                                        RSTORE    41
      DTCF(I)=DTCF(J)                                                    RSTORE    42
      IEOS(I)=IEOS(J)                                                    RSTORE    43
      MAT(I)=MAT(J)                                                      RSTORE    44
      ME(I)=ME(J)                                                        RSTORE    45
      C1(I)=C1(J)                                                        RSTORE    46
      S1(I)=S1(J)                                                        RSTORE    47
```

```
      C2(I)=C2(J)                                                       RSTORE    48
      S2(I)=S2(J)                                                       RSTORE    49
      SWV(I)=SWV(J)                                                     RSTORE    50
      VMN(I)=VMN(J)                                                     RSTORE    51
      GAMMA(I)=GAMMA(J)                                                 RSTORE    52
      ALP(I)=ALP(J)                                                     RSTORE    53
      A(I)=A(J)                                                         RSTORE    54
      BR(I)=BR(J)                                                       RSTORE    55
      BA(I)=BA(J)                                                       RSTORE    56
      VBO(I)=VBO(J)                                                     RSTORE    57
      VBSW(I)=VBSW(J)                                                   RSTORE    58
      FS(I)=FS(J)                                                       RSTORE    59
      GS(I)=GS(J)                                                       RSTORE    60
      HS(I)=HS(J)                                                       RSTORE    61
      SI(I)=SI(J)                                                       RSTORE    62
      SJ(I)=SJ(J)                                                       RSTORE    63
      CV(I)=CV(J)                                                       RSTORE    64
      SP(I)=SP(J)                                                       RSTORE    65
      USP(I)=USP(J)                                                     RSTORE    66
      PS(I)=PS(J)                                                       RSTORE    67
      BUDV(I)=BUDV(J)                                                   RSTORE    68
      DO 100 M=1,NGC                                                    RSTORE    69
  100 GC(M,I)=GC(M,J)                                                   RSTORE    70
      DO 200 M=1,NDW                                                    RSTORE    71
  200 DWDT(M,I)=DWDT(M,J)                                               RSTORE    72
      DO 300 M=1,NCF                                                    RSTORE    73
  300 CF(M,I)=CF(M,J)                                                   RSTORE    74
      Z(I)=Z(J)                                                         RSTORE    75
      E(I)=E(J)                                                         RSTORE    76
      VCJ(I)=VCJ(J)                                                     RSTORE    77
      PCJ(I)=PCJ(J)                                                     RSTORE    78
      PM(I)=PM(J)                                                       RSTORE    79
      ND(I)=ND(J)                                                       RSTORE    80
      DRO(I)=DRO(J)                                                     RSTORE    81
      NLOC(I)=NLOC(J)                                                   RSTORE    82
      SR(I)=SR(J)                                                       RSTORE    84
      ES(I)=ES(J)                                                       RSTORE    85
      A1(I)=A1(J)                                                       RSTORE    86
      A2(I)=A2(J)                                                       RSTORE    87
      A3(I)=A3(J)                                                       RSTORE    88
      EM(I)=EM(J)                                                       RSTORE    89
      IRV(I)=IRV(J)                                                     RSTORE    90
   10 CONTINUE                                                          RSTORE    92
      RETURN                                                            RSTORE    93
      END                                                               RSTORE    94
```

## RSTORE

Shifts region I+1 data back to I (where it should be) for each region I.

Local Variables

K = do loop index.

NGC = the first dimension in the GC array - the number of gas constants allowed. (NGC is set in the parameter statement.)

NCF = the first dimension in the CF array = the number of parameters allowed in the 8-parameter fit. (NCF is set in the parameter statement.)

NDW = the first dimension in the DWDT array = the number of Forest Fire constants allowed for each material. (NDW is set in the parameter statement.)

```
      SUBROUTINE EOSDSK(I)                                                   EOSDSK     2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,              PARAM      2
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                       PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTA8=1,NTA8=MTA8*3742                         PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                             PARAM      5
      COMMON/XEOS/IX                                                         XEOS       2
      COMMON/EOSN/IEOS(ML),ME(ML)                                            EN         2
      DATA IST/1/                                                            EOSDSK     6
      IF(IST.EQ.2)GO TO 2                                                    EOSDSK     7
      IST=2                          1st time through                        EOSDSK     8
      CALL QASSIGN(22,5HSES2L,0,0)    QASSIGN all EOS files                  EOSDSK    10
      CALL QASSIGN(23,4HHML8,0,0)                                            EOSDSK    12
    2 CONTINUE                                                               EOSDSK    13
      IX=I+1                                                                 EOSDSK    14
      ITYPE=IEOS(IX)                                                         EOSDSK    15
      GO TO (11,12,13,14),ITYPE                                              EOSDSK    16
   11 CALL RHOM                 Call the appropriate subroutine to read      EOSDSK    17
      RETURN                     and store data for that type of EOS         EOSDSK    18
   12 CALL RBLDUP                                                            EOSDSK    19
      RETURN                                                                 EOSDSK    20
   13 CALL RPOLY                                                             EOSDSK    21
      RETURN                                                                 FOSDSK    22
   14 CALL RSESAME                                                           EOSDSK    23
      RETURN                                                                 EOSDSK    24
      END                                                                    FOSDSK    25
```

## EOSDSK

Switching routine that assigns units for EOS files and calls routines to read them. (Data from EOS files can then be overridden by namelist reads.)

### Local Variables

IST = flag to determine whether this is the first call to EOSDSK.

IX = I+1 = data is stored in region I+1 during setup.

ITYPE = IEOS for that region.

### Notes

IX is used in all of the called subroutines.

```
      SUBROUTINE RHOM
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO
     +(ML),TO(ML),POW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),
     +MAT(ML),UD(ML),UT(ML),CTCF(ML),QO(ML),TMLT(ML),TMC(ML)
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),
     +GAMMA(ML),ALP(ML)
      COMMON/BRNO/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML)
     +,MSFF
      COMMON/GASC/GC(NGC,ML)
      COMMON/SPC/SP(ML),USP(ML)
     +,XISP(ML)
      COMMON/FGHIJC/FS(ML),GS(ML),HS(ML),SI(ML),SJ(ML),CV(ML)
      COMMON/BRNS/A(ML),BR(ML),BA(ML),VBO(ML),VBSW(ML)
      COMMON/EOSN/IEOS(ML),ME(ML)
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)
     +,BUR,BUO
      COMMON/XEOS/IX
      DIMENSION DAT(84),IDAT(84),NAM(3)
      EQUIVALENCE(DAT,IDAT)
      DATA LUN/23/     Unit # for HMLB
      I=IX
      NNA=3*MAT(I)-1   Location in index of 1st word address for material # MAT(I)
      CALL RDISK(LUN,NAM,3,NNA)]- Read the FWA plus two words of comment
      IF(UNIT(LUN))1,1,1       ]
    1 CONTINUE
      CALL RDISK(LUN,DAT,84,NAM(1))] Read the 84 words of HOM data
      IF(UNIT(LUN))2,2,2           ]
    2 CONTINUE
      IBRN(I)=IDAT(4)                           ]  Convert the SIN type
      IF(IBRN(I).LE.1)IBRN(I)=IBRN(I)+1         ]   burn index
      IF(DAT(64).EQ.0..AND.IBRN(I).EQ.1)IBRN(I)=0]
      IF(ME(I).EQ.0)GO TO 10] If more data is to be read from namelist, then
      IBRN(1)=IBRN(I)       ]  the region 1 values must be set to override
      I=1                   ]  the default values
   10 CONTINUE
      NV(I)=IDAT(5)+1
      IF(NV(I).EQ.3)NV(I)=0
      XV(I)=DAT(6)
      TO(I)=DAT(9)
      C1(I)=DAT(12)
      S1(I)=DAT(13)
      SWV(I)=DAT(14)
      C2(I)=DAT(15)             Load the data into the proper constants
      S2(I)=DAT(16)
      FS(I)=DAT(17)
      GS(I)=DAT(18)
      HS(I)=DAT(19)
      SI(I)=DAT(20)
      SJ(I)=DAT(21)
      GAMMA(I)=DAT(22)
      CV(I)=DAT(23)
      ALP(I)=DAT(25)
      SP(I)=DAT(26)
      USP(I)=DAT(27)
      YO(I)=DAT(30)
      XMU(I)=DAT(31)
```

```
      PLAP(I)=DAT(32)
      VMN(I)=DAT(34)
      Z(I)=DAT(36)
      E(I)=DAT(37)
      TMLT(I)=E(I)
      TMC(I)=Z(I)
      VCJ(I)=DAT(38)
      ND(I)=IDAT(41)
      PCJ(I)=DAT(42)
      PM(I)=DAT(43)
      DO 110 K=1,20
110   DWDT(K,I)=DAT(K+43)
      DO 111 K=1,17
111   GC(K,I)=DAT(K+63)
      A(I)=DAT(64)
      BR(I)=DAT(65)
      BA(I)=DAT(66)
      VBO(I)=DAT(67)
      VBSW(I)=DAT(68)
      ROW(I)=DAT(7)
      GC(18,I)=10.  ]  Default limits on the region of validity of two fits in MIX
      GC(19,I)=-20. ]
      IF(IBRN(I).EQ.4)CALL GASLM(I)   Calculate the actual limits of validity
      RETURN
      END
```

## RHOM

Reads EOS file HMLB to get HOM EOS data.

### Local Variables

I = IX = temporary region # in which data is stored.

NNA = location in HMLB of the 1st word address for the location of data for the material # MAT(I).

NAM = three words in the index for material # MAT(I). Word #1 is the 1st word address described above. Words 2 and 3 are Hollerith data giving a label for that material.

DAT = the actual data (84 words) for that HOM material. The order corresponds to that in the SIN input deck for the same material.

IDAT = equivalenced to DAT, so that integer data can be retrieved without conversion.

JJ = an index such that data is stored in region 1 only if more namelist data is to be read (ME=1).

### Notes

IBRN and NV require conversion from their corresponding SIN values. Data is stored directly in region I+1 if no changes are to be made (ME = 0). Data is stored in region 1 if changes are to be made by namelist (ME ≠ 0). After changes are made, all of the region 1 data is copied to region I+1.

```
      SUBROUTINE RBLDUP                                                   RBLDUP     2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,           PARAM      2
     +NUMV=10,MQC=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                    PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTA8=1,NTA8=MTA8*3742                      PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                          PARAM      5
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)                                   BUP        2
     +,BUR,BUD                                                            BUP        3
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                           MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)     MCELL      3
     +,W(MCL)                                                             MCELL      4
      LEVEL 2,R                                                           MCELL      5
      COMMON/DVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UTI,UFI,NADD,NM,   MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                      MCELL      7
      COMMON/MTSC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                    MCELL      8
      LEVEL 2,TIME                                                        MCELL      9
      COMMON/INTT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO   INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),  INIT       3
     +MAT(ML),UD(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)              INIT       4
      COMMON/XEOS/IX                                                      XEOS       2
      COMMON/EOSN/TEOS(ML),ME(ML)                                         EN         2
      DIMENSION DAT(84),IDAT(84),NAM(3)                                   RBLDUP     9
      EQUIVALENCE(DAT,IDAT)                                               RBLDUP    10
      DATA LUN/23/                                                        RBLDUP    11
      I=IX                                                                RBLDUP    12
      NNA=3*MAT(I)-1                    ]                                 RBLDUP    13
      CALL RDISK(LUN,NAM,3,NNA)         ]                                 RBLDUP    14
      IF(UNIT(LUN))1,1,1                ]  See RHOM                       RBLDUP    15
    1 CONTINUE                          ]                                 RBLDUP    16
      CALL RDISK(LUN,DAT,84,NAM(1))     ]                                 RBLDUP    17
      IF(UNIT(LUN))2,2,2                ]                                 RBLDUP    18
    2 CONTINUE                                                            RBLDUP    19
      BUA=DAT(81)               ]                                         RBLDUP    20
      BUB=DAT(82)               ]                                         RBLDUP    21
      BUMAX=DAT(83)             ]                                         RBLDUP    22
      IF(ME(I).NE.0)I=1         ]                                         RBLDUP    23
      BUDV(I)=DAT(84)           ]  Load the data into the proper constants RBLDUP   24
      IBRN(I)=2                 ]                                         RBLDUP    25
      ROW(I)=DAT(7)             ]                                         RBLDUP    26
      NV(I)=IDAT(5)+1           ]                                         RBLDUP    27
      IF(NV(I).EQ.3)NV(I)=0     ]                                         RBLDUP    28
      XV(I)=DAT(6)              ]                                         RBLDUP    29
      RETURN                                                              RBLDUP    30
      END                                                                 RBLDUP    31
```

## RBLDUP

Reads EOS file HMLB to get data for the buildup EOS and burn model.

Local Variables

I = IX = temporary region # in which data is stored.

NNA = location in HMLB of the 1st word address for the location of data for the material # MAT(I).

NAM = the 3 words in the index for material # MAT(I). Work #1 is the 1st word address described above. Words 2 and 3 are Hollerith data giving a label for that material.

DAT = the actual data (84 words) for that HOM material. The order corresponds to that in the SIN input deck for the same material.

IDAT = equivalenced to DAT, so that integer data can be retrieved without conversion.

JJ = an index such that data is stored in region 1 only if more namelist data is to be read (ME=1).

Notes

NV requires conversion from the corresponding SIN value. See notes for RHOM.

```
      SUBROUTINE RPOLY                                                   RPOLY
      PRINT 900                                                          RPOLY
      WRITE(6,900)                                                       RPOLY
      WRITE(8,900)                                                       RPOLY
      WRITE(9,900)                  Output message and STOP              RPOLY
  900 FORMAT(1X,20H********************/                                 RPOLY
     +1X,20H*  NO LIBRARY FOR  */                                        RPOLY
     +1X,20H* IFOS=3 MATERIALS */                                        RPOLY
     +1X,20H********************)                                        RPOLY
      STOP                                                               RPOLY
      END                                                                RPOLY
```

## RPOLY

Dummy routine because a library is not provided for the eight-parameter fit constants.

```
      SUBROUTINE RSESAME                                                RSESAME    2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLOWDT=20*ML,          PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                   PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                     PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                         PARAM      5
      LEVEL 2,DC                                                        RSESAME    5
      COMMON/SESDAT/DC(NSD)                                             RSESAME    6
      COMMON/S2DIR/LCMX,NREG,LCFW(ML,1)                                 S2DIR      2
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT       3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)             INIT       4
      COMMON/XEOS/IX                                                    XEOS       2
      DIMENSION ZB(3)                                                   RSESAME   10
      DATA LCNT/1/                                                      RSESAME   11
      DATA LU1,LU2/21,22/   Unit # for SES2L                            RSESAME   12
      IR=IX-1   Region # is what it will be after RSTORE is called      RSESAME   13
      IF(LCNT.GT.1) GO TO 10                                            RSESAME   14
      NREG=ML                                                           RSESAME   15
      LCMX=NSD                   Initialize                             RSESAME   16
      DO 5 I=1,NREG                                                     RSESAME   17
      LCFW(I,1)=0                                                       RSESAME   18
 5    CONTINUE                                                          RSESAME   19
 10   CALL GETINV(IR,MAT(IX),1,DC,LCNT,LU2,IFL,ZB)   Store data         RSESAME   20
      IF(IFL.LT.0) GO TO 90                                             RSESAME   21
      IF(IFL.GT.0) GO TO 100                                            RSESAME   22
      PRINT 20,MAT(IX)                                                  RSESAME   23
 20   FORMAT(* SESAME MATERIAL * ,I5, * NOT FOUND *)                    RSESAME   24
      RETURN                                                            RSESAME   25
 90   PRINT 95,MAT(IX)                                                  RSESAME   26
 95   FORMAT(* NOT ENOUGH STORAGE SPACE TO LOAD SESAME MATERIAL *,      RSESAME   27
     $ I5)                                                              RSESAME   28
      RETURN                                                            RSESAME   29
C . . MATERIAL LOADED                                                   RSESAME   30
 100  ROW(IX)=DC(LCFW(IR,1)+1)   Get density from SESAME                RSESAME   31
      ROW(1)=ROW(IX)                                                    RSESAME   32
      RETURN                                                            RSESAME   33
      END                                                               RSESAME   46
```

## RSESAME

Reads data from disk for SESAME materials.

### Local Variables

I = do loop index.

IFL = error flag, see GETINV.

IR = region #.

LCNT = position in array for storing tables, see GETINV.

LU2 = unit # for reading SESAME library.

ZB = output array from GETINV that is not used.

### Notes

Calls GETINV of the SESAME package to initialize data. If SESAME is not defined (*DEFINE SESAME not in the update input file), a message is returned to the terminal and all print files that a compilation including SESAME is required to run a problem with a SESAME EOS. Execution of the program then terminates. Sample update commands to define SESAME are given below:

```
*ID SMDF
*B MAIN.1
*DEFINE SESAME
```

```
      SUBROUTINE JMNMX(NMAX)                                              JMNMX      2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,           PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                    PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                      PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                          PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                           MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)     MCELL      3
     +,W(MCL)                                                             MCELL      4
      LEVEL 2,R                                                           MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,   MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),TJK                      MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                    MCELL      8
      LEVEL 2,TIME                                                        MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),YV(ML),NV(ML),VO(ML),PO   INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),  INIT       3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)              INIT       4
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                                 MN         2
      COMMON/ES/IE(ML2),NME                                               ESM        2
      DATA IST/1/,NMC/1/                                                  JMNMX      8
      IF(IST.NE.1) GO TO 2                                                JMNMX      9
      IST=2                                                               JMNMX     10
      IF(TIME.NE.0..AND.NM1.EQ.1)GO TO 2  Don't set KMIN,KMAX for a restart JMNMX   11
      DO 1 I=NM1,NM                        unless there are changes       JMNMX     12
      KMAX(I)=JMAX(I)                                                     JMNMX     13
      KMIN(I)=JMIN(I)                                                     JMNMX     14
    1 CONTINUE                                                            JMNMX     15
      NME=NM                                                              JMNMX     16
    2 CONTINUE                                                            JMNMX     17
      JM=0                                                                JMNMX     18
      IF(NMAX.GT.KMAX(NM))NMAX=KMAX(NM)                                   JMNMX     19
      DO 10 I=NMC,NM                                                      JMNMX     20
      IF(JM.NE.0) GO TO 10                                                JMNMX     21
      IF(NMAX.GT.KMAX(I)) GO TO 10                                        JMNMX     22
      JM=I                                       Reset JMAX's to correspond JMNMX   23
   10 CONTINUE                                    to NMAX active cells    JMNMX     24
      IF(JM.EQ.0) JM=NMC                                                  JMNMX     25
      IF(JM.NE.NMC) JMAX(NMC)=KMAX(NMC)                                   JMNMX     26
      NMC=JM                                                              JMNMX     27
      JMAX(NMC)=NMAX                                                      JMNMX     28
      IF(NMAX.LT.JMIN(NMC)) JMAX(NMC)=JMIN(NMC)                           JMNMX     29
      RETURN                                                              JMNMX     30
      END                                                                 JMNMX     31
```

## JMNMX(NMAX)

Sets indices to determine the minimum and maximum cell numbers for each material. Also sets indices for the last region with a cell turned on and the last cell turned on.

Local Variables

NMAX = cell # of the last cell turned on (subroutine argument).

IST = index that is 1 the 1st time through and 2 thereafter.

I = do loop index = region #.

JM = variable used to find the region which NMAX is in and reset NMC if necessary.

Notes

The first time through, KMIN and KMAX are set to JMIN and JMAX which were determined in SETUP. If the problem has been restarted (TIME $\neq$ 0 the 1st time through) and no changes have been made (NM1 = 1), then it is not necessary to change KMIN and KMAX. In the current usage, KMIN is always the same as JMIN. KMAX is the same as JMAX except for the last region that is turned on. In that case, JMAX is the last cell turned on in the region and KMAX is the last cell in the region.

Each time through, a test is made to determine which region NMAX is in. JMAX for that region is set to NMAX. Sometimes NMAX will be the cell number of an artificial cell used for voids. In that case, JMAX of the next region is set to JMIN. Also, the index NMC (which is the region number of the last active region) is reset accordingly. See SHFT for the effects of spalling and rezoning on the cell numbering.

```
      SUBROUTINE HEI(I)                                                   HEI        2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,           PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                    PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTA8=1,NTA8=MTA8*3742                      PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                          PARAM      5
      COMMON/ES/IE(ML2),NME                                               ESM        2
      COMMON/GASC/GC(NGC,ML)                                              GC         2
      COMMON/8UX/8UA,8U8,8UMAX,8UDV(ML)                                   8UP        2
     +,8UR,8UD                                                            8UP        3
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO   INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),I8RN(ML),PLAP(ML),DRO(ML),  INIT       3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)              INIT       4
      COMMON/EOSN/IEOS(ML),ME(ML)                                         EN         2
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                           MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)     MCELL      3
     +,W(MCL)                                                             MCELL      4
      LEVEL 2,R                                                           MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,   MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                      MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                    MCELL      8
      LEVEL 2,TIME                                                        MCELL      9
      DIMENSION G(1)                                                      HEI       10
      EQUIVALENCE (G,GC)                                                  HEI       11
      IM=I-1                                                              HEI       12
      JMN=JMIN(IM)                                                        HEI       13
      JMX=JMAX(IM)                                                        HEI       14
      GO TO (1,2,99,99),IEOS(I)                                           HEI       15
    1 CONTINUE   HOM EOS                                                  HEI       16
      K=IM*NGC                                                            HEI       17
      XMT=0.                                                              HEI       18
      DO 11 J=JMN,JMX                                                     HEI       19
   11 XMT=XMT+XM(J)    Total mass                                         HEI       20
      X=G(K+19)                                                           HEI       21
      EE(3*ML+IM)=(EXP(G(K+6)+X*(G(K+7)+X*(G(K+8)+X*(G(K+9)+X*G(K+10)     HEI       22
     +))))-G(K+17))*XMT   Reference energy for very large volumes         HEI       23
      RETURN                                                              HEI       24
    2 CONTINUE   BLDUP EOS                                                HEI       25
      E8=0.                                                               HEI       26
      DO 10 J=JMN,JMX                                                     HEI       27
   10 E8=E8+XM(J)/(T(J)*T(J)-1.)   Σ Mj/(γj² - 1)                         HEI       28
      EE(3*ML+IM)=E8*8UDV(I)/2    Σ MjD²/[2(γj² - 1)]                     HEI       29
      RETURN                                                              HEI       30
   99 EE(ML*3+IM)=0.   No shift calculated                                HEI       31
      RETURN                                                              HEI       32
      END                                                                 HEI       33
```

# HEI

Calculates the total internal energy of a region of solid HE relative to the energy of its products at infinite expansion at T = 0.

Local Variables

EB = the sum over cells in the region of $M_j/(\gamma^2 - 1)$.

IM = I - 1 = the actual region # (EOS data is still stored in the region # + 1 when HEI is called).

JMN, JMX = minimum and maximum cell # in the region.

K = index to locate the data for this region in G.

G = gas constants for a HOM gas.

X = $\ell n$ P at which the reference energy is set to a constant (see GASLM).

XMT = total "mass" in a region.

Notes

The total internal energy at infinite expansion for HE products is calculated for each region that is an HE. This constant is used to shift the total energy and total internal energy calculations for those regions. The shifted energies are relative to the infinite expansion energy and, therefore, reflect how much energy has not been transferred from the HE to other regions in the system.

For the HOM EOS the reference energy becomes a constant calculated in GASLIM for very large volumes. This value is used for the infinite expansion energy.

For Buildup EOS, the infinite expansion energy becomes

$$K = -\frac{D^2}{2(\gamma^2 - 1)} \quad ,$$

because the $\gamma$-law energy term goes to zero at large volume. Since $\gamma$ is not the same for all cells, the sum of the energy in each cell is calculated.

```
      SUBROUTINE GASLM(I)                                                  GASLM    2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,            PARAM    2
     +NUMV=10,MQC=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                     PARAM    3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                       PARAM    4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                           PARAM    5
      COMMON/GASC/GC(NGC,ML)                                               GC       2
      EQUIVALENCE (GC,G)                                                   GASLM    5
      DIMENSION G(MLGC)                                                    GASLM    6
      K=(I-1)*NGC                                                          GASLM    7
      IF(G(K+15).EQ.0.)GO TO 15                   General case for -1/β    GASLM    8
      TU=G(K+14)/(4*G(K+15))                                               GASLM    9
      G(K+18)=-TU+SQRT(TU*TU-G(K+13)/(6*G(K+15)))                          GASLM   10
      GO TO 14                                                             GASLM   11
   15 CONTINUE                                                             GASLM   12
      IF(G(K+14).EQ.0.)GO TO 14                                            GASLM   13
      G(K+18)=G(K+13)/(3*G(K+14))   Special case for U = 0                 GASLM   14
   14 CONTINUE                                                             GASLM   15
      IF(G(K+10).EQ.0.)GO TO 10                                            GASLM   16
      P=0.75*G(K+9)/G(K+10)                                                GASLM   17
      Q=0.5*G(K+8)/G(K+10)                                                 GASLM   18
      R=0.25*G(K+7)/G(K+10)                                                GASLM   19
      A=(3*Q-P*P)/3                              General case for ℓn I     GASLM   20
      B=(2*P*P*P-9*P*Q+27*R)/27                                            GASLM   21
      SBA=SQRT(B*B/4+A*A*A/27)                                             GASLM   22
      OT=1./3.                                                             GASLM   23
      BT=-B/2                                                              GASLM   24
      BTMS=BT-SBA                                                          GASLM   25
      BTPS=BT+SBA                                                          GASLM   26
      G(K+19)=SIGN(1.,BTPS)*ABS(BTPS)**OT+SIGN(1.,BTMS)*ABS(BTMS)**OT      GASLM   27
     +-P/3                                                                 GASLM   28
      RETURN                                                               GASLM   29
   10 CONTINUE                                                             GASLM   30
      IF(G(K+9).EQ.0.)GO TO 9  Special case for O = 0                      GASLM   31
      G(K+19)=(-2*G(K+8)+SQRT(4*G(K+8)**2-12*G(K+9)*G(K+7)))/(6*G(K+9))    GASLM   32
      RETURN                                                               GASLM   33
    9 CONTINUE                                                             GASLM   34
      IF(G(K+8).EQ.0.)GO TO 8                                              GASLM   35
      G(K+19)=G(K+7)/(2*G(K+8))   Special case for O = N = 0               GASLM   36
    8 CONTINUE                                                             GASLM   37
      RETURN                                                               GASLM   38
      END                                                                  GASLM   39
```

## GASLM(I)

Finds limits for the region in which two of the analytic fits in GAS are reasonable.

### Local Variables

G = equivalenced to GC but only one index.

K = Index to find the proper location in the G array for region I.

TU,P,Q,R,A,B = defined in notes below.

$$SBA = \sqrt{\frac{B^2}{4} + \frac{A^3}{27}} .$$

OT = 1/3.

BT = -B/2.

BTMS = BT - SBA.

### Notes

Two of the analytic fits in GAS can cause problems in regions of high specific volume (typically $\approx 10\ V_0 - 100\ V_0$). In most problems the volumes will be small enough for the fits to be good during the time of interest. However, in cases where an HE decomposes a few percent and then expands, most of the expansion is done by the product gas. This situation can lead to very large volumes for the gas even though the specific volume of the mixture is still $\approx 2\ V_0$.

The fits that cause problems are

$$-\frac{1}{\beta} = R + 2S(\ell n\ V) + 3T(\ell n\ V)^2 + 4U(\ell n\ V)^3 \tag{1}$$

and

$$\ell n\ I_i' = K + L(\ell n\ P_i) + M(\ell n\ P_i)^2 + N(\ell n\ P_i)^3 + O(\ell n\ P_i)^4 \quad . \tag{2}$$

For a typical choice of constants, the fit to $-\frac{1}{\beta}$ has a relative minimum and a relative maximum. For large V, the cubic term dominates and $-\frac{1}{\beta}$ becomes very large. In the Grüneisen EOS,

$$P = \frac{1}{\beta V}(I - I_i) + P_i \quad , \tag{3}$$

a large $-\frac{1}{\beta}$ gives large vlaues of $|P|$ in a region where it should be small. We have artificially chosen V at the second extremum (denoted $V_2$) to be the largest value of V at which the fit in Eq. (1) will be used. For larger V, the value of $-\frac{1}{\beta}$ at $V_2$ is used.

The extrema can be found at the zeros of $\frac{d}{dV}(-\frac{1}{\beta})$, i.e., solve the equation

$$\frac{d}{dV}\left(R + 2S(\ell n\ V) + 3T(\ell n\ V)^2 + 4U(\ell n\ V)^3\right) = 0 \tag{4}$$

to find the values of $\ell n$ V at the extrema. Eq. (4) simplifies to

$$\frac{S}{6U} + \frac{T}{2U}\ell n\ V + (\ell n\ V)^2 = 0 \quad , \tag{5}$$

which has the solutions

$$\ell n\ V = \frac{-\frac{T}{2U} \pm \sqrt{\left(\frac{T}{2U}\right)^2 - 4\left(\frac{S}{6U}\right)}}{2} \quad . \tag{6}$$

We will use the greater of the two which is

$$\ell n\ V_2 = -\frac{T}{4U} + \sqrt{\left(\frac{T}{4U}\right)^2 - \frac{S}{6U}} \quad . \tag{7}$$

For a typical choice of constants, the fit to $\ell n\ I_i$ has one extremum, which is found by solving

$$\frac{d}{d(\ell n\ P)}\left[K + L(\ell n\ P) + M(\ell n\ P)^2 + N(\ell n\ P)^3 + O(\ell n\ P)^4\right] = 0 \tag{8}$$

or

$$L + 2M(\ell n\ P) + 3N(\ell n\ P)^2 + 4O(\ell n\ P)^3 = 0 \quad . \tag{9}$$

In the CRC Standard Math Tables we see that the general cubic equation, $y^3 + Py^2 + Qy + R = 0$, may be put in the form

$$x^3 + Ax + B = 0 \quad , \tag{10}$$

where

$$x = y + \frac{P}{3} \quad , \tag{11}$$

$$A = \frac{1}{3}(3Q - P^2) \quad , \tag{12}$$

and

$$B = \frac{1}{27}(2P^3 - 9PQ + 27R) \quad . \tag{13}$$

The real solution for the case where there is one real root is given by

$$x = \sqrt[3]{-\frac{B}{2} + \sqrt{\frac{B^2}{4} + \frac{A^3}{27}}} + \sqrt[3]{-\frac{B}{2} - \sqrt{\frac{B^2}{4} + \frac{A^3}{27}}} \quad . \tag{14}$$

```
      FUNCTION BLDSM(DR,I)                                              BLDSM     2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM     2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM     3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM     4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM     5
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)                                 BUP       2
     +,BUR,BUD                                                          BUP       3
      DATA IL/0/                                                        BLDSM     5
      IF(BUD.LE.0.)GO TO 2                                              BLDSM     6
      IF(I.EQ.IL)GO TO 1                                                BLDSM     7
      IL=I                                    ]                         BLDSM     8
      XX=BUB/(BUMAX-BUA)                      |                         BLDSM     9
      B=XX-BUD                                |  Set up smoothing       BLDSM    10
      X=(3*BUB+SQRT(9*BUB*BUB-8*(BUMAX-BUA)*BUB*B))/ |  function        BLDSM    11
     +(4*(BUMAX-BUA))                         |                         BLDSM    12
      A=BUB/(2*(X-B)*X*X)                     ]                         BLDSM    13
    1 CONTINUE                                                          BLDSM    14
      IF(DR.LT.B.OR.DR.GT.X)GO TO 2                                     BLDSM    15
      BLDSM=BUMAX-A*(DR-B)**2  Evaluate smoothing function              BLDSM    16
      RETURN                                                            BLDSM    17
    2 CONTINUE                                                          BLDSM    18
      BLDSM=BUA+BUR/DR                 ]  Usual functional form of γ    BLDSM    19
      IF(BLDSM.GT.BUMAX)BLDSM=BUMAX    ]                                BLDSM    20
      RETURN                                                            BLDSM    21
      END                                                               BLDSM    22
```

## BLDSM

Calculates the $\gamma$ for each cell using the buildup model. The transition from constant $\gamma_{max}$ to the $\gamma = A + B/R$ form is smoothed out with a parabola which joins both curves, leaving the first derivative continuous.

Local Variables

A = a in notes.

B = b in notes.

IL = last region #.

X = X in notes.

XX = $X_0$ in notes.

Notes

The $\gamma$ used in the buildup model is a function of the form

$$\gamma = \text{Min}\ (A + B/R,\ \gamma_{max}) \quad . \tag{1}$$

This function has a discontinuous first derivative at

$$X_0 = \frac{B}{\gamma_{max} - A} \quad . \tag{2}$$

For some cases this can cause a perturbation in the numerical solution that is eliminated by smoothing out the discontinuity. A convenient way to smooth out the discontinuity is to find a parabola with the following properties.

(1) It intersects the line $\gamma = \gamma_{max}$ at $R = X_0 - \text{BUD} \equiv b$ with zero slope; that is,

$$f(R) = \gamma_{max} - a(R - b)^2 \tag{3}$$

is a function with the desired property. (2) It intersects $\gamma = A + B/R$ at some point X, such that

$$\gamma_{max} - a(X - b)^2 = A + B/X \quad . \tag{4}$$

(3) At this point X, the derivatives of the two curves are equal; that is

$$-2a(X - b) = -\frac{B}{X^2} \quad . \tag{5}$$

Combining Eqs. (4) and (5), we get

$$X = \frac{3B + \sqrt{9B^2 - 8(\gamma_{max} - A)bB}}{4(\gamma_{max} - A)} \tag{6}$$

and

$$a = \frac{B}{2X^2(X - b)} \quad . \tag{7}$$

Equation (3) is then used to calculate $\gamma$ from b to X.

```
      SUBROUTINE PRNT                                                  PRNT       2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT       3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT       4
      COMMON/EOSN/IEOS(ML),ME(ML)                                       EN         2
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                         VD         2
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL      3
     +,W(MCL)                                                           MCELL      4
      LEVEL 2,R                                                         MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL      7
      COMMON/MISC/TIME,ICYCL,DI,NCL,IA,BU,BUI,F2,F3,JS                  MCELL      8
      LEVEL 2,TIME                                                      MCELL      9
      COMMON/UCJC/UCJ,JJ,NMAX,RCJ,DCJ                                   UC         2
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                               MN         2
      COMMON/ES/IE(ML2),NME                                             ESM        2
      WRITE(8,4)     TAPE8 = XOUT                                       PRNT      11
    4 FORMAT(1H1)                                                       PRNT      12
      WRITE(8,6)LABEL   Problem label                                   PRNT      13
    6 FORMAT(1X,8A10)                                                   PRNT      14
      WRITE(8,5)TIME,DT,ICYCL   Time (μs), Δt, cycle #                  PRNT      15
    5 FORMAT(//1X,5HTIME=,1PE16.5,1X,4HDT= ,1PE12.5,1X,7HCYCLE= ,I10//) PRNT      16
      PRINT 5,TIME,DT,ICYCL   PRINT → OUTPUT                            PRNT      17
      CALL ESUM   Calculate energy sums                                 PRNT      18
      DO 20 I=1,NME                                                     PRNT      19
      WRITE(8,96)I,EE(I),EE(I+ML),EE(I+2*ML)] Total, kinetic, and internal PRNT   20
      PRINT 96,I,EE(I),EE(I+ML),EE(I+ML*2) ]  energy for each region    PRNT      21
   20 CONTINUE                                                          PRNT      22
      WRITE(8,95)EE(ML),EE(2*ML),EE(3*ML)] Total, kinetic, and internal PRNT      23
      PRINT 95,EE(ML),EE(2*ML),EE(3*ML)  ]  energy for the problem      PRNT      24
   95 FORMAT(1X,14H TOTAL ENERGY=,1PE12.5,23H TOTAL INTERNAL ENERGY=,   PRNT      25
     +1PE12.5,22H TOTAL KINETIC ENERGY=,1PE12.5)                        PRNT      26
   96 FORMAT(1X,10H MATERIAL ,I2,8H ENERGY=,1PE12.5,                    PRNT      27
     +17H INTERNAL ENERGY=,1PE12.5,16H KINETIC ENERGY=,1PE12.5)         PRNT      28
      WRITE(8,97)   Cell quantity labels                                PRNT      29
   97 FORMAT(1H0,2X,10HJ IEOS MAT,4X,1HM,10X,1HR,10X,1HV,10X,1HU,10X    PRNT      30
     +,1HI,10X,1HT,10X,1HP,10X,1HQ,9X,2HSX,9X,2HSZ,10X,1HW/)            PRNT      31
      IF(BU.EQ.0.)GO TO 10                                              PRNT      32
      WRITE(8,80)R(1),U(1)    Piston radius and velocity                PRNT      33
   80 FORMAT(3X,10H1 PISTON  ,10X,2(1PE10.3,12X))                       PRNT      34
      WRITE(8,81)  Region separator                                     PRNT      35
   81 FORMAT(1X,131(1H-))                                               PRNT      36
   10 CONTINUE                                                          PRNT      37
      NMP=NMC                                                           PRNT      38
      IF(ICYCL.EQ.0)NMP=NM    Print all cells for cycle 0               PRNT      39
      DO 11 I=1,NMP   Print all active cells                            PRNT      40
      JMN=JMIN(I)                                                       PRNT      41
      JMX=JMAX(I)                                                       PRNT      42
      IF(ICYCL.EQ.0)JMX=KMAX(I)                                         PRNT      43
      II=IE(I)                                                          PRNT      44
      DO 12 J=JMN,JMX                                                   PRNT      45
      WRITE(8,82)J,IEOS(II),MAT(II),XM(J),R(J),V(J),U(J),XI(J),T(J)     PRNT      46
     +,P(J),Q(J),SX(J),SZ(J),W(J)    Print cell quantities              PRNT      47
   82 FORMAT(1X,2I3,I5,11(1PE10.3,1X))                                  PRNT      48
   12 CONTINUE                                                          PRNT      49
```

```
      JMP=JMX+1                                                          PRNT
      IF(II.EQ.IF(I+1))GO TO 13                                          PRNT
      WRITE(8,81)  Region separator                                      PRNT
      IF(IV(I).NE.-1)WRITE(8,84)JMP,R(JMP),U(JMP)  Radius and velocity at the  PRNT
      GO TO 14                                      outside of a void ini-     PRNT
   13 CONTINUE                                      tially in the problem      PRNT
      WRITE(8,83)  Interface due to spalling                             PRNT
   83 FORMAT(1X,131(1H.))                                                PRNT
      IF(IV(I).NE.-1)WRITE(8,85)JMP,R(JMP),U(JMP)  Radius and velocity at the  PRNT
   14 CONTINUE                                      outside of a void due to   PRNT
                                                    spalling
   84 FORMAT(1X,I3,5H VOID,2X,2(12X,1PE10.3)/1X,131(1H-))               PRNT
   85 FORMAT(1X,I3,5H VOID,2X,2(12X,1PE10.3)/1X,131(1H.))               PRNT
   11 CONTINUE                                                           PRNT
      IF(NMAX.LT.NCL-1)GO TO 15                                          PRNT
      WRITE(8,86)NCL,R(NCL),U(NCL)   Innermost radius and velocity       PRNT
   86 FORMAT(1X,I3,7H INSIDE,12X,2(1PE10.3,12X))                         PRNT
      NCLP=NCL+1                                                         PRNT
      IF(BUI.NE.0.)WRITE(8,87)NCLP,R(NCLP),U(NCLP)   Inside piston       PRNT
   87 FORMAT(1X,131(1H-)/1X,I3,7H PISTON,12X,2(1PE10.3,12X))             PRNT
   15 CONTINUE                                                           PRNT
      WRITE(8,5)TIME,DT,ICYCL                                            PRNT
  900 FORMAT(1X,I3,11E11.4)                                              PRNT
      RETURN                                                             PRNT
      END                                                                PRNT
```

## PRNT

Makes a cycle printout including time, $\Delta t$, cycle #, region and total energies, and cell quantities for active cells.

Local Variables

I = region #.

II = IE(I) = original region #.

J = cell #.

JMN,JMX = JMIN(I),JMAX(I) = minimum and maximum active cell # in a region; for cycle #0, JMX is the maximum cell # in a region.

JMP = JMX+1 = cell # of inside radius of a region.

NCLP = NCL+1 = cell # of the inside piston radius.

NMP = NMC for any cycle # except 0; NMP = NM for cycle #0 so that all cells are printed with their initial conditions.

Notes

The cell quantity lines are longer than 120 characters, so the LONG. option should be used in ALLOUT for printing XOUT or OUTPUT. The dashed region separators are for the original regions in the problem. The dotted region separators are for new regions created by spalling.

```
      SUBROUTINE ESUM                                                   ESUM
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),TFLAG(MCL)   MCELL
     +,W(MCL)                                                           MCELL
      LEVEL 2,R                                                         MCELL
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL
      LEVEL 2,TIME                                                      MCELL
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT
      COMMON/ES/IE(ML2),NME                                             ESM
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                               MN
      COMMON/PWORK/PW,PWI                                               PWORK
      DATA PI/3.1415926535/                                             ESUM
      M3=ML*3                                                           ESUM
      GF=PI*IA*2        ] Geometry-dependent factor                     ESUM
      IF(IA.EQ.0)GF=1.  ]                                               ESUM
      DO 10 I=1,M3                                                      ESUM
   10 EE(I)=0.                                                          ESUM
      DO 20 I=1,NM                                                      ESUM
      JMN=JMIN(I)                                                       ESUM
      JMX=KMAX(I)                                                       ESUM
      IT=IE(I)   Original region #; sums are over original regions      ESUM
      II=IT+ML                                                          ESUM
      IK=II+ML                                                          ESUM
      DO 21 J=JMN,JMX                                                   ESUM
      EE(II)=XM(J)*XI(J)+EE(II)  Sum of term proportional to internal energy ESUM
   21 EE(IK)=XM(J)*(U(J)+U(J+1))**2+EE(IK) Sum of term proportional to  ESUM
   20 CONTINUE                                    kinetic energy        ESUM
      IJ=IE(NM)                                                         ESUM
      DO 23 I=1,IJ                                                      ESUM
      II=I+ML                                                           ESUM
   23 EE(II)=EE(II)+EE(II+2*ML)  Add energy shift for HE's from HEI(q.v.) ESUM
      ITWML=2*ML                                                        ESUM
      ITHML=3*ML                                                        ESUM
      DO 22 I=1,IJ                                                      ESUM
      IML=I+ML                                                          ESUM
      ITML=IML+ML                                                       ESUM
      EE(IML)=EE(IML)*GF   Scale internal energy to proper value        ESUM
      EE(ITML)=EE(ITML)*GF/8  Scale kinetic energy to proper value      ESUM
      EE(I)=EE(IML)+EE(ITML)  Total energy for the region               ESUM
      EE(ML)=EE(ML)+EE(I)   Total energy for the problem                ESUM
      EE(ITWML)=EE(ITWML)+EE(IML)   Total internal energy for the problem ESUM
      EE(ITHML)=EE(ITHML)+EE(ITML)   Total kinetic energy for the problem ESUM
   22 CONTINUE                                                          ESUM
      IFML=4*ML                                                         ESUM
      EE(IFML+1)=GF*PW  ] Scale piston work                             ESUM
      EE(IFML+2)=GF*PWI ]                                               ESUM
      EE(ML)=EE(ML)+EE(IFML+1)+EE(IFML+2) Add work done by the pistons to the ESUM
      RETURN                                  total energy of the problem ESUM
      END                                                               ESUM
```

## ESUM

Calculates kinetic, internal, and total energies for each region and for the whole problem.

### Local Variables

GF = geometry factor = 1,2$\pi$,4$\pi$ for $\alpha$ = 1,2,3.

I = do loop index (usually region #).

IFML = 4*ML.

II = original region # + ML = index for internal energy.

IJ = original # of regions.

IK = original region # + 2*ML = index for kinetic energy.

IML = same as II.

IT = original region #.

ITML = same as IK.

ITWML = 2*ML = index for total internal energy.

ITHML = 3*ML = index for total kinetic energy.

J = cell #.

JMN,JMX = Minimum and maximum active cell #'s.

M3 = 3*ML.

### Notes

The variable EE contains energy sums in cell quantities in the following order:

| | |
|---|---|
| 1 to ML-1 | total energy for region 1 to ML-1 |
| ML | total energy for the problem plus work done on pistons |
| ML+1 to 2*ML-1 | internal energy for region 1 to ML-1 |
| 2*ML | total internal energy for the problem |
| 2*ML+1 to 3*ML-1 | kinetic energy for region 1 to ML-1 |
| 3*ML | total kinetic energy for the problem |

| | |
|---|---|
| 3*ML+1 to 4*ML-1 | shift of the energy zero for HE's so that at infinite expansion of the products, the energy will be zero |
| 4*ML+1 | work done on the outside piston |
| 4*ML+2 | work done on the inside piston |

ML is set in a parameter statement and is usually 21, the number of allowed materials plus one.

The specific internal energy in a cell is in the units Mbar-$cm^3$/g, which is $10^{12}$ erg/g. The internal energy in a cell is just the mass of the cell times the specific internal energy. The mass in variable XM is not the actual mass of the cell. For $\alpha = 1$ it is the mass per unit area so that XM(J)*XI(J) is the energy per unit area in cell J. For $\alpha = 2$, XM is the mass per unit length per unit angle. So, $2\pi$*XM(J)*XI(J) is the energy per unit length. For $\alpha = 3$, XM is the mass per unit solid angle. So, $4\pi$*XM(J)*XI(J) is the energy in cell J.

The kinetic energy is calculated from $\frac{1}{2} mv^2$. For mass in grams and velocity in cm/μs, the unit of kinetic energy is $10^{12}$ erg = Mbar-$cm^3$, which agrees with the unit for internal energy. There is also a geometry factor of $1, 2\pi$, and $4\pi$ of $\alpha = 1,2,3$, respectively. Also, as above, the energy is per unit area for $\alpha = 1$ and per unit length for $\alpha = 2$. A cell-centered velocity is used to calculate the kinetic energy in a cell.

The energies for a region are calculated for the original region even if it is later split into more than one region. For HE's, the internal energy has a constant added so that the energy zero is shifted to that of the products expanded to infinite volume. For the total energy of the problem, the work done on pistons is also included so that the total energy should be constant to a good approximation.

```
      SUBROUTINE WDUMP                                                       WDUMP
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,              PARAM
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                       PARAM
     +MXDUMP=30,NDX=2*MXDUMP+2,MTA8=1,NTA8=MTA8*3742                         PARAM
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                             PARAM
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                              MCELL
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),TFLAG(MCL)        MCELL
     +,W(MCL)                                                                MCELL
      LEVEL 2,R                                                              MCELL
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,      MCELL
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                         MCELL
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                       MCELL
      LEVEL 2,TIME                                                           MCELL
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VD(ML),PO      INIT
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),     INIT
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)                 INIT
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),              US
     +GAMMA(ML),ALP(ML)                                                      US
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML)     BRD
     +,MSFF                                                                  BRD
      COMMON/GASC/GC(NGC,ML)                                                 GC
      COMMON/FGHTJC/FS(ML),GS(ML),HS(ML),SI(ML),SJ(ML),CV(ML)                FG
      COMMON/UCJC/UCJ,JJ,NMAX,RCJ,DCJ                                        UC
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                              VD
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                                    MN
      COMMON/BRNS/A(ML),BR(ML),BA(ML),VBO(ML),VBSW(ML)                       BRN
      COMMON/EOSN/IEOS(ML),ME(ML)                                            EN
      COMMON/NSPLT/NOSPLT(ML2)                                               NSP
      COMMON/SPC/SP(ML),USP(ML)                                              SPLC
     +,XISP(ML)                                                              SPLC
      COMMON/POLYC/CF(NCF,ML),PS(ML)                                         PLC
      COMMON/GAS/FI(1003),DI(MQL)                                            GS
      LEVEL 2,FI                                                             GS
      COMMON/LEV/DMPNO                                                       GS
      LEVEL 2,DMPNO                                                          GS
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)                                      BUP
     +,BUR,BUD                                                               BUP
      COMMON/RLC/RC(ML),RP(ML),RLV(ML),PH1,DV1,DV2                           RLC
      DIMENSION XINDX(NDX),INDX(NDX)                                         WDUMP
      EQUIVALENCE (XINDX(1),INDX(1))                                         WDUMP
      DATA IST/1/                                                            WDUMP
      IF(IST.NE.1)GO TO 2   Initialize                                       WDUMP
      IST=2                                                                  WDUMP
      I=1                                                                    WDUMP
      J=0                                                                    WDUMP
      L1=LOCF(DV2)-LOCF(NDF)+1       SCM length                              WDUMP
      L2=LOCF(JS)-LOCF(R(1))+1       LCM length                              WDUMP
      L12=L1+L2  Total length per dump                                       WDUMP
      LEN=(L12+2)*MXDUMP+2  Total length of file to allow MXDUMP restart dumps WDUMP
      INDX(1)=L12                                                            WDUMP
      INDX(2)=0                                                              WDUMP
      CALL QASSIGN(1,5HDUMPO,0,0) ]                                          WDUMP
      CALL FAMSIZ(1,LEN)          ] Set up DUMPO file                        WDUMP
      CALL FAMWATT(1,1)           ]                                          WDUMP
    2 CONTINUE                                                               WDUMP
      J=J+1                  ]                                               WDUMP
      INDX(2)=J              ] The index has the form:                       WDUMP
      I=I+2                  ]  L12, last dump #, cycle1, time1, ..., cycleMXDUMP, WDUMP
      INDX(I)=ICYCL          ]  timeMXDUMP                                   WDUMP
      XINDX(I+1)=TIME        ]                                               WDUMP
```

```
      CALL WDISK(1,XINDX,NDX,0)  Write the index                        WDUMP
      IF(UNIT(1))10,10,10                                               WDUMP
   10 CONTINUE                                                          WDUMP
      N1=NDX+L12*(J-1)]  Find the locations for the SCM and LCM dumps   WDUMP
      N2=N1+L1        ]                                                 WDUMP
      CALL WDISK(1,NDF,L1,N1)  Write the SCM dump                       WDUMP
      IF(UNIT(1))11,11,11                                               WDUMP
   11 CONTINUE                                                          WDUMP
      CALL WDISK(1,R,L2,N2)  Write the LCM dump                         WDUMP
      IF(UNIT(1))12,12,12                                               WDUMP
   12 CONTINUE                                                          WDUMP
      PRINT 100,J,ICYCL,TIME                                            WDUMP
  100 FORMAT(1X,4HDUMP,I4,9H AT CYCLE,I10,6H,TIME=,E12.5)               WDUMP
      IF(J.LT.MXDUMP)RETURN                                             WDUMP
      PRINT 101                      ]                                  WDUMP
      WRITE(9,101)                   ]  If this is the last allowed dump, STOP  WDUMP
  101 FORMAT(16H LAST DUMP: STOP)    ]                                  WDUMP
      STOP                                                              WDUMP
      END                                                               WDUMP
```

## WDUMP

Writes a restart dump (all of the necessary data to restart the problem at a given cycle). (Inactive regions may be replaced with new setup information so that two different problems that start out the same may be restarted at a time before they differ without completely rerunning the problem.)

### Local Variables

I = location in the index where the cycle # at this dump is stored = 2*J.

J = dump #.

L1 = # of words in SCM to be dumped.

L2 = # of words in LCM to be dumped.

L12 = L1 + L2.

LEN = total length of the dump file required to make MXDUMP dumps.

N1 = starting location for writing the SCM data.

N2 = starting location for writing the LCM data.

### Notes

All of the variables to be dumped are in common blocks. The order in which they are stored in memory is specified by the order they appear in MAIN. The SCM and LCM data are separated so they are dumped separately. The number of words of SCM and LCM data are each determined by using the standard function LOCF (which returns the address for a given variable) to get the location of the first variable in the first common block and the last variable in the last common block to be dumped. The index is written in the following order beginning with word 0 of the dump file: L12, last dump #, 1st cycle #, 1st time, 2nd cycle #, 2nd time, ···, last cycle #, last time. If the last allowed dump (MXDUMP is usually 30) is made, the problem is stopped.

```
      SUBROUTINE RDUMP(I)                                                RDUMP     2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,          PARAM     2
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                   PARAM     3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                     PARAM     4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                         PARAM     5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                          MCELL     2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)    MCELL     3
     +,W(MCL)                                                            MCELL     4
      LEVEL 2,R                                                          MCELL     5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFT,NADD,NM,  MCELL     6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                     MCELL     7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                   MCELL     8
      LEVEL 2,TIME                                                       MCELL     9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO  INIT      2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT      3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)             INIT      4
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),          US        2
     +GAMMA(ML),ALP(ML)                                                  US        3
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML) BRD       2
     +,MSFF                                                              BRD       3
      COMMON/GASC/GC(NGC,ML)                                             GC        2
      COMMON/FGHTJC/FS(ML),GS(ML),HS(ML),SI(ML),SJ(ML),CV(ML)            FG        2
      COMMON/UCJC/UCJ,JJ,NMAX,RCJ,DCJ                                    UC        2
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                          VD        2
      COMMON/MNMY/KMAX(ML2),KMIN(ML2),NMC                                MN        2
      COMMON/BRNS/A(ML),BR(ML),BA(ML),VBO(ML),VBSW(ML)                   BRN       2
      COMMON/EOSN/IEOS(ML),ME(ML)                                        EN        2
      COMMON/NSPLT/NDSPLT(ML2)                                           NSP       2
      COMMON/SPC/SP(ML),USP(ML)                                          SPLC      2
     +,XISP(ML)                                                          SPLC      3
      COMMON/POLYC/CF(NCF,ML),PS(ML)                                     PLC       2
      COMMON/GAS/FI(1003),DI(MOL)                                        GS        2
      LEVEL 2,FI                                                         GS        3
      COMMON/LEV/DMPNO                                                   GS        4
      LEVEL 2,DMPNO                                                      GS        5
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)                                  BUP       2
     +,BUR,BUD                                                           BUP       3
      COMMON/RLC/RC(ML),RP(ML),RLV(ML),PH1,DV1,DV2                       RLC       2
      DIMENSION L(2)                                                     RDUMP    21
      CALL QASSIGN(2,5HDUMPI,0,0)  Restart dump read from file DUMPI     RDUMP    22
      CALL RDISK(2,L,2,0)    Read the first two words of the index       RDUMP    23
      IF(UNIT(2))10,10,10                                                RDUMP    24
   10 CONTINUE                                                           RDUMP    25
      IF(L(2).LT.I)GO TO 991   Are there I dumps                         RDUMP    26
      L1=LOCF(DV2)-LOCF(NDF)+1   SCM length                              RDUMP    27
      L2=LOCF(JS)-LOCF(R(1))+1   LCM length                              RDUMP    28
      L12=L1+L2   Total length                                           RDUMP    29
      N1=NDX+L12*(I-1)   SCM address                                     RDUMP    30
      N2=N1+L1    LCM address                                            RDUMP    31
      IF(L12.NE.L(1))GO TO 992   If L12 ≠ L(1), then dimensions don't match  RDUMP    32
      CALL RDISK(2,NDF,L1,N1)   Read SCM variables                       RDUMP    33
      IF(UNIT(2))11,11,11                                                RDUMP    34
   11 CONTINUE                                                           RDUMP    35
      CALL RDISK(2,R,L2,N2)  Read LCM variables                          RDUMP    36
      IF(UNIT(2))12,12,12                                                RDUMP    37
   12 CONTINUE                                                           RDUMP    38
      RETURN                                                             RDUMP    39
  991 CONTINUE                                                           RDUMP    40
      PRINT 101,I                                                        RDUMP    41
  101 FORMAT(5H DUMP,I5,21H DOES NOT EXIST! STOP)                        RDUMP    42
```

```
      WRITE(9,101)I                                                       RDUMP
      GO TO 999                                                           RDUMP
  992 CONTINUE                                                            RDUMP
      PRINT 102    In this case check for differences in the parameter statement   RDUMP
      WRITE(9,102)  or for any changes in any common block               RDUMP
  102 FORMAT(33H DUMP LENGTH DOES NOT MATCH, STOP)                        RDUMP
  999 CONTINUE                                                            RDUMP
      STOP                                                                RDUMP
      END                                                                 RDUMP
```

## RDUMP

Reads the restart dump and stores all of the data in the appropriate locations.

Local Variables

L = array containing the 1st two words of the dump index in the file DUMPI (see WDUMP).

L1 = # of words in SCM that should be read.

L2 = # of words in LCM that should be read.

L12 = L1 + L2, which = L(1) for a valid restart.

N1 = first word address for the SCM part of the dump to be read.

N2 = first word address for the LCM part of the dump to be read.

Notes

If L12 $\neq$ L(1), then the number of words in the dump does not match the variables into which the data is to be stored. Then either the code has been changed incorrectly between the two runs or one of the array sizes in the PARAMETER statement has been changed. In either case, the code would not run properly and is terminated.

```
      SUBROUTINE OUTGAS                                                 OUTGAS     2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL      3
     +,W(MCL)                                                           MCELL      4
      LEVEL 2,R                                                         MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL      8
      LEVEL 2,TIME                                                      MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT      3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT       4
      COMMON/GAS/FI(1003),DI(MQL)                                       GS         2
      LEVEL 2,FI                                                        GS         3
      COMMON/LEV/DMPND                                                  GS         4
      LEVEL 2,DMPND                                                     GS         5
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                               MN         2
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                         VD         2
      COMMON/FS/IF(ML2),NME                                             ESM        2
      DIMENSION DATIN(MCL,NUMV)                                         OUTGAS    10
      EQUIVALENCE (DATIN,R)                                             OUTGAS    11
      DIMENSION IFI(1003),IDI(MQL)                                      OUTGAS    12
      EQUIVALENCE (FI,IFI),(DI,IDI)                                     OUTGAS    13
      DIMENSION ITMP(100)                                               OUTGAS    14
      DATA (ISTFLAG=1)                                                  OUTGAS    15
      DATA IBLANK/10H          /                                        OUTGAS    16
      DATA IMC/5/                                                       OUTGAS    17
      GO TO (1,2),ISTFLAG        Initialize                             OUTGAS    18
    1 ISTFLAG=2                                                         OUTGAS    19
      IFI(1)=1003                                                       OUTGAS    20
      IFI(2)=0                                                          OUTGAS    21
      IFI(3)=1002                                                       OUTGAS    22
      NUMVP=NUMV+2   Total # of variables including region flag and time OUTGAS   23
C                                                                       OUTGAS    24
C       THIS CODE WRITES A RANDOM M2C FILE                              OUTGAS    25
C          NO CONVERSION IS NECESSARY IN GAS                            OUTGAS    26
C                                                                       OUTGAS    27
C       COMPUTE THE # OF PACKED WORDS PER CELL                          OUTGAS    28
      NWPC=(NUMVP-1)/3+1   # of packed words per cell                   OUTGAS    29
      NAF=2                                                             OUTGAS    30
      NA=3                                                              OUTGAS    31
      IF(IJK.LE.0)IJK=1    Use every IJKth cell                         OUTGAS    32
      NCLP=NCL+55    Allow 55 extra cells from splits and spalls        OUTGAS    33
      IF(NCLP.GE.MCL) NCLP=MCL-1   Don't allow for more than dimensioned OUTGAS   34
      NVN=100+NWPC*((NCLP+IJK-1)/IJK)   # words per dump                OUTGAS    35
      LENGTH=500*NVN+1003+512   Total # of words needed for 500 dumps   OUTGAS    36
      CALL FAMST7(3,LENGTH)   Make GASSIN that length                   OUTGAS    37
      CALL FAMWATT(3,1)                                                 OUTGAS    38
C     THE MAGIC FIRST HUNDRED WORDS                                     OUTGAS    39
      IDI(3)=1                                                          OUTGAS    40
      IDI(5)=NWPC   # words per cell                                    OUTGAS    41
      IDI(6)=3   # variables per word                                   OUTGAS    42
      IDI(10)=14   # of fraction bits in packed word                    OUTGAS    43
      IDI(11)=5   # of exponent bits in packed word                     OUTGAS    44
      CALL DATEH(IDATE) ]                                               OUTGAS    45
      IDI(81)=IDATE     ] Get the date and put it on graphs             OUTGAS    46
```

```
      IDI(82)=IBLANK   Label is currently blank
      IDI(83)=IBLANK
      IDI(90)=LABEL(1)   Put the 1st 30 characters of LABEL in
      IDI(91)=LABEL(2)    the classification words
      IDI(92)=LABEL(3)
      IDI(93)=1
      IDI(94)=1
    2 CONTINUE
      IF(IFI(2).GE.500) RETURN
C
      DO 10 I=1,NMC
      J=JV(I)
      IF(IV(I))10,11,12
   11 P(J)=0.
      Q(J)=0.
      SX(J)=0.
      SZ(J)=0.    Set cell quantities to 0 in the void cell for open voids
      W(J)=0.
      V(J)=0.
      XI(J)=0.
      GO TO 10
   12 P(J)=(P(J+1)+P(J-1))/2
      Q(J)=(Q(J+1)+Q(J-1))/2
      SX(J)=(SX(J+1)+SX(J-1))/2
      SZ(J)=(SZ(J+1)+SZ(J-1))/2   Interpolate cell quantities for a closed void
      W(J)=(W(J+1)+W(J-1))/2
      V(J)=(V(J+1)+V(J-1))/2
      XI(J)=(XI(J+1)+XI(J-1))/2
   10 CONTINUE
      CALL ESUM  Calculate energy sums
      DMPNO=TIME
      ITMP(2)=ICONV(DMPNO)  2nd variable is time
      N =NA+2
      DI(1)=DMPNO   Dump # is time
      IDI(2)=NCLP
C     FILE INDEX
      NAF=NAF+2    # words in file index
      FI(NAF)=DMPNO   time word
      IFI(NA)=IFI(3)+1  FWA of dump
      IFI(2)=IFI(2)+1    # of dumps
      IFI(3)=IFI(3)+NVN   Last word address of dump
C     WRITE OUT TO DISK THE FIRST 1003 WORDS, FILE INDEX
      CALL WDISK(3,FI,1003,0)
      IF(UNIT(3)) 152,152,152
  152 CONTINUE
C       PACK 'EM
      IIA=100
      I=1
      DO 210 L=1,NCL,IJK
      IF(L.GT.KMAX(I))I=I+1
      ITMP(1)=IFLAG(L)/64   1st variable is a region #
C       SHIFT 'EM TO 20 BIT WORDS
      DO 220 K=3,NUMVP
  220 ITMP(K)=ICONV(DATIN(L,K-2))   Convert to 20-bit words
      II=IE(I)
      IF(XMU(II).EQ.0.)ITMP(9)=ICONV(W(L))  Store W in SZ's place if μ = 0
      DO 230 K=1,NUMVP,3
      IIA=IIA+1
  230 DI(IIA)=SHIFT(ITMP(K),40).OR.SHIFT(ITMP(K+1),20).OR.ITMP(K+2)
  210 CONTINUE  Put three 20-bit words together to make one 60-bit word
```

```
      IF(IIA.GE.NVN) GO TO 400                                        OUTGAS
      IO=IIA-NWPC+1                                                   OUTGAS
      IP=IIA+1                      Fill unused cells with data       OUTGAS
      DO 300 L=IP,NVN                from the last cell used          OUTGAS
      DI(L)=DI(IO)                                                    OUTGAS
      IO=IO+1                                                         OUTGAS
  300 CONTINUE                                                        OUTGAS
  400 CONTINUE                                                        OUTGAS
C   WRITE OUT TO DISK THE DATA                                        OUTGAS
      CALL WDISK(3,DT,NVN,IFI(NA))                                    OUTGAS
      IF(UNIT(3)) 340,340,340                                         OUTGAS
  340 CONTINUE                                                        OUTGAS
      RETURN                                                          OUTGAS
      END                                                             OUTGAS
```

# OUTGAS

Makes a GAS dump to file GASSIN which includes most cell quantities. GASSIN may be postprocessed to give on the Tektronix/film/fiche any cell variable as a function of any other cell variable (e.g., pressure vs radius) at a given time, time plot a cell variable for a given cell, r-t plots of interfaces, cell positions for each cell, contour plots of a cell variable in r-t space, etc.

Local Variables

IBLANK = 10 blank Hollerith characters used in the plot label.

IDI = array containing the first 100 words in each dump.

IFI = array containing the first 1003 words of GASSIN which contains the file index.

IIA = counter to keep track of the index for DI.

IJK = only dump every IJKth cell (usually IJK = 1).

IO, IP = indices used to copy the packed data for the last cell into the remaining space for a data dump.

ISTFLAG = flag that is 1 for the 1st time through and 2 otherwise.

ITMP = array to contain the 20-bit words before they are combined 3 to a word.

J = cell # for voids.

K = do loop index.

L = do loop index.

LENGTH = length of the file GASSIN that will hold 500 dumps.

NA = index to give the location in IFI that the 1st word address for the current dump is stored.

NAF = NA+1 = index to give the location in IFI that the dump time for the current dump is stored.

NCLP = # of cells allowed per dump = initial NCL plus 55 to allow for rezoning and spalling.

NUMVP = # of variables stored per cell.

NVN = # of words per dump.

NWPC = # of words per cell.

Notes

The cell variables are stored with the following variable numbers.

1 = region index

2 = time

3 = radius

4 = velocity

5 = specific internal energy

6 = specific volume

7 = pressure

8 = stress deviator in the X direction

9 = stress deviator in the Z direction or mass fraction if $\mu = 0$

10 = energy sums (see ESUM)

11 = temperature

12 = viscosity

Space is provided for 55 extra cells to be added due to space splits and/or spalling. Data from the last word is repeated in the unused cells. This is to allow the new cells added to also be plotted. The repetition of the last cell data is necessary for two-dimensional and r-t plots of all cells to avoid extraneous lines. Details of the GAS file and how to run GAS are given in LTSS-523.

```
      FUNCTION ICONV(X)
      LEVEL 2,X
C     FORM A 20 BIT FLOATING POINT WORD FOR MAGEE MOVIE
      DATA IFA/1700000000B/,IFB/3777777B/,IFC/37777776B/
      ISIGN=0
      IF(X.LT.0.) ISIGN=1
      JS=SHIFT(ABS(X),-33)-IFA
      IF(JS.GT.IFB) JS=IFB   Maximum allowed value of JS
      IF(JS.LT.0) JS=0   Minimum allowed value of JS
      ICONV=(JS.AND.IFC).OR.ISIGN  20-bit word with sign in bit 0
      RETURN
      END
```

## ICONV(X)

Takes a 60-bit floating point word and converts it to a 20-bit floating point word.

### Local Variables

IFA = bit pattern to shift the bias of the exponent.

IFB = maximum allowed integer value of the 20-bit word = 20 bits of 1's.

IFC = the 19 bits for exponent and integer coefficient are 1's.

ISIGN = the sign of X.

JS = the 20-bit word before the sign bit is set correctly.

### Notes

The 60-bit floating point word has the following structure:

1 sign bit at bit #59.

11 exponent bits at bits 48-58, with a bias of 2000B.

48 integer coefficient bits at bits 0-47.

The desired 20-bit floating point structure is as follows:

5 exponent bits at bits 15-19, with a bias of 100B.

14 integer coefficient bits at bits 1-14.

1 sign bit at bit #0.

The 20-bit word has about 4 significant figures. Negative numbers are stored in 1's complement form for the 60-bit word but not for the 20-bit word. The sign of X is stored in ISIGN. The absolute value of X is then shifted to the right 33 bits and IFA is subtracted from this value to give JS. This puts the exponent bits at 15-25 and the integer coefficient bits at 0-14. Subtracting IFA shifts the exponent bias from 2000B to 100B. By using only 14 bits of the integer coefficient, there is also an effective shift of 34 bits or 42B. If JS > IFB, then a 5-bit exponent is not sufficient and the maximum allowed 20-bit word is used. If JS < 0, then the exponent is too small and a value of 0 is used. Then the 0 bit is set with the sign bit and the 20-bit word conversion is complete. Bits 20-59 are all 0. The largest 20-bit word is 2777776B which is $2^{16} - 2 = 65534$. The smallest positive 20-bit word is 0060000B which is $2^{-16} \doteq 1.5 * 10^{-5}$.

An an exercise, the interested reader can follow the conversion of a floating point 1.0 from the 60-bit octal work 17204 00000 00000 00000 B to the 20-bit octal word 2040000B.

```
      SUBROUTINE DIFEQ                                                  DIFEQ      2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTA8=1,NTA8=MTA8*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL      3
     +,W(MCL)                                                           MCELL      4
      LEVEL 2,R                                                         MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NAND,NM, MCELL      6
     +IALPH,NDELT,LABFL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL      8
      LEVEL 2,TIME                                                      MCELL      9
      GO TO (1,2),NDF  Index to determine type of differencing scheme   DIFEQ      5
    1 CALL HYDRO   HYDROX                                               DIFEQ      6
      RETURN                                                            DIFEQ      7
    2 CALL SINX   SIN                                                   DIFEQ      8
      RETURN                                                            DIFEQ      9
      END                                                               DIFEQ     10
```

## DIFEQ

Switching routine to determine the type of difference equation scheme to be used in the main hydro cycle. Default is HYDRO.

Notes

| NDF | Type of Difference Equations |
|---|---|
| 1 | Hydrox (see HYDRO) |
| 2 | SIN (see SINX) |

Any other type of differencing scheme may be added by extending the computed go to statement list and adding the subroutine call and the subroutine.

```
      SUBROUTINE HYDRO
C
C     PERFORMS THE BASIC HYDRONAMICS CYCLE
C     CHANGES IN THE DIFFERENCE EQUATIONS MAY BE MADE
C     EXCLUSIVELY IN THIS SUBROUTINE
C
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)
     +,W(MCL)
      LEVEL 2,R
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS
      LEVEL 2,TIME
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)
      COMMON/UCJC/UCJ,JJ,NMAX,RCJ,DCJ
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC
      COMMON/EOSN/IEOS(ML),ME(ML)
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV
      COMMON/ES/IE(ML2),NME
R^(α-1)dσ/dM DSDM(J,JM)=2*R(J)**IA*(P(JM)+Q(JM)-P(J)-Q(J))/(XM(J)+XM(JM))
(α-1)Vφ/R VFR(J,JM)=IA*(V(J)+V(JM))*(2*(SX(J)+SX(JM))+SZ(J)+SZ(JM))/(4*R(J))
     φ PH(J,JM)=2*SX(J)+SZ(J)+XM(J)*(2*(SX(J)-SX(JM))-SZ(J)+SZ(JM))/
     1 (XM(J)+XM(JM))
      VV(J,JM)=V(J)+XM(J)*(V(J)-V(JM))/(XM(J)+XM(JM))  V
      DO 10 I=1,NMC
      JMN=JMIN(I)
      JMX=JMAX(I)
      DO 10 J=JMN,JMX
      XI(J)=XI(J)-V(J)*DT*0.5*((P(J)+Q(J))*(U(J+1)-U(J))/
     1  (R(J+1)-R(J))+IA*(P(J)+Q(J)-2*SX(J)-SZ(J))*(U(J+1)+U(J))/
     2  (R(J+1)+R(J)))  Part of ΔI that uses old R,V,U
   10 CONTINUE
      DO 20 I=1,NMC
      JMN=JMIN(I)
      JMX=JMAX(I)
      IF(I.NE.1)JMN=JMN+1
      IF(JMN.GT.JMX)GO TO 20
      DO 21 J=JMN,JMX
      JM=J-1
      U(J)=U(J)-DT*DSDM(J,JM)                    ] ∂u/∂t = -R^(α-1) ∂σ/∂M - (α - 1) Vφ/R
      IF(INTX(I).GT.2) U(J)=U(J)-DT*VFR(J,JM)    ]
      R(J)=R(J)+DT*U(J)  ∂R/∂t = u
   21 CONTINUE
      IF(JMAX(I).NE.KMAX(I))GO TO 20
      IF(I.EQ.NMC.AND.NMC.NE.NM)GO TO 20
      IF(IV(I))31,32,33  Treat interface as a special case
   31 J=JMX+1      No artificial void cell
      GO TO 22
   32 J=JMX+1
      JP=J+1                                            ]
      U(J)=U(J)-DT*2*R(J)**IA*(P(JMX)+Q(JMX))/XM(JMX)   ]  Open void:
      R(J)=R(J)+DT*U(J)                                 ]  free surfaces
      U(JP)=U(JP)+DT*2*R(JP)**IA*(P(JP)+Q(JP))/XM(JP)   ]
```

```
      R(JP)=R(JP)+DT*U(JP)
      GO TO 20
33    J=JMX+2   Closed void
22    CONTINUE
      GO TO (1,2,3,4,5),INTX(I)  Boundary conditions depend on μ's
5     U(J)=U(J)-DT*VFR(J,JMX)
1     U(J)=U(J)-DT*DSDM(J,JMX)
      R(J)=R(J)+DT*U(J)
      IF(IV(I).EQ.-1)GO TO 20
      U(J-1)=U(J)
      R(J-1)=R(J)
      GO TO 20
3     D=IA*(PH(JMX,JMX-1)*VV(JMX,JMX-1)*XM(JMX)/(XM(J)
     1 +XM(JMX)))/R(J)
      GO TO 6
2     D=IA*PH(J,J+1)*VV(J,J+1)*XM(J)/((XM(J)+XM(JMX))*R(J))
      GO TO 6
4     FM=XM(J)/(XM(J)+XM(JMX))
      D=IA*(PH(J,J+1)*VV(J,J+1)*FM+PH(JMX,JMX-1)*VV(JMX,JMX-1)*(1-FM))
     1 /R(J)
6     U(J)=U(J)-DT*(D+DSDM(J,JMX))
      R(J)=R(J)+U(J)*DT
      IF(IV(I).EQ.-1)GO TO 20
      U(J-1)=U(J)
      R(J-1)=R(J)
20    CONTINUE
      CALL BNDR1   Check other boundary conditions
      MR=1
      MRP=1
      DO 30 I=1,NMC
      JMN=JMIN(I)
      JMX=JMAX(I)
      II=IE(I)
      DO 30 J=JMN,JMX
      MRM=MR
      MR=MRP
      MRP=IFLAG(J+1)/64    New volume
      V(J)=F2*(R(J)-R(J+1))*(R(J)**IA+R(J+1)**IA+F3*R(J)*R(J+1))/XM(J)
      XI(J)=XI(J)-V(J)*DT*0.5*((P(J)+Q(J))*(U(J+1)-U(J))/
     1  (R(J+1)-R(J))+IA*(P(J)+Q(J)-2*SX(J)-SZ(J))*(U(J+1)+U(J))/
     2  (R(J+1)+R(J)))       Part of ΔI that uses new R,V,U
      IF(XMU(II).EQ.0.)GO TO 12
      SX(J)=SX(J)-4*XMU(II)*DT*((U(J+1)-U(J))/(R(J+1)-R(J))
     1  -IA*0.5*(U(J+1)+U(J))/(R(J+1)+R(J)))/3
      IF(IALPH.EQ.2)GO TO 13
      SZ(J)=-SX(J)/2             New Stress deviators
      GO TO 12
13    SZ(J)=SZ(J)+2*XMU(II)*DT*((U(J+1)-U(J))/(R(J+1)-R(J))
     1  +(U(J+1)+U(J))/(R(J+1)+R(J)))/3
12    CONTINUE
C     XI(J)=XI(J)+DT*((XL(MR)+XL(MRP)))*(T(J+1)-T(J))*R(J+1)**IA/(R(J+2)
C    1  -R(J))-(XL(MR)+XL(MRM))*(T(J)-T(J-1))*R(J)**IA/(R(J+1)-R(J-2))
C    2  )/XM(J)
30    CONTINUE
      CALL VISC   New Q's
      DO 60 I=1,NMC
      II=IE(I)
60    IF(IBRN(II).GT.0)CALL BURN(I)  New W's
      CALL EOS  New P's and T's
      DO 80 I=1,NM
```

```
      II=IE(I)                                                          HYDRO
      JMN=JMIN(I)                                                       HYDRO
      JMX=JMAX(I)                                                       HYDRO
      IF(IV(I).EQ.2)CALL RLEOS(I)  Special for high-velocity impact     HYDRO
   80 CONTINUE                                                          HYDRO
      CALL BNDR2  More boundary conditions (an entry point in BNDR1)    HYDRO
      RETURN                                                            HYDRO
      END                                                               HYDRO
```

# HYDRO

The main hydro cycle using the HYDROX difference equations. New values of radii, velocities, specific volumes, specific internal energies, and stress deviators are calculated. Subroutines are called to get new pressures, temperatures, and artificial viscosities.

## Statement Functions

DSDM(J,JM) = difference form of $R^{\alpha-1} \frac{\partial\sigma}{\partial M}$.

VFR(J,JM) = difference form of $\frac{(\alpha - 1)V\phi}{R}$, where $\phi = 2S_x + S_z$.

PH(J,JM) = interpolation formula to evaluate $\phi$ at a region boundary where $\phi$ is discontinuous.

VV(J,JM) = interpolation formula to evaluate V at a region boundary where V is discontinuous.

## Local Variables

I = do loop index for region #.

$$\left.\begin{array}{l}\text{JMN = JMIN for region I}\\ \text{JMX = JMAX for region I}\end{array}\right\}\text{ minimum and maximum cell \#'s in a region.}$$

J = do loop index for cell #.

JM = J - 1.

JP = J + 1.

D = the last term inside brackets of Eq. 8.

FM = $M_j/(M_j + M_{j+1})$.

MRM = region # for cell J - 1.

MR = region # for cell J.

MRP = region # for cell J + 1.

II = IE(I) = original region # for region I.

Care must be taken that interfaces between materials are treated properly. The differential equation for acceleration is given by

$$\frac{\partial u}{\partial t} = -R^{\alpha-1}\frac{\partial \sigma}{\partial M} - (\alpha - 1)\frac{\phi}{R\rho} \quad . \tag{1}$$

Now, consider an interface between material 1 and material 2 at radius $R_0$. The difference in the limit of Eq. (1) as R tends to $R_0$ from region 1 and region 2 is given by

$$-\frac{\partial u_1}{\partial t} + \frac{\partial u_2}{\partial t} = R^{\alpha-1}\left(\frac{\partial \sigma_1}{\partial M_1} - \frac{\partial \sigma_2}{\partial M_2}\right) + (\alpha - 1)\left(\frac{\phi_1}{R\rho_1} - \frac{\phi_2}{R\rho_2}\right) = 0 \quad , \tag{2}$$

where the subscripts denote the region from which the limit is taken. The difference in acceleration is zero provided the two regions are in contact.

For the case $\phi_1 = \phi_2 = 0$, Eq. (2) implies $\partial\sigma_1/\partial M_1 = \partial\sigma_2/\partial M_2$ at $R_0$. The usual difference equation schemes don't encounter any problems at the interface because $\partial\sigma/\partial M$ is continuous.

For the case $\phi_1 \neq \phi_2$, set $\Delta = (\phi_2/R^\alpha\rho_2 - \phi_1/R^\alpha\rho_1)(\alpha - 1)$. We then have

$$\frac{\partial \sigma_1}{\partial M_1} - \frac{\partial \sigma_2}{\partial M_2} = \Delta \quad . \tag{3}$$

That is, $\partial\sigma/\partial M$ is not continuous across the interface. Therefore, care must be taken so that the interface will be treated properly by the difference equations.

Let j be the cell number of the cell touching the interface in region 1 and j+1 for region 2. Equation (3) can be written in difference form as

$$\frac{\sigma_j - \sigma_{j+\frac{1}{2}}}{\frac{1}{2}M_j} - \frac{\sigma_{j+\frac{1}{2}} - \sigma_{j+1}}{\frac{1}{2}M_{j+1}} = \Delta \quad , \tag{4}$$

where $\sigma_{j+\frac{1}{2}}$ is unknown. Solving for $\sigma_{j+\frac{1}{2}}$, we have

$$\sigma_{j+\frac{1}{2}} = (M_j + M_{j+1})^{-1}\left(M_{j+1}\sigma_j + M_j\sigma_{j+1} - \left(\frac{\Delta}{2}\right)M_jM_{j+1}\right) \quad . \tag{5}$$

The difference form for $\partial\sigma_1/\partial M_1$ then becomes

$$\frac{\partial\sigma_1}{\partial M_1} = \frac{2}{M_j + M_{j+1}} (\sigma_j - \sigma_{j+1} + \frac{\Delta}{2} M_{j+1}) \quad . \tag{6}$$

The difference form of the acceleration in region 1 at the boundary is given by

$$\frac{\partial u_1}{\partial t} = - \left[\frac{2R^{\alpha-1}}{M_j + M_{j+1}} (\sigma_j - \sigma_{j+1} + \frac{\Delta}{2} M_{j+1}) + (\alpha - 1) \frac{\phi_1 V_1}{R}\right] \quad . \tag{7}$$

The velocity equation can then be written in the form

$$u_{j+\frac{1}{2}}^{n+\frac{1}{2}} = u_{j+\frac{1}{2}}^{n-\frac{1}{2}} - \Delta t\left\{(R_{j+\frac{1}{2}})^{\alpha-1}\ 2(\sigma_j - \sigma_{j+1})/(M_j + M_{j+1}) + \frac{(\alpha - 1)}{R_{j+\frac{1}{2}}}\Phi\right\} \quad , \tag{8}$$

where

$$\Phi = \phi_1 V_1\left(\frac{M_j}{M_j + M_{j+1}}\right) + \phi_2 V_2\left(\frac{M_{j+1}}{M_j + M_{j+1}}\right) \quad , \tag{9}$$

with the subscript 1 and 2 denoting the region from which the limit is taken approaching the interface. These limits are taken by extrapolation.

```
      SUBROUTINE SINX                                                  SINX      2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,        PARAM     2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                 PARAM     3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                   PARAM     4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                       PARAM     5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                        MCELL     2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)  MCELL     3
     +,W(MCL)                                                          MCELL     4
      LEVEL 2,R                                                        MCELL     5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL     6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                   MCELL     7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                 MCELL     8
      LEVEL 2,TIME                                                     MCELL     9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT      2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT    3
     +MAT(ML),UD(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)           INIT      4
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),        US        2
     +GAMMA(ML),ALP(ML)                                                US        3
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                              MN        2
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                        VD        2
      COMMON/ES/IE(ML2),NME                                            ESM       2
      COMMON/UCJC/UCJ,JJ,NMAX,RCJ,DCJ                                  UC        2
      DIMENSION UU(MCL),VO(MCL)                                        SINX     11
      DSDM(J,JM)=2*R(J)**IA*(P(JM)+Q(JM)-P(J)-Q(J))/(XM(J)+XM(JM))     SINX     12
      VFR(J,JM)=IA*(V(J)+V(JM))*(2*(SX(J)+SX(JM))+SZ(J)+SZ(JM))/(4*R(J)) SINX   13
      PH(J,JM)=2*SX(J)+SZ(J)+XM(J)*(2*(SX(J)-SX(JM))-SZ(J)+SZ(JM))/    SINX     14
     +(XM(J)+XM(JM))                                                   SINX     15
      VV(J,JM)=V(J)+XM(J)*(V(J)-V(JM))/(XM(J)+XM(JM))                  SINX     16
      NCLP=NCL+1                                                       SINX     17
      DO 10 J=1,NCLP                                                   SINX     18
      UU(J)=U(J)      Save old u's                                     SINX     19
   10 CONTINUE                                                         SINX     20
      DO 20 I=1,NMC                                                    SINX     21
      JMN=JMIN(I)                                                      SINX     22
      JMX=JMAX(I)                                                      SINX     23
      IF(I.NE.1)JMN=JMN+1                                              SINX     24
      DO 21 J=JMN,JMX                                                  SINX     25
      JM=J-1                                                           SINX     26
      U(J)=U(J)-DT*DSDM(J,JM)                                          SINX     27
      IF(INTX(I).GT.2) U(J)=U(J)-DT*VFR(J,JM)                          SINX     28
      R(J)=R(J)+DT*U(J)                                                SINX     29
   21 CONTINUE                                                         SINX     30
      IF(JMAX(I).NE.KMAX(I))GO TO 20                                   SINX     31
      IF(I.EQ.NMC.AND.NMC.NE.NM)GO TO 20                               SINX     32
      IF(IV(I))31,32,33                                                SINX     33
   31 J=JMX+1                                                          SINX     34
      GO TO 22                                                         SINX     35
   32 J=JMX+1                                                          SINX     36
      JP=J+1                                                           SINX     37
      U(J)=U(J)-DT*2*R(J)**IA*(P(JMX)+Q(JMX))/XM(JMX)                  SINX     38
      R(J)=R(J)+DT*U(J)                                                SINX     39
      U(JP)=U(JP)+DT*2*R(JP)**IA*(P(JP)+Q(JP))/XM(JP)                  SINX     40
      R(JP)=R(JP)+DT*U(JP)                                             SINX     41
      GO TO 20                                                         SINX     42
   33 J=JMX+2                                                          SINX     43
   22 CONTINUE                                                         SINX     44
      GO TO (1,2,3,4,5),INTX(I)                                        SINX     45
    5 U(J)=U(J)-DT*VFR(J,JMX)                                          SINX     46
    1 U(J)=U(J)-DT*DSDM(J,JMX)                                         SINX     47
      R(J)=R(J)+DT*U(J)                                                SINX     48
```

```
      IF(IV(I).EQ.-1)GO TO 20                                              SINX
      U(J-1)=U(J)                                                          SINX
      R(J-1)=R(J)                                                          SINX
      GO TO 20                                                             SINX
3     D=IA*(PH(JMX,JMX-1)*VV(JMX,JMX-1)*XM(JMX)/(XM(J)                     SINX
     1 +XM(JMX)))/R(J)                                                     SINX
      GO TO 6                                                              SINX
2     D=IA*PH(J,J+1)*VV(J,J+1)*XM(J)/((XM(J)+XM(JMX))*R(J))                SINX
      GO TO 6                                                              SINX
4     FM=XM(J)/(XM(J)+XM(JMX))                                             SINX
      D=IA*(PH(J,J+1)*VV(J,J+1)*FM+PH(JMX,JMX-1)*VV(JMX,JMX-1)*(1-FM))     SINX
     1 /R(J)                                                               SINX
6     U(J)=U(J)-DT*(D+DSDM(J,JMX))                                         SINX
      R(J)=R(J)+U(J)*DT                                                    SINX
      IF(IV(I).EQ.-1)GO TO 20                                              SINX
      U(J-1)=U(J)                                                          SINX
      R(J-1)=R(J)                                                          SINX
20    CONTINUE                                                             SINX
      CALL BNDR1                                                           SINX
      DO 13 I=1,NMC                                                        SINX
      JMN=JMIN(I)                                                          SINX
      JMX=JMAX(I)                                                          SINX
      II=IE(I)                                                             SINX
      IF(IV(I).NE.0)GO TO 101                                              SINX
      JI=JV(I)                                                             SINX
      P(JI)=0.                                                             SINX
      Q(JI)=0.                                                             SINX
  101 CONTINUE                                                             SINX
      DO 12 J=JMN,JMX                                                      SINX
      VO(J)=V(J)        Save old V's                                       SINX
      V(J)=((R(J)+R(J+1))/2)**IA*(R(J)-R(J+1))/XM(J)                       SINX
      JM=J-1                                                               SINX
      IF(J.EQ.JMN.AND.IV(I-1).GE.1.AND.I.NE.1)JM=J-2                       SINX
      JP=J+1                                                               SINX
      IF(J.EQ.KMAX(I).AND.IV(I).GE.1)JP=J+2                                SINX
      IF(J.EQ.2.AND.BU.NE.0.)JM=2                                          SINX
      IF(JJ.EQ.J+1)JP=J                                                    SINX
      XI(J)=XI(J)-DT*(((XM(J)*(P(JM)+Q(JM))+XM(JM)*(P(J)+Q(J)))            SINX
     1/(XM(J)+XM(JM)))*U(J)*R(J)**IA-((XM(JP)*(P(J)+Q(J))                  SINX
     2+XM(J)*(P(JP)+Q(JP)))/(XM(J)+XM(JP)))*U(J+1)*R(J+1)**IA)             SINX
     3/XM(J)   Δ energy - Δ kinetic energy                                 SINX
     4+((UU(J+1)+UU(J))**2-(U(J)+U(J+1))**2)/8                             SINX
      IF(XMU(II).EQ.0.) GO TO 12                                           SINX
      SX(J)=SX(J)+2*XMU(II)*(DT*(U(J)-U(J+1))/(R(J+1)-R(J))+2*(V(J)-       SINX
     +VO(J))/(3*(VO(J)+V(J))))                                             SINX
      SZ(J)=-SX(J)/2                                                       SINX
12    CONTINUE                                                             SINX
   13 CONTINUE                                                             SINX
      CALL VISC                                                            SINX
      DO 60 I=1,NMC                                                        SINX
      II=IE(I)                                                             SINX
   60 IF(IBRN(II).GT.0)CALL BURN(I)                                        SINX
      CALL EOS                                                             SINX
      DO 80 I=1,NM                                                         SINX
      II=IE(I)                                                             SINX
      JMN=JMIN(I)                                                          SINX
      JMX=JMAX(I)                                                          SINX
      IF(IV(I).EQ.2)CALL RLEOS(I)                                          SINX
   80 CONTINUE                                                             SINX
      CALL BNDR2                                                           SINX
      RETURN                                                               SINX
      END                                                                  SINX
```

SINX

The main hydro cycle using the SIN difference equations. New values of radii, velocities, specific volumes, specific internal energies, and stress deviators are calculated. Subroutines are called to get new pressures, temperatures, and artificial viscosities.

Local Variables (Those variables that are the same as in HYDRO are not repeated here.)

JI = JV(I) = cell # of a void.

NCLP = NCL + 1.

UU = velocity from previous cycle.

VO = volume from previous cycle.

Notes

The difference equation for acceleration is identical with HYDROX. The internal energy equation, however, is based on the total energy equation

$$\frac{\partial E}{\partial t} = - \frac{(\sigma u R^{\alpha-1})}{\partial M} ,$$

which is essentially the rate of PdV work being done on the cell. The rate of change of kinetic energy is then subtracted to get the rate of change of internal energy.

```
      SUBROUTINE EOS                                                    EOS        2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL      3
     +,W(MCL)                                                           MCELL      4
      LEVEL 2,R                                                         MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL      8
      LEVEL 2,TIME                                                      MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT      3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT       4
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                               MN         2
      COMMON/EOSN/IEOS(ML),ME(ML)                                       EN         2
      COMMON/ES/TE(ML2),NME                                             ESM        2
      COMMON/EOSCOM/SR(ML),ES(ML),A1(ML),A2(ML),A3(ML),EM(ML),IRV(ML)   EOSCOM     2
      COMMON/SESIN/II,IDT,RPT4,XIPT4,IBR,IFL                            EOS       11
      COMMON/SESOUT/PPT4(3),TPT4(3)                                     EOS       12
      IBR=0                                                             EOS       14
      IDT=1                                                             EOS       15
      I=1                                                               EOS       16
   20 CONTINUE                                                          EOS       17
      II=IE(I)     Original region #                                    EOS       18
      IS=IEOS(II)  EOS type                                             EOS       19
      JMN=JMIN(I)                                                       EOS       20
      JMX=JMAX(I)                                                       EOS       21
      GO TO (1,2,3,4),IS                                                EOS       22
    1 CONTINUE                                                          EOS       23
      JS=0                                                              EOS       24
      DO 11 J=JMN,JMX                                                   EOS       25
      CALL HOM(II,J)     HOM EOS                                        EOS       26
      IF(IBRN(II).NE.0)GO TO 11                                         EOS       27
      IF(P(J).LT.-0.005)CALL SPEOS(I,J)   Spall?                        EOS       28
      IF(XMU(II).GT.0.)CALL EPP(II,J)   Elastic - perfectly plastic     EOS       29
   11 CONTINUE                                                          EOS       30
      GO TO 10                                                          EOS       31
    2 CONTINUE                                                          EOS       32
      DO 12 J=JMN,JMX                                                   EOS       33
   12 CALL BLDUP(II,J)   Buildup EOS                                    EOS       34
      GO TO 10                                                          EOS       35
    3 CONTINUE                                                          EOS       36
      JS=0                                                              EOS       37
      DO 13 J=JMN,JMX                                                   EOS       38
      CALL POLY(II,J)   Eight-parameter polynomial EOS                  EOS       39
      IF(P(J).LT.-0.005)CALL SPEOS(I,J)   Spall?                        EOS       40
      IF(XMU(II).GT.0.)CALL EPP(II,J)   Elastic - perfectly plastic     EOS       41
   13 CONTINUE                                                          EOS       42
      GO TO 10                                                          EOS       43
    4 DO 14 J=JMN,JMX                                                   EOS       45
      XIPT4=XI(J)          ]                                            EOS       46
      RPT4=1./V(J)         ]   Input for SESAME                         EOS       47
      IFL=MOD(IFLAG(J),64) ]                                            EOS       48
      CALL T4EOSA     SESAME EOS                                        EOS       49
      IF(MOD(IFLAG(J),64).EQ.1) GO TO 140                               EOS       50
      IF(IRV(I).EQ.0.AND.IFL.EQ.1) IFLAG(J)=IFLAG(J)+1                  EOS       51
  140 P(J)=PPT4(1)    Output pressure                                   EOS       52
```

```
      T(J)=TOT4(1)    Output temperature                                    EOS        53
      IF(XMU(II).GT.0.)CALL EPP(II,J)   Elastic - perfectly plastic         EOS        54
      IF(P(J).LT.-0.005)CALL SPEOS(I,J)  Spall?                             EOS        55
   14 CONTINUE                                                              EOS        56
   10 CONTINUE                                                              EOS        61
      IF(JS.NE.0)CALL SL(I)   Spall if indicated                            EOS        62
      I=I+1                                                                 EOS        63
      IF(I.LE.NMC) GO TO 20   Spalling changes the # of regions, so a do loop  EOS     64
      RETURN                                                 is not used    EOS        65
      END                                                                   EOS        66
```

# EOS

Switching routine to call the appropriate equation of state. The spalling and elastic-plastic treatments are also called if turned on.

Local Variables

I = region #.

II = original region #.

IS = EOS #.

JMN,JMX = minimum and maximum active cell #.

```
      SUBROUTINE PTEOS(I,PI,TII,VI,XII)                                 PTEOS       2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM       2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM       3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM       4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM       5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL       2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL       3
     +,W(MCL)                                                           MCELL       4
      LEVEL 2,R                                                         MCELL       5
      COMMON/OVL/NDF,NT,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL       6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL       7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL       8
      LEVEL 2,TIME                                                      MCELL       9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO  INIT        2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT        3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT        4
      COMMON/EOSN/IEOS(ML),ME(ML)                                       EN          2
      COMMON/ES/TE(ML2),NME                                             ESM         2
      COMMON/SESIN/TI,IDT,RPT4,XIPT4,IBR,IFL                            PTEOS       9
      COMMON/SESOUT/PPT4(3),TPT4(3)                                     PTEOS      10
      IDT=1                                                             PTEOS      11
      IBR=1                                                             PTEOS      12
      II=IE(I)                                                          PTEOS      14
      J=JMIN(I)                                                         PTEOS      15
      PT=P(J)   ]                                                       PTEOS      16
      TT=T(J)   ]                                                       PTEOS      17
      VT=V(J)   ]  Set cell quantities in temporary storage             PTEOS      18
      XIT=XI(J) ]                                                       PTEOS      19
      V(J)=VI   ]  Replace by input values                              PTEOS      20
      XI(J)=XII ]                                                       PTEOS      21
      IS=IEOS(II)                                                       PTEOS      22
      GO TO (1,2,3,4),IS    Call EOS                                    PTEOS      23
    1 CALL HOM(II,J)                                                    PTEOS      24
      GO TO 10                                                          PTEOS      25
    2 CALL BLDUP(II,J)                                                  PTEOS      26
      GO TO 10                                                          PTEOS      27
    3 CALL POLY(II,J)                                                   PTEOS      28
      GO TO 10                                                          PTEOS      29
    4 DO 14 J=JMN,JMX                                                   PTEOS      31
      XIPT4=XI(J)                                                       PTEOS      32
      PPT4=1./V(J)                                                      PTEOS      33
      IFL=MOD(IFLAG(J),64)                                              PTEOS      34
      CALL T4EOS#                                                       PTEOS      35
      P(J)=PPT4(1)                                                      PTEOS      36
   14 CONTINUE                                                          PTEOS      37
   10 CONTINUE                                                          PTEOS      42
      PI=P(J)   ]  Output P,T                                           PTEOS      43
      TII=T(J)  ]                                                       PTEOS      44
      P(J)=PT                                                           PTEOS      45
      T(J)=TT   ]                                                       PTEOS      46
      V(J)=VT   ]  Put back original cell quantities                    PTEOS      47
      XI(J)=XIT ]                                                       PTEOS      48
      RETURN    ]                                                       PTEOS      49
      END                                                               PTEOS      50
```

## PTEOS

Controls calls to EOS subroutines with energy and volume as input rather than region # and cell #.

### Local Variables

IS = EOS type.

J = cell # used for EOS calls.

| | | |
|---|---|---|
| PT | temporary storage for | P(J) |
| TT | | T(J) |
| VT | | V(J) |
| XIT | | XI(J). |

### Notes

EOS calls are made by specifying region # and cell # in this code. In the special treatment of high-velocity void collapse (see RL), it is necessary to call the EOS with arbitrary values of V and I. This is done by saving the cell quantities for a given cell, replacing them with arbitrary values, calling the EOS, returning P,T, and then putting the original cell quantities back in their proper place.

```
      SUBROUTINE HOM(I,J)                                                HOM       2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,          PARAM     2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                   PARAM     3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                     PARAM     4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                         PARAM     5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                          MCELL     2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)    MCELL     3
     +,W(MCL)                                                            MCELL     4
      LEVEL 2,R                                                          MCELL     5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,  MCELL     6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                     MCELL     7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                   MCELL     8
      LEVEL 2,TIME                                                       MCELL     9
      COMMON/INIT/RTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VD(ML),PD   INIT      2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT      3
     +MAT(ML),UD(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)             INIT      4
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                                MN        2
      COMMON/EOSN/IEOS(ML),ME(ML)                                        EN        2
      DATA GASW/0.02/                                                    HOM       8
      DATA SOLW/0.999/                                                   HOM       9
C FIXUP FOR FOREST FIRE                                                  HOM      10
      IF(IBRN(I).EQ.0) GO TO 11   All solid                              HOM      11
      IF(W(J).LT.GASW)GO TO 12   Treat as all gas for W < 0.02           HOM      12
      IF(IBRN(I).EQ.2)GO TO 13   Special for CJ burn                     HOM      13
      IF(W(J).GT.SOLW) GO TO 11   Treat as all solid for W > 0.999       HOM      14
      CALL MIX(I,J)   Mixture of gas and solid                           HOM      15
      GO TO 10                                                           HOM      16
11    CONTINUE                                                           HOM      17
      CALL USUP(I,J)   All solid                                         HOM      18
      GO TO 10                                                           HOM      19
12    CALL GAS(I,J)   All gas                                            HOM      20
      GO TO 10                                                           HOM      21
   13 CONTINUE                                                           HOM      22
      IF(W(J).GT.SOLW)GO TO 14                                           HOM      23
      CALL GAS(I,J)                                                      HOM      24
      P(J)=(1.-W(J))*P(J)   No solid EOS for CJ burn                     HOM      25
      GO TO 10                                                           HOM      26
   14 P(J)=PO(I)                                                         HOM      27
10    CONTINUE                                                           HOM      28
      RETURN                                                             HOM      29
      END                                                                HOM      30
```

# HOM

Switching routine for deciding which type of EOS is used for a cell for the HOM EOS (e.g., determines whether a material is a solid, gas, or mixture).

## Local Variables

GASW = mass fraction below which a material is treated as all gas (W = 0).

SOLW = mass fraction above which a material is treated as all solid (W = 1).

## Notes

If a CJ burn is used for an HE, the MIX EOS is not used for partially decomposed HE. Instead, the GAS EOS is used with the pressure weighted by 1-W. For other burn methods (except for sharp shock which never allows partially burned HE), partially decomposed HE (GASW<W<SOLW) is treated in the MIX EOS.

```
      SUBROUTINE USUP(I,J)                                                USUP
C     EQUATION OF STATE FOR A SOLID USING USUP FIT                        USUP
C     FOR TWO PHASE FE TYPE EQUATION OF STATE                             USUP
C     I.E., C AND S IN THE RELATION US=C+S*UP CHANGE                      USUP
C     AT SPECIFIC VOLUME SWV                                              USUP
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,           PARAM
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                    PARAM
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                      PARAM
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                          PARAM
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                           MCELL
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)     MCELL
     +,W(MCL)                                                             MCELL
      LEVEL 2,R                                                           MCELL
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NAND,NM,   MCELL
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                      MCELL
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                    MCELL
      LEVEL 2,TIME                                                        MCELL
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO   INIT
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),  INIT
     +MAT(ML),UO(ML),UT(ML),CTCF(ML),QD(ML),TMLT(ML),TMC(ML)              INIT
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),           US
     +GAMMA(ML),ALP(ML)                                                   US
      COMMON/FGHIJC/FS(ML),GS(ML),HS(ML),SI(ML),SJ(ML),CV(ML)             FG
      COMMON/BRNS/A(ML),BR(ML),BA(ML),VBO(ML),VBSW(ML)                    BRN
      DATA CF/1.39528394E-5/                                              USUP
      IF(V(J).LT.VBSW(I)) GO TO 30   Barnes EOS for V < VBSW              USUP
      IF(V(J).GT.VO(I))GO TO 40  Grüneisen EOS (with P = 0 reference) for USUP
      IF(V(J).GT.SWV(I))GO TO 11                                  tension USUP
      IF(V(J).LT.VMN(I))GO TO 12                                          USUP
      V(J)=VMN(I)                   2nd USUP fit used for phase change    USUP
12    C=C2(I)                                                             USUP
      S=S2(I)                                                             USUP
      GO TO 20                                                            USUP
11    C=C1(I)                                                             USUP
      S=S1(I)                                                             USUP
20    VOMV=VO(I)-V(J)                                                     USUP
      HP=((C/(VO(I)-S*VOMV))**2)*VOMV+PO(I)                               USUP
      HE=(HP+PO(I))*VOMV*0.5              P,I on the Hugoniot             USUP
      P(J)=HP+(XI(J)-HE)*GAMMA(I)/V(J)  Shift off the Hugoniot with constant γ USUP
C     IF NO HEAT CAPACITY SKIP TEMP CALCULATION                           USUP
      IF(CV(I))22,22,21                                                   USUP
21    ALNV=ALOG(V(J))                                                     USUP
      T(J)=(XI(J)-HE)*23890./CV(I)+EXP(FS(I)+ALNV*(GS(I)+ALNV*(HS(I)      USUP
     1+ALNV*(SI(I)+ALNV*SJ(I)))))    Temperature fit                      USUP
   22 RETURN                                                              USUP
   30 CONTINUE                                                            USUP
      CALL BEQST(I,J)   Barnes EOS                                        USUP
      RETURN                                                              USUP
   40 CONTINUE                                                            USUP
      IF(ALP(I))51,51,52                                                  USUP
51    P(J)=PO(I)                                                          USUP
      T(J)=TO(I)    Default values for tension if α < 0                   USUP
      RETURN                                                              USUP
   52 P(J)=(GAMMA(I)*(XI(J)+(1.-V(J)/VO(I))*CV(I)*CF/ALP(I)))/V(J)        USUP
      T(J)=XI(J)*23890./CV(I)+TO(I)  EOS for tension                      USUP
      RETURN                                                              USUP
      END                                                                 USUP
```

## USUP

USUP EOS allows for two USUP fits with a phase change. At high density the Barnes EOS is used. In tension, the Grüneisen EOS with the P=0 line as the standard curve is used.

### Local Variables

C,S = constants used in USUP fit. $U_S = C + SU_P$ where $U_S$ is the shock velocity and $U_P$ is the particle velocity.

VOMV = $V_0 - V$, where $V_0$ is the initial specific volume ($cm^3/g$).

HP = pressure on the Hugoniot for volume V.

HE = energy on the Hugoniot for volume V.

ALNV = $\ell n(V)$

CF = conversion factor in the Grüneisen EOS.

### Notes

USUP EOS: For many materials, a plot of $U_S$ versus $U_P$ data is a straight line to a good approximation over the range of interest. The data is then fit to the equation

$$U_S = C + SU_P \quad . \tag{1}$$

This equation combined with the Rankine-Hugoniot equations gives sufficient information to determine the Hugoniot pressure, $P_H$, and Hugoniot specific internal energy, $I_H$, on the Hugoniot as a function of the density, $\rho$, initial density, $\rho_0$, initial pressure, $P_0$, and the initial specific internal energy, $I_0$. The Rankine-Hugoniot equations (also called jump conditions) are given from conservation of mass, momentum, and energy, and from the assumption that the shape of the shock front is constant in time. (For a derivation of these equations see, for example, Courant and Freidrichs, Sec. 54.) The jump conditions are

$$\rho_0 U_S = \rho(U_S - U_P) \quad , \tag{2}$$

$$P_H - P_0 = \rho_0 U_S U_P \quad , \tag{3}$$

$$\left[(I_H - I_0) - \frac{U_P^2}{2}\right]\rho_0 U_S = P_0 U_P \quad . \tag{4}$$

After some algebraic manipulation, the four equations yield the results:

$$P_H = \left(\frac{C}{V_0 - S(V_0 - V)}\right)^2 (V_0 - V) + P_0 \quad , \tag{5}$$

and

$$I_H = \frac{1}{2}(V_0 - V)(P_H + P_0) \quad , \tag{6}$$

where

$$V_0 = \frac{1}{\rho_0} \text{ and } V = \frac{1}{\rho} \quad . \tag{7}$$

Of course, P and I are required off the Hugoniot too. The Grüneisen gamma is defined as $\gamma = V(\partial P/\partial I)_V$. If $\gamma$ is known, a Taylor expansion around the Hugoniot values for a given specific volume yields (to first order)

$$P = P_H + (I - I_H)\left(\frac{\partial P}{\partial I}\right)_V = P_H + \frac{(I - I_H)\gamma}{V} \quad . \tag{8}$$

For this subroutine, $\gamma$ is assumed to be constant.

The temperature on the Hugoniot, $T_H$, is calculated by the method of Walsh and Christian. For the USUP fit, an analytic solution for the Hugoniot temperature can be found, but it involves an exponential integral. So, $\ell n(T_H)$ is fit to a polynomial in $\ell n(V)$, that is,

$$\ell n\, T_H = F + G(\ell n\, V) + H(\ell n\, V)^2 + I(\ell n\, V)^3 + J(\ell n\, V)^4 \quad , \tag{9}$$

with the assumption that $C_V = \left(\frac{\partial I}{\partial T}\right)_V$ is constant, we obtain for temperatures off the Hugoniot

$$T = T_H + \frac{I - I_H}{C_V} \quad . \tag{10}$$

This temperature calculation uses approximately 10% of the CPU time per cell per cycle. So, the temperature should not be calculated unless it will be used (for example, melt criteria, mixture of solid and gas products, etc.).

In tension, the reference pressure, $P_r$, is zero instead of $P_H$. At $V = V_0$ the values of T and I are denoted $T_1$ and $I_1$, respectively. Using

$$\left(\frac{\partial T}{\partial I}\right)_P = \frac{1}{C_P} \quad , \tag{11}$$

and the assumption that $C_P$ and $C_V$ are indistinguishable and $C_V$ is constant, we have for the reference temperature

$$T_r = T_1 + \frac{I_r - I_1}{C_V} \quad . \tag{12}$$

For P = 0,

$$\left(\frac{\partial I}{\partial V}\right)_P = \frac{C_P}{3\alpha V} \quad . \tag{13}$$

With a different assumption about $C_V$,

$$C_P/3\alpha V = C_V/3\alpha V_0 = \text{constant} \quad , \tag{14}$$

we have

$$I_r = I_1 + \frac{C_V}{3\alpha V_0}(V - V_0) \quad . \tag{15}$$

For this reference energy and $P_r = 0$, the Grüneisen EOS becomes

$$P = \frac{\gamma}{V}\left(I - I_1 - \frac{C_V}{3\alpha V_0}(V - V_0)\right) \quad . \tag{16}$$

Also,

$$T = T_r + \frac{I - I_r}{C_V} \quad . \tag{17}$$

In order to have continuous P and T at $I = 0$, $V = V_0$, we find

$$I_1 = \frac{-P_0 V_0}{\gamma} \tag{18}$$

and

$$T_1 = T_0 - \frac{P_0 V_0}{C_V \gamma} \quad . \tag{19}$$

It can be shown that $\left(\frac{\partial P}{\partial V}\right)_S$ will be continuous at $V = V_0$, $P = P_0$ for

$$\gamma C_V \doteq 3\alpha C^2 \quad . \tag{20}$$

If this relation is not satisfied, there will be a "bend" in the isentrope.

For further details, see the memo "Consistent EOS Input for HOM" by J. N. Johnson.

```
      SUBROUTINE GAS(I,J)                                                GAS       2
C   EQUATION OF STATE FOR GAS ONLY                                       GAS       3
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,          PARAM     2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                   PARAM     3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTA8=1,NTA8=MTA8*3742                     PARAM     4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                         PARAM     5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                          MCELL     2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)    MCELL     3
     +,W(MCL)                                                            MCELL     4
      LEVEL 2,R                                                          MCELL     5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,  MCELL     6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                     MCELL     7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                   MCELL     8
      LEVEL 2,TIME                                                       MCELL     9
      COMMON/GASC/GC(NGC,ML)                                             GC        2
      DIMENSION G(MLGC)                                                  GAS       7
      EQUIVALENCE (GC,G)                                                 GAS       8
      K=(I-1)*NGC                                                        GAS       9
  110 ALNV=ALOG(V(J))    ℓn V                                            GAS      10
      ALNPI=G(K+1)+ALNV*(G(K+2)+ALNV*(G(K+3)+ALNV*(G(K+4)+ALNV*G(K+5)))  GAS      11
     1))    ℓn Pi                                                        GAS      12
      ALNII=G(K+6)+ALNPI*(G(K+7)+ALNPI*(G(K+8)+ALNPI*(G(K+9)+ALNPI*G(K   GAS      13
     1+10))))   ℓn Ii'                                                   GAS      14
      ALNTI=G(K+11)+ALNV*(G(K+12)+ALNV*(G(K+13)+ALNV*(G(K+14)+ALNV*G(K   GAS      15
     1+15))))     ℓn Ti                                                  GAS      16
      EI=EXP(ALNII)-G(K+17)     Ii                                       GAS      17
      P(J)=EXP(ALNPI)+(EI-XI(J))/V(J)*(G(K+12)+ALNV*(G(K+13)+G(K+13)+    GAS      18
     1ALNV*(3.*G(K+14)+ALNV*4.*G(K+15))))     P = Pi + (I - Ii)/βV       GAS      19
      T(J)=EXP(ALNTT)+(XI(J)-EI)*23890./G(K+16)    T = Ti + (I - Ii)/Cv  GAS      20
      RETURN                                                             GAS      21
      END                                                                GAS      22
```

## GAS

Calculates the EOS for gases using analytic fits to the results of the BKW code. By special choice of constants, a $\gamma$-law gas EOS may be calculated.

Local Variables

G = one-dimensional array equivalenced to the two-dimensional array GC.

K = index to locate the data for region I in G.

NGC = parameter = # of gas constants per region.

ALNV = $\ell n\ V_g$.

ALNPI = $\ell n\ P_i$.

ALNII = $\ell n\ I_i'$.

ALNTI = $\ell n\ T_i$.

EI = $I_i' - Z = I_i$.

Notes

The BKW equation of state for the gaseous products is

$$PV_g/RT = 1 + xe^{\beta x} \quad , \tag{1}$$

where $V_g$ is the molar volume of the gaseous products and

$$x = \frac{\kappa k}{V_g(T + \theta)^{\alpha}} \quad , \tag{2}$$

where

$$k = \sum_g x_i k_i \quad , \tag{3}$$

with $x_i = \pi_i/n_g$ being the mole fraction of gaseous compound i, $k_i$ is a constant covolume for that compound, and $\kappa,\alpha,\beta$ are constants fit to reproduce detonation data. Also, there may be solid products such as graphite. The mole fractions will vary with volume and temperature. The equilibrium composition is calculated by minimization of the Gibbs free energy. Space does not permit a complete discussion of the EOS calculation in the BKW code. The interested reader is referred to Appendix E of Numerical Modeling of Detonations by C. L. Mader. Analytic fits are made in the BKW code to reference values of pressure ($P_i$), specific internal energy ($I_i$) and temperature ($T_i$) on the adiabat going through the CJ point. The fits used are given by

$$\ell n\ P_i = G_1 + G_2(\ell n\ V) + G_3(\ell n\ V)^2 + G_4(\ell n\ V)^3 + G_5(\ell n\ V)^4 \quad , \tag{4}$$

$$\ell n\ I_i' = G_6 + G_7(\ell n\ P_i) + G_8(\ell n\ P_i)^2 + G_9(\ell n\ P_i)^3 + G_{10}(\ell n\ P_i)^4 \quad , \tag{5}$$

$$\ell n\ T_i = G_{11} + G_{12}(\ell n\ V) + G_{13}(\ell n\ V)^2 + G_{14}(\ell n\ V)^3 + G_{15}(\ell n\ V)^4 \quad , \tag{6}$$

$$I_i = \exp\ (\ell n\ I_i') - Z \quad , \tag{7}$$

where Z is a constant such that $I_i$ has the same energy zero as the solid EOS. Also, Z can be used to keep $I_i'$ positive when making a fit.

The Grüneisen EOS is

$$P = P_i + \frac{\gamma}{V}(I - I_i) \quad , \tag{8}$$

where

$$\gamma = V\left(\frac{\partial P}{\partial I}\right)_V \quad . \tag{9}$$

We can use the thermodynamic relation

$$\left(\frac{\partial P}{\partial I}\right)_V = \left(\frac{\partial P}{\partial S}\right)_V\left(\frac{\partial S}{\partial I}\right)_V = -\frac{1}{T}\left(\frac{\partial T}{\partial V}\right)_S \tag{10}$$

to write $\gamma$ in terms of the function $\beta$ defined by

$$-\frac{1}{\beta} = \left(\frac{\partial \ \ln T}{\partial \ \ln V}\right)_S = \frac{V}{T}\left(\frac{\partial T}{\partial V}\right)_S \quad , \tag{11}$$

giving

$$\gamma = \frac{1}{\beta} \quad . \tag{12}$$

We can readily evaluate $\beta$ since $\ln T_i$ on the adiabat is fit as a function of $\ln V$. The result is

$$-\frac{1}{\beta} = G_{12} + 2G_{13}\ln V + 3G_{14}(\ln V)^2 + 4G_{15}(\ln V)^3 \quad . \tag{13}$$

The pressure is then given by

$$P = P_i + \frac{I - I_i}{\beta V} \quad . \tag{14}$$

With the assumption of constant $C_V$, the temperature is

$$T = T_i + \frac{(I - I_i)}{C_V} \quad . \tag{15}$$

```
      SUBROUTINE SSBGAS(I,J)                                              SSBGAS
C     SHARP SHOCK BURN                                                    SSBGAS
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,           PARAM
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                    PARAM
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                      PARAM
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                          PARAM
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                           MCELL
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)     MCELL
     +,W(MCL)                                                             MCELL
      LEVEL 2,R                                                           MCELL
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,   MCELL
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TO(ML),IJK                      MCELL
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                    MCELL
      LEVEL 2,TIME                                                        MCELL
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO   INIT
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),  INIT
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)              INIT
      COMMON/GASC/GC(NGC,ML)                                              GC
      COMMON/ES/IE(ML2),NME                                               ESM
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)                                   BUP
     +,BUR,BUD                                                            BUP
      COMMON/EOSN/IEOS(ML),ME(ML)                                         EN
      DIMENSION G(MLGC)                                                   SSBGAS
      EQUIVALENCE (GC,G)                                                  SSBGAS
      II=IE(I)                                                            SSBGAS
      IF(IEOS(I).EQ.2)GO TO 2                                             SSBGAS
      K=(I-1)*NGC                                                         SSBGAS
  110 ALNV=ALOG(V(J))    ℓn V                                             SSBGAS
      ALNPI=G(K+1)+ALNV*(G(K+2)+ALNV*(G(K+3)+ALNV*(G(K+4)                 SSBGAS
     ++ALNV*G(K+5))))    ℓn Pi                                            SSBGAS
      ALNII=G(K+6)+ALNPI*(G(K+7)+ALNPI*(G(K+8)+ALNPI*(G(K+9)              SSBGAS
     ++ALNPI*G(K+10))))    ℓn Ii                                          SSBGAS
      EI=EXP(ALNTI)-G(K+17)    Ii                                         SSBGAS
  111 SIP=EXP(ALNPI)    Pi                                                SSBGAS
      GAV=(G(K+12)+ALNV*(G(K+13)*2+ALNV*(3*G(K+14)+ALNV*4*G(K+15))))      SSBGAS
     +/V(J)    -1/βV                                                      SSBGAS
      P(J)=(SIP+FT*GAV)/(1+0.5*(VO(I)-V(J))*GAV)    P on Hugoniot         SSBGAS
      XI(J)=0.5*P(J)*(VO(I)-V(J))    I on Hugoniot                        SSBGAS
      RETURN                                                              SSBGAS
    2 CONTINUE  Buildup EOS                                               SSBGAS
      WPCJ=BUDV(I)*ROW(I)/(T(J)+1.)    PCJ                                SSBGAS
      WVCJ=T(J)*VO(I)/(T(J)+1.)    VCJ                                    SSBGAS
      WBI=T(J)*(T(J)-2.32)/(T(J)-.66)    1/β                              SSBGAS
      WK=-0.5*BUDV(I)/(T(J)*T(J)-1.)    I∞                                SSBGAS
      WPI=WPCJ*(WVCJ/V(J))**T(J)    Pi                                    SSBGAS
      P(J)=(WPI-(WK+WPI*V(J)/(T(J)-1))/(WBI*V(J)))/    P on Hugoniot      SSBGAS
     +(1-(VO(II)-V(J))/(2*WBI*V(J)))                                      SSBGAS
      XI(J)=P(J)*(VO(II)-V(J))/2    I on Hugoniot                         SSBGAS
      RETURN                                                              SSBGAS
      END                                                                 SSBGAS
```

## SSBGAS

Calculates the pressure and specific internal energy for a cell that has just been burned using the sharp-shock burn method. The pressure and specific internal energy are calculated on the Hugoniot for the HE products at the given volume.

Local Variables

G = one-dimensional array equivalenced to the two-dimensional array GC

K = index to locate the data for region I in G

NGC = parameter = # of gas constants per region

ALNV = $\ell n\ V_g$

ALNPI = $\ell n\ P_i$

ALNII = $\ell n\ I'_i$

EI = $I'_i - Z = I_i$

$S_{IP} = P_i$

GAV = $-\frac{1}{\beta V}$

Notes

In the sharp shock burn (see subroutine SSB for more details) a cell is compressed to CJ volume without going through the hydrodynamic equations. So, it is necessary to find the Hugoniot pressure, $P_H$, and specific internal energy, $I_H$, from the jump conditions and the equation of state. The jump condition for specific internal energy is

$$I_H = \frac{1}{2}(V_0 - V)(P_H + P_0) \quad . \tag{1}$$

For detonations, $P_0$ is negligible. The equation of state for the HE products is

$$P = \frac{1}{\beta V}(I - I_i) + P_i \quad , \tag{2}$$

where $I_i$ and $P_i$ are functions of volume only (see subroutine GAS). By using $I_H$ from Eq. (1) for I in Eq. (2) and solving for $P = P_H$, we have

$$P_H = \frac{P_i - I_i/\beta V}{1 - (V_0 - V)/2\beta V} \quad . \tag{3}$$

Then using the value of $P_H$, $I_H$ follows from Eq. (1).

```
      SUBROUTINE MIX(I,J)                                                MIX        2
C  EQUATION OF STATE FOR MIXTURE OF SOLID AND GAS                        MIX        3
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                         PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                          MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)    MCELL      3
     +,W(MCL)                                                            MCELL      4
      LEVEL 2,R                                                          MCELL      5
      COMMON/DVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,  MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                     MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                   MCELL      8
      LEVEL 2,TIME                                                       MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO  INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),PRO(ML), INIT       3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)             INIT       4
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),          US         2
     +GAMMA(ML),ALP(ML)                                                  US         3
      COMMON/GASC/GC(NGC,ML)                                             GC         2
      COMMON/FGHIJC/FS(ML),GS(ML),HS(ML),SI(ML),SJ(ML),CV(ML)            FG         2
      DIMENSION G(MLGC)                                                  MIX       10
      EQUIVALENCE (GC,G)                                                 MIX       11
      DIMENSION VIT(10)                                                  MIX       12
      DATA VIT(3)/1.E-5/  Tolerance for solution: pressure equilibrium within  MIX 13
      DATA VIT(10)/0./                                  10^-5 Mbar       MIX       14
      DATA VGSS/0.65/                                                    MIX       15
      XLM=1.01*VO(I)*(S1(I)-1.)/S1(I)  1.01 times V at which Hugoniot pressure  MIX 16
      K=(I-1)*NGC                                       is infinite      MIX       17
  210 OMW=1.-W(J)                                                        MIX       18
      OMWR=1./OMW                                                        MIX       19
      IF (V(J).LT.VO(I)) GO TO 230   Iterate on Vs for V < V0; Vg for V > V0  MIX  20
      WR=1./W(J)                                                         MIX       21
      VIT(1)=(V(J)-W(J)*VO(I)*VGSS)*OMWR  1st guess for Vg: assume Vs =  MIX       22
      VIT(2)=0.998   Ratio to get next point                  VGSS*VO(I) MIX       23
C     IBR=1 FOR ITERATION ON VG                                          MIX       24
      IBR=1                                                              MIX       25
  215 CALL LFB (X,F,VIT)  One step of iteration                          MIX       26
      IF (VIT(10)) 900,260,220   Error, solution, continue iteration     MIX       27
  220 IF (X.LE.0.) GO TO 225   For fixup when X = Vg becomes unphysical  MIX       28
      VG=X                  ]                                            MIX       29
      VS=(V(J)-OMW*VG)*WR   ] New values for Vg,Vs                       MIX       30
      IF (VS.LE.0.) GO TO 225   For fixup                                MIX       31
      GO TO 250                                                          MIX       32
C     SET VS=VG=VOLUME WHEN GET IN TROUBLE                               MIX       33
  225 VS=V(J) ]                                                          MIX       34
      VG=V(J) ]  Attempted fixup when iteration gives unphysical values  MIX       35
      X=V(J)  ]                                                          MIX       36
      GO TO 250                                                          MIX       37
  230 VIT(1)=V(J)    1st guess for Vs                                    MIX       38
      VIT(2)=.999    Ratio to get 2nd point                              MIX       39
C     IBR=2 FOR ITERATION ON VS                                          MIX       40
      IBR=2                                                              MIX       41
  235 CONTINUE                                                           MIX       42
      IF(F.LT.10.0.AND.X.GT.XLM)GO TO 236 ]  Don't allow Vs to get near or     MIX 43
      X=(X+VIT(4))/2                      ]  below the singularity in pressure MIX 44
      GO TO 240                           ]                              MIX       45
  236 CALL LFB(X,F,VIT)   One step of iteration                          MIX       46
      IF (VIT(10)) 900,260,240   Error, solution, continue iteration     MIX       47
  240 IF (X.LE.0.) GO TO 225   For fixup                                 MIX       48
```

```
      VS=X                        ] New values for Vs,Vg
      VG=(V(J)-W(J)*VS)*OMWR      ]
      IF (VG.LE.0.) GO TO 225
C     CALCULATE TEMPERATURE/PRESSURE DIFFERENCE FOR MIXTURE ITERATION
  250 VOMV=VO(I)-VS
      HP=((C1(I)/(VO(I)-S1(I)*VOMV))**2)*VOMV   PH
      HE = (HP-PO(I))*VOMV*0.5   IH
      ALNV=ALOG(VS)   ln Vs
      HT=EXP(FS(I)+ALNV*(GS(I)+ALNV*(HS(I)+ALNV*(SI(I)+ALNV*SJ(I))))) TH
      ALNV=ALOG(VG)   ln Vg
      ALNPI=G(K+1)+ALNV*(G(K+2)+ALNV*(G(K+3)+ALNV*(G(K+4)+ALNV*G(K+5))))
      IF(ALNPI.LT.G(K+19))ALNPI=G(K+19)  ln Pi, limit of validity on ln Ii fit
      EI=EXP(G(K+6)+ALNPI*(G(K+7)+ALNPI*(G(K+8)+ALNPI*(G(K+9)+ALNPI*G(
     1K+10)))))-
     1G(K+17)   Ii
      PI=EXP(ALNPI)   Pi
      TI=EXP(G(K+11)+ALNV*(G(K+12)+ALNV*(G(K+13)+ALNV*(G(K+14)+ALNV*G(
     1K+15)))))   Ti
      IF(ALNV.GT.G(K+18))ALNV=G(K+18)  Limit of validity of 1/β fit
      BETER=-(G(K+12)+ALNV*(G(K+13)+G(K+13)+ALNV*(3.*G(K+14)+4.*ALNV*G
     1(K+15))))   1/β
      TEMP=-G(K+16)*BETER/VG   -C'v V/βVg
      TEMP1=GAMMA(I)*CV(I)/VS   γCv/Vs
      F=-(HT*TEMP1+TI*TEMP)*4.18585182E-5   -[(THγsCv/Vs) - (TiC'v/βVg)]
      TEMP=TEMP+TEMP1   Equilibrium temperature
      VSTO=(CV(I)-G(K+16))*W(J)+G(K+16)   WCv + (1 - W)C'v
      F=((OMW*G(K+16)*TI+W(J)*CV(I)*HT)*4.18585182E-5+(EI-HE)*W(J)-EI+
     1XI(J))*TEMP/VSTO+F-PI+HP   Pressure difference = fi(x)
      GO TO (215,235),IBR
C     HAVE FOUND A SOLUTION FOR THE MIXTURE
C     GET THE TEMPERATURE AND PRESSURE
  260 VARST=((((TI-HT)*G(K+16)*4.18585182E-5+XI(J)*OMWR-EI)*CV(I)+HE*G
     1(K+16))*OMW/VSTO)-HE   I - IH
      P(J)=HP+VARST*GAMMA(I)/VS  ] Output P,T
      T(J)=HT+VARST*23890./CV(I) ]
      RETURN
C     ERROR IN HOM ITERATION      SET IND TO -1
  900 IND=-1
      PRINT 901,TIMF,I,J,X,F,VIT,VS,VG,IBR
  901 FORMAT(1X,10H HOM ERROR,E12.5,2I5,4E12.5/10E12.5,I5)
      RETURN
      END
```

## MIX (I,J)

Calculates pressure and temperature for a mixture of solid and gas where temperature and pressure are assumed to be in equilibrium. The equations of state for the solid and gas are described more fully in USUP and GAS, respectively.

Local Variables

VIT = array of dimension 10 used by LFB (q.v.) for the iterative solution.

VGSS = relative specific volume assumed for the solid for the first guess when iteration is on $V_g$.

K = index to locate the data for region I in G.

OMW = 1 - W.

OMWR = $(1 - W)^{-1}$.

WR = 1/W.

IBR = 1 for iteration on $V_g$; 2 for iteration on $V_s$.

X = current value of the iteration variable: updated by LFB.

F = current value of the iteration function: calculated in MIX.

VG = $V_g$.

VS = $V_s$.

HP = $P_H$.

HE = $I_H$.

HT = $T_H$.

ALNV = $\ell n\ V_g$ or $\ell n\ V_s$.

ALNPI = $\ell n\ P_i$.

EI = $I_i$.

PI = $P_i$.

TI = $T_i$.

BETER = $1/\beta$.

TEMP = $-C_V'/\beta V_g$.

TEMP1 = $\gamma_s C_V / V_s$.

VSTO = $WC_V + (1 - W)C_V'$.

VARST = $I - I_H$ in equilibrium.

IND = error index.

Notes

The solid equations used are

$$P_H = \frac{C^2(V_0 - V_s)}{[V_0 - S(V_0 - V_s)]^2} \quad , \tag{1}$$

$$\ell n\ T_H = F_s + G_s\ \ell n\ V_s + H_s(\ell n\ V_s)^2 + I_s(\ell n\ V_s)^3 + J_s(\ell n\ V_s)^4 \quad , \tag{2}$$

$$I_H = \frac{1}{2} P_H(V_0 - V_s) \quad , \tag{3}$$

$$P_s = \frac{\gamma_s}{V_s}(I_s - I_H) + P_H \quad , \tag{4}$$

$$T_s = T_H + \frac{I_s - I_H}{C_V} \quad , \tag{5}$$

where the s subscript denotes quantities associated with the solid.

The gas equations used are

$$\ell n\ P_i = A + B\ \ell n\ V_g + C(\ell n\ V_g)^2 + D(\ell n\ V_g)^3 + E(\ell n\ V_g)^4 \quad , \tag{6}$$

$$\ell n\ I_i' = K + L(\ell n\ P_i) + M(\ell n\ P_i)^2 + N(\ell n\ P_i)^3 + O(\ell n\ P_i)^4 \quad , \tag{7}$$

$$I_i = I_i' - Z \quad , \tag{8}$$

$$\ell n\ T_i = Q + R\ \ell n\ V_g + S(\ell n\ V_g)^2 + T(\ell n\ V_g)^3 + U(\ell n\ V_g)^4 \quad , \tag{9}$$

$$-\frac{1}{\beta} = R + 2S(\ell n\ V_g) + 3T(\ell n\ V_s)^2 + 4U(\ell n\ V_g)^3 \quad , \tag{10}$$

$$P_g = \frac{1}{\beta V_g} (I_g - I_i) + P_i \quad , \tag{11}$$

$$T_g = T_i + \frac{I_g - I_i}{C_V'} \quad , \tag{12}$$

where the g subscript denotes quantities associated with the gas.

The equilibrium conditions are

$$P = P_g = P_s \quad , \tag{13}$$

$$T = T_g = T_s \quad , \tag{14}$$

where P and T are the pressure and temperature of the mixture.

Two more relations are easily derived from the definition of mass fraction:

$$V = WV_s + (1 - W)V_g \quad , \tag{15}$$

$$I = WI_s + (1 - W)I_g \quad , \tag{16}$$

where V and I are the specific volume and specific internal energy of the mixture.

Multiplying Eq. (5) by $WC_V$ and Eq. (12) by $(1 - W)C_V'$ we have, after adding and substituting Eq. (14) and Eq. (16),

$$T = \frac{[I - (WI_H + (1 - W)I_i) + WC_V T_H + (1 - W)C_V' T_i]}{WC_V + (1 - W)C_V'} \quad , \tag{17}$$

which is an expression for the equilibrium temperature as a function of $V_s$ and $V_g$ (which are related by Eq. (15)).

Combining Eqs. (4), (11), and (13), we have

$$\frac{\gamma_s}{V_s} (I_s - I_H) + P_H = \frac{1}{\beta V_g} (I_g - I_i) + P_i \quad . \tag{18}$$

Combining Eqs. (5), (12), and (14), we have

$$I_s - I_H = C_V(T - T_H) \tag{19}$$

and

$$I_g - I_i = C_V'(T - T_i) \quad . \tag{20}$$

So the equation for pressure equilibrium (with equilibrium temperature T from Eq. (17)) may be written in the form

$$f(V_s,V_g) = P_H - P_i + \left(\frac{\gamma_s C_V}{V_s} - \frac{C_V'}{\beta V_g}\right)T - \left(\frac{\gamma_s C_V T_H}{V_s} - \frac{C_V' T_i}{\beta V_g}\right) = 0 \quad . \tag{21}$$

Using Eq. (16), this equation can be reduced to either of two functions of one variable:

$$f_1(V_g) = f\left(\frac{V - (1 - W)V_g}{W} , V_g\right) = 0 \tag{22}$$

or

$$f_2(V_s) = f\left(V_s , \frac{V - WV_s}{1 - W}\right) = 0 \quad . \tag{23}$$

Since the pressure for the mixture is always positive, $V_s$ will not get much larger than $V_0$ (thermal expansion at low pressure). So, for $V > V_0$, $f_2(V_s)$ will be very sensitive to the value of $V_s$. Therefore, Eq. (22) is solved for $V > V_0$ and Eq. (23) is solved for $V < V_0$. The solution is found by an iterative technique, basically the secant method, described in LFB. The method requires two starting points to be stored in VIT(1) and VIT(2) where $x_1$ = VIT(1) and $x_2$ = VIT(1) * VIT(2). For iteration on $V_s$, VIT(1) is chosen to be V. For iteration on $V_g$, VIT(1) is chosen such that $V_s$ is $V_0$ * VGSS. Sometimes the iteration will begin to diverge and give unphysical values such as negative $V_s$ or $V_g$. The standard fixup is to set $V_g = V_s = V$ which does not always work. Currently, if the iteration fails to converge, the old values of P,T are used and an error message is printed. Usually, for cases where the iteration fails to converge for several cycles for the same cell, the problem will become unstable and an error (usually due to a negative volume) will occur, stopping the problem.

Two of the fits used in the GAS equation of state are not always accurate for large $V_g$. The constants G(K + 18) and G(K + 19) provide limits on the valid regions of the fits. Further detail can be found in GASLM.

The function f is the difference in pressure of the gas and solid when they are in thermal equilibrium. So, the value of VIT(3) (which is the tolerance allowed in a solution) is the absolute limit of the accuracy in megabars. The standard limit is $10^{-5}$ Mbar.

```
      SUBROUTINE LF8(XP,FP,TX)
C      TX(1)     INITIAL GUESS
C      TX(2)     RATIO TO GET SECOND POINT
C      TX(3)     ZERO DEFINITION
C      TX(10)    COUNT OF NUMBER OF ITERATIONS
C                SET TO ZERO ON SOLUTION
C                SET TO NEGATIVE OF COUNT ON ERROR
C      FP        =FUNCTION(XP)
C      WHEN A SOLUTION IS FOUND, XP IS THE ROOT
C
C      ERROR EXITS OCCUR FOR
C             1. TOO MANY ITERATIONS, .GT.CNTMAX
C             2. TWO SUCESSIVE XP S OR FP S ARE EQUAL
      DIMENSION TX(10)
      DATA CNTMAX /100./
      IF (TX(10).LE.0.) GO TO 1
      TX(10)=TX(10)+1.
      IF (TX(10)-3.) 2,3,4
C  ENTRY FIRST TIME THROUGH
    1 TX(10)=1.
      IF (TX(1).EQ.0.) TX(1) = 1.
      XP=TX(1)  x0
C  GO GET F(XP)   f(x0)
      RETURN
C  ENTRY SECOND TIME THROUGH
    2 TX(9)=FP  f(x0)
      TX(8)=XP  x0
      TX(5)=FP
      IF (ABS(FP).LT.TX(3)) GO TO 18   Solution?
      XP=TX(1)*TX(2)    x1
C  GO GET F(XP)
      RETURN
C  ENTRY THIRD TIME THROUGH
    3 TX(5)=FP   f(x1)
      TX(6)=XP   x1
      TX(4)=XP   x1
      TX(7)=FP   f(x1)
      IF (ABS(FP).LT.TX(3)) GO TO 18   Solution?
      XP=TX(5)-TX(7)*(TX(6)-TX(8))/(TX(7)-TX(9))   x2
C  GO GET F(XP)   f(x2)
      RETURN
C  ENTRY FOR FOURTH AND SUCEEDING TIMES THROUGH
    4 IF (TX(10).GT.CNTMAX) GO TO 99
      TX(4)=XP   xi
      TX(5)=FP   f(xi)
      T=TX(4)-TX(5)   xi - xi-1
      IF (T.EQ.0.) GO TO 99
      IF (ABS(FP).LT.TX(3)) GO TO 18   Solution
      R=TX(5)-TX(7)    f(xi) - f(xi-1)
      IF (R.EQ.D.) GO TO 99
      XP=TX(4)-TX(5)*(T/R)   xi+1 by Eq. (1)
      IF (TX(5)*TX(7).LT.0.) GO TO 11 ]
      IF (TX(5)*TX(9).GE.0.) GO TO 11 ]  See notes
      IF (XP.GT.TX(4)) GO TO 6        ]
      IF (XP.GT.TX(8)) GO TO 10       ]
    8 XP=TX(4)-TX(5)*(TX(4)-TX(8))/(TX(5)-TX(9))   xi+1 by Eq. (2)
   10 TX(7)=TX(5) ]
      TX(6)=TX(4) ]  See notes
C  GO GET F(XP)   ]
      RETURN
```

```
    6 IF (XP.GT.TX(8)) GO TO 8                                  LF8    62
      GO TO 10                                                  LF8    63
   11 TX(9)=TX(7)                                               LF8    64
      TX(8)=TX(6)      See notes                                LF8    65
      GO TO 10                                                  LF8    66
C  HAVE FOUND A SOLUTION                                        LF8    67
   18 TX(10)=0.    Index for solution                           LF8    68
      TX(1)=XP                                                  LF8    69
      TX(4)=XP                                                  LF8    70
      RETURN                                                    LF8    71
C  AN ERROR HAS OCCURED                                         LF8    72
C  SET COUNT NEGATIVE AND EXIT                                  LF8    73
   99 TX(10)=-TX(10)   Index for error                          LF8    74
      RETURN                                                    LF8    75
      END                                                       LF8    76
```

## LFB(XP,FP,TX)

A two-point iteration scheme to find the zero of a function of one variable. The iteration is a slightly modified form of the secant method. This method is faster than Newton-Raphson iteration for the case where the time required to evaluate the derivative is longer than 0.44 of the time required to evaluate the function.

### Local Variables

XP = estimated value of the root from the previous iterative step.

FP = value of the function at XP.

TX = array containing current and previous values of XP and FP. Also, TX(1) = initial guess for XP; TX(2) = ratio to get second XP; TX(3) = error limit, TX(10) = count of iterations.

T = TX(4) - TX(6).

R = TX(5) - TX(7).

### Notes

The secant method for finding a root of $f(x) = 0$ is given by

$$x_{i+1} = x_i - \left(\frac{x_i - x_{i-1}}{y_i - y_{i-1}}\right) y_i \quad , \tag{1}$$

where $y_i = f(x_i)$. Two points, $x_0$ and $x_1$, are required to begin the iteration.

The secant method is used in LFB with the restriction that if $y_i y_{i-1} > 0$ and $y_i y_{i-2} < 0$ with $x_{i+1}$ not between $x_i$ and $x_{i-2}$, then

$$x_{i+1} = x_i - \left(\frac{x_i - x_{i-2}}{y_i - y_{i-2}}\right) y_i \quad . \tag{2}$$

This modification treats the case where a root is known to be between $x_i$ and $x_{i-2}$ from the fact that $y_i y_{i-2} < 0$. If $y_i y_{i-1} > 0$, then there is the possibility that $x_{i+1}$ from Eq. (1) is not between $x_i$ and $x_{i-2}$. (If that is the case, then Eq. (2) is used for $x_{i+1}$, which will then give $x_{i+2}$ between $x_i$ and $x_{i-2}$.) This modification helps avoid divergence of the solution in some cases. When Eq. (2) is used, $x_{i-1}$ is replaced by $x_{i-2}$ and $y_{i-1}$ is replaced by $y_{i-2}$. The relation of $x_i$ and $y_i$ to TX in the code is as follows

TX(4) = $x_i$

TX(5) = $y_i$

TX(6) = $x_{i-1}$

TX(7) = $y_{i-1}$

TX(8) = $x_{i-2}$

TX(9) = $y_{i-2}$

```
      SUBROUTINE BEQST(I,J)                                                  BEQST     2
C     BARNES SOLID EQUATION OF STATE                                         BEQST     3
C     ROUTINE SETS VOLUMES GREATER THAN VO TO VO AND ENERGIES LESS           BEQST     4
C     THAN ZERO DEGREE ENERGY TO ZERO DEGREE ENERGY                          BEQST     5
      REAL  N,NU,IC                                                          BEQST     6
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,              PARAM     2
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                      PARAM     3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTA8=1,NTA8=MTA8*3742                        PARAM     4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                             PARAM     5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                              MCELL     2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)        MCELL     3
     +,W(MCL)                                                                MCELL     4
      LEVEL 2,R                                                              MCELL     5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UT,UF,UTI,UFI,NADD,NM,      MCELL     6
     +IALPH,NDFLT,LABFL(8),NDUMP,IDMP,NM1,TD(ML),IJK                        MCELL     7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                       MCELL     8
      LEVEL 2,TIME                                                           MCELL     9
      COMMON/BRNS/A(ML),BR(ML),BA(ML),VBO(ML),VBSW(ML)                       BRN       2
      DIMENSION E1BR(ML),EXBR(ML)                                            BEQST    10
C     VOLUME CAN NOT BE GREATER THAN VO                                      BEQST    11
      DATA E1BR,EXBR/ML*0.,ML*0./                                            BEQST    12
      IF(E1BR(I).NE.0.)GO TO 10                                              BEQST    13
      E1BR(I)=E1(BR(I)) ]                                                    BEQST    14
      EXBR(I)=EXP(BR(I))] Constant for a given material                      BEQST    15
   10 CONTINUE                                                               BEQST    16
      VV=V(J)                                                                BEQST    17
      IF(VV.GT.VBO(I)) VV=VBO(I)                                             BEQST    18
      N=VBO(I)/VV     η                                                      BEQST    19
      CRN = N**(1./3.)   η^(1/3)                                             BEQST    20
      NU = 1. - 1./CRN   ν                                                   BEQST    21
      EBRN=EXP(BR(I)*NU)   exp(b_r ν)                                        BEQST    22
      EBAN=EXP(BA(I)*NU)   exp(b_a ν)                                        BEQST    23
      PC=A(I)*(EBRN*CRN**5-EBAN*CRN*CRN)  P_c                                BEQST    24
      GM=-1./3.+1./18.*((EBRN*(8.*BR(I)*CRN**4    γ                          BEQST    25
     ++18.*CRN**5+BR(I)*BR(I)*N)+EBAN*                                       BEQST    26
     +(-2.*BA(I)*CRN-BA(I)*BA(I)))/(EBRN*(CRN**5                             BEQST    27
     ++1./3.*BR(I)*CRN**4)-EBAN*1./3.*CRN*BA(I)))                            BEQST    28
      BON=BR(I)/CRN   b_r η^(-1/3)                                           BEQST    29
      IC=3./2.*A(I)*VBO(I)*(EBRN*(CRN**2-BR(I)      I_c                      BEQST    30
     +*CRN)+BR(I)-1.+(2./BA(I))*(1.-EBAN)                                    BEQST    31
     +-BR(I)*BR(I)*EXBR(I)*(E1BR(I)-E1(BON)))                                BEQST    32
C     ENERGY CAN NOT BE LESS THAN ZERO DEGREE ENERGY                         BEQST    33
      IF(XI(J).LT.IC)XI(J)=IC                                                BEQST    34
      P(J)=PC+GM*(XI(J)-IC)/VV  Pressure output                              BEQST    35
      PZ=PC                                                                  BEQST    36
      RETURN                                                                 BEQST    37
      END                                                                    BEQST    38
```

BEQST(I,J)

The Barnes EOS is used for the high-pressure region where the USUP fit becomes unphysical.

Local Variables

VV = AMIN (V(J),VBO(I)).

N = $\eta = \frac{V_0}{V}$, where $V_0$ = VBO(I), V = V(J), and N is declared real.

CRN = $\eta^{1/3}$.

NU = $\nu = 1 - \eta^{-1/3}$. NU is declared real.

EBRN = $e^{b_r\nu}$.

EBAN = $e^{b_a\nu}$.

PC = $P_c = A\eta^{2/3}(\eta e^{b_r\nu} - e^{b_a\nu})$ = pressure on the zero-degree isotherm.

GM = $\gamma$ calculated by the Dugdale-MacDonald formula using $P_c$.

BON = $b_r\eta^{-1/3}$.

IC = $I_c$ = specific internal energy of the zero-degree isotherm. IC is declared real.

Notes

The Morse potential has been modified by Barnes to give the proper (i.e., $P \propto \rho^{5/3}$ for free-electron gas) form at very high pressures. The pressure on the zero-degree isotherm, $P_c$, is given by

$$P_c = a\eta^{2/3}\left(\eta e^{b_r\nu} - e^{b_a\nu}\right) , \quad (1)$$

where $\eta = V_0/V$ , $\nu = 1 - \eta^{-1/3}$ , and a, $b_r$, and $b_a$ are constants. The constants a and $b_r$ are usually chosen such that the repulsive term matches the Thomas-Fermi-Dirac (TFD) pressure at $\eta = 1$ and $\eta = 10$, respectively. The TFD pressure is typically fit within 2% over the range $1 < \eta < 15$ by this repulsive term. Given a and $b_r$, then $b_a$ is chosen to match the experimental isothermal bulk modulus, $B_0$, using

$$B_0 = \left(\frac{dP_c}{d\eta}\right)_{\eta=1} = \frac{1}{3}\, a(3 + b_r - b_a) \quad . \tag{2}$$

The specific internal energy on the zero-degree isotherm, $I_c$, is given by

$$I_c = -\int_{V_0}^{V} P_c\, dV \quad . \tag{3}$$

Changing variables to $\eta = V_0/V$ gives

$$I_c = V_0 \int_1^{\eta} \frac{P_{c(\eta)}}{\eta^2}\, d\eta = aV_0 \int_1^{\eta} \left(\eta^{-1/3} e^{b_r\nu} - \eta^{-4/3} e^{b_a\nu}\right) d\eta \quad . \tag{4}$$

Further change of variables to $x = \eta^{-1/3} = 1 - \nu$ gives

$$\begin{aligned} I_c &= -3aV_0 \int_1^{\eta^{-1/3}} \frac{xe^{b_r(1-x)} - x^{+4} e^{b_a(1-x)}}{x^4}\, dx \\ &= +\frac{3}{2}\, a V_0 \Bigg[-1 + b_r + e^{b_r\nu}\left(\eta^{2/3} - b_r\eta^{1/3}\right) + \frac{2}{b_a}\left(1 - e^{b_a\nu}\right) \\ &\qquad -b_r^2\left(\left[-E_i(-b_r)\right] - \left[-E_i(-b_r\eta^{-1/3})\right]\right)\Bigg] \quad , \end{aligned} \tag{5}$$

where $E_i(x) = \int_{-\infty}^{x} \frac{e^t}{t}\, dt$ is the exponential integral.

The Grüneisen $\gamma$ is computed using the Dugdale-MacDonald formula:

$$\gamma = -\frac{1}{3} - \frac{V}{2}\, \frac{\partial^2(PV^{2/3})/\partial V^2}{\partial(PV^{2/3})\partial V} \quad , \tag{6}$$

or in terms of $\eta$,

$$\gamma = -\frac{1}{3} + \frac{\eta^2 \frac{\partial}{\partial\eta}(PV^{2/3}) + \frac{\eta^3}{2}\frac{\partial^2}{\partial\eta^2}(PV^{2/3})}{\eta^2 \frac{\partial(PV^{2/3})}{\partial\eta}} \quad , \tag{7}$$

where

$$PV^{2/3} = AV_0^{2/3}(\eta e^{b_r\nu} - e^{b_a\nu}) \quad . \tag{8}$$

After evaluating the derivatives and multiplying numerator and denominator by $\eta^{-1/3}$, we have

$$\gamma = -\frac{1}{3} + \frac{1}{18}\,\frac{e^{b_r\nu}\left(18\eta^{5/3} + 8\,b_r\eta^{4/3} + b_r^2\eta\right) + e^{b_a\nu}\left(-2b_a\eta^{1/3} - b_a^2\right)}{e^{b_r\nu}\left(\eta^{5/3} + \frac{1}{3}\,b_r\eta^{4/3}\right) + e^{b_a\nu}\left(-\frac{b_a}{3}\,\eta^{1/3}\right)} \quad . \tag{9}$$

The pressure is then evaluated using the Grüneisen EOS with $P_c$ and $I_c$ as reference pressure and specific internal energy, respectively,

$$P = P_c + \frac{\gamma\eta}{V_0}(I - I_c) \quad . \tag{10}$$

Care must be taken in choosing the volume, VBSW, below which the Barnes EOS is used instead of the USUP EOS. The Barnes $\gamma$ depends on volume and the USUP $\gamma$ is constant. So, for a given volume the two $\gamma$'s are generally not equal. That means there is only one value of I which gives a continuous pressure at VBSW. Therefore, a value of VBSW chosen to give a continuous pressure on the Hugoniot will not give a continuous pressure along a typical adiabat. So, VBSW may need to be different for different problems. The error due to a wrong choice of VBSW will be enhanced as VBSW decreases. This is due primarily to the increased values of I with smaller volume.

The usual choice of VBSW is to match pressures at the Hugoniot. Using the Hugoniot energy equation

$$E_H = \frac{1}{2}(V_0 - V)P_H = \frac{1}{2}V_0((\eta - 1)/\eta)P_H \tag{11}$$

and the Grüneisen EOS, we can solve for the Hugoniot pressure

$$P_H = \frac{P_c - \gamma(\eta/V_0)E_c}{1 - \gamma(\eta - 1)/2} \tag{12}$$

and compare with USUP Hugoniot to find the proper choice for VBSW.

```
      SUBROUTINE BLDUP(I,J)                                               BLDUP      2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,           PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                    PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                      PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                          PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                           MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)     MCELL      3
     +,W(MCL)                                                             MCELL      4
      LEVEL 2,R                                                           MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,   MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                      MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                    MCELL      8
      LEVEL 2,TIME                                                        MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO   INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),  INIT       3
     +MAT(ML),UD(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)              INIT       4
      COMMON/EOSM/IEOS(ML),ME(ML)                                         EN         2
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)                                   BUP        2
     +,BUR,BUD                                                            BUP        3
      DATA WMAX/0.99/                                                     BLDUP      8
      DATA TL/0./,IL/0/                                                   BLDUP      9
      IF(W(J).GE.WMAX)GO TO 11  Set P = 0 for W > 0.99                    BLDUP     10
      IF(T(J).EQ.TL.AND.I.EQ.IL)GO TO 20  Don't recalculate constants unless   BLDUP     11
                                                               necessary
      TL=T(J)                                                             BLDUP     12
      IL=I                                                                BLDUP     13
      WPCJ=BUDV(I)*ROW(I)/(T(J)+1.)  PCJ                                  BLDUP     14
      WVCJ=T(J)*VO(I)/(T(J)+1.)    VCJ                                    BLDUP     15
      WBI=T(J)*(T(J)-2.32)/(T(J)-.66)   1/β                               BLDUP     16
      WK=-0.5*BUDV(I)/(T(J)*T(J)-1.)   IB                                 BLDUP     17
   20 WPI=WPCJ*(WVCJ/V(J))**T(J)    Reference pressure                    BLDUP     18
      PG=WBI*((XT(J)-WK)/V(J)-WPI/(T(J)-1.))+WPI   Pressure for W = 0     BLDUP     19
      P(J)=PG*(1-W(J))   Scaled pressure                                  BLDUP     20
      GO TO 10                                                            BLDUP     21
11    CONTINUE                                                            BLDUP     22
      P(J)=0.                                                             BLDUP     23
10    CONTINUE                                                            BLDUP     24
      RETURN                                                              BLDUP     25
      END                                                                 BLDUP     26
```

## BLDUP

Calculates the equation of state to be used with the buildup burn model. The EOS is that of a $\gamma$-law gas but the $\gamma$ is not necessarily the same for all cells in a given material.

### Local Variables

TL = the value of T(J) = $\gamma$ for cell J from the last call to BLDUP.

IL = the value of I = original region # from the last call to BLDUP.

WPCJ = $P_{CJ}$ for this cell.

WVCJ = $V_{CJ}$ for this cell.

WBI = $1/\beta$ for this cell.

WK = $I_\infty$ for this cell.

WPI = the reference pressure, $P_r$, which is on the isentrope going through the CJ point.

PG = the actual pressure, $P_g$, which assumes W = 0.

### Notes

A number of explosives have been shown to have an effective CJ pressure that varies with distance of run. An effective way of modeling this experimentally observed phenomena is to use a $\gamma$-law EOS for the gas products where $\gamma$ is a function of the distance of run. (See Mader's book for more details.) To a very good approximation, $\gamma$ has been found to fit the data with the functional form

$$\gamma = A + \frac{B}{X} \quad , \tag{1}$$

where X is the distance of run. The detonation velocity, D, is essentially constant and is assumed to be constant. For short distances of run, $\gamma$ is not allowed to exceed a maximum value $\gamma_{max}$ because the functional form for $\gamma$ becomes inappropriate for small values of X. The buildup model is designed to describe conditions in which the HE is underdriven but promptly detonates. For a sufficiently small input shock, the buildup to detonation requires a non-negligible distance of run and should be calculated with the Forest Fire burn model (see FOREST). The constants for PBX-9404 are A = 2.68, B = 1.39, D = 0.88, and $\gamma_{max}$ = 3.7.

From the jump conditions and the definition of $\gamma$ (see, for example, Fickett and Davis for details), the CJ pressure and volume can be expressed as

$$P_{CJ} = \frac{\rho_0 D^2}{\gamma + 1} \tag{2}$$

and

$$V_{CJ} = \frac{\gamma V_0}{\gamma + 1} \quad , \tag{3}$$

where D is the detonation velocity and $\gamma$ is evaluated at the CJ point.

For buildup EOS we use a $\gamma$-law gas as the reference curve with the constant-$\beta$ EOS off the isentrope. For a given cell, $\gamma$ is a constant calculated from Eq. (1). With the condition that the reference curve goes through the CJ point, we have for the reference pressure

$$P_i = P_{CJ} V_{CJ}^{\gamma} V^{-\gamma} \quad . \tag{4}$$

The corresponding reference energy is then

$$I_i = \frac{P_i V}{\gamma - 1} + K \quad , \tag{5}$$

where K is a constant appropriate for the zero of energy used. In our case, the energy is defined as zero for the solid explosive. For this zero of energy, the specific internal energy at the CJ point is

$$I_{CJ} = \frac{P_{CJ}(V_0 - V_{CJ})}{2} \quad . \tag{6}$$

Substitution of Eq. (6) in Eq. (5) gives

$$K = -\frac{-P_{CJ}V_{CJ}}{\gamma - 1} + \frac{P_{CJ}}{2}(V_0 - V_{CJ}) = -\frac{D^2}{2(\gamma^2 - 1)} \quad . \tag{7}$$

The constant-$\beta$ EOS ($\beta$ is the inverse of the Grüneisen $\Gamma$) is given by

$$P = P_i + \frac{1}{\beta V}(I - I_i) \quad . \tag{8}$$

One can determine $\beta$ at the CJ point from experiment using the following relations (see Fickett and Davis, p. 30):

$$\alpha\beta\gamma = 1 + \alpha \tag{9}$$

and

$$\frac{\partial \ln D}{\partial \ln \rho_0} = \frac{\gamma - 1 - \alpha}{2 + \alpha} \quad , \tag{10}$$

to give

$$\beta = \frac{\gamma - \frac{\partial \ln D}{\partial \ln \rho_0}}{\gamma\left(\gamma - \frac{5}{3} - \frac{\partial \ln D}{\partial \ln \rho_0}\right)} \quad . \tag{11}$$

For PBX-9404, $\partial \ln D/\partial \ln \rho_0$ is about 0.66.

Either CJ burn or sharp-shock burn may be used with the buildup model using the CJ volume from Eq. (3) that can vary from cell to cell as $\gamma$ varies through Eq. (1).

```
      SUBROUTINE SPEOS(I,J)                                              SPEOS     2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,          PARAM     2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                   PARAM     3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                     PARAM     4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                         PARAM     5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                          MCELL     2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)    MCELL     3
     +,W(MCL)                                                            MCELL     4
      LEVEL 2,R                                                          MCELL     5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,  MCELL     6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                     MCELL     7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                   MCELL     8
      LEVEL 2,TIME                                                       MCELL     9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO  INIT      2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT      3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)             INIT      4
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),          US        2
     +GAMMA(ML),ALP(ML)                                                  US        3
      COMMON/SPC/SP(ML),USP(ML)                                          SPLC      2
     +,XISP(ML)                                                          SPLC      3
      COMMON/FGHIJC/FS(ML),GS(ML),HS(ML),SI(ML),SJ(ML),CV(ML)            FG        2
      COMMON/EOSN/IEOS(ML),ME(ML)                                        EN        2
      COMMON/ES/TE(ML2),NME                                              ESM       2
      II=IE(I)                                                           SPEOS    11
      IF(SP(II).LT.0.0001)RETURN    Don't spall for SP < 10^-4           SPEOS    12
      JMP=JMAX(I)+1                                                      SPEOS    13
      DPDX=P(J)/(R(J)-R(JMP))  Calculate pressure gradient               SPEOS    14
      IF(J.LE.(JMIN(I)+1)) RETURN   New regions must have at least 2 cells  SPEOS 15
      IF(J.GE.(JMP-2)) RETURN                                            SPEOS    16
      TM=SP(II)*SQRT(-DPDX)                                              SPEOS    17
      SPLP=-TM   Gradient spall pressure                                 SPEOS    18
      IF(TM.GT.USP(II))SPLP=-USP(II)   Ultimate spall pressure           SPEOS    19
      IF(P(J).GT.SPLP)RETURN  Spall?                                     SPEOS    20
      JS=J                                                               SPEOS    21
      RETURN                                                             SPEOS    22
      END                                                                SPEOS    23
```

## SPEOS

Determines whether a cell should spall by using the gradient spall model. As a special case, a constant spall pressure may be specified.

### Local Variables

II = IE(I) = original region #.

JMP = JMAX(I)+1 = index for the inside radius of the region I.

DPDX = $\frac{dP}{dx}$ assuming pressure is linear and the inside surface is a P = 0.

TM = negative of the gradient spall term.

SPLP = spall pressure.

### Notes

An empirical model* that has been found to fit experimental data for spalling is the gradient spall model. The spall pressure is of the form

$$P_s = -A\sqrt{\frac{dP}{dx}} \quad , \qquad (1)$$

where $\frac{dP}{dx}$ is the pressure gradient and A is a constant for a given material which is denoted SP(II) in the code for original region II. A locally evaluated numerical value of $\frac{dP}{dx}$ requires a very smooth solution for P vs x in order to be accurate. To avoid possible problems, $\frac{dP}{dx}$ has been approximated by the form

$$\frac{dP}{dx} \cong \frac{\Delta P}{\Delta x} = \frac{P}{x - x_0} \quad , \qquad (2)$$

where P is evaluated at x and $x_0$ is the inside surface of the region in question. The assumptions in this approximation are that the inside surface

*B. R. Breed, Charles L. Mader, Douglas Venable, J. Appl. Phys. 38, 3271 (1967).

of the region is a free surface with $P_0 = 0$, that the pressure is a linear function of distance between x and $x_0$, and that the direction of motion is toward the inside.

An additional assumption is made that there is an ultimate spall pressure (-USP(II) in the code) at which the material spalls regardless of the stress gradient. So if $P_s$ from Eq. (1) is less than -USP(II), then the spall pressure is set at -USP(II). If a cell meets the criteria for spalling, the flag JS is set to the cell #. The actual spalling is done elsewhere in the code. Since the code requires at least 2 cells in a region, spalling that would create a one-cell region is not allowed.

```
      SUBROUTINE POLY(I,J)                                                 POLY        2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,            PARAM       2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                     PARAM       3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                       PARAM       4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                           PARAM       5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                            MCELL       2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)      MCELL       3
     +,W(MCL)                                                              MCELL       4
      LEVEL 2,R                                                            MCELL       5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,    MCELL       6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                       MCELL       7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                     MCELL       8
      LEVEL 2,TIME                                                         MCELL       9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PD    INIT        2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),ORO(ML),   INIT        3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)               INIT        4
      COMMON/POLYC/CF(NCF,ML),PS(ML)                                       PLC         2
    1 CONTINUE                                                             POLY        7
      VV=VO(I)/V(J)-1.   η                                                 POLY        8
      A=VV*(CF(1,I)+CF(2,I)*ABS(VV))  A                                    POLY        9
      B=CF(3,I)+VV*(CF(4,I)+VV*CF(5,I))   B                                POLY       10
      C=CF(6,I)+CF(7,I)*VV   C                                             POLY       11
      XIJ=XI(J)*ROW(I)   ε                                                 POLY       12
      P(J)=(A+XIJ*(B+C*XIJ))/(XIJ+CF(8,I))  Pressure output                POLY       13
10    CONTINUE                                                             POLY       14
      RETURN                                                               POLY       15
      END                                                                  POLY       16
```

## POLY(I,J)

An eight-parameter fit to the equation of state that is basically a polynomial in two variables divided by a linear function in one of the variables. The two variables are related to specific volume and specific internal energy.

### Local Variables

VV = $\eta = \dfrac{V_0 - V}{V}$ (not the same $\eta$ as in CUSUP).

A = A in the notes (a function of $\eta$ only).

B = B in the notes (a function of $\eta$ only).

C = C in the notes (a function of $\eta$ only).

XIJ = $\rho_0 I$ = initial density * specific internal energy = $\varepsilon$ in the notes.

### Notes

The coefficients CF(J,I) will be denoted $C_J$ in the notes. The eight-parameter fit for the pressure is given by

$$P = \frac{A + B\varepsilon + C\varepsilon^2}{\varepsilon + C_8} \quad , \tag{1}$$

where

$$A = \eta C_1 + \eta|\eta| C_2 \quad , \tag{2}$$

$$B = C_3 + \eta C_4 + \eta^2 C_5 \quad , \tag{3}$$

$$C = C_6 + \eta C_7 \quad , \tag{4}$$

with

$$\eta = \frac{V_0 - V}{V} \tag{5}$$

and

$$\varepsilon = \rho_0 I \quad . \tag{6}$$

It can be shown that, like the USUP EOS, this EOS has a maximum possible compression on the Hugoniot. However, unlike the USUP EOS, adiabats are well described at higher compressions. For most (if not all) materials the $C_i$'s are all positive. A positive $C_8$ means the pressure is finite for finite $\eta$ and $\varepsilon$. (The Hugoniot pressure goes to infinity at maximum compression because $\varepsilon$ does.)

```
      SUBROUTINE VISC
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)
     +,W(MCL)
      LEVEL 2,R
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK
      COMMON/MTSC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS
      LEVEL 2,TIME
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),
     +MAT(ML),UD(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC
      COMMON/XCOM/R1,R2,DR1,DR2,WO,NCI,DR,ZI
      COMMON/FS/TF(ML2),NMF
      DO 10 I=1,NMC
      II=IE(I)
      JMN=JMIN(I)
      JMX=JMAX(I)
      DO 10 J=JMN,JMX
      DU=U(J+1)-U(J)    ΔU
      IF(ABS(P(J)).LT.QO(II))GO TO 11
      IF(DU.LT.0.0.AND.NV(I).NE.0)GO TO 11
      IF(NV(I)-1)1,2,3
1     Q(J)=4*XV(I)*DU/(3*V(J)*XM(J))    "Real" (negative Q allowed)
      GO TO 10
    2 Q(J)=ABS(XV(I)*(0.5*(U(J)+U(J+1))-UO(I))*DU/V(J))
      GO TO 10
    3 Q(J)=ABS(XV(I)*DU/V(J))    Landshoff
      GO TO 10
11    Q(J)=0.
10    CONTINUE
      RETURN
      END
```

# VISC

Computes the viscosity for all cells using either "real," PIC, or Landshoff-type viscosity.

## Local Variables

DU = U(J) - U(J-1) = the change in velocity across a cell.

I = region #.

J = cell #.

JMN,JMX = minimum and maximum active cell #'s in a region.

## Notes

For a sufficiently small time step, the difference equations coupled with an equation of state will lead to cell quantities that follow an adiabat, i.e., the entropy will not change. However, in a shock, entropy is not conserved. Without some dissipative mechanism it is impossible to satisfy the energy and momentum jump conditions at a shock front at the same time. That is, the proper pressure jump on the adiabat gives the wrong energy jump and vice versa. (However, for small amplitude shocks the adiabat and the Hugoniot are almost the same since it can be shown that the difference between the two is of third order in the specific volume change, e.g., see Courant and Friedrichs, p. 142.) This and other things lead to oscillations in the cell quantities after the shock front passes.

For most problems, the real viscosity is too small by several orders of magnitude to supply the needed entropy change across the shock front. Two "artificial viscosity" treatments are included in the code. Both are proportional to the velocity change, ΔU, across a cell. The PIC type is also proportional to the change in cell velocity from initial conditions. This effectively scales the viscosity with shock strength so that the shock is smeared over about the same number of cells independent of strength.

```
      SUBROUTINE BURN(I)                                                BURN       2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT      3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT       4
      COMMON/ES/IE(ML2),NME                                             ESM        2
      II=IE(I)                                                          BURN       6
      GO TO (1,2,3,4,5,6,7),IBRN(II)   Index for type of burn           BURN       7
    1 CALL ARH(I)   Arrhenius                                           BURN       8
      GO TO 10                                                          BURN       9
    2 CALL CJ(I)   CJ                                                   BURN      10
      GO TO 10                                                          BURN      11
    3 CALL SSB(I)   Sharp shock                                         BURN      12
      GO TO 10                                                          BURN      13
    4 CALL FOREST(I)   Forest Fire (pressure)                           BURN      14
      GO TO 10                                                          BURN      15
    5 CALL FFT(I)   Forest Fire (temperature)                           BURN      16
      GO TO 10                                                          BURN      17
    6 CALL FFI(I)   Forest Fire (energy)                                BURN      18
      GO TO 10                                                          BURN      19
    7 CALL GLTW(I)   Gamma-law Taylor wave                              BURN      20
   10 CONTINUE                                                          BURN      21
      RETURN                                                            BURN      22
      END                                                               BURN      23
```

## BURN(I)

Switching routine to determine type of burn to be used.

Notes

| IBRN | Type of Burn |
|---|---|
| 1 | Arrhenius |
| 2 | CJ |
| 3 | Sharp shock |
| 4 | Forest Fire |
| 5 | Forest Fire rate as a function of temperature |
| 6 | Forest Fire rate as a function of internal energy |
| 7 | Gamma-law Taylor wave |

Any other type of burn may be added by extending the computed go to statement last and adding the subroutine call and the subroutine.

```
      SUBROUTTNF ARH(I)                                                 ARH      2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM    2
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM    3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM    4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM    5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL    2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL    3
     +,W(MCL)                                                           MCELL    4
      LEVEL 2,R                                                         MCELL    5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL    6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL    7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL    8
      LEVEL 2,TIME                                                      MCELL    9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT     2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT    3
     +MAI(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT     4
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML) BRD     2
     +,MSFF                                                             BRD      3
      COMMON/ES/IE(ML2),NME                                             ESM      2
      II=IE(I)                                                          ARH      8
      JMN=JMIN(I)                                                       ARH      9
      JMX=JMAX(I)                                                       ARH     10
      IF(E(II).LT.0.0001) GO TO 11  Don't calculate if constants say don't burn   ARH     11
      IF(Z(II).LT.0.0001) GO TO 11                                      ARH     12
      DO 10 J=JMN,JMX                                                   ARH     13
      IF(W(J).EQ.0.)GO TO 10  No more to burn                           ARH     14
      IF(T(J).LT.0.0001) GO TO 10  Avoid underflow                      ARH     15
      W(J)=W(J)*(1.-DT*Z(II)*EXP(-E(II)/(1.9865*T(J)))) W^-1 dW/dt = Ze^-E/RT   ARH     16
      IF(W(J).LT.0.6)W(J)=0.   Burn the rest of it                      ARH     17
   10 CONTINUE                                                          ARH     18
   11 CONTINUF                                                          ARH     19
      RETURN                                                            ARH     20
      END                                                               ARH     21
```

## ARH(I)

Calculates the decomposition due to an Arrhenius rate law for region I.

### Local Variables

II = IE(I) = original region #.

JMN = minimum cell #.

JMX = maximum cell #.

J = do loop index = cell #.

### Notes

The Arrhenius rate for burn is given by

$$\frac{1}{W}\frac{dW}{dt} = Ze^{-E/RT} \quad , \tag{1}$$

where W is the mass fraction of undecomposed explosive, Z is a frequency factor ($\mu s^{-1}$), E is the activation energy in cal/mole, R is the gas constant (1.9865 cal/mole-K), and T is the temperature (K). This corresponds to a thermally activated process where the barrier height is EK/R and the frequency of attempts to cross the barrier is Z.

```
      SUBROUTINE CJ(I)                                                   CJ
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,          PARAM
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                   PARAM
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                     PARAM
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                         PARAM
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                          MCELL
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)    MCELL
     +,W(MCL)                                                            MCELL
      LEVEL 2,R                                                          MCELL
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFT,NADD,NM,  MCELL
     +IALPH,NOELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),TJK                     MCELL
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                   MCELL
      LEVEL 2,TIME                                                       MCELL
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO  INIT
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)             INIT
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML) BRD
     +,MSFF                                                              BRD
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                                MN
      COMMON/EOSN/IEOS(ML),ME(ML)                                        EN
      COMMON/ES/IE(ML2),NME                                              ESM
      II=IE(I)                                                           CJ
      JMN=JMIN(I)                                                        CJ
      JMX=JMAX(I)                                                        CJ
      DO 10 J=JMN,JMX                                                    CJ
      WO=W(J)                                                            CJ
      IF(IEOS(II).EQ.2) VCJ(II)=T(J)*VO(II)/(T(J)+1)  Buildup VCJ        CJ
      W(J)=1.-(VO(II)-V(J))/(VO(II)-VCJ(II))   CJ burn                   CJ
      IF(W(J).LT.0.02)W(J)=0.                                            CJ
      IF(WO.GT.W(J))GO TO 10   Don't unburn                              CJ
      W(J)=WO                                                            CJ
      IF(WO.LT.0.9)W(J)=0.                                               CJ
   10 CONTINUE                                                           CJ
      RETURN                                                             CJ
      END                                                                CJ
```

# CJ(I)

Calculates the decomposition of a detonating HE using the CJ burn model.

## Local Variables

II = IE(I) = original region #.

JMN = minimum cell #.

JMX = maximum cell #.

J = do loop index = cell #.

WO = old value of W(J).

## Notes

The CJ burn model assumes the burn fraction, W, varies linearly with the specific volume, V, between $V_0$ and $V_{CJ}$; that is,

$$W = 1 - \frac{V_0 - V}{V_0 - V_{CJ}} = \frac{V - V_{CJ}}{V_0 - V_{CJ}} \quad . \tag{1}$$

For the buildup EOS (see BLDUP), $V_{CJ}$ is a function of position and is given by

$$V_{CJ} = \frac{\gamma V_0}{\gamma + 1} \quad , \tag{2}$$

where $\gamma$ depends on position and is stored in the temperature variable.

Several constraints are made on the value of W. If $W < 0.02$ the rest is burned. If $W_{old} < W_{new}$ (i.e., the cell expands) then the old W is still used to prevent "unburning." If this occurs for $W < 0.9$ then the cell is completely burned.

The burn is normally initiated by a piston with the CJ particle velocity. After the detonation is well started (3rd cell in has burned), the piston velocity is switched to the escape velocity of the products.

The CJ burn is appropriate for a case where the HE promptly detonates. Otherwise, the Forest Fire model (see FOREST) will be appropriate.

```
      SUBROUTINE SSB(I)
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)
     +,W(MCL)
      LEVEL 2,R
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS
      LEVEL 2,TIME
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML)
     +,MSFF
      COMMON/UCJC/UCJ,JJ,NMAX,RCJ,DCJ
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC
      COMMON/ES/IE(ML2),NME
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)
     +,BUR,BUD
      COMMON/EOSN/IEOS(ML),ME(ML)
      DATA ISSB,JJ/1,1/
      II=IE(I)
      IF(JJ.GT.KMAX(I))GO TO 50   New region
      IF(JJ.LT.KMIN(I))JJ=KMIN(I)
      IF(ISSB-4)30,35,35   4 cycles per cell
   30 ISSB=ISSB+1
      P(JJ)=0.
      Q(JJ)=0.
      RETURN
   35 ISSB=1
      W(JJ)=0.   Burn the cell
      Q(JJ)=0.
      CALL SSBGAS(II,JJ)   Get CJ P,T
      IF(IA.EQ.0.AND.IEOS(II).EQ.2)UCJ=-DCJ/(T(JJ+1)+1)   Build up UCJ
      IF(IA.EQ.0)GO TO 42
      DCJ=VO(II)*SQRT(P(JJ)/(VO(II)-V(JJ)))  ]  Converging geometry
      UCJ=-SQRT(P(JJ)*(VO(II)-V(JJ)))        ]
   42 DT=((R(JJ+1)-R(JJ+2))/DCJ*0.25)   4 cycles for detonation to cross the cell
      JJ=JJ+1
      NMAX=JJ   Next cell
      IF(JJ.LE.KMAX(I))U(JJ)=UCJ   Particle velocity for next cell
      RCJ=R(JJ)   Radius
      CALL JMNMX(NMAX)   Activate the next cell
      IF(NMC.EQ.I)RETURN
      IF(II.NE.IE(I))RETURN
   50 DT=DTO(II+1)   Next region
      IBRN(II)=1
      II=IE(I+1)
      IF(IBRN(II).NE.3)GO TO 51
      IF(IEOS(II).EQ.2)GO TO 52
      UCJ=-E(II)              ]
      DCJ=VCJ(II)             ]
      GO TO 53                ]
   52 DCJ=SQRT(BUDV(II))      ]  Initial UCJ,DCJ for the next region
      JMN=JMIN(I+1)           ]
      UCJ=-DCJ/(T(JMN)+1)     ]
```

```
53 CONTINUE
   JJ=KMIN(I+1)-1
   U(JJ+1)=UCJ
   ISS8=0
   GO TO 42
51 NMAX=NMAX+10   Next region not sharp shock
   CALL JMNMX(NMAX)
   RETURN
   END
```

## SSB

Calculates the decomposition of an HE using a sharp shock model. All of the HE is burned at the shock front.

Local Variables

II = IE(I) = original region #.

ISSB = index incremented by one each cycle to keep track of when a cell should be burned.

Notes

The sharp shock burn runs with the following constraints on the cell at the shock front.

1. The inside boundary is held at a fixed position.
2. The outside boundary is moved at the CJ particle velocity.
3. The viscosity and pressure are set to zero until the cell is burned.
4. The cell is compressed for the time it takes for the wave to cross the cell at the detonation velocity. (The time step is set so that this takes four cycles.)
5. When the cell is burned, the internal energy and pressure are set on the Hugoniot for that volume.

For converging geometry the detonation velocity and particle velocity are no longer constant. The first cell is burned with the slab geometry values and the rest are calculated with a new detonation velocity $D_{CJ}$, particle velocity $U_{CJ}$, and time step $\Delta t$ that are calculated from

$$D_{CJ} = V_0[P_{CJ}/(V_0 - V_{CJ})]^{0.5} \quad , \tag{1}$$

$$U_{CJ} = -[P_{CJ}(V_0 - V_{CJ})]^{0.5} \quad , \tag{2}$$

$$\Delta t = (\Delta X/D_{CJ})0.25 \quad , \tag{3}$$

where

$P_{CJ}$ = Hugoniot P, $V_{CJ}$ = V of cell just burned.

When the shock arrives at a new explosive, the new input CJ detonation and particle velocities are used to compress the new explosive.

The advantages of a sharp shock burn are that one can obtain excellent Taylor waves by using a small number of cells to describe multiple layers of explosives in plane or converging geometry. The disadvantages are that it requires more information and more artificial constraints than do the CJ volume or Arrhenius-burn techniques. The sharp shock burn can give incorrect results for systems in diverging geometry and for systems that are overdriven or significantly underdriven.

```
      SUBROUTINE FOREST(I)                                              FOREST     2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL      3
     +,W(MCL)                                                           MCELL      4
      LEVEL 2,R                                                         MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL      8
      LEVEL 2,TIME                                                      MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VD(ML),PO INIT       2
     +(ML),TO(ML),RDW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT      3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT       4
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML) BRD       2
     +,MSFF                                                             BRD        3
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                               MN         2
      COMMON/ES/IE(ML2),NME                                             ESM        2
      II=IE(I)                                                          FOREST     9
      JMN=JMIN(I)                                                       FOREST    10
      JMX=JMAX(I)                                                       FOREST    11
      JL=JMN                                                            FOREST    12
      DO 10 J=JMN,JMX                                                   FOREST    13
      IF(W(J).LE.0.) GO TO 11  All burned                               FOREST    14
      PPQ=P(J)+AMAX1(0.,Q(J))  Pressure dependent rate                  FOREST    15
      IF(PPQ.LT.PM(II)) GO TO 10  Minimum pressure to calculate burn    FOREST    16
      IF(PPQ.GT.PCJ(II)) GO TO 11  Burn all if PPQ > PCJ                FOREST    17
      IF(MSFF.EQ.0)GO TO 20  Multiple shock Forest Fire?                FOREST    18
      IF(SX(J).NE.0.)GO TO 21  Has 1st shock passed?                    FOREST    19
      IF(J.NE.JMX)GO TO 22  Special at JMX for reflected shocks         FOREST    20
      JP=JMIN(I+1)                                                      FOREST    21
      PP=P(JP)+AMAX1(0.,Q(JP))  If the pressure in the next material exceeds   FOREST    22
      IF(PP.LT.SX(JL))GO TO 22  that in the nearest cell flagged as the        FOREST    23
      DO 33 JJ=JL,JMX           shock having passed, then those cells not      FOREST    24
   33 SX(JJ)=SX(JL)             flagged as the shock having passed are flagged FOREST    25
      GO TO 21                  with that pressure in SX.                      FOREST    26
   22 CONTINUE                                                          FOREST    27
      PQL=SZ(J)    P + Q from previous cycle is in SZ                   FOREST    28
      SZ(J)=PPQ                                                         FOREST    29
      IF(PPQ.GT.PQL)GO TO 20                                            FOREST    30
      SX(J)=PPQ    Shock has passed; set SX                             FOREST    31
   21 CONTINUE                                                          FOREST    32
      JL=J                                                              FOREST    33
      IF(PPQ.LT.SX(J))SX(J)=PPQ  Pressure used to determine rate, may decrease FOREST 34
      PPQ=SX(J)                                         but not increase       FOREST 35
   20 CONTINUE                                                          FOREST    36
      SUM=0.                                                            FOREST    37
      NDA=ND(II)                                                        FOREST    38
      DO 1000 N=1,NDA                                                   FOREST    39
 1000 SUM=SUM*PPQ+DWDT(N,II)   Polynomial fit                           FOREST    40
      IF(SUM.GT.100.)SUM=100.   Avoid overflow                          FOREST    41
      RATE=EXP(SUM)  Rate                                               FOREST    42
      W(J)=W(J)*(1.-DT*RATE)  New W                                     FOREST    43
      IF(W(J).LT.0.05) W(J)=0.                                          FOREST    44
      GO TO 10                                                          FOREST    45
   11 W(J)=0.                                                           FOREST    46
10    CONTINUE                                                          FOREST    47
      RETURN                                                            FOREST    48
      END                                                               FOREST    49
```

## FOREST(I)

Calculates the decomposition using the Forest Fire burn model. This model is appropriate for cases that require a non-negligible distance of run to detonation for the given input shock strength.

Local Variables

II = IE(I) = original region #.

JMN = minimum cell #.

JMX = maximum cell #.

J = do loop index = cell #.

PPQ = P + Max (Q,0).

SUM = temporary variable used to sum the $A_n P^n$ terms in the rate fit.

NDA = # of rate constants = n + 1.

RATE = $-\frac{1}{W}\frac{dW}{dt}$ from the fit.

Notes

This discussion of the Forest Fire rates is a condensed version of Appendix B in LA-7245. An even more thorough discussion of Forest Fire is in LA-6259. Several assumptions are made: (1) the Pop plot ($\ln$ P vs $\ln$ run of distance to detonation) is a straight line given by

$$\ln(\text{run}) = a_1 + a_2 \ln(P - a_3) \quad . \qquad (1)$$

where $a_1$, $a_2$, and $a_3$ are constants depending on the HE, and P is the input shock pressure; (2) the "single-curve buildup hypothesis" that the pressure grows along a unique line in (time, distance, state) space during buildup to detonation and that the Pop plot gives that line; (3) the "reactive Hugoniot" is assumed to be described by a USUP fit (several USUP fits are allowed but are not treated in this derivation)

$$U_S = C + SU_P \quad , \qquad (2)$$

which gives from the jump conditions for $P_0 = 0$,

$$P = \rho_0 U_S U_P \quad , \tag{3}$$

$$V = V_0(U_S - U_P)/U_S \quad , \tag{4}$$

$$I = U_P^2/2 \quad ; \tag{5}$$

(4) the HOM EOS is used with the functional form

$$P = H(V,I,W) \quad . \tag{6}$$

The Lagrange coordinates $m,\tau$ are related to the Eulerian coordinates $x,t$ by

$$\frac{\partial}{\partial m} = \frac{1}{\rho}\frac{\partial}{\partial x} \tag{7}$$

and

$$\frac{\partial}{\partial \tau} = \frac{\partial}{\partial t} + U\frac{\partial}{\partial x} \quad . \tag{8}$$

Letting subscripts denote partial derivatives with respect to the subscript, the fluid flow equations are

$$U_\tau = -P_m \quad , \tag{9}$$

$$V_\tau = U_m \quad , \tag{10}$$

and

$$I_\tau = -PV_\tau \quad . \tag{11}$$

The derivatives of shock front quantities are denoted by the symbol °. The equations for these derivatives are

$$\mathring{P} \equiv \frac{dP(x_s(t),t)}{dt} = P_m m_\tau + P_\tau = \rho_0 U_S P_m + P_\tau = \frac{dP}{d_{run}} U_S \quad , \tag{12}$$

where $x_s(t)$ is the shock position, and

$$\mathring{U} = \rho_0 U_S U_m + U_\tau = \frac{dU}{dP}\frac{dP}{dt} = \frac{\mathring{P}V_0}{C + 2SU} \quad . \tag{13}$$

Finally, the total derivative of the equation is given by

$$P_\tau = H_V V_\tau + H_I I_\tau + H_W W_\tau \quad , \tag{14}$$

which can be solved for $W_\tau$, the desired quantity.

The usual calculation is for $W_\tau$ as a function of pressure. The pressure-dependent rate has proved useful, but there is nothing in the calculation that says the rate is a function of pressure only. The solutions are for rates on the Hugoniot and any variable will do (P, T, V, or I). Once off the Hugoniot, assumptions about what variable (or variables) determine the rate leads to different rates for a given state. For this derivation we will assume the rate is a function of pressure only.

Given P, the run is given by Eq. (1). The particle velocity can be obtained using Eq. (2) and Eq. (3) to give

$$U_P = \frac{-\rho_0 C + \sqrt{(\rho_0 C)^2 + 4P\rho_0 S}}{2\rho_0 S} \quad . \tag{15}$$

We have $U_S$ from Eq. (2) and Eq. (5). Having $U_S$ and $U_P$, we substitute in Eq. (4) and Eq. (5) to get V and I. Then, knowing P, V, and I, we solve P = H(V,I,W) for W. Now, with V, I, and W, we calculate the derivatives $H_V$, $H_I$, and $H_W$ at that state point.

The shock front derivatives can be calculated using Eq. (1), Eq. (12), Eq. (13), and the already calculated value of $U_P$. The usual assumption $P_m = 0$ is then made. Eq. (12) becomes

$$P_\tau = \mathring{P} \tag{16}$$

and Eq. (13) combines with Eq. (9) and Eq. (10) to give

$$V_\tau = \frac{\mathring{U}}{\rho_0 U_S} \quad . \tag{17}$$

We already know $I_\tau$ from Eq. (11). Using the known values of $P_\tau$, $H_V$, $H_\tau$, $H_I$, $I_\tau$, and $H_W$, we have

$$W_\tau(P) = \frac{H_V V_\tau + H_I I_\tau - P_\tau}{H_W} \quad . \tag{18}$$

The function $\ell n\ (-\frac{1}{W} W_\tau)$ is then fit to a polynomial of the form

$$\ell n \left(-\frac{W_\tau}{W}\right) \cong \sum_{i=0}^{n} A_i P^i \quad . \tag{19}$$

The constants DWDT(N,II) are related to the $A_i$'s by

$$\mathrm{DWDT(N,II)} = A_{n-(N-1)} \quad . \tag{20}$$

The value of P + Q (denoted PPQ) is used instead of P to calculate the rate. The fluid flow equations are solved using P + Q instead of P, so this is consistent. If PPQ < PM(II), then the rate would be so small it would not be worthwhile to calculate it. If PPQ > PCJ(II), where PCJ is the CJ pressure, then the cell is all burned. For W < 0.05 the cell is also all burned. Except for the above restrictions, the new mass fraction is given by

$$W_{new} = W_{old}\left(1 - \Delta t \exp\left(\sum_{i=0}^{n} A_i (P + Q)^i\right)\right) \quad , \tag{21}$$

which is a linearized version of

$$\Delta W = W_{new} - W_{old} = \int_{t_{old}}^{t_{new}} W(- \frac{1}{W} W_\tau)\, d\tau \ , \qquad (22)$$

with $\ell n \left(- \frac{W_\tau}{W}\right)$ replaced by the fit.

```
      SUBROUTINE FFT(I)                                                  FFT         2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,          PARAM       2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                   PARAM       3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,KTAB=MTAB*3742                     PARAM       4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                         PARAM       5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                          MCELL       2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)    MCELL       3
     +,W(MCL)                                                            MCELL       4
      LEVEL 2,R                                                          MCELL       5
      COMMON/DVL/NDF,NI,NP,NG,TENO,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,  MCELL       6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                     MCELL       7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                   MCELL       8
      LEVEL 2,TIME                                                       MCELL       9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VD(ML),PO  INIT        2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT        3
     +MAT(ML),UD(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)             INIT        4
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML) BRD         2
     +,MSFF                                                              BRD         3
      COMMON/ES/IE(ML2),NME                                              ESM         2
      II=IE(I)                                                           FFT         8
      JMN=JMIN(I)                                                        FFT         9
      JMX=JMAX(I)                                                        FFT        10
      DO 10 J=JMN,JMX                                                    FFT        11
      IF(W(J).LE.0.)GO TO 11   All burned                                FFT        12
      TP=T(J)  Temperature-dependent rate                                FFT        13
      IF(TP.LT.PM(II))GO TO 10   Minimum temperature to calculate burn   FFT        14
      IF(TP.GT.PCJ(II))GO TO 11   Burn it all                            FFT        15
      SUM=0.                                                             FFT        16
      NDA=ND(II)                                                         FFT        17
      DO 1000 N=1,NDA                                                    FFT        18
 1000 SUM=SUM*TP+DWDT(N,II)   Polynomial fit                             FFT        19
      IF(SUM.GT.100.)SUM=100.   Avoid overflow                           FFT        20
      RATE=EXP(SUM)   Rate                                               FFT        21
      W(J)=W(J)*(1.-DT*RATE)   New W                                     FFT        22
      IF(W(J).LT.0.05)W(J)=0.                                            FFT        23
      GO TO 10                                                           FFT        24
   11 W(J)=0.                                                            FFT        25
   10 CONTINUE                                                           FFT        26
      RETURN                                                             FFT        27
      END                                                                FFT        28
```

## FFT

The Forest Fire rate is calculated as a function of temperature.

### Local Variables

See FOREST.

### Notes

The minimum temperature to calculate the rate is stored in PM and maximum temperature at which W is set to zero is stored in PCJ.

## FFI

The Forest Fire rate is calculated as a function of specific internal energy.

### Local Variables

See FOREST.

### Notes

The minimum specific internal energy to calculate the rate is stored in PM and maximum temperature at which W is set to zero is stored in PCJ.

```
      SUBROUTINE FFI(I)                                                 FFI       2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM     2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM     3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM     4
     +,NSM=4,NWPM=3729,NSD=NSM*NWPM+132,ML2=100)                        PARAM     5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL     2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL     3
     +,W(MCL)                                                           MCELL     4
      LEVEL 2,R                                                         MCELL     5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL     6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL     7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL     8
      LEVEL 2,TIME                                                      MCELL     9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT      2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),INIT      3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT      4
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML)BRD       2
     +,MSFF                                                             BRD       3
      COMMON/ES/TE(ML2),NME                                             ESM       2
      II=IE(I)                                                          FFI       8
      JMN=JMIN(I)                                                       FFI       9
      JMX=JMAX(I)                                                       FFI      10
      DO 10 J=JMN,JMX                                                   FFI      11
      IF(W(J).LE.0.)GO TO 11  All burned                                FFI      12
      E=XI(J)    Internal energy-dependent rate                         FFI      13
      IF(E.LT.PM(II))GO TO 10  Minimum energy to calculate rate         FFI      14
      IF(E.GT.PCJ(II))GO TO 11  Burn it all                             FFI      15
      SUM=0.                                                            FFI      16
      NDA=ND(II)                                                        FFI      17
      DO 1000 N=1,NDA                                                   FFI      18
 1000 SUM=SUM*E+DWDT(N,II)   Polynomial fit                             FFI      19
      IF(SUM.GT.100.)SUM=100.   Avoid overflow                          FFI      20
      RATE=EXP(SUM)  Rate                                               FFI      21
      W(J)=W(J)*(1.-DT*RATE)  New W                                     FFI      22
      IF(W(J).LT.0.05)W(J)=0.                                           FFI      23
      GO TO 10                                                          FFI      24
   11 W(J)=0.                                                           FFI      25
   10 CONTINUE                                                          FFI      26
      RETURN                                                            FFI      27
      END                                                               FFI      28
```

```
      SUBROUTINE GLTW(I)                                                GLTW
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL
     +,W(MCL)                                                           MCELL
      LEVEL 2,R                                                         MCELL
      COMMON/DVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TO(ML),IJK                    MCELL
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL
      LEVEL 2,TIME                                                      MCELL
      COMMON/BUX/BUA,BUR,BUMAX,BUDV(ML)                                 BUP
     +,BUR,BUD                                                          BUP
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),INIT
     +MAT(ML),UD(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT
      COMMON/ES/TE(ML2),NME                                             ESM
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                               MN
      COMMON/EOSN/IEOS(ML),ME(ML)                                       EN
      COMMON/GASC/GC(NGC,ML)                                            GC
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML) BRD
     +,MSFF                                                             BRD
      COMMON/UCJC/X1,I1,NMAX,X2,X3                                      GLTW
      II=IE(I)                                                          GLTW
      JMN=JMIN(I)                                                       GLTW
      JMX=JMAX(I)                                                       GLTW
      RO=R(JMX+1)                                                       GLTW
      G=-GC(2,II)            ]  γ                                       GLTW
      IF(IEOS(II).EQ.2)G=T(JMX)]                                        GLTW
      GP=G+1    γ + 1                                                   GLTW
      GM=G-1    γ - 1                                                   GLTW
      DV=VCJ(II)                         ]  D                           GLTW
      IF(IEOS(II).EQ.2)DV=SQRT(BUDV(II))]                               GLTW
      DV2=DV*DV    D^2                                                  GLTW
      UCJ=DV/GP    U_CJ                                                 GLTW
      DRO=R(JMN)-R(JMN+1)                                               GLTW
      UL=-UCJ*GP/GM                                                     GLTW
      U(JMX+1)=-UCJ   Start with CJ conditions at shock front and calculate outward GLTW
      PCJ=DV2*ROW(II)/GP    P_CJ                                        GLTW
      CJV=G/(GP*ROW(II))    V_CJ                                        GLTW
      CCJ=UCJ*G     C_CJ                                                GLTW
      DLR=V(JMX)*XM(JMX)                   ]                            GLTW
      IF(IA.NE.0)DLR=DLR/(R(JMX+1)**IA)]  Get new radius from V and M   GLTW
      R(JMX)=R(JMX+1)+DLR                 ]                             GLTW
      DR=R(JMN)-RO   ΔR                                                 GLTW
      DT=DR/DV    Time                                                  GLTW
      J=JMX+1             ]                                             GLTW
      DO 10 JJ=JMN,JMX]  Calculate cell quantities starting with the inside GLTW
      J=J-1               ]                                             GLTW
      W(J)=0.   All burned                                              GLTW
      U(J)=UCJ*(2*(R(J)-RO)/DR-1)]  Particle velocity                   GLTW
      IF(U(J).GT.UF)U(J)=UF      ]                                      GLTW
      CC=-GM*(U(J+1)+UL)/(2*G*UCJ)   Local C                            GLTW
      P(J)=PCJ*CC**(2*G/GM)                ]                            GLTW
      V(J)=CJV*CC**(-2/GM)                 ]  P,V,I solutions for that cell GLTW
      XI(J)=P(J)*V(J)/GM-DV2/(2*GP*GM)]                                 GLTW
      DLR=V(J)*XM(J)                  ]  Get new radius from V and M    GLTW
      IF(IA.NE.0)DLR=DLR/(R(J)**IA)]                                    GLTW
```

```
      R(J-1)=R(J)+DLR                                                            GLTW
   10 CONTINUE                                                                   GLTW
      IF(J.NE.2)GO TO 11                                                         GLTW
      R(1)=R(2)                                                                  GLTW
      U(1)=U(2)    Piston conditions if the HE was in region 1                   GLTW
      BU=UF                                                                      GLTW
      TIME=TIME+DT   Set the time to include the GLTW detonation time            GLTW
   11 CONTINUE                                                                   GLTW
      IBRN(II)=1                                                                 GLTW
      IF(I.EQ.1)RETURN                                                           GLTW
      NMAX=JMAX(I-1)   Special for other materials outside GLTW HE               GLTW
      CALL JMNMX(NMAX)                                                           GLTW
      NCI=JMX-JMN+1                                                              GLTW
      NT=NCI*12*(R(NMAX-1)-R(NMAX))/DRO                                          GLTW
      DT=DT/NT   Set time step                                                   GLTW
      NCLT=NCL                                                                   GLTW
      NCL=NMAX                                                                   GLTW
      BUI=UF   Effect of HE treated as a piston                                  GLTW
      KMAX(I-1)=KMAX(I-1)+1                                                      GLTW
      NMC=I-1                                                                    GLTW
      DO 20 II=1,NT                                                              GLTW
      TIME=TIME+DT    Run hydro for the outside regions to catch up with GLTW HE GLTW
      CALL DIFEQ                                                                 GLTW
   20 CONTINUE                                                                   GLTW
      NCL=NCLT                                                                   GLTW
      BUI=0.   Get rid of piston                                                 GLTW
      KMAX(I-1)=KMAX(I-1)-1                                                      GLTW
      NMAX=NMAX+NCI+3        Set up for normal hydro                             GLTW
      CALL JMNMX(NMAX)                                                           GLTW
      RETURN                                                                     GLTW
      END                                                                        GLTW
```

## GLTW

An entire region of explosive is burned using the gamma-law Taylor-wave description.

Local Variables

CC = C in notes = sound speed.

CJV = $V_{CJ}$ in notes = CJ volume.

DLR = width of a cell computed from its mass and specific volume.

DR = $\Delta R$ in notes = width of region.

DV = D in notes = detonation velocity.

DV2 = $D^2$.

G = $\gamma$ in notes.

GM = $\gamma - 1$.

GP = $\gamma + 1$.

II = IE(I) = original region #.

J = cell # being computed.

JJ = do loop index.

JMN = JMIN(I).

JMX = JMAX(I).

NCI = # of cells in the first region outside the GLTW HE.

NCLT = temporary storage of NCL.

NT = # of time steps used to allow the first region outside the GLTW HE to respond.

RO = inside radius of the GLTW HE.

UCJ = $U_{CJ}$ in notes = CJ particle velocity.

UL = $-U_{CJ}\left(\frac{\gamma + 1}{\gamma - 1}\right)$.

Notes

Following Fickett and Davis, we will outline the gamma-law Taylor-wave solution for a detonation. The equation of state in a rarefaction wave following the detonation is restricted to the CJ isentrope. The pressure and sound speed are then functions of density only. For a $\gamma$-law gas they are given by

$$P = P_{CJ}\left(\frac{V_{CJ}}{V}\right)^{\gamma} \tag{1}$$

and

$$C = C_{CJ}\left(\frac{P}{P_{CJ}}\right)^{(\gamma-1)/2\gamma} = \left(\frac{dP}{d\rho}\right)^{1/2} \quad . \tag{2}$$

With the addition of two characteristic equations,

$$\frac{dP}{du} = \rho C \tag{3}$$

and

$$U + C = x/t \quad , \tag{4}$$

a solution can be found for $U(x/t)$ where x is the Eulerian distance and t is the time. From the jump conditions the CJ state has the following relations.

$$U_{CJ} = \frac{D}{\gamma + 1} \tag{5}$$

$$P_{CJ} = \frac{\rho_0 D^2}{\gamma + 1} \tag{6}$$

and

$$V_{CJ} = \frac{V_0 \gamma}{\gamma + 1} \quad . \tag{7}$$

Using Eqs. (1) and (2), we can solve Eq. (3) for

$$P = P_{CJ}\left[\frac{1 + (\gamma - 1)(U - U_{CJ})}{2C_{CJ}}\right]^{2\gamma/\gamma-1} \quad . \tag{8}$$

Combining Eqs. (2), (4), and (8), we have

$$U = \left(\frac{2}{\gamma + 1}\right)\left(\frac{x}{t}\right) - U_{CJ} \quad . \tag{9}$$

By integrating the PdV work from infinite volume, one can find the specific internal energy on the CJ isentrope to be

$$I = \frac{P_V}{\gamma - 1} - \frac{P_{CJ}V_{CJ}}{\gamma - 1} + \frac{P_{CJ}}{2}\left(V_0 - V_{CJ}\right) \quad , \tag{10}$$

where the last term sets the zero of energy relative to the solid. The time is then set to

$$t = \frac{\Delta R}{D} \quad , \tag{11}$$

where $\Delta R$ is the width of the $\gamma$-law explosive. That is, the state is evaluated when the detonation has just crossed the entire region. For a given cell, we can use Eqs. (1), (8), (9), (10), and (11) to find V, P, U, and I. Then, from the mass and V, the radius of the inside of the next cell can be evaluated. For U greater than UF, the final piston velocity U is reset to the final piston velocity with the corresponding values for V, P, and I.

It must be remembered that Eq. (1) is the equation of state only on the CJ isentrope. Provision is made for either the HOM EOS or the Buildup EOS for subsequent evaluation of the EOS. However, the buildup EOS must not be used when the region is thick enough to require a nonconstant $\gamma$.

The GLTW is useful for burning a large region of HE in slab geometry with a minimum of computation. However, it has also proved useful in cylindrical or spherical geometry under certain restrictions. Since GLTW is designed for slab geometry, it can only be used for a small enough region that convergence can be ignored. This small region can be used instead of a piston to initiate HE burn for a larger region. In some cases the usual piston initiation can be sufficiently zoning-dependent to be noticeable. By burning three cells of a large HE region using GLTW, the zoning dependence is minimal.

Provision is made for materials outside the GLTW HE. A separate calculation is made for the outside regions for the total detonation time of the GLTW HE. A piston with the final piston velocity UF acts on the outside regions from the inside.

```
      SUBROUTINE BNDR1                                                  BNDR1      2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL      3
     +,W(MCL)                                                           MCELL      4
      LEVEL 2,R                                                         MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL      8
      LEVEL 2,TIME                                                      MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT       3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT       4
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                         VD         2
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                               MN         2
      COMMON/UCJC/UCJ,JJ,NMAX,RCJ,DCJ                                   UC         2
      COMMON/NSPLT/NOSPLT(ML2)                                          NSP        2
      COMMON/PWORK/PW,PWI                                               PWORK      2
      COMMON/ES/TE(ML2),NME                                             ESM        2
      COMMON/SPC/SP(ML),USP(ML)                                         SPLC       2
     +,XISP(ML)                                                         SPLC       3
      DATA IFLP/0/,IFLPI/0/                                             BNDR1     13
      IF(IBRN(NMC).NE.3)GO TO 20                                        BNDR1     14
      IF(JJ.LE.2)GO TO 20                                               BNDR1     15
      RCJ=RCJ+DT*UCJ             Sharp shock burn                       BNDR1     16
      R(JJ)=RCJ                                                         BNDR1     17
      U(JJ)=UCJ                                                         BNDR1     18
   20 CONTINUE                                                          BNDR1     19
      IFLP=IFLP-1                                                       BNDR1     20
      IFLPI=IFLPI-1                                                     BNDR1     21
      IF(BU.EQ.0.) GO TO 1   Skip if no outside piston                  BNDR1     22
      RO=R(1)                                                           BNDR1     23
      R(1)=R(1)+DT*BU                                                   BNDR1     24
      U(1)=BU               Outside piston boundary condition           BNDR1     25
      R(2)=R(1)                                                         BNDR1     26
      DVP=F2*0.5*(R(1)-RO)*(R(1)**IA+RO**IA+F3*R(1)*RO)                 BNDR1     27
      PW=PW+(P(2)+Q(2))*DVP   Work done by the piston                   BNDR1     28
      IF(IFLP.LT.0)PW=PW+XM(2)*((U(2)+U(3))**2-(U(1)+U(3))**2)          BNDR1     29
      IFLP=1                                                            BNDR1     30
      U(2)=U(1)                                                         BNDR1     31
    1 CONTINUE                                                          BNDR1     32
      IF(BUI.EQ.0.)GO TO 11    Skip if no inside piston                 BNDR1     33
      RO=R(NCL+1)                                                       BNDR1     34
      R(NCL+1)=R(NCL+1)+DT*BUI                                          BNDR1     35
      U(NCL+1)=BUI                 Inside piston boundary condition     BNDR1     36
      IF(R(NCL).GT.R(NCL+1))GO TO 11                                    BNDR1     37
      R(NCL)=R(NCL+1)                                                   BNDR1     38
      DVPI=F2*0.5*(R(NCL)-RO)*(R(NCL)**IA+RO**IA+F3*R(NCL)*RO)          BNDR1     39
      PWI=PWI+(P(NCL)+Q(NCL))*DVPI   Work done by piston                BNDR1     40
      IFLPI=1                                                           BNDR1     41
      U(NCL)=U(NCL+1)                                                   BNDR1     42
11    CONTINUE                                                          BNDR1     43
      IF(R(NCL).GT.1.E-10.OR.IA.EQ.0) GO TO 5 Special for converging geometry BNDR1     44
      R(NCL)=1.E-10  If R(NCL) is 0, the difference equations won't allow it to move BNDR1     45
      NOSPLT(NM)=0  Don't rezone after R(NCL) hits the origin           BNDR1     46
      U(NCL)=0.   Don't move until it goes into tension                 BNDR1     47
    5 CONTINUE                                                          BNDR1     48
```

```
      IF(NNV.EQ.0)RETURN   Skip if no voids
      DO 2 I=1,NMC
      IF(IV(I).LT.0) GO TO 2   No void
      J=JV(I)
      IF(J.GT.JMAX(NMC)) GO TO 2   Void not active
      IF(IV(I).GE.1)GO TO 3   Go check for closed void in tension
      IF(R(J).GT.R(J+1))GO TO 2   Open void still open
      DU=U(J)-U(J+1)   Void closes with relative velocity DU
      PRINT 100,TIME,R(J),DU,I,J,MAT(I),MAT(I+1)
      WRITE(8,100)TIME,R(J),DU,I,J,MAT(I),MAT(I+1)
  100 FORMAT(21H1VOID COLLAPSE! TIME=,1PE12.5,8H RADIUS=,1PE12.5,
     +4H DU=,1PE12.5,3H I=,I3,3H J=,I5,17H MATERIAL NUMBERS,2I6)
      IF(DU.LT.-0.5)GO TO 4   Special treatment for high velocity
      DR=R(J)-R(J+1)
      R(J)=(R(J)+R(J+1))/2     Calculate R,U for newly closed interface
      R(J+1)=R(J)                as if materials are the same
      U(J)=(U(J)+U(J+1))/2
      U(J+1)=U(J)
      IV(I)=1   Closed void flag
      GO TO 2
    4 CALL RL(I)   Special treatment for high velocity
      IV(I)=2
      GO TO 2
    3 II=IE(I)
      IF(P(J+1).LT.XISP(II).AND.P(J-1).LT.XISP(II))IV(I)=0   Open void?
      IF(IV(I).EQ.0)PRINT 101,TIME,R(J),I,J,MAT(I),MAT(I+1)
      IF(IV(I).EQ.0)WRITE(8,101)TIME,R(J),I,J,MAT(I),MAT(T+1)
  101 FORMAT(18H1VOID OPENS! TIME=,1PE12.5,8H RADIUS=,1PE12.5,
     +3H I=,I3,3H J=,I5,17H MATERIAL NUMBERS,2I6)
    2 CONTINUE
      RETURN
      ENTRY BNDR2
      IF(IFLP.EQ.1)PW=PW+(P(2)+Q(2))*DVP             Work done by pistons
      IF(IFLPT.EQ.1)PWI=PWI+(P(NCL)+Q(NCL))*DVPI
      RETURN
      END
```

## BNDR1

Calculates several special boundary conditions such as an applied piston.

### Local Variables

I = do loop index = region #.

J = JV(I) = cell# for outside free surface of open void.

DU = relative velocity of the two free surfaces bounding an open void.

DR = position difference of the two free surfaces bounding a just closed void. The positions were calculated assuming the void would not close.

### Notes

For sharp-shock burn, the cell currently being burned has its outside cell boundary moving at the computed CJ particle velocity instead of the results of hydrodynamics. A piston on the inside of the problem and a piston on the outside of the problem are allowed. (Pistons are only calculated when their velocity is not set to 0.) Piston positions are calculated using the input piston velocities BU and BUI. The inside and outside radii of the problem are calculated assuming they are free surfaces. If a radius is not within the limits of the corresponding piston then it is reset to the piston value and the velocity is reset to the piston velocity. For spherical and cylindrical geometry the origin cannot be crossed (yielding negative radii?). If it is, the inside radius is reset to $10^{-10}$. A negligible but positive value is used in order to allow the inside surface to move under tension. Otherwise, the difference equations will not allow it to move under any circumstances. Also, a flag is reset to turn off rezoning for the inside region after the inside surface hits the origin.

If there are voids, a check is made whether an open void just closed or a closed void should open. A closed void is allowed to open under tension.

When a void closes, the two free surfaces "overshoot" each other. An extrapolation is made back to the point of contact. The two velocities are reset to the average velocity. This velocity is correct only for identical materials. However, the difference equations will bring the interface velocity to its proper value unless the relative velocity is too great. For large relative velocities, the artificial viscosity treatments currently in use in the code will dump too much internal energy into the cells bounded by the interface before the cell boundaries have time to respond. That is, the difference in velocity across a cell next to a just-closed interface becomes instantaneously finite. When this difference is large (and viscosity depends on this difference in velocity) the above-mentioned problem occurs. A special treatment is made in this case that is described in RL.

```
      SUBROUTINE SL(I)                                                    SL
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,           PARAM
     +NUMV=10,MQI=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                    PARAM
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                      PARAM
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                          PARAM
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                           MCELL
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)     MCELL
     +,W(MCL)                                                             MCELL
      LEVEL 2,R                                                           MCELL
      COMMON/DVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,   MCELL
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                      MCELL
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                    MCELL
      LEVEL 2,TIME                                                        MCELL
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO   INIT
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),  INIT
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)              INIT
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                           VD
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                                 MN
      COMMON/NSPLT/NOSPLT(ML2)                                            NSP
      COMMON/ES/TE(ML2),NME                                               ESM
      IPM=NM+1                                                            SL
      J=JS   Cell # to spall                                              SL
      IF(IPM.GE.ML2)RETURN   Don't make more regions than storage allows  SL
      II=IE(I)                                                            SL
      CALL OUTGAS   GAS dump before spall                                 SL
      PRINT 100,TIME,R(J),I,J,MAT(II)                                     SL
      WRITE(8,100)TIME,R(J),I,J,MAT(II)                                   SL
  100 FORMAT(13H1SPALL! TIME=,1PE12.5,8H RADIUS=,1PE12.5,3H I=,I3,        SL
     +3H J=,I5,16H MATERIAL NUMBER,I6)                                    SL
      CALL PRNT   Print out before spall                                  SL
      DO 10 II=I,NM                                                       SL
      IP=IPM                                                              SL
      IPM=IP-1                                                            SL
      IE(IP)=IE(IPM)         ]                                            SL
      IV(IP)=IV(IPM)         ]                                            SL
      JV(IP)=JV(IPM)         ]                                            SL
      INTX(IP)=INTX(IPM)     ]                                            SL
      JMAX(IP)=JMAX(IPM)     ]Shift variables that characterize the region up by 1  SL
      KMAX(IP)=KMAX(IPM)     ]                                            SL
      JMIN(IP)=JMIN(IPM)     ]                                            SL
      KMIN(IP)=KMIN(IPM)     ]                                            SL
      NOSPLT(IP)=NOSPLT(IPM) ]                                            SL
   10 CONTINUE                                                            SL
      NM=NM+1   ]  ' One new region                                       SL
      NMC=NMC+1 ]                                                         SL
      CALL SHFT(I,J,1)   Shift cell quantities up by 1                    SL
      NNV=NNV+1   A new void                                              SL
      IV(I)=0       that is open                                          SL
      JP=J+1                                                              SL
      XM(J)=0. ]  Void cell                                               SL
      JV(I)=J  ]                                                          SL
      IT=1                  ]                                             SL
      IF(INTX(I).GE.2) IT=5 ]   Set interface flag                        SL
      INTX(I)=IT            ]                                             SL
      JMAX(I)=J-1   ]                                                     SL
      KMAX(I)=J-1   ]  Set region boundaries                              SL
      JMIN(I+1)=JP  ]                                                     SL
      KMIN(I+1)=JP  ]                                                     SL
      NOSPLT(I)=0   Turn off rezoning                                     SL
      R(JP)=R(J)                                                          SL
```

```
      IFLAG(J)=IFLAG(J-1)+64
      JMN=JP
      JMX=KMAX(NM)
      DO 12 K=JMN,JMX          Set flags so that GAS plots will show
      IFLAG(K)=IFLAG(K)+128     spall on interface plots
   12 CONTINUE
      IFLAG(NCL)=IFLAG(NCL)+128
      CALL OUTGAS   GAS dump after spall
      JS=0   Turn off spall flag
      RETURN
      END
```

## SL(I)

Does all the bookkeeping required to create a spall.

### Local Variables

IPM = NM+1 = # of regions after the spall (which splits one region into two); also, IPM is decremented in the do loop where data is shifted.

J = JS = cell # to spall.

II = IE(I) = original region #; also do loop index.

IP = previous value of IPM in the do loop where data is shifted.

JP = J+1 = cell # next to the created void.

IT = temporary variable to calculate INTX.

### Notes

When a spall occurs, a single region is split into two regions with an open void between them. The open void requires a new cell to be created.

```
      SUBROUTINE SPLTCHK                                                SPLTCHK    2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL      3
     +,W(MCL)                                                           MCELL      4
      LEVEL 2,R                                                         MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UIT,UFT,NADD,NM, MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),TJK                    MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL      8
      LEVEL 2,TIME                                                      MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PD INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT      3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT       4
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                               MN         2
      COMMON/NSPLT/NOSPLT(ML2)                                          NSP        2
      COMMON/ES/IE(ML2),NME                                             ESM        2
      DATA DRMN/0.001/                                                  SPLTCHK    9
      DATA N/4/                                                         SPLTCHK   10
      IF(KMAX(NM).GE.MCL-1-N)RETURN   Don't overrun storage             SPLTCHK   11
      DO 10 I=1,NMC                                                     SPLTCHK   12
      IF(NOSPLT(I).LE.0) GO TO 10   Is rezoning allowed?                SPLTCHK   13
      J=KMAX(I)                                                         SPLTCHK   14
      DLR=R(J)-R(J+1)                                                   SPLTCHK   15
      II=IE(I)                                                          SPLTCHK   16
      DR=5*DRO(II)      If the innermost cell has ΔR > 4ΔR0, then rezone SPLTCHK  17
      IF(DLR.LT.DR) GO TO 10                                            SPLTCHK   18
   12 CONTINUE                                                          SPLTCHK   19
      NCI=KMAX(I)-KMIN(I)+1   Don't split cells outside the region      SPLTCHK   20
      IF(N.GT.NCI)N=NCI                                                 SPLTCHK   21
      CALL OUTGAS                                                       SPLTCHK   22
      CALL SHFT(I,J,N)   Shift cells inside of the split by N           SPLTCHK   23
   11 CALL SPLIT(I,J,N)  Split the last N cells in half                 SPLTCHK   24
      CALL PRNT                                                         SPLTCHK   25
      CALL OUTGAS                                                       SPLTCHK   26
   10 CONTINUE                                                          SPLTCHK   27
      RETURN                                                            SPLTCHK   28
      END                                                               SPLTCHK   29
```

## SPLTCHK

Checks whether rezoning is required in a region and if so calls subroutines to do the rezoning.

### Local Variables

N = # of cells to be split if rezoning is required.

I = do loop index = region #.

J = cell # of innermost cell in region I.

DLR = width of cell J.

DR = 5 times initial width of cell J.

NCI = # of cells in region I.

### Notes

Rezoning is not checked for unless NOSPLT(I) > 0. Currently, the only criterion included is that the innermost cell of a region gets wider than five times the initial width. Other criteria are available for particular types of problems. The GAS dump before and after the rezoning is to avoid extraneous lines in certain types of plots.

```
      SUBROUTINE SHFT(I,J,N)                                                SHFT       2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,             PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                      PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                        PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                            PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                             MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)       MCELL      3
     +,W(MCL)                                                               MCELL      4
      LEVEL 2,R                                                             MCELL      5
      COMMON/DVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,     MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                        MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                      MCELL      8
      LEVEL 2,TIME                                                          MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO     INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),I8RN(ML),PLAP(ML),DRO(ML),    INIT       3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)                INIT       4
      COMMON/UCJC/UCJ,JJ,NMAX,RCJ,DCJ                                       UC         2
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                             VD         2
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                                   MN         2
      JJ=NCL+1                                                              SHFT       9
      R(JJ+N)=R(JJ)                                                         SHFT      10
      U(JJ+N)=U(JJ)                                                         SHFT      11
      DO 10 K=J,NCL                                                         SHFT      12
      JJ=JJ-1                                                               SHFT      13
      JN=JJ+N                                                               SHFT      14
      R(JN)=R(JJ)                                                           SHFT      15
      U(JN)=U(JJ)                                                           SHFT      16
      P(JN)=P(JJ)                                                           SHFT      17
      V(JN)=V(JJ)          Shift cell quantities up by N for cell #'s > J   SHFT      18
      XI(JN)=XI(JJ)                                                         SHFT      19
      SX(JN)=SX(JJ)                                                         SHFT      20
      SZ(JN)=SZ(JJ)                                                         SHFT      21
      W(JN)=W(JJ)                                                           SHFT      22
      XM(JN)=XM(JJ)                                                         SHFT      23
      IFLAG(JN)=IFLAG(JJ)                                                   SHFT      24
      T(JN)=T(JJ)                                                           SHFT      25
      Q(JN)=Q(JJ)                                                           SHFT      26
   10 CONTINUE                                                              SHFT      27
      DO 11 K=I,NM                                                          SHFT      28
      KMAX(K)=KMAX(K)+N                                                     SHFT      29
      JMAX(K)=JMAX(K)+N     1st KMIN, KMAX, JMIN, JMAX reflect the shifted cells SHFT 30
      IF(I.EQ.K) GO TO 11                                                   SHFT      31
      JMIN(K)=JMIN(K)+N                                                     SHFT      32
      KMIN(K)=KMIN(K)+N                                                     SHFT      33
   11 CONTINUE                                                              SHFT      34
      DO 12 K=1,NM                                                          SHFT      35
   12 IF(JV(K).GT.J) JV(K)=JV(K)+N  Shift void cell index                   SHFT      36
      NCL=NCL+N        N new cells                                          SHFT      37
      NMAX=NMAX+N                                                           SHFT      38
      RETURN                                                                SHFT      39
      END                                                                   SHFT      40
```

## SHFT

Shifts all cells with cell # ⩾ J up by N. Used when new cells are created in the middle of the problem; e.g., for spall and rezoning.

### Local Variables

JJ = index of cell being shifted.

JN = index of cell to which cell # JJ quantities are being shifted.

K = do loop index.

### Notes

Any new quantities that are tied to a cell # should be included in this subroutine.

```
      SUBROUTINE SPLIT(I,J,N)                                          SPLIT
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,        PARAM
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NOW=20,NCF=8,                 PARAM
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                   PARAM
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                       PARAM
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                        MCELL
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)  MCELL
     +,W(MCL)                                                          MCELL
      LEVEL 2,R                                                        MCELL
      COMMON/DVL/NCF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,  MCELL
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                   MCELL
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                 MCELL
      LEVEL 2,TIME                                                     MCELL
      JJ=J+1                                                           SPLIT
      JP=J+N+1                                                         SPLIT
      R(JP)=R(JJ)                                                      SPLIT
      U(JP)=U(JJ)                                                      SPLIT
      DO 10 K=1,N                                                      SPLIT
      JP=JP-2                                                          SPLIT
      JJ=JJ-1                                                          SPLIT
      IFLAG(JP)=IFLAG(JJ)                                              SPLIT
      IFLAG(JP+1)=IFLAG(JJ)                                            SPLIT
      R(JP+1)=(R(JJ)+R(JJ+1))/2                                        SPLIT
      R(JP)=R(JJ)                                                      SPLIT
      U(JP+1)=(U(JJ)+U(JJ+1))/2                                        SPLIT
      U(JP)=U(JJ)                                                      SPLIT
      JM=JJ-1                                                          SPLIT
      XMF=0.5*XM(JJ)/(XM(JJ)+XM(JM))                                   SPLIT
      DV=XMF*(V(JJ)-V(JM))                                             SPLIT
      V(JP+1)=V(JJ)+DV                                                 SPLIT
      V(JP)=V(JJ)-DV                                                   SPLIT
      DV=XMF*(P(JJ)-P(JM))                                             SPLIT
      P(JP+1)=P(JJ)+DV                                                 SPLIT
      P(JP)=P(JJ)-DV                                                   SPLIT
      DV=XMF*(XI(JJ)-XI(JM))                                           SPLIT
      XI(JP+1)=XI(JJ)+DV                            Interpolate        SPLIT
      XI(JP)=XI(JJ)-DV                                cell quantities  SPLIT
      DV=XMF*(SX(JJ)-SX(JM))                                           SPLIT
      SX(JP+1)=SX(JJ)+DV                                               SPLIT
      SX(JP)=SX(JJ)-DV                                                 SPLIT
      DV=XMF*(SZ(JJ)-SZ(JM))                                           SPLIT
      SZ(JP+1)=SZ(JJ)+DV                                               SPLIT
      SZ(JP)=SZ(JJ)-DV                                                 SPLIT
      DV=XMF*(W(JJ)-W(JM))                                             SPLIT
      W(JP+1)=W(JJ)+DV                                                 SPLIT
      W(JP)=W(JJ)-DV                                                   SPLIT
      DV=XMF*(T(JJ)-T(JM))                                             SPLIT
      T(JP+1)=T(JJ)+DV                                                 SPLIT
      T(JP)=T(JJ)-DV                                                   SPLIT
      DV=XMF*(Q(JJ)-Q(JM))                                             SPLIT
      Q(JP+1)=Q(JJ)+DV                                                 SPLIT
      Q(JP)=Q(JJ)-DV                                                   SPLIT
      CF=XM(JJ)                                                        SPLIT
      DR=R(JP+1)-R(JP+2)                                               SPLIT
      R1=R(JP+2)                                                       SPLIT
      R2=R(JP+1)                                                       SPLIT
      XM(JP+1)=DR*F2*(R1**IA+R2**IA+F3*R1*R2)/V(JP+1)                  SPLIT
      DR=R(JP)-R(JP+1)                                                 SPLIT
      R1=R2                                                            SPLIT
      R2=R(JP)                                                         SPLIT
```

```
      XM(JP)=DR*F2*(R1**IA+R2**IA+F3*R1*R2)/V(JP)                    SPLIT   52
      CF=CF/(XM(JP)+XM(JP+1))                                        SPLIT   53
      XM(JP)=XM(JP)*CF        Be sure mass is conserved exactly      SPLIT   54
      XM(JP+1)=XM(JP+1)*CF                                           SPLIT   55
   10 CONTINUE                                                       SPLIT   56
      RETURN                                                         SPLIT   57
      END                                                            SPLIT   56
```

# SPLIT

Splits N cells starting at cell #J into two cells. All cell quantities are linearly interpolated and conservation of mass is explicitly required.

## Local Variables

JJ = cell # being split.

JP = cell # to which the split is made.

JM = JJ - 1.

CF = (1) mass of cell being split, (2) ratio of computed mass to original mass.

DR = cell width.

## Notes

Cells are divided in half and the new cell quantities are linear (in Lagrangian coordinates) interpolations of the old cell quantities. The cell masses are adjusted from the interpolation such that mass is conserved exactly.

```
      SUBROUTINE EPP(I,J)
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)
     +,W(MCL)
      LEVEL 2,R
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS
      LEVEL 2,TIME
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRD(ML),
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)
      COMMON/BRND/Z(ML),E(ML),VCJ(ML),DWDT(NDW,ML),PCJ(ML),PM(ML),ND(ML)
     +,MSFF
      IF(TMLT(I).LE.0.)GO TO 10
      TM=TMLT(I)+TMC(I)*(VO(I)-V(I))/VO(I)   Kennedy melt law
      IF(TM.GT.T(J))GO TO 10
      SX(J)=0.
      SZ(J)=0.
      RETURN
   10 CONTINUE
      GO TO (1,3,1),IALPH
1     IF(P(J).LT.0.)GO TO 2   No correction for negative pressures
      IF(PLAP(I).LT.YO(I))GO TO 2  No correction if PLAP < 2/3 Y0
      X=P(J)/PLAP(I)              ]  Correction to get on the hydrostat
      P(J)=P(J)-YO(I)*AMIN1(1.,X) ]
2     IF(ABS(SX(J)).LE.YO(I))GO TO 4   On the yield surface?
      SX(J)=SIGN(YO(I),SX(J))
4     P(J)=P(J)+SX(J)   Total stress in the x-direction
      RETURN
3     CONTINUE
      IF(P(J).LT.0.)GO TO 5
      IF(PLAP(I).LT.YO(I))GO TO 5
      X=P(J)/PLAP(I)
      P(J)=P(J)-YO(I)*AMIN1(1.,X)
    5 F=2*(SX(J)*(SX(J)+SZ(J))+SZ(J)*SZ(J))   Yield surface for α = 2 is
      TTY2=1.5*YO(I)*YO(I)                                     different
      IF(F.LT.TTY2)GO TO 4
      FT=SQRT(TTY2/F)
      SX(J)=FT*SX(J)]  Sx and Sz are independent
      SZ(J)=SZ(J)*FT]
      GO TO 4
      RETURN
      END
```

## EPP

An elastic — perfectly plastic model with the von Mises yield model and an optional correction term to put shock data fit equations of state on the hydrostat.

Local Variables

X = P/PLAP. See below for usage.

Notes

In the elastic region, consider the linear stress-strain relation (Hooke's Law),

$$\sigma_{ij} = 2\mu\varepsilon_{ij} + \lambda(\varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33}) \quad . \tag{1}$$

The hydrostat is given by $\varepsilon_{ii} = -\Delta V/3V$ for all i and

$$P_{hydrostat} = -\left(\frac{2\mu}{3} + \lambda\right)\frac{\Delta V}{V} \quad . \tag{2}$$

For one-dimensional compression we have $\varepsilon_{11} = -\Delta V/V$ with all other $\varepsilon_{ii} = 0$. This gives

$$P_{Hugoniot} = \sigma_{11} = -(2\mu + \lambda)\ \frac{\Delta V}{V} \tag{3}$$

and

$$\sigma_{22} = \sigma_{33} = -\lambda\frac{\Delta V}{V} \quad , \tag{4}$$

with

$$\sigma_{11} - \sigma_{22} = -\ 2\mu\ \frac{\Delta V}{V} \quad . \tag{5}$$

The value determined experimentally is $P_{Hugoniot}$ in single shock compression. For strains larger than at the yield point, $\sigma_{11} - \sigma_{22}$ is assumed to remain constant at $Y_0$, the yield stress. The difference between the Hugoniot stress and the hydrostat is

$$P_{Hug} - P_{hydro} = -\frac{4}{3}\mu\frac{\Delta V}{V} \tag{6}$$

in the elastic region. At the yield point, $-\Delta V^Y = Y_0 V/2\mu$ and $P_{Hug}^Y = ((2\mu+\lambda)/2\mu)Y_0$. The difference between $P_{Hug}$ and $P_{hydro}$ can then be written as

$$P_{Hug} - P_{hydro} = \frac{2}{3}Y_0\frac{P_{Hug}}{P_{Hug}^Y} = -\frac{4}{3}\mu\frac{\Delta V}{V} \quad . \tag{7}$$

In a similar manner, we can write for strains larger than at the yield point,

$$P_{Hug} - P_{hydro} = \frac{2}{3}Y_0 \quad . \tag{8}$$

These last two formulas are found to be in satisfactory agreement with experiment for real materials in the region of interest. They are then used to calculate the hydrostat for equations of state fit to Hugoniot data. In the code $P_{Hug}^Y$ is designated PLAP and is input data. For PLAP < $\frac{2}{3}Y_0$, it is assumed that the EOS pressure is the hydrostat.

Next we define the stress deviators by

$$S_i = \sigma_i - P \quad , \tag{9}$$

where

$$P = \frac{\sigma_1 + \sigma_2 + \sigma_3}{3} \tag{10}$$

and

$$\sigma_i = 2\mu\varepsilon_i + \lambda(\varepsilon_1 + \varepsilon_2 + \varepsilon_3) \quad , \tag{11}$$

which is Eq. (1) with a change of notation to take advantage of the fact that $\sigma_{ij}$ is diagonal for the coordinate systems we are considering.

Substitution of Eq. (11) in Eqs. (9) and (10) gives

$$S_i = 2\mu(\varepsilon_i + \frac{\Delta V}{3V}) \quad , \tag{12}$$

with

$$P = -\left(\frac{2\mu}{3} + \lambda\right) \frac{\Delta V}{V} \quad , \tag{13}$$

where we have used

$$-\frac{\Delta V}{V} = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 \quad . \tag{14}$$

The differential form of Eq. (12),

$$\dot{S}_i = 2\mu(\dot{\varepsilon}_i + \frac{\dot{V}}{3V}) \quad , \tag{15}$$

is used in the code to calculate the stress deviators.

The yield stress is calculated using the von Mises yield criteria. This model assumes that yielding occurs when the distortion energy is the same as the distortion energy at yield for simple tension (see, e.g., Mendelson, Plasticity: Theory and Application for further details). This criterion can be written as

$$\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right] = Y_0^2 \quad , \tag{16}$$

which can be rewritten in terms of the stress deviators to give

$$S_1^2 + S_2^2 + S_2^2 = \frac{2}{3} Y_0^2 \quad . \tag{17}$$

For $\alpha = 1$ and $\alpha = 3$, directions 2 and 3 are equivalent so we have

$$S_2 = S_3 = -\frac{1}{2} S_1 \quad , \tag{18}$$

which simplifies Eq. (17) to yield

$$S_1 = \frac{2}{3} Y_0 \quad . \tag{19}$$

Equation 15 need only be evaluated in the 1-direction, which is the x-direction for $\alpha = 1$ and the r-direction (still denoted x in the code) for $\alpha = 3$. The value for $\dot{\varepsilon}_1$ is obtained from

$$\dot{\varepsilon}_1 = - \frac{\partial U}{\partial R} \quad , \tag{20}$$

which is the same for all $\alpha$.

For $\alpha = 2$, none of the directions are equivalent, so we have

$$S_2 = -S_1 - S_3 \quad , \tag{21}$$

which is substituted into Eq. (17) to give

$$f \equiv 2(S_1^2 + S_3^2 + S_1 S_3) = \frac{2}{3} Y_0^2 \quad . \tag{22}$$

The value for $\dot{\varepsilon}_1$ is again obtained from Eq. (20). However, the value for $\dot{\varepsilon}_3$ is zero because the 3-direction, which is the z-direction, does not have any motion in cylindrical geometry. When the yield condition, Eq. (22), is met, the stress deviators in the 1- and 3-directions are scaled to lie on the yield surface.

```
      SUBROUTINE DELT                                                   DELT      2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM     2
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM     3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM     4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM     5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL     2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL     3
     +,W(MCL)                                                           MCELL     4
      LEVEL 2,R                                                         MCELL     5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL     6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL     7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL     8
      LEVEL 2,TIME                                                      MCELL     9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO INIT      2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT      3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)            INIT      4
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                         VD        2
      COMMON/MNMX/KMAX(ML2),KMIN(ML2),NMC                               MN        2
      COMMON/BRNS/A(ML),BR(ML),BA(ML),VBO(ML),VBSW(ML)                  BRN       2
      COMMON/EOSN/IEOS(ML),ME(ML)                                       EN        2
      COMMON/ES/TE(ML2),NME                                             ESM       2
      DATA IDV/O/                                                       DELT     11
      NMCE=IE(NMC)                                                      DELT     12
      IF(JMAX(NMC)-JMIN(NMC).LE.2.AND.NMC.GT.1)NMCE=TE(NMC-1)           DELT     13
      IF(DTO(NMCE).LE.DT)DTF=DTO(NMCE)                                  DELT     14
      IF(NDELT.EQ.1.OR.IBRN(NMC).EQ.3)RETURN                            DELT     15
      DTC=DTD(NMCE)                                                     DELT     16
      DLT=DTO(NMCE)                                                     DELT     17
      JI=0                                                              DELT     18
      DO 10 I=1,NMC                                                     DELT     19
      II=IE(I)                                                          DELT     20
      JMN=JMIN(I)                                                       DELT     21
      JMX=JMAX(I)                                                       DELT     22
      VM=V(JMN)                                                         DELT     23
      DRM=R(JMN)-R(JMN+1)                                               DELT     24
      DRV=DRM                                                           DELT     25
      JJ=JMN                                                            DELT     26
      JR=JMN                                                            DELT     27
      JMN=JMN+1                                                         DELT     28
      DO 11 J=JMN,JMX            Find the smallest ΔR in the region     DELT     29
      DR=R(J)-R(J+1)                                                    DELT     30
      IF(DR.GT.DRM)GO TO 12                                             DELT     31
      JR=J                                                              DELT     32
      DRM=DR                                                            DELT     33
   12 CONTINUE                                                          DELT     34
      IF(V(J).GE.VM)GO TO 11                                            DELT     35
      JJ=J                                                              DELT     36
      VM=V(J)                    Find the smallest V in the region      DELT     37
      DRV=DR                                                            DELT     38
   11 CONTINUE                                                          DELT     39
      DTC=DTCF(NMCF)*DRM/C(II,JR)   Δt for the smallest R               DELT     40
      IF(DTC.LT.DLT)DLT=DTC    Is it smaller than DLT?                  DELT     41
      IF(JR.EQ.JJ)GO TO 10    Don't check the same cell twice           DELT     42
      DTC=DTCF(NMCE)*DRV/C(II,JJ)   Δt for the smallest V               DELT     43
      IF(DTC.LT.DLT)DLT=DTC   Is it smaller than DLT?                   DELT     44
   10 CONTINUE                                                          DELT     45
      J=KMAX(NM)+1                                                      DELT     46
      IF(IA.EQ.0)GO TO 20          Check for last cell about to hit the origin   DELT     47
      IF(R(J).LE.1.E-10)GO TO 20    (α = 2 or 3) and whether the time step is    DELT     48
      IF(U(J).GE.O.)GO TO 20        small enough                                 DELT     49
```

```
      DTV=-R(J)/U(J)
      IF(DTV.GT.5*DT)GO TO 20
      DLTV=-.05*(R(J-1)-R(J))/U(J-1)
      IF(DLTV.GT.DLT)GO TO 20
      DT1=DLTV
      IDV=101
   20 CONTINUE
      IF(NNV.LE.0) GO TO 40
      DO 30 I=1,NMC
      IF(IV(I).NE.0) GO TO 30
      J=JV(I)
      DRV=R(J)-R(J+1)
      DU=U(J)-U(J+1)
      IF(DU.GT.-0.01)GO TO 30
      DTV=-DRV/DU                          Check for void about to collapse
      IF(DTV.GT.5*DT)GO TO 30
      DRM=R(J-1)-R(J)
      DRP=R(J+1)-R(J+2)
      DRT=AMIN1(DRP,DRM)
      DTV=0.05*DRT/ABS(DU)
      IF(DTV.GT.DLTV)GO TO 30
      IDV=101
      DLTV=DTV
      DT1=DLTV
   30 CONTINUE
   40 CONTINUE
      IDV=IDV-1
      IF(DLT.LT.DTD(NMCE))DTF=DLT
      IF(IDV.GE.1)DTF=DT1                  Pick the smallest time step
      IF(DLT.LT.DTF)DTF=DLT
      DTR=DTF/DT
      IF(DTR.LT.0.8.OR.DTR.GT.1.2)DT=DTF   Don't make small changes in Δt
      RETURN
      END
```

## DELT

Calculates the time step to be used. The time step may be input data or may be evaluated from several criteria in order to keep the problem numerically stable.

Local Variables

IDV = index used to count the number of time steps to use the void collapse time step.

NMCE = the region from which the maximum allowed time step is obtained.

DTC = $\Delta t$ calculated from the sound speed.

DLT = variable used to keep track of the lowest calculated $\Delta t$.

II = IE(I) = original region #.

JMN = JMIN(I).

JMX = JMAX(I).

VM = variable used to find the smallest volume in the region.

DRM = variable used to find the smallest $\Delta r$ in the region.

JJ = cell # of cell with V = VM.

JR = cell # of cell with $\Delta r$ = DRM.

J = last cell # in the problem/void cell #.

DTV = time step required for cell with $\Delta r$ = DRT to collapse 5% when void collapses - also, approximate time required for the inner surface of the problem to reach the origin.

DLTV = time step required to collapse the last cell 5% when it hits the origin.

DT1 = DLTV when small enough to be needed = collapse time step.

DRV = width of void.

DU = relative velocity at which a void is closing.

DTV = approximate time required for a void to close.

DRM = $\Delta r$ for cell that is outside bound of a void.

DRP = $\Delta r$ for a cell that is inside bound of a void.

DRT = smaller of the two.

DTR = ratio of computed time step to the current time step.

Notes

Several methods are used to compute the time step. For a sharp shock burn, $\Delta t$ is fixed in SSB (q.v.) and is not changed by DELT. Each region can be assigned a maximum time step, DT0, which is in effect when that region is the last active region. For NDELT = 1 in the INP namelist, $\Delta t$ is set to DT0 for the last active region. The time step can also be calculated from a constant times the time it takes a sound wave to cross a cell at the local sound speed. That is, $\Delta t = a(\Delta r/c)$, where a is a constant (DTCF in the code with default value of 1/2), $\Delta r$ is the cell width, and c is the sound speed in that cell. This check is made for the densest cell and the cell with the smallest width for each material. (See Chap. 12 of Richtmyer and Morton, Differential Methods for Initial-Value Problems, 2nd Ed., for a discussion of the stability criteria for the difference equations.)

In converging geometry, for the inside surface about to collapse at the origin, the time step is temporarily (100 cycles) required to be no greater than 1/20 of the time required for the outside of the innermost cell to reach the origin at its current velocity. The same criteria is used for collapsing voids, except that the relative velocity of the two surfaces is used along with the smaller of the two cell widths for the cells touching the void.

```
      FUNCTION C(I,J)                                                   C          2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL      3
     +,W(MCL)                                                           MCELL      4
      LEVEL 2,R                                                         MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL      8
      LEVEL 2,TIME                                                      MCELL      9
      COMMON/EOSN/IEOS(ML),ME(ML)                                       EN         2
      IS=IEOS(I)                                                        C          6
      VJ=V(J)                                                           C          7
      PJ=P(J)                                                           C          8
      XIJ=XI(J)                                                         C          9
      GO TO (1,2,3,4),IS    Pick sound speed subroutine according to EOS type  C  10
    1 CALL CUSUP(VJ,PJ,CC,I)                                            C         11
      C=CC                                                              C         12
      RETURN                                                            C         13
    2 CALL CBLDUP(I,J,CC)                                               C         14
      C=CC                                                              C         15
      RETURN                                                            C         16
    3 CALL CPOLY(VJ,PJ,XIJ,CC,I)                                        C         17
      C=CC                                                              C         18
      RETURN                                                            C         19
    4 CALL CSES(VJ,XIJ,CC,I,J)                                          C         20
      C=CC                                                              C         21
      RETURN                                                            C         22
      END                                                               C         23
```

<u>C</u>

Switching function subroutine to pick the appropriate sound speed subroutine.

<u>Local Variables</u>

IS = IEOS(I) = equation-of-state type.

VJ = V(J).

PJ = P(J).

XIJ = XI(J).

CC = sound speed.

C = sound speed.

```
      SUBROUTINE CUSUP(VC,PC,C,I)                                             CUSUP      2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,               PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                        PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTA8=1,NTA8=MTA8*3742                          PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                              PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                               MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)         MCELL      3
     +,W(MCL)                                                                 MCELL      4
      LEVEL 2,R                                                               MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,       MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                          MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                        MCELL      8
      LEVEL 2,TIME                                                            MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO       INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),      INIT       3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TPLT(ML),TMC(ML)                  INIT       4
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),               US         2
     +GAMMA(ML),ALP(ML)                                                       US         3
      COMMON/BRNS/A(ML),BR(ML),BA(ML),VBD(ML),VBSW(ML)                        BRN        2
      IF(IBRN(I).NE.0)GO TO 40                                                CUSUP      8
      IF(VC.GT.SWV(I)) GO TO 11                                               CUSUP      9
   12 C=C2(I)                                                                 CUSUP     10
      S=S2(I)                   Pick appropriate USUP constants               CUSUP     11
      GO TO 20                                                                CUSUP     12
   11 C=C1(I)                                                                 CUSUP     13
      S=S1(I)                                                                 CUSUP     14
   20 ETA=(VO(I)-VC)/VO(I)                                                    CUSUP     15
      IF(PC.LE.0.) GO TO 30   If in tension, use C from USUP constants        CUSUP     16
      VV=VC/VO(I)                                                             CUSUP     17
      HP=(C/(1-S*ETA))**2*ETA/VO(I)  Hugoniot pressure                        CUSUP     18
      CSQ=(VV*C)**2*(1+S*ETA)*(1-GAMMA(I)*ETA/(2*VV))                         CUSUP     19
     +/(1-S*ETA)**3+GAMMA(I)*VO(I)*HP/2+VC*(PC-HP)*(GAMMA(I)+1)   C^2         CUSUP     20
      IF(CSQ .LE. 0) GO TO 30  If C^2 < 0, use C from USUP constants          CUSUP     21
      C=SQRT(CSQ)                                                             CUSUP     22
   30 CONTINUE                                                                CUSUP     23
      RETURN                                                                  CUSUP     24
   40 C=1.   Used for HE's                                                    CUSUP     25
      RETURN                                                                  CUSUP     26
      END                                                                     CUSUP     27
```

CUSUP(VC,PC,C,I)

Calculates the sound speed for a USUP EOS with constant Grüneisen $\gamma$.

Local Variables

VC = specific volume, V, for which C is to be calculated (subroutine argument).

PC = pressure, P, for which C is to be calculated (subroutine argument).

C = sound speed = output of subroutine (subroutine argument).

I = initial region number from which to get EOS constants (subroutine argument).

S = S in USUP fit $U_S = C + SU_P$

ETA = $\eta = \dfrac{V_0 - V}{V_0}$ .

VV = $\dfrac{V}{V_0} = 1 - \eta$ .

HP = Hugoniot pressure at specific volume V.

CSQ = $C^2$.

Notes

The sound velocity C is given by

$$C^2 = \left(\frac{\partial P}{\partial \rho}\right)_S = -V^2\left(\frac{\partial P}{\partial V}\right)_S \quad , \tag{1}$$

which can be rewritten as

$$C^2 = -V^2\left[\left(\frac{\partial P}{\partial I}\right)_V\left(\frac{\partial I}{\partial V}\right)_S + \left(\frac{\partial P}{\partial V}\right)_I\left(\frac{\partial V}{\partial V}\right)_S\right] = V^2\left[P\left(\frac{\partial P}{\partial I}\right)_V - \left(\frac{\partial P}{\partial V}\right)_I\right] . \tag{2}$$

The Grüneisen EOS is given by

$$P = P_H + \frac{\gamma}{V}(I - I_H) \quad , \tag{3}$$

where

$$P_H = \rho_0 C_0^2 \eta/(1 - S\eta)^2 \quad , \tag{4}$$

$$I_H = \frac{P_H \eta V_0}{2} \quad , \tag{5}$$

with $P_0$ and $I_0$ ignored, and

$$\eta = \frac{V_0 - V}{V_0} \quad . \tag{6}$$

Since $P_H$ is a function of volume only, the first term in Eq. (2) is readily evaluated using Eq. (3) to give

$$P\left(\frac{\partial P}{\partial I}\right)_V = P\,\frac{\gamma}{V} \quad , \tag{7}$$

where $\gamma$ is assumed to be constant.

The second term in Eq. (2) is more readily evaluated if we rewrite Eq. (3) in the form

$$P = P_H\left(1 - \frac{\gamma\eta V_0}{2V}\right) + \frac{I\gamma}{V} \quad . \tag{8}$$

We then have

$$\left(\frac{\partial P}{\partial V}\right)_I = \frac{\partial P_H}{\partial V}\left(1 - \frac{\eta\gamma V_0}{2V}\right) + \frac{P_H \gamma V_0}{2V^2} - \frac{I\gamma}{V^2} \quad , \tag{9}$$

where

$$\frac{\partial P_H}{\partial V} = \frac{\partial P_H}{\partial \eta}\,\frac{\partial \eta}{\partial V} = -\left(\frac{C_0}{V_0}\right)^2\left(\frac{1 + S\eta}{(1 - S\eta)^3}\right) \tag{10}$$

and I can be written from Eq. (8) as

$$I = \frac{V}{\gamma}\left[P - P_H\left(1 - \frac{\eta\gamma V_0}{V}\right)\right] \quad . \tag{11}$$

Substitution of Eqs. (6), (7), (9), (10), and (11) in Eq. (2) yields

$$C^2 = V^2\left\{\frac{P\gamma}{V} - \frac{\partial P_H}{\partial V}\left(1 - \frac{\eta\gamma V_0}{2V}\right) - P_H\,\frac{\gamma V_0}{2V^2} + \frac{1}{V}\left[P - P_H\left(1 - \frac{\eta\gamma V_0}{V}\right)\right]\right\}$$

$$= \left(\frac{C_0 V}{V_0}\right)^2 \frac{(1 + S\eta)}{(1 - S\eta)^3}\left(1 - \frac{\eta\gamma V_0}{2V}\right) + (P - P_H)(\gamma + 1) + P_H\,\frac{\gamma V_0}{2} \quad . \tag{12}$$

If $C^2 < 0$ or $P < 0$, then C is set to $C_0$, the sound speed in the uncompressed solid. The value of $C_0$ and S that is used is allowed to be one of two sets of values depending on the volume. See USUP for more details.

```
      SUBROUTINE CBLDUP(I,J,CC)                                          CBLDUP     2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,          PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                   PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                     PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                         PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                          MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)    MCELL      3
     +,W(MCL)                                                            MCELL      4
      LEVEL 2,R                                                          MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,  MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                     MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                   MCELL      8
      LEVEL 2,TIME                                                       MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO  INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML), INIT       3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)             INIT       4
      COMMON/BUX/BUA,BUB,BUMAX,BUDV(ML)                                  BUP        2
     +,BUR,BUD                                                           BUP        3
      COMMON/ES/TE(ML2),NME                                              ESM        2
      JMN=JMIN(I)                                                        CBLDUP     8
      JMX=JMAX(I)                                                        CBLDUP     9
      II=IE(I)                                                           CBLDUP    10
      IF(W(J).GT.0.99)GO TO 10                                           CBLDUP    11
      IF(W(JMX).GT.0.99.OR.W(JMN).GT.0.99)GO TO 10  If still burning, use D CBLDUP 12
      B=(T(J)-.66)/(T(J)*(T(J)-2.32))   β                                CBLDUP    13
      B=(B+1)/B                                                          CBLDUP    14
      VV=T(J)*VO(II)/(V(J)*(T(J)+1))                                     CBLDUP    15
      PI=(ROW(II)*BUDV(II)*VV**T(J))/(T(J)+1)    Reference pressure      CBLDUP    16
      CC=SQRT((B*(P(J)/(1-W(J))-PI)+T(J)*PI)*V(J))   C                   CBLDUP    17
      IF(CC.LT.0.2)CC=0.2   Lower limit on C                             CBLDUP    18
      RETURN                                                             CBLDUP    19
   10 CONTINUE                                                           CBLDUP    20
      CC=SQRT(BUDV(II))   Use the detonation velocity for C              CBLDUP    21
      RETURN                                                             CBLDUP    22
      END                                                                CBLDUP    23
```

## CBLDUP

Calculates the sound speed for a buildup EOS in cell J.

Local Variables

II = IE(I) = original region #.

JMN = JMIN(I).

JMX = JMAX(I).

$B = \dfrac{\beta + 1}{\beta}$ .

VV = $V_{CJ}/V$.

PI = $P_i$ = reference pressure.

CC = sound speed.

Notes

The buildup EOS (see BLDUP for details) is given by

$$P = \left[\frac{1}{\beta V}\left(I - K - \frac{P_i V}{\gamma - 1}\right) + P_i\right](1 - W) \quad , \tag{1}$$

where $P_i$ is the reference pressure,

$$P_i = P_{CJ} * \left(\frac{V_{CJ}}{V}\right)^{\gamma} = \frac{\rho_0 D^2}{\gamma + 1}\left(\frac{\gamma V_0}{(\gamma + 1)V}\right)^{\gamma} \quad , \tag{2}$$

and $\beta$ is given by

$$\beta = \frac{\gamma - 0.66}{\gamma(\gamma - 2.32)} \quad , \tag{3}$$

and K is a constant for fixed $\gamma$ (note $\gamma$ is stored in T(J)).

The square of the sound speed, $C^2$, is (see CUSUP)

$$C^2 = V^2\left[P\left(\frac{\partial P}{\partial I}\right)_V - \left(\frac{\partial P}{\partial V}\right)_I\right] \quad . \tag{4}$$

Combining Eq. (1) and Eq. (4) we find

$$C^2 = \left(\frac{\beta + 1}{\beta^2}\right)\left(I - K - \frac{P_i V}{\gamma - 1}\right) + \gamma P_i V = \left(\frac{\beta + 1}{\beta}\right)\left(\frac{P}{1 - W} - P_i\right)V + \gamma P_i V \quad . \qquad (5)$$

Now, if W(J) is too near 1, there is the possibility of numerical problems. Also, the calculated sound speed will be small. Since the buildup burn model is designed for prompt detonation, the detonation velocity is used for the sound speed when W(J) > 0.99. It is also used if either the innermost or outermost cell of the region has W > 0.99. We have arbitrarily set the lower limit of the sound speed at 0.2 cm/μs.

```
      SUBROUTINE CPOLY(VC,PC,XC,C,I)                                         CPOLY      2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,              PARAM      2
     +NUMV=10,MOL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                       PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                         PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                              PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                              MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)        MCELL      3
     +,W(MCL)                                                                MCELL      4
      LEVEL 2,R                                                              MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,      MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                         MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                       MCELL      8
      LEVEL 2,TIME                                                           MCELL      9
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO      INIT       2
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),     INIT       3
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)                 INIT       4
      COMMON/POLYC/CF(NCF,ML),PS(ML)                                         PLC        2
      VP=VO(I)/VC                                                            CPOLY      7
      VV=VP-1.      η                                                        CPOLY      8
      VP=-VP/VC      η'                                                      CPOLY      9
      A=VV*(CF(1,I)+CF(2,I)*ABS(VV))    A                                    CPOLY     10
      B=CF(3,I)+VV*(CF(4,I)+VV*CF(5,I))   B                                  CPOLY     11
      C=CF(6,I)+CF(7,I)*VV   C                                               CPOLY     12
      AP=VP*(CF(1,I)+2*CF(2,I)*ABS(VV))    A'                                CPOLY     13
      BP=VP*(CF(4,I)+2*CF(5,I)*ABS(VV))    B'                                CPOLY     14
      CP=VP*CF(7,I)   C'                                                     CPOLY     15
      XIC=XC*ROW(I)    ε                                                     CPOLY     16
      DPDV=((PC*ROW(I)*(B+2*XIC*C)-(A+XIC*(B+XIC*C))/(XIC+CF(8,I)))          CPOLY     17
     +-(AP+XIC*(BP+XIC*CP)))/(XIC+CF(8,I))                                   CPOLY     18
      IF(DPDV.LT.0.)DPDV=CF(1,I)/CF(8,I)  -(∂P/∂V)s                          CPOLY     19
      C=VC*SQRT(DPDV)    C                                                   CPOLY     20
      RETURN                                                                 CPOLY     21
      END                                                                    CPOLY     22
```

## CPOLY(VC,PC,XC,C,I)

Calculates the sound speed at specific volume VC, pressure PC, and specific internal energy XC for the eight-parameter fit EOS in subroutine POLY.

Local Variables

$VP = \frac{\partial \eta}{\partial V}$ .

$VV = \frac{V_0 - V}{V} = \eta$.

A,B,C = see notes.

$AP = \frac{\partial A}{\partial V}$ .

$BP = \frac{\partial B}{\partial V}$ .

$CP = \frac{\partial C}{\partial V}$ .

$XIC = \rho_0 I = \varepsilon$ .

$DPDV = \left.\frac{\partial P}{\partial V}\right)_S$ .

C = sound speed, $C^2 = -V^2\left(\frac{\partial P}{\partial V}\right)_S$ .

Notes

As shown in CUSUP, the square of the sound velocity can be written

$$C^2 = -V^2\left(\frac{\partial P}{\partial V}\right)_S = V^2\left[P\left(\frac{\partial P}{\partial I}\right)_V - \left(\frac{\partial P}{\partial V}\right)_I\right] . \tag{1}$$

The POLY EOS is

$$P = \frac{A + B\varepsilon + C\varepsilon^2}{\varepsilon + C_8} , \tag{2}$$

where

$$A = \eta C_1 + |\eta|\eta C_2 , \tag{3}$$

$$B = C_3 + \eta C_4 + \eta^2 C_5 , \tag{4}$$

and

$$C = C_6 + \eta C_7 \quad , \tag{5}$$

with

$$\eta = \frac{V_0 - V}{V} \tag{6}$$

and

$$\varepsilon = \rho_0 I \quad . \tag{7}$$

The bracket is readily evaluated to give

$$-\left(\frac{\partial P}{\partial V}\right)_S = P\rho_0\left(\frac{B + 2\varepsilon C}{\varepsilon + C_8} - \frac{A + B\varepsilon + C\varepsilon^2}{(\varepsilon + C_8)^2}\right) - \frac{A' + \varepsilon B' + \varepsilon^2 C'}{\varepsilon + C_8} \quad , \tag{8}$$

where

$$A' = \frac{\partial A}{\partial V} = \frac{\partial A}{\partial \eta}\frac{\partial \eta}{\partial V} = (C_1 + 2C_2|\eta|)\eta' \quad , \tag{9}$$

$$B' = \frac{\partial B}{\partial \eta}\frac{\partial \eta}{\partial V} = (C_4 + \eta C_5)\eta' \tag{10}$$

$$C' = \frac{\partial C}{\partial \eta}\eta' = C_7\eta' \quad , \tag{11}$$

with

$$\eta' = \frac{\partial \eta}{\partial V} = -\frac{V_0}{V^2} \quad . \tag{12}$$

A check is made whether $-\left(\frac{\partial P}{\partial V}\right)_S$ is positive. If it is not, the $\eta = 0$, $I = 0$ value (initial conditions) is used, which is

$$-\left(\frac{\partial P}{\partial V}\right)_S^0 = -\frac{A'(\eta=0)}{C_8} = \frac{C_1}{C_8 V_0} \quad . \tag{13}$$

The sound velocity is then

$$c = V\sqrt{-\left(\frac{\partial P}{\partial V}\right)_S} \quad . \tag{14}$$

```
      SUBROUTINE CSES(VJ,XIJ,CC,I,J)                                        CSES       2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,             PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                      PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                        PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                            PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                             MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)       MCELL      3
     +,W(MCL)                                                               MCELL      4
      LEVEL 2,R                                                             MCELL      5
      COMMON/DVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,     MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                        MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                      MCELL      8
      LEVEL 2,TIME                                                          MCELL      9
      COMMON/ES/IE(ML2),NME                                                 ESM        2
      COMMON/SESIN/II,IDT,RPT4,XIPT4,IBR,IFL                                CSES       7
      COMMON/SESOUT/PPT4(3),TPT4(3)                                         CSES       8
      CC=.5    Default value                                                CSES       9
      IBR=1    Output PPT4 only                                             CSES      10
      IDT=1                                                                 CSES      11
      II=IE(I)                  ]                                           CSES      12
      XIPT4=XIJ                 ]  Set up input                             CSES      13
      RPT4=1./VJ                ]                                           CSES      14
      IFL=MOD(IFLAG(J),64)      ]                                           CSES      15
      CALL T4EOSA   Call SESAME                                             CSES      16
      C2=VJ*VJ*PPT4(1)*PPT4(3)+PPT4(2)   C^2                                CSES      17
      IF(C2.GT.0.)CC=SQRT(C2)   C                                           CSES      18
      RETURN                                                                CSES      20
      END                                                                   CSES      21
```

## CSES

Calculates the sound speed for a Sesame EOS.

### Local Variables

CC = sound speed = C.

C2 = $C^2$.

### Notes

The output of the Sesame EOS call in PPT4 is

$$
\begin{aligned}
PPT4(1) &= P \quad , \\
PPT4(2) &= \left(\frac{\partial P}{\partial \rho}\right)_I \quad , \qquad (1) \\
PPT4(3) &= \left(\frac{\partial P}{\partial I}\right)_\rho \quad .
\end{aligned}
$$

The sound speed is given by

$$
C^2 = V^2\left[P\left(\frac{\partial P}{\partial I}\right)_V - \left(\frac{\partial P}{\partial V}\right)_I\right] = V^2 P\left(\frac{\partial P}{\partial I}\right)_\rho + \left(\frac{\partial P}{\partial \rho}\right)_I \quad , \qquad (2)
$$

which in terms of the PPT4 array is

$$
C^2 = V^2 * PPT4(1) * PPT4(3) + PPT4(2) \quad . \qquad (3)
$$

If the calculated $C^2$ is negative, then the default value of C = 1/2 is used.

```
      SUBROUTINE RLEOS(I)                                               RLEOS      2
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,         PARAM      2
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,                  PARAM      3
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742                    PARAM      4
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)                        PARAM      5
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),                         MCELL      2
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)   MCELL      3
     +,W(MCL)                                                           MCELL      4
      LEVEL 2,R                                                         MCELL      5
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM, MCELL      6
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK                    MCELL      7
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS                  MCELL      8
      LEVEL 2,TIME                                                      MCELL      9
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV                         VD         2
      COMMON/RLC/RC(ML),RP(ML),RLV(ML),PH1,DV1,DV2                      RLC        2
      JP=JV(I)+1                                                        RLEOS      7
      IF(V(JP-2).LT.RLV(I).OR.V(JP).LT.RLV(I+1))GO TO 10                RLEOS      8
      JM=JP-2                                                           RLEOS      9
      Q(JP)=0.                                                          RLEOS     10
      Q(JM)=0.                                                          RLEOS     11
      P(JP)=V(JP)*RC(I+1)+RP(I+1)]   Rayleigh line EOS                  RLEOS     12
      P(JM)=V(JM)*RC(I)+RP(I)    ]                                      RLEOS     13
      RETURN                                                            RLEOS     14
   10 CONTINUE                                                          RLEOS     15
      IV(I)=1   Go back to normal EOS                                   RLEOS     16
      RETURN                                                            RLEOS     17
      END                                                               RLEOS     18
```

# RLEOS(I)

The Rayleigh line in P-V space is used as an equation of state for the initial compression of the two cells touching an interface that has just become a closed void when the relative velocity of the two surfaces was large.

## Local Variables

JP = JV(I) + 1 = cell # of inside cell touching the interface.

JM = JP - 2 = cell # of outside cell touching the interface.

## Notes

The Rayleigh line is a straight line in P-V space that passes through the initial state $(P_0,V_0)$ and the final state $(P_f,V_f)$ for a shock. That is

$$P = P_0 + (P_f - P_0)\left(\frac{V_0 - V}{V_0 - V_f}\right) \quad . \tag{1}$$

The change in energy across the shock front using the Rayleigh line EOS is

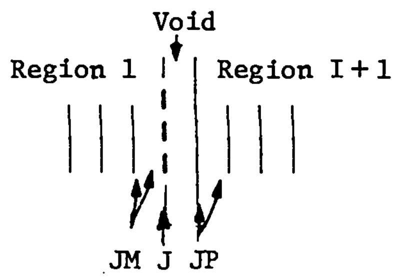
$$\Delta I = -\int_{V_0}^{V_f} P dV = P_0(V_0 - V_f) + \frac{(P_f - P_0)(V_0 - V_f)}{2} = \frac{(P_f + P_0)(V_0 - V_f)}{2} \quad , \tag{2}$$

which is equal to the energy change from the jump conditions. That is, without artificial viscosity, energy is conserved across the shock front using the Rayleigh line EOS. $P_f$ and $V_f$ are determined in RL using the jump conditions and equations of state for the two materials that collide. The RLEOS is used until one of the materials reaches a preset specific volume. RLV(I), at which time a flag is set to switch back to the normal EOS with artificial viscosity. Since the materials may be different, $V_f$ can be different and the RLEOS would then be different. By the time the RLEOS is no longer used, the artificial viscosity is no longer so large that excessive energy is dumped in the cell before the difference equations have time to respond.

```
      SUBROUTINE RL(I)
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)
     +,W(MCL)
      LEVEL 2,R
      COMMON/DVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UT,UF,UII,UFI,NADD,NM,
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS
      LEVEL 2,TIME
      COMMON/RLC/RC(ML),RP(ML),RLV(ML),PH1,DV1,DV2
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV
      JP=JV(I)+1
      J=JP-1
      JM=JP-2
      DU=U(J)-U(JP)
      UV2=U(J)-DU/2           Initial guesses for interface velocity
      GB=G(UV2,I)
      UV3=U(J)-DU*.45
      IC=0
   10 CONTINUE
      IC=IC+1
      UV1=UV2
      UV2=UV3
      IF(IC.GT.100)GO TO 11
      GA=GB                                    Iterate to find the velocity
      GB=G(UV3,I)                               at which Hugoniot pressures
      UV3=(GB*UV1-GA*UV2)/(GB-GA)               match
      IF(UV3.LT.U(J).OR.UV3.GT.U(JP))GO TO 12
      IF(ABS((UV3-UV2)/UV3).GT.0.001)GO TO 10
      RC(I)=(P(JM)-PH1)/DV1
      RC(I+1)=(P(JP)-PH1)/DV2
      RP(I)=P(JM)-RC(I)*V(JM)
      RP(I+1)=P(JP)-RC(I+1)*V(JP)
      DR=R(J)-R(JP)
      A=(UV3-U(JP))/(U(J)-U(JP))
      RV=A*R(J)+(1-A)*R(JP)              Set the Rayleigh line EOS parameters
      U(J)=UV3                            and the interface velocity
      U(JP)=U(J)
      R(J)=RV
      R(JP)=RV
      RLV(I)=V(JM)-0.9*DV1
      RLV(I+1)=V(JP)-0.9*DV2
      RETURN
   12 CONTINUE
      UV3=U(JP)-DU*.01
      UV2=U(J)+DU*.01          Fixup attempt if the iteration
      GB=G(UV2,I)               gets out of range
      GO TO 10
   11 CONTINUE
      PRINT 1,TIME,I,J
    1 FORMAT(18H RL ERROR AT TIME=,E13.4,4H I=,I3,4H J=,I4)
      DR=R(J)-R(JP)
      R(J)=R(J)+U(J)*DR/DU         Iteration failed:
      R(JP)=R(J)                    try the low-velocity method
      U(J)=(U(J)+U(JP))/2
      U(JP)=U(J)
      IV(I)=1
      RETURN
      END
```

## RL(I)

Calculates parameters for the Rayleigh line EOS. This primarily consists of iteration to find the interface velocity which sends shock waves into both materials with the same final pressure.




### Local Variables

JM,J,JP - see illustration.

After the collision, R(J) = R(JP) is the interface, U(J) = U(JP), and the void is closed.

DU = relative velocity of the two free surfaces.

UV1,UV2,UV3 = $U_{i-2}, U_{i-1}, U_i$. See notes.

GA,GB = $g_i, g_{i+1}$. See notes.

IC = count of # of iterations (only 100 are allowed).

DR = cell width.

RV = position of the interface after collision.

DTV = time the void has actually been closed.

### Notes

The jump conditions (see USUP) can be combined to give the equations

$$I - I_0 = \frac{(P + P_0)(V_0 - V)}{2} \tag{1}$$

and

$$U_P = \sqrt{(P - P_0)(V_0 - V)} \quad , \tag{2}$$

where the subscript indicates initial quantities in the unshocked material and $U_P$ is the particle velocity of the shocked material relative to the particle velocity of the unshocked material. Combining Eqs. (1) and (2), we have

$$I - I_0 = \frac{U_P^2}{2} + P_0(V_0 - V) \quad . \tag{3}$$

Now the shocks we are considering are from the collision of two free surfaces. At the free surface $P_0 = 0$, but $P_0$ can be nonzero at the center of a cell touching the collision interface. However, $P_0$ will be much less than P for the high-velocity collisions we are considering and will be set to zero in Eq. (3).

For the collision we are considering, the interface particle velocity, U, is the same for both materials. (We will designate the region outside the interface by superscript 1 and the inside region by 2.) By our choice of coordinates we will have the relation

$$U_0^{(2)} > U > U_0^{(1)} \quad , \tag{4}$$

where $U_0^{(1)}$ and $U_0^{(2)}$ are the free-surface velocities of regions 1 and 2 just before the collision. The corresponding particle velocities to be used with the jump conditions are

$$U_P^{(1)} = U - U_0^{(1)} \tag{5}$$

and

$$U_P^{(2)} = U_0^{(2)} - U \quad . \tag{6}$$

What we need to find is a value of U such that the Hugoniot pressure is the same on both sides of the interface. For a given value of U, we have a fixed value of I for each region from Eqs. (3), (5), and (6) (with $P_0 = 0$). Given I, V can be varied until a point on the Hugoniot (using Eq. (1) and the EOS) is found. We then define a function g(U)to be the difference in Hugoniot pressures for regions 1 and 2 when the interface velocity is U. The zero of g(U) is found using the secant method,

$$U_{i+1} = U_i - \frac{(U_i - U_{i-1})}{(g_i - g_{i-1})} g_i = \frac{U_{i-1}g_i - U_i g_{i-1}}{g_i - g_{i-1}} \quad . \tag{7}$$

Two initial values are set as

$$U_0 = U_0^{(1)} + \frac{1}{2}\left(U_0^{(2)} - U_0^{(1)}\right) \tag{8}$$

and

$$U_1 = U_0^{(1)} + .45\left(U_0^{(2)} - U_0^{(1)}\right) \quad , \tag{9}$$

where $U_0$ is the correct value if the two materials are the same and in the same state. If the iteration goes out of the range in Eq. (4), then new initial values

$$U_0 = U_0^{(1)} \tag{10}$$

and

$$U_1 = U_0^{(2)} \tag{11}$$

are tried. The iteration is allowed to proceed for 100 steps. If a solution has not been found, then an error message is printed and the low-velocity collision procedure is used.

When a solution for U is found, the parameters $R_C$ and $R_P$ for RLEOS are set. The Rayleigh line EOS can be written as

$$P = P_0 + (P_f - P_0) \frac{V_0 - V}{V_0 - V_f} = VR_C + R_P \quad , \tag{12}$$

where

$$R_C = \frac{P_0 - P_f}{V_0 - V_f} \tag{13}$$

and

$$R_P = P_0 + \frac{(P_f - P_0)V_0}{V_0 - V_f} = P_0 - R_C V_0 \quad . \tag{14}$$

A third parameter, $R_\ell$, is the volume at which a flag is set in RLEOS to switch back to the normal EOS plus viscosity. It is given by

$$R_\ell = V_0 - 0.9(V_0 - V_f) \quad . \tag{15}$$

The interface velocity, U(J) = U(JP), is set to the solution value for U. The position of the interface, R(J) = R(JP), is found by a linear extrapolation back to the point of collision followed by uniform motion at velocity U.

```
 FUNCTION G(UV,I)
 PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
+NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
+MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
+,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
 COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),
+P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)
+,W(MCL)
 LEVEL 2,R
 COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,
+IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK
 COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS
 LEVEL 2,TIME
 COMMON/RLC/RC(ML),RP(ML),RLV(ML),PH1,DV1,DV2
 COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV
 COMMON/EOSN/IEOS(ML),ME(ML)
 J=JV(I)
 JP=J+1
 JM=J-1     Given the interface velocity, calculate:
 XI1=0.5*(UV-U(J))**2+XI(JM)   Internal energies
 XI2=0.5*(U(JP)-UV)**2+XI(JP)
 PH1=PH(XI1,I,JM)
 IP=I+1                        Hugoniot pressures
 PH2=PH(XI2,IP,JP)
 DV1=2*XI1/PH1                 Volume changes
 DV2=2*XI2/PH2
 G=PH1-PH2                     Pressure difference (= iteration function)
 RETURN
 END
```

## G(UV,I)

Given a value for the interface particle velocity, UV, the difference in the corresponding Hugoniot pressures of the two bounding cells is calculated.

Local Variables

JM,J,JP - see RL.

XI1 = specific internal energy for region 1. (See RL for definition of regions.)

XI2 - specific internal energy for region 2.

IP = I+1 = region # for region 2. (I = region # for region 1.)

PH2 = Hugoniot pressure for region 2.

Notes

We will reproduce here Eqs. (3), (5), and (6) from RL (with $P_0$ set to zero).

$$I = I_0 + \frac{U_P^2}{2} \quad . \tag{1}$$

$$U_P^{(1)} = U - U_0^{(1)} \quad . \tag{2}$$

$$U_P^{(2)} = U_0^{(2)} - U \quad . \tag{3}$$

These equations give the specific internal energy on the Hugoniot $I_H^{(1)}$ and $I_H^{(2)}$ for regions 1 and 2 consistent with an interface velocity U. The Hugoniot pressures $P_H^{(i)}$ for those energies are calculated in PH. The corresponding volume changes are calculated using

$$I - I_0 = \frac{(P + P_0)(V_0 - V)}{2} \quad , \tag{4}$$

and the function g is given by

$$g(U) = P_H^{(1)} - P_H^{(2)} \quad . \tag{5}$$

```
      FUNCTION PH(XIH,I,J)
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      COMMON/CELL/R(MCL),U(MCL),V(MCL),XI(MCL),
     +P(MCL),SX(MCL),SZ(MCL),EE(MCL),T(MCL),Q(MCL),XM(MCL),IFLAG(MCL)
     +,W(MCL)
      LEVEL 2,R
      COMMON/OVL/NDF,NI,NP,NG,TEND,TP(ML),TG(ML),UI,UF,UII,UFI,NADD,NM,
     +IALPH,NDELT,LABEL(8),NDUMP,IDMP,NM1,TD(ML),IJK
      COMMON/MISC/TIME,ICYCL,DT,NCL,IA,BU,BUI,F2,F3,JS
      LEVEL 2,TIME
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)
      COMMON/RLC/RC(ML),RP(ML),RLV(ML),PH1,DV1,DV2
      COMMON/VOID/INTX(ML2),JV(ML2),IV(ML2),NNV
      COMMON/USUPC/C1(ML),S1(ML),C2(ML),S2(ML),SWV(ML),VMN(ML),
     +GAMMA(ML),ALP(ML)
      COMMON/EOSN/IEOS(ML),ME(ML)
      COMMON/ES/IE(ML2),NME
      II=IE(I)
      JM=JV(I)-1
      IS=IEOS(I)
      GO TO (1,2,2,2),IS
    1 CONTINUE   USUP EOS
      C=C1(II)
      S=S1(II)
      UP=SQRT(2*(XIH-XI(J)))
      VP=VO(II)*(1.-UP/(C+S*UP))  Volume on Hugoniot for Up
      IF(VP.LT.VMN(II).OR.VP.LT.SWV(II))GO TO 2
      PH=UP*(C+S*UP)/VO(II)  Hugoniot pressure
      GO TO 10
    2 CONTINUE
C     GENERAL EOS   ITERATE TO FIND HUGONIOT
      DVB=.01*V(J)
      DVC=2*DVB            ]
      VI=V(J)-DVB          ]  Initialize
      CALL PTEOS(I,PH,TI,VI,XIH)
      GB=PH-2*XIH/DVB      ]
      IC=0                 ]
   20 CONTINUE
      IC=IC+1                              ]
      IF(IC.GT.100)GO TO 99                ]
      DVA=DVB                              ]
      GA=GB                                ]
      DVB=DVC                              ]
      VI=V(J)-DVC                          ]  Secant method iteration
      CALL PTEOS(I,PH,TI,VI,XIH)           ]
      GB=PH-2*XIH/DVB                      ]
      DVC=(GB*DVA-GA*DVB)/(GB-GA)          ]
      IF(ABS((DVC-DVB)/DVC).GT.0.001)GO TO 20
   10 CONTINUE
      RETURN
   99 CONTINUE
      PRINT 990
  990 FORMAT(35H WARNING: FAILURE TO CONVERGE IN PH )
      RETURN
      END
```

## PH

For a given specific internal energy, the volume on the Hugoniot and the Hugoniot pressure are determined.

### Local Variables

II = IE(I) = original region #.

IS = IEOS(I) = EOS type.

C,S = in USUP fit $U_S = C + SU_P$.

UP = $U_P$ = particle velocity relative to unshocked material.

VP = $V_H$ from USUP EOS.

PH = $P_H$ = calculated pressure in the iteration to find the Hugoniot pressure.

DVA,DVB,DVC = $\Delta V_{i-1}$, $\Delta V_i$, $\Delta V_{i+1}$.

VI = $V_i$.

GA,GB = $g_{i-1}, g_i$ ; $g = P_H - \frac{2\Delta I}{\Delta V}$ , g is 0 for the Hugoniot.

IC = iteration count.

### Notes

The USUP fit is treated as a special case because the Hugoniot pressure can be calculated directly as a function of the particle velocity , UP. The material is assumed to be in the initial state in order to allow this simplification. The error involved in this assumption should be small. The particle velocity is calculated using

$$\Delta I = \frac{U_P^2}{2} \quad . \tag{1}$$

From the jump condition for conservation of mass,

$$\rho_0 U_S = \rho(U_S - U_P) \quad , \tag{2}$$

we have, with a little algebra,

$$V = \left(1 - \frac{U_P}{U_S}\right) V_0 \quad . \tag{3}$$

The jump condition for conservation of momentum,

$$P_H = P_0 + \rho_0 U_S U_P \quad , \tag{4}$$

directly gives the Hugoniot pressure in terms of $U_P$ provided a USUP fit is used, i.e.,

$$P_H = P_0 + \rho_0 U_P (C + S U_P) \quad . \tag{5}$$

However, if the final volume in Eq. (3) is small enough that the initial USUP fit is not used in the USUP EOS, then the general Hugoniot iteration scheme described below is used.

For a general EOS, we define a function, g(I,V) such that it is zero everywhere along the Hugoniot, i.e.,

$$g = P(I,V) - 2\,\frac{I - I_0}{V_0 - V} + P_0 \quad , \tag{6}$$

which, as can be seen from the jump condition,

$$I - I_0 = \frac{(P + P_0)(V_0 - V)}{2} \quad , \tag{7}$$

satisfies the condition that g(I,V) is zero on the Hugoniot. In this subroutine, the specific internal energy I is given. So we are looking for the zero of g as a function of V only. The secant method,

$$\Delta V_{i+1} = \frac{\Delta V_{i-1} g_i - \Delta V_i g_{i-1}}{g_i - g_{i-1}} \quad , \tag{8}$$

is used with the initial values

$$\Delta V_0 = 0.01\ V_0 \tag{9}$$

and

$$\Delta V_1 = 0.02\ V_0 \quad , \tag{10}$$

where

$$\Delta V = V_0 - V \quad . \tag{11}$$

Iteration is allowed for a maximum of 100 cycles with a relative convergence criteria of $10^{-3}$ in $\Delta V$.

## E. SESAME Tabular Equation-of-State Subroutines

The following set of subroutines are used for reading and interpolating the SESAME equation-of-state tables. Further information may be found in Sec. VI.B.

```
      SUBROUTINE MATCHK(MID,NRS,LOC,TBLS,IFLG)
C-----------------------------------------------------------------------
C
C  SUBROUTINE   MATCHK(MID,NRS,LOC,TBLS,IFLG)
C
C  PURPOSE      TO CHECK IF A MATERIAL HAS BEEN
C               PREVIOUSLY LOADED
C
C  ARGUMENTS    MID          (INPUT)   SESAME MATERIAL ID
C               NRS          (INPUT)   NUMBER OF REGIONS
C               LOC          (INPUT)   ARRAY OF FIRST WORD LOCATIONS
C                                      IN TABLE STORAGE ARRAY FOR
C                                      FOR EACH REGION
C               TBLS         (INPUT)   TABLE STORAGE ARRAY
C               IFLG         (OUTPUT)  =0 MATERIAL NOT PREVIOUSLY LOADED
C                                      GT.0 LOCATION OF TABLE IF LOADED
C                                      ALREADY
C
C  REMARKS      NONE
C
C  EXTERNALS    NONE
C
C  PROGRAMMER   J.ABDALLAH,JR.
C
C  DATE         26 APRIL 1979
C
C-----------------------------------------------------------------------
      LEVEL 2,TBLS
      DIMENSION LOC(1),TBLS(1)
      IFLG=0
      DO 100 J=1,NRS
      LC=LOC(J)
      IF(LC.LE.0) GO TO 100
      ITEST=TBLS(LC)
      IF(MID.EQ.ITEST) GO TO 200
  100 CONTINUE
      RETURN
  200 IFLG=LC
      RETURN
      END
```

```
      SUBROUTINE TABFCH(MID,TID,LIB,A,LEN,IFLAG)
C-----------------------------------------------------------------------
C
C   SUBROUTINE    TABFCH(MID,TID,LIB,A,LEN,IFLAG)
C
C   PURPOSE       TO FETCH A GIVEN TABLE FOR A GIVEN MATERIAL
C                 FROM A SESAME II LIBRARY
C
C   ARGUMENTS     MID        (INPUT)   MATERIAL ID
C                 TID        (INPUT)   TABLE NO.- IF 0.0 MATERIAL INDEX
C                                      IS RETURNED
C                 LIB        (INPUT)   LIBRARY FILE UNIT NO.
C                 A          (OUTPUT)  ARRAY FOR TABLE STORAGE
C                 LEN        (INPUT)   NO. OF WORDS IN A AVAILABLE
C                 IFLAG      (OUTPUT)  =0 IF TABLE COULD NOT BE LOCATED
C                                      GT. 0=NO. OF WORDS IN TABLE RETURNED
C                                      LT. 0 - NO. OF ADDITIONAL
C                                      WORDS NEEDED
C
C   REMARKS       A RANDOM I/O TECHNIQUE IS USED TO LOCATE AND LOAD
C                 THE SPECIFIED TABLE FROM THE SESAME II LIBRARY.
C                 THE MATERIAL INDEX AND ITS ADDRESS ARE TO SAVED
C                 TO HASTEN THE FETCHING OF ANOTHER TABLE FOR THE SAME
C                 MATERIAL AND LIBRARY FILE IN SUBSEQUENT CALLS TO
C                 TABFCH.
C
C   EXTERNALS     INBUFR
C
C   PROGRAMMER    J.ABDALLAH.,JR.
C
C   DATE          24 APRIL 1979
C
C-----------------------------------------------------------------------
      LEVEL 2,A
      DIMENSION A(1),HINDEX(35)
      DATA HINDEX(1)/0.0/
      DATA LIBLST/0/
      IFLAG=0
C . . FIND NO. MATERIALS ON LIBRARY
      IF(LIB.NE.LIBLST) GO TO 50
      IDLAST=HINDEX(1)
      IF(IDLAST.EQ.MID) GO TO 230
 50   LIBLST=LIB
      NW=1
      IF(LEN.LT.NW) GO TO 999
      CALL INBUFR(LIB,A,1,1,IER)
      N=A(1)
      NW=N+N+N
      IF(LEN.LT.NW) GO TO 999
      CALL INBUFR(LIB,A,NW,5,IER)
C . . FIND ADDRESS OF MATERIAL FILE
      DO 100 J=1,N
      ITEST=A(J)
      IF(ITEST.NE.MID) GO TO 100
      NW=A(J+N)
      IAD=A(J+N+N)
      GO TO 200
 100  CONTINUE
      RETURN
C . . GET MATERIAL INDEX
```

```
 200  IF(LEN.LT.NW) GO TO 999                                     SESAME
      IADX=IAD                                                    SESAME
      CALL INBUFR(LIB,A,NW,IADX,IER)                              SESAME
      DO 210 J=1,NW                                               SESAME
      HINDEX(J)=A(J)                                              SESAME
 210  CONTINUE                                                    SESAME
      IF(TID.EQ.0.0) GO TO 500                                    SESAME
 230  N=HINDEX(5)                                                 SESAME
      IAD=IADX+6+N+N                                              SESAME
      DO 300 J=1,N                                                SESAME
      NW=HINDEX(5+J+N)                                            SESAME
      IF(TID.NE.HINDEX(5+J)) GO TO 250                            SESAME
      GO TO 400                                                   SESAME
 250  IAD=IAD+NW+1                                                SESAME
 300  CONTINUE                                                    SESAME
      RETURN                                                      SESAME
 400  IF(LEN.LT.NW) GO TO 999                                     SESAME
C . . READ REQUESTED TABLE                                         SESAME
      CALL INBUFR(LIB,A,NW,IAD,IER)                               SESAME
 500  IFLAG=NW                                                    SESAME
      RETURN                                                      SESAME
 999  IFLAG=LEN-NW                                                SESAME
      RETURN                                                      SESAME
      END                                                         SESAME
```

```
      SUBROUTINE INBUFR(LU,Z,NW,IAD,IFLG)                               SESAME
C-----------------------------------------------------------------------SESAME
C                                                                       SESAME
C  SUBROUTINE   INBUFR(LU,Z,NW,IAD,IFLG)                                SESAME
C                                                                       SESAME
C  PURPOSE      RANDOM I/O READ                                         SESAME
C                                                                       SESAME
C  ARGUMENTS    LU    (INPUT)  UNIT NO.                                 SESAME
C               Z     (OUTPUT)  STORAGE AREA WHERE DAT IS RETURNED      SESAME
C               NW    (INPUT)  NO. OF WORDS TO BE READ                  SESAME
C               IAD   (INPUT)  STARTING DISK ADDRESS OF DATA            SESAME
C               IFLG  (OUTPUT) 0=NORMAL                                 SESAME
C                              1=EOF ENCOUNTERED                        SESAME
C                             -1=ERROR                                  SESAME
C                                                                       SESAME
C  REMARKS      NONE                                                    SESAME
C                                                                       SESAME
C  EXTERNALS    RDISK                                                   SESAME
C                                                                       SESAME
C  PROGRAMMER   J.ABDALLAH,JR.                                          SESAME
C                                                                       SESAME
C  DATE         1 MAY 1979                                              SESAME
C                                                                       SESAME
C-----------------------------------------------------------------------SESAME
      LEVEL 2,Z                                                         SESAME
      CALL RDISK(LU,Z,NW,IAD)                                           SESAME
      IF(UNIT(LU)) 10,20,30                                             SESAME
 10   IFLG=1                                                            SESAME
      RETURN                                                            SESAME
 20   IFLG=0                                                            SESAME
      RETURN                                                            SESAME
 30   IFLG=-1                                                           SESAME
      RETURN                                                            SESAME
      END                                                               SFSAME
```

```
      FUNCTION DPACK(A,B)
C-----------------------------------------------------------------------
C
C  FUNCTION   DPACK
C
C  PURPOSE    TO DOUBLE PACK ARGUMENTS A AND B INTO A SINGLE WORD
C
C  REMARKS    SYSTEM DEPENDENT SHIFT FUNCTION
C
C  PROGRAMMER  J.ABDALLAH,JR.
C
C  DATE        1 MAY 1979
C
C-----------------------------------------------------------------------
      EQUIVALENCE (I1,X1),(I2,X2)
      DATA MASK/77777777770000000000B/
      X1=A
      X2=B
      I1=I1.AND.MASK
      I2=I2.AND.MASK
      I2=SHIFT(I2,30)
      I1=I1.OR.I2
      DPACK=X1
      RETURN
      END
```

```
      FUNCTION ISRCHK(X,TBLS,N,K,NSFT)
C-----------------------------------------------------------------------
C
C  FUNCTION:     ISRCHK(X,TBLS,N,K,NSFT)
C
C  PURPOSE:      FIND INDEX OF X IN AN ARRAY TBLS.  TABLE VALUES
C                NEED NOT BE CONTIGUOUS AND CAN BE IN EITHER
C                ASCENDING OR DESCENDING ORDER.
C
C  ARGUMENTS:    X    (INPUT) - VALUE TO BE LOCATED
C                TBLS (INPUT) - TABLE TO BE SEARCHED
C                N    (INPUT) - NUMBER OF VALUES TO BE SEARCHED
C                K    (INPUT) - SPACING BETWEEN VALUES IN TABLE
C                THE VALUE OF THE FUNCTION = INDEX I, WHERE
C                  TBLS(1+K*(I-1)).LE.X.LT.TBLS(1+K*I), OR
C                  TBLS(1+K*(I-1)).GE.X.GT.TBLS(1+K*I), OR
C                  I=0 OR I=N IF X IS OUTSIDE RANGE OF TABLE.
C                NSFT (INPUT) - NO. OF BITS THE TABLE VALUES ARE
C                               TO BE SHIFTED
C
C  REMARKS:      TBLS CAN BE DECLARED LCM ON THE CDC 7600.
C
C  EXTERNALS:    SHIFT.
C
C  PROGRAMMER:   G. I. KERLEY, T-4., J.ABDALLAH,JR.
C
C  DATE:         19 NOVEMBER 1976,REVISED 6 JULY 1979
C
C-----------------------------------------------------------------------
      LEVEL 2,TBLS
      DIMENSION TBLS(1)
      ISRCHK = 0
      J = N+1
      KI = 1-K
      S1=TBLS(1)
      S1=SHIFT(S1,NSFT)
      S=TBLS(KI+K*N)
      S=SHIFT(S,NSFT)
      S=S-S1
 1    IF(J-ISRCHK.EQ.1) RETURN
      JP = .5*(J+ISRCHK)
      S1=TBLS(KI+K*JP)
      S1=SHIFT(S1,NSFT)
      IF(S*(X-S1).LT.0.0) GO TO 2
      ISRCHK = JP
      GO TO 1
 2    J = JP
      GO TO 1
      END
```

```
      SUBROUTINE T4INTP
C-----------------------------------------------------------------------
C
C  SUBROUTINE:  T4INTP
C
C  PURPOSE:     INTERPOLATE FOR A FUNCTION Z(X,Y) AND ITS
C               DERIVATIVES FROM TABLES LOCATED IN ARRAY TBLS.
C
C               THE ROUTINE REQUIRES COMMON BLOCKS,
C               COMMON/RTBLK2/LOCX,IX,NX,LOCY,IY,NY,LOCZ,NZ,NSFT,
C                              X,Y,Z(3),IP,IDS,ZZ
C                 LOCX = LOCATION OF X VECTOR
C                 IX   = INDEX OF X VECTOR
C                 NX   = LENGTH OF X VECTOR
C                 LOCY = LOCATION OF Y VECTOR
C                 IY   = INDEX OF Y VECTOR
C                 NY   = LENGTH OF Y VECTOR
C                 LOCZ = LOCATION OF Z(X,Y) ARRAY
C                 NZ   = SPACING OF Z ARRAY
C                 NSFT = BIT SHIFT PARAMETER
C               X,Y  (INPUT) - INDEPENDENT VARIABLES
C               Z   (OUTPUT) - VECTOR OF LENGTH 3, WHERE
C                 Z(1) = VALUE OF FUNCTION
C                 Z(2) = X DERIVATIVE OF FUNCTION
C                 Z(3) = Y DERIVATIVE OF FUNCTION
C               ZZ  (IN/OUT) - COEFFICIENT VECTOR OF LENGTH 16
C               IP    (INPUT) - BRANCH PARAMETER
C                 IP.EQ.0, USE INPUT COEFFICIENTS IN ZZ
C                 IP.NE.0, CALCULATE ZZ VECTOR FIRST
C               IDS  (INPUT) - DISPLACEMENT INTO ZZ FOR COEFFS.
C                              TO BE USED
C               COMMON/INTORD/IFN.
C               IFN  (INPUT) - INTERPOLATION TYPE
C                 IFN.NE.1, RATIONAL FUNCTION
C                 IFN.EQ.1, BILINEAR
C               COMMON/SESDAT/TBLS
C                 TBLS IS THE TABLE STORAGE ARRAY
C
C
C  REMARKS:     UNLESS BILINEAR FORM IS SPECIFIED, ROUTINE
C               USES RATIONAL FUNCTION METHOD WITH QUADRATIC
C               ESTIMATE OF DERIVATIVES AT THE MESH POINTS.
C               TBLS CAN BE DECLARED LCM ON THE CDC 7600.
C
C        *****  SYSTEM DEPENDENT FEATURE.  THE Z-ARRAY CAN BE
C        *****  DOUBLE PACKED.  PARAMETER NSFT SPECIFIES THE
C        *****  NUMBER OF BITS TO BE SHIFTED WHEN UNPACKING THE
C        *****  RIGHT HALF OF THE WORD.  THIS ROUTINE USES
C        *****  THE LASL SHIFT FUNCTION
C
C  EXTERNALS:   NONE, BUT A SEARCH ROUTINE MUST BE CALLED
C               FIRST, TO COMPUTE INDICES IX AND IY.
C
C  PROGRAMMER:  G. I. KERLEY, T-4., J. ABDALLAH,T-4.
C
C  DATE:        01 AUG 1979
C
C-----------------------------------------------------------------------
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
```

```
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      LEVEL 2,TBLS
      COMMON/RTBLK2/LOCX,IX,NX,LOCY,IY,NY,LOCZ,NZ,NSFT,X,Y,Z(3),
     $ IP,IDS,ZZ(32)
      COMMON/INTORD/IFN
      COMMON/SESDAT/TBLS(NSD)
C  CALCULATE COEFFICIENTS FOR RATIONAL FUNCTION INTERPOLATION
      IF(IFN.EQ.1) GO TO 13
      IF(IP.EQ.0) GO TO 8
      I = LOCX+IX-1
      IZ = LOCZ+NZ*(IX-1+NX*(IY-1))
      KZ = NZ
      IBR = IX
      NBR = NX-IX
      ZZ(IDS+4) = TBLS(I)
      DO 7 K=1,4
      KI=IDS+K-1
      IF(K.LT.4) GO TO 1
      IZ = IZ+NZ
      GO TO 4
1     IF(K.LT.3) GO TO 2
      ZZ(IDS+6) = D
      I = LOCY+IY-1
      KZ = KZ*NX
      IZ = IZ-KZ
      IBR = IY
      NBR = NY-IY
      ZZ(IDS+5) = TBLS(I)
      GO TO 3
2     IF(K.LT.2) GO TO 3
      IZ = IZ+NX*NZ
      GO TO 4
3     D = TBLS(I+1)-TBLS(I)
4     ZZ(KI)=SHIFT(TBLS(IZ),NSFT)
      S=SHIFT(TBLS(IZ+KZ),NSFT)
      S = (S-ZZ(KI))/D
      IF(NBR.EQ.1) GO TO 5
      SP=SHIFT(TBLS(IZ+KZ+KZ),NSFT)
      SP = (SP-D*S-ZZ(KI))/(TBLS(I+2)-TBLS(I+1))
      G2 = (SP-S)/(TBLS(I+2)-TBLS(I))
      IF(IBR.GT.1) GO TO 5
      IF(S*(S-D*G2).LE.0.) G2=S/D
      G1 = G2
      GO TO 6
5     DM = TBLS(I)-TBLS(I-1)
      SM=SHIFT(TBLS(IZ-KZ),NSFT)
      SM = (ZZ(KI)-SM)/DM
      G1 = (S-SM)/(D+DM)
      IF(NBR.EQ.1) G2=G1
      IF(IBR.GT.2) GO TO 6
      IF(SM*(SM-DM*G1).LE.0.) G1=(S-SM-SM)/D
6     IF(G2.NE.0.) G1=G1/G2
      ZZ(KI+8) = G1
7     ZZ(KI+12) = G2
      ZZ(IDS+7)=D
      ZZ8=ZZ(IDS+7)
      ZZ7=ZZ(IDS+6)
      ZZ(IDS+2)=(ZZ(IDS+1)-ZZ(IDS))/ZZ8
      ZZ(IDS+1)=(ZZ(IDS+3)-ZZ(IDS))/ZZ7
```

```
      ZZ(IDS+3)=(S-ZZ(IDS+2))/ZZ7                                       SESAME
      ZZ(IDS+12)=ZZ(IDS+12)/ZZ8                                         SESAME
      ZZ(IDS+13)=ZZ(IDS+13)/ZZ8                                         SESAME
      ZZ(IDS+14)=ZZ(IDS+14)/ZZ7                                         SESAME
      ZZ(IDS+15)=ZZ(IDS+15)/ZZ7                                         SESAME
C  EVALUATE RATIONAL FUNCTION FROM PRECALCULATED COEFFICIENTS           SESAME
 8    QX = X-ZZ(IDS+4)                                                  SESAME
      RX = ZZ(IDS+6)-QX                                                 SESAME
      QY = Y-ZZ(IDS+5)                                                  SESAME
      RY = ZZ(IDS+7)-QY                                                 SESAME
      IF(RX.NE.0.) GO TO 9                                              SESAME
      W1 = 1.                                                           SESAME
      W2 = 1.                                                           SESAME
      GO TO 10                                                          SESAME
 9    W1 = 1.-1./(1.+ABS(ZZ(IDS+8)*QX/RX))                              SESAME
      W2 = 1.-1./(1.+ABS(ZZ(IDS+9)*QX/RX))                              SESAME
 10   F1 = ZZ(IDS+12)*(W1+ZZ(IDS+8)*(1.-W1))                            SESAME
      F2 = ZZ(IDS+13)*(W2+ZZ(IDS+9)*(1.-W2))                            SESAME
      Z(2) = ZZ(IDS+6)*(RY*(F1-ZZ(IDS+12))*W1+QY*(F2-ZZ(IDS+13))*W2)    SESAME
      G1 = RY*F1+QY*F2                                                  SESAME
      IF(RY.NE.0) GO TO 11                                              SESAME
      W1 = 1.                                                           SESAME
      W2 = 1.                                                           SESAME
      GO TO 12                                                          SESAME
11    W1 = 1.-1./(1.+ABS(ZZ(IDS+10)*QY/RY))                             SESAME
      W2 = 1.-1./(1.+ABS(ZZ(IDS+11)*QY/RY))                             SESAME
 12   F3 = ZZ(IDS+14)*(W1+ZZ(IDS+10)*(1.-W1))                           SESAME
      F4 = ZZ(IDS+15)*(W2+ZZ(IDS+11)*(I.-W2))                           SESAME
      Z(3) = ZZ(IDS+7)*(RX*(F3-ZZ(IDS+14))*W1+QX*(F4-ZZ(IDS+15))*W2)    SESAME
      G2 = RX*F3+QX*F4                                                  SESAME
      ZZ2=ZZ(IDS+1)                                                     SESAME
      ZZ3=ZZ(IDS+2)                                                     SESAME
      ZZ4=ZZ(IDS+3)                                                     SESAME
      Z(1) = ZZ(IDS)+(ZZ2+ZZ4*QY-RX*G1)*QX+(ZZ3-RY*G2)*QY               SESAME
      Z(2) = Z(2)+ZZ2+QY*(ZZ4+RY*(F3-F4))+(QX-RX)*G1                    SESAME
      Z(3) = Z(3)+ZZ3+QX*(ZZ4+RX*(F1-F2))+(QY-RY)*G2                    SESAME
      RETURN                                                            SESAME
C  CALCULATE COEFFICIENTS FOR BILINEAR INTERPOLATION                    SESAME
 13   IF(IP.EQ.0) GO TO 14                                              SESAME
      I=LOCX+IX                                                         SESAME
      IND=IDS+4                                                         SESAME
      ZZ(IND)=TBLS(I-1)                                                 SESAME
      DX=TBLS(I)-ZZ(IND)                                                SESAME
      J=LOCY+IY                                                         SESAME
      IND=IDS+5                                                         SESAME
      ZZ(IND)=TBLS(J-1)                                                 SESAME
      DY=TBLS(J)-ZZ(IND)                                                SESAME
      IZ=LOCZ+NZ*(IX-1+NX*(IY-1))                                       SESAME
      ZZ(IDS)=SHIFT(TBLS(IZ),NSFT)                                      SESAME
      IND=IDS+1                                                         SESAME
      ZZ(IND)=SHIFT(TBLS(IZ+NZ),NSFT)                                   SESAME
      ZZ(IND)=(ZZ(IND)-ZZ(IDS))/DX                                      SESAME
      IZ=IZ+NZ*NX                                                       SESAME
      IND=IDS+2                                                         SESAME
      ZZ(IND)=SHIFT(TBLS(IZ),NSFT)                                      SESAME
      ZZ(IND)=(ZZ(IND)-ZZ(IDS))/DY                                      SESAME
      IND=IDS+3                                                         SESAME
      ZZ(IND)=SHIFT(TBLS(IZ+NZ),NSFT)                                   SESAME
      ZZ(IND)=(ZZ(IND)-ZZ(IDS)-ZZ(IDS+1)*DX-ZZ(IDS+2)*DY)/(DX*DY)       SESAME
C  EVALUATE BILINEAR FUNCTION FROM PRECALCULATED COEFFICIENTS           SESAME
```

```
14    QX = X-ZZ(IDS+4)                                                SESAME   411
      QY = Y-ZZ(IDS+5)                                                SESAME   412
      Z(2) = ZZ(IDS+1)+ZZ(IDS+3)*QY                                   SESAME   413
      Z(3) = ZZ(IDS+2)+ZZ(IDS+3)*QX                                   SESAME   414
      Z(1) = ZZ(IDS)+Z(2)*QX+ZZ(IDS+2)*QY                             SESAME   415
      RETURN                                                          SESAME   416
      END                                                             SESAME   417
```

```
      SUBROUTINE GETINV(IR,MID,IDT,TBLS,LCNT,LU,IFL,Z8)
C-----------------------------------------------------------------------
C
C  SUBROUTINE   GETINV(IR,MID,IDT,TBLS,LCNT,LU,IFL,Z8)
C
C  PURPOSE      TO LOAD INVERTED (ENERGY BASED) SESAME II
C               EOS TABLES
C
C  ARGUMENTS    IR         (INPUT)   REGION NO.
C               MID        (INPUT)   SESAME MATERIAL ID
C               IDT        (INPUT)   DATA TYPE INDICATOR
C               TBLS       (INPUT)   TABLE STORAGE ARRAY
C               LCNT       (IN/OUT)  POSITION IN ARRAY FOR STORING TABLES
C               LU         (INPUT)   SESAME LIBRARY UNIT NO.
C               IFL        (OUTPUT)  ERROR FLAG
C                                    2=MATERIAL ALREADY LOADED
C                                    1=SUCCESSFUL LOADING
C                                    0=DATA NOT FOUND
C                                    LT.0 FOR - THE NO. OF EXTRA WORDS
C                                    NEEDED FOR LOADING
C               Z8         (OUTPUT)  ATOMIC CHARGE,CHARGE**2,AND MASS
C                                    Z8(1)=Z
C                                    Z8(2)=Z**2
C                                    Z8(3)=A
C
C  REMARKS      UNITS - ENERGY      MBAR*CC/GM
C                       TEMP        DEGREES KELVIN
C                       DENSITY     GRAMS/CC
C                       PRESSURE    MBAR
C
C  EXTERNALS    MATCHK,TABFCH,INV301
C
C  PROGRAMMER   J.ABDALLAH,JR.
C
C  DATE         13 JUNE 1979
C
C-----------------------------------------------------------------------
C
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSO=NSM*NWPM+132,ML2=100)
      LEVEL 2,TBLS
      COMMON/S2DIR/LCMX,NRS,LCFW(ML,1)
      DIMENSION Z8(3),TBLS(1)
C . . UNIT CONVERSION FACTORS
      DATA TFAC,RFAC,PEFAC/1.0,1.0,.01/
      CALL MATCHK(MID,NRS,LCFW(1,IDT),TBLS(1),IFL)
      IF(IFL.EQ.0) GO TO 10
      LCFW(IR,IDT)=IFL
      IFL=2
      RETURN
 10   NL=LCMX-LCNT-1
C . . FETCH EOS TABLES
      CALL TABFCH(MID,201.,LU,TBLS(LCNT+2),NL,IFL)
      IF(IFL.LE.0) RETURN
      Z8(1)=TBLS(LCNT+2)
      Z8(2)=Z8(1)*Z8(1)
      Z8(3)=TBLS(LCNT+3)
      TBLS(LCNT+1)=TBLS(LCNT+4)
```

```
      CALL TABFCH(MID,301.,LU,TBLS(LCNT+2),NL,IFL)                      SESAME
      IF(IFL.LE.0) RETURN                                               SESAME
      TBLS(LCNT)=FLOAT(MID)                                             SESAME
      CALL PERTCB(IR,TBLS(LCNT),ZB(1),ZB(3))                            SESAME
      NR=TBLS(LCNT+2)                                                   SESAME
      NT=TBLS(LCNT+3)                                                   SESAME
      NRT=NR*NT                                                         SESAME
      LOCP=LCNT+3+NR+NT                                                 SESAME
C . . CONVERT TO DESIRED UNITS                                           SESAME
      DO 30 I=1,NT                                                      SESAME
      TBLS(3+I+LCNT+NR)=TFAC*TBLS(3+I+LCNT+NR)                          SESAME
      DO 30 J=1,NR                                                      SESAME
      IF(I.GT.1) GO TO 20                                               SESAME
      TBLS(3+J+LCNT)=TBLS(3+J+LCNT)*RFAC                                SESAME
   20 LOCP=LOCP+1                                                       SESAME
      TBLS(LOCP)=PEFAC*TBLS(LOCP)                                       SESAME
      TBLS(LOCP+NRT)=PEFAC*TBLS(LOCP+NRT)                               SESAME
   30 CONTINUE                                                          SESAME
C . . WINDOW TABLES HERE AND RESET VALUES OF NR NT AND                  SESAME
C     NRT IF WINDOWING IS NEEDED                                        SESAME
C . . INVERT TABLES                                                     SESAME
C . . CHECK TO SEE IF THERE IS ENOUGH ROOM TO INVERT THE TABLES         SESAME
C     NINV IS THE LAST LOCATION NEEDED FOR TABLE INVERSION              SESAME
      NINV=LCNT+3+2*NRT+2*NR+4*NT                                       SESAME
      IF(NINV.LE.LCMX) GO TO 40                                         SESAME
      IFL=LCMX-NINV                                                     SESAME
      RETURN                                                            SESAME
   40 RO=TBLS(LCNT+1)                                                   SESAME
      LOC=LCNT+2                                                        SESAME
      CALL INV301(TBLS,LOC,RO,LDS)                                      SESAME
C . . DOUBLE PACK DEPENDENT VARIABLES                                   SESAME
      LOCP=LCNT+3+NR+NT+NR                                              SESAME
      DO 50 I=1,NRT                                                     SESAME
      LOCP=LOCP+1                                                       SESAME
      PTEM=TBLS(LOCP)                                                   SESAME
      TTEM=TBLS(LOCP+NRT)                                               SESAME
      TBLS(LOCP)=DPACK(PTEM,TTEM)                                       SESAME
   50 CONTINUE                                                          SESAME
C . . WRAP UP                                                           SESAME
      LCFW(IR,IDT)=LCNT                                                 SESAME
      LCNT=LCNT+2+LDS-NRT                                               SESAME
      IFL=1                                                             SESAME
      RETURN                                                            SESAME
      END                                                               SESAME
```

```
      SUBROUTINE RATFN1
C-----------------------------------------------------------------
C
C   SUBROUTINE:  RATFN1
C
C   PURPOSE:     INTERPOLATE FOR A FUNCTION Y(X) AND ITS
C                DERIVATIVE FROM TABLES LOCATED IN ARRAY TBLS.
C
C                THE ROUTINE ALSO REQUIRES COMMON BLOCKS,
C                COMMON/RTBLK1/LOCX,KX,LOCY,KY,I,N,IP,X,Y(2)
C                  LOCX = LOCATION OF X VECTOR
C                  KX   = SPACING OF X VECTOR
C                  LOCY = LOCATION OF Y VECTOR
C                  KY   = SPACING OF Y VECTOR
C                  I    = INDEX OF X AND Y VECTORS
C                  N    = LENGTH OF X AND Y VECTORS
C                X     (INPUT) - INDEPENDENT VARIABLE
C                Y    (OUTPUT) - VECTOR OF LENGTH 2, WHERE
C                  Y(1) = VALUE OF FUNCTION
C                  Y(2) = DERIVATIVE OF FUNCTION
C                IP    (INPUT) - BRANCH PARAMETER
C                  IP.EQ.0, USE INPUT COEFFICIENTS IN YY
C                  IP.NE.0, CALCULATE YY VECTOR FIRST
C                COMMON/INTORD/IFN
C                IFN  (INPUT) - INTERPOLATION TYPE
C                  IFN.NE.1, RATIONAL FUNCTION
C                  IFN.EQ.1, LINEAR
C                COMMON/SESDAT/TBLS
C                 TBLS (INPUT) - TABLE STORAGE ARRAY
C
C
C   REMARKS:     UNLESS LINEAR FORM IS SPECIFIED, ROUTINE
C                USES RATIONAL FUNCTION METHOD WITH QUADRATIC
C                ESTIMATE OF DERIVATIVES AT THE MESH POINTS.
C                TBLS CAN BE DECLARED LCM ON THE CDC 7600.
C
C   EXTERNALS:   NONE, BUT A SEARCH ROUTINE MUST BE CALLED
C                FIRST, TO COMPUTE INDEX I.
C
C   PROGRAMMER:  G. I. KERLEY, T-4.
C
C   DATE:        18 JULY 1979
C
C-----------------------------------------------------------------
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      LEVEL 2,TBLS
      DIMENSION YY(6)
      COMMON/SESDAT/TBLS(NSD)
      COMMON/INTORD/IFN
      COMMON/RTBLK1/LOCX,KX,LOCY,KY,I,N,IP,X,Y(2)
      IF(IFN.EQ.1) GO TO 6
      IF(IP.EQ.0) GO TO 3
C   CALCULATE COEFFICIENTS FOR RATIONAL FUNCTION INTERPOLATION
      IX = LOCX+KX*(I-1)
      IY = LOCY+KY*(I-1)
      YY(3) = TBLS(IX)
      YY(4) = TBLS(IX+KX)-YY(3)
```

```
      YY(1) = TBLS(IY)                                                  SESAME
      YY(2) = (TBLS(IY+KY)-YY(1))/YY(4)                                 SESAME
      IF(I.EQ.N-1) GO TO 1                                              SESAME
      SP = (TBLS(IY+KY+KY)-TBLS(IY+KY))/(TBLS(IX+KX+KX)-TBLS(IX+KX))   SESAME
      YY(6) = (SP-YY(2))/(TBLS(IX+KX+KX)-YY(3))                         SESAME
      IF(I.GT.1) GO TO 1                                                SESAME
      IF(YY(2)*(YY(2)-YY(4)*YY(6)).LE.0.) YY(6)=YY(2)/YY(4)             SESAME
      YY(5) = YY(6)                                                     SESAME
      GO TO 2                                                           SESAME
 1    DM = YY(3)-TBLS(IX-KX)                                            SESAME
      SM = (YY(1)-TBLS(IY-KY))/DM                                       SESAME
      YY(5) = (YY(2)-SM)/(YY(4)+DM)                                     SESAME
      IF(I.EQ.N-1) YY(6)=YY(5)                                          SESAME
      IF(I.GT.2) GO TO 2                                                SESAME
      IF(SM*(SM-DM*YY(5)).LE.0.) YY(5)=(YY(2)-SM-SM)/YY(4)              SESAME
 2    IF(YY(6).NE.0.) YY(5)=YY(5)/YY(6)                                 SESAME
C  EVALUATE RATIONAL FUNCTION FROM PRECALCULATED COEFFICIENTS           SESAME
 3    Q = X-YY(3)                                                       SESAME
      R = YY(4)-Q                                                       SESAME
      IF(R.NE.0.) GO TO 4                                               SESAME
      W = 1.                                                            SESAME
      GO TO 5                                                           SESAME
 4    W = 1.-1./(1.+ABS(YY(5)*Q/R))                                     SESAME
 5    F = YY(6)*(W+YY(5)*(1.-W))                                        SESAME
      Y(1) = YY(1)+Q*(YY(2)-R*F)                                        SESAME
      Y(2) = YY(2)+(Q-R)*F+YY(4)*W*(F-YY(6))                            SESAME
      RETURN                                                            SESAME
C  CALCULATE COEFFICIENTS FOR LINEAR INTERPOLATION                      SESAME
 6    IF(IP.EQ.0) GO TO 7                                               SESAME
      IX = LOCX+KX*(I-1)                                                SESAME
      IY = LOCY+KY*(I-1)                                                SESAME
      YY(3) = TBLS(IX)                                                  SESAME
      YY(1) = TBLS(IY)                                                  SESAME
      YY(2) = (TBLS(IY+KY)-YY(1))/(TBLS(IX+KX)-YY(3))                   SESAME
C  CALCULATE LINEAR ESTIMATE FROM PRECALCULATED COEFFICIENTS            SESAME
 7    Y(1) = YY(1)+YY(2)*(X-YY(3))                                      SESAME
      Y(2) = YY(2)                                                      SESAME
      RETURN                                                            SESAME
      END                                                               SESAME
```

```
      SUBROUTINE T4DATI
C-----------------------------------------------------------------
C
C   SUBROUTINE:  T4DATI
C
C   PURPOSE:      SEARCH/INTERPOLATE FOR PRESSURE AND TEMPERATURE
C                 AS FUNCTIONS OF REGION, DENSITY AND ENERGY,
C                 USING PACKED SESAME 2 DATA STRING OF TYPE 302
C
C                 COMMON/SESIN/IR,IDT,R,E,IBR,IFL
C                 COMMON/SESOUT/P(3),T(3)
C                 IR   (INPUT) - MATERIAL REGION  NUMBER
C                 IDT  (INPUT) - DATA TYPE INDICATOR
C                 R    (INPUT) - DENSITY
C                 E    (INPUT) - INTERNAL ENERGY
C                 P,T (OUTPUT) - PRESSURE, TEMPERATURE VECTORS
C                   P(1),T(1) = PRESSURE AND TEMPERATURE
C                   P(2),T(2) = DENSITY DERIVATIVES
C                   P(3),T(3) = ENERGY DERIVATIVES
C                 IBR  (INPUT) - 0=COMPUTE BOTH P AND T
C                                1=COMPUTE P ONLY
C                                2=COMPUTE T ONLY
C                 COMMON/SESDAT/TBLS
C                 TBLS (INPUT) - TABLE STORAGE ARRAY
C
C   REMARKS:      ADAPTED FROM T-4 SESAME 2 ROUTINES S2EOSI AND
C                 LA302A.  PRESSURE AND TEMPERATURE ARE PACKED.
C                 THE SEARCH INDICES AND INTERPOLATION CONSTANTS
C                 ARE SAVED AND REUSED, IF POSSIBLE.
C
C           *****  SYSTEM DEPENDENT FEATURE.  THE CONSTANT NSFT
C           *****  IN STATEMENT 60 SHOULD BE SET TO 1/2 THE BIT
C           *****  LENGTH.  FOR A CDC 7600, NSFT = 30.
C
C   EXTERNALS:    RATFN1 (1-D INTERPOLATION ROUTINE)
C                 T4INTP (2-D INTERPOLATION ROUTINE)
C
C   PROGRAMMER:   G. I. KERLEY AND B. I. BENNETT, T-4.
C                 J. ABDALLAH,JR.
C
C   DATE:         2 AUGUST 1978
C
C-----------------------------------------------------------------
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      LEVEL 2,TBLS
      COMMON/S2DIR/LCMX,NREG,LCFW(ML,1)
      COMMON/RTBLK1/LOCR,KX,LOCE,KY,IRX,N,ISAME,RX1,PX1(2)
      COMMON/RTBLK2/LOCX,IX,NX,LOCY,IY,NY,LOCZ,NZ,NSFT,
     $ RX2,ET,PX2(3),INT,IDS,ZZ(32)
      COMMON/SESIN/IR,IDT,R,E,IBR,IFL
      COMMON/SESOUT/P(3),T(3)
      COMMON/SESDAT/TBLS(NSD)
      DATA LOCLST,IP,IT/0,1,1/
C   LOC IS POINTER TO START OF DATA STRING FOR REGION IR
      LOC = LCFW(IR,IDT)+2
C   TEST TO SEE IF THE MATERIAL IS THE SAME AS LAST CALL
      IF(LOC.EQ.LOCLST) GO TO 5
```

```
C  THE FOLLOWING OPERATIONS DO NOT NEED TO BE REPEATED
      NX = TBLS(LOC)
      NY = TBLS(LOC+1)
      N = NX
      LOCR = LOC+2
      KX = 1
      LOCX = LOCR
      LOCY = LOCX+NX
      LOCE = LOCY+NY
      KY = 1
      LOCZ = LOCE+NX
      NZ = 1
C  UNLESS A NEW REGION HAS BEEN ENTERED
      LOCLST=LOC
      IXLAST = 0
      IYLAST = 0
      LOCI = LOCX+NX/2-1
      LOCJ = LOCY+NY/2-1
      LOCNX=LOCX+NX-2
      LOCNY=LOCY+NY-2
C  SEARCH FOR DENSITY INDEX
 5    IF(R.LT.TBLS(LOCI)) GO TO 15
 10   IF(R.LT.TBLS(LOCI+1)) GO TO 20
      IF(LOCI.EQ.LOCNX) GO TO 20
      LOCI=LOCI+1
      GO TO 10
 15   IF(LOCI.EQ.LOCX) GO TO 20
      LOCI=LOCI-1
      IF(R.LT.TBLS(LOCI)) GO TO 15
 20   IX=LOCI-LOCX+1
C  INTERPOLATE FOR ENERGY ON COLD CURVE.  IF ISAME = 0, DENSITY
C  INDEX IS THE SAME AS IN THE LAST CALL TO THIS ROUTINE
      IRX = IX
      ISAME = IA3S(IX-IXLAST)
      RX1=R
      CALL RATFN1
      ET = AMAX1(0.,E-PX1(1))
      DECOR = PX1(2)
      RX2=R
C  SEARCH FOR ENERGY INDEX
      IF(ET.LT.TBLS(LOCJ)) GO TO 35
 30   IF(ET.LT.TBLS(LOCJ+1)) GO TO 40
      IF(LOCJ.EQ.LOCNY) GO TO 40
      LOCJ=LOCJ+1
      GO TO 30
 35   IF(LOCJ.EQ.LOCY) GO TO 40
      LOCJ=LOCJ-1
      IF(ET.LT.TBLS(LOCJ)) GO TO 35
 40   IY=LOCJ-LOCY+1
C  IF ISAME = 0,  DENSITY AND TEMPERATURE INDICES ARE
C  THE SAME AS IN THE LAST CALL TO THIS ROUTINE
      ISAME = ISAME+IABS(IY-IYLAST)
      IP = MIN0(1,IP+ISAME)
      IT = MIN0(1,IT+ISAME)
      IXLAST = IX
      IYLAST = IY
      IDS=(IDT-1)*32+1
      IF(IBR.EQ.2) GO TO 50
C  PRESSURE CALCULATION
      NSFT = 0
```

```
      INT=IP
      CALL T4INTP
      P(1)=PX2(1)
      P(2)=PX2(2)-DECDR*PX2(3)
      P(3)=PX2(3)
      IP = 0
      IF(IBR.EQ.1) RETURN
C  TEMPERATURE CALCULATION
 50   NSFT = 30
      INT=IT
      IOS=IDS+16
      CALL T4INTP
      T(1)=PX2(1)
      T(2)=PX2(2)-DECDR*PX2(3)
      T(3)=PX2(3)
      IT = 0
      RETURN
      END
```

```
      SUBROUTINE T4RTPE(IR,IDT,TBLS,R,T,P,E,IFL)
C----------------------------------------------------------------------
C
C  SUBROUTINE T4RTPE(IR,IDT,TBLS,R,T,P,E,IFL)
C
C  PURPOSE     TO FIND PRESSURE AND ENERGY AS FUNCTIONS
C              OF DENSITY AND TEMPERATURE FROM A
C              SESAME TYPE 302 TABLE USING NEWTONS METHOD.
C
C  ARGUMENTS   IR         (INPUT)    REGION NO.
C              IDT        (INPUT)    DATA TYPE FOR 302 TABLES
C              TBLS       (INPUT)    TABLE STORAGE ARRAY
C              T          (INPUT)    TEMPERATURE
C              P          (OUTPUT)   PRESSURE
C              E          (OUTPUT)   ENERGY
C              IFL        (OUTPUT)   OUTPUT FLAG
C                                    =1 FOR SUCCESS
C                                    =0 FOR FAILURE
C
C  REMARKS      NONE
C
C  EXTERNALS    T4EOSA
C
C  PROGRAMMER   J.ABDALLAH,JR.
C
C  DATE         5 JULY 1979
C
C-------------------------------------------------------------------------
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      LEVEL 2,TBLS
      COMMON/S2DIR/LCMX,NREG,LCFW(ML,1)
      DIMENSION TBLS(1)
      COMMON/SESIN/IRXX,IDTX,RX,EX,IBR,IFLX
      COMMON/SESOUT/ZP(3),ZT(3)
      IBR=0
      IFLX=1
      RX=R
      IRXX=IR
      IDTX=IDT
      LOC=LCFW(IR,IDT)
      NR=TBLS(LOC+2)
      NE=TBLS(LOC+3)
C . . GET INITIAL GUESS ON ENERGY
C . . FIND CLOSEST DENSITY INDEX
      LOCX=LOC+4
      IXR=1
      DELS=ABS(R-TBLS(LOCX))
      IF(NR.EQ.1) GO TO 20
      DO 10 J=2,NR
      LOCX=LOCX+1
      DEL=ABS(R-TBLS(LOCX))
      IF(DEL.GT.DELS) GO TO 10
      IRX=J
      DELS=DEL
 10   CONTINUE
C . . FIND THE ENERGY INDEX ASSOCIATED WITH THE CLOSEST TEMP
 20   LOCX=LOC+3+NR+NE+NR+IRX
```

```
      DELS=TBLS(LOCX)
      DELS=SHIFT(DELS,30)
      DELS=ABS(T-DELS)
      IEX=1
      IF(NE.EQ.1) GO TO 40
      DO 30 J=2,NE
      LOCX=LOCX+NR
      DEL=TBLS(LOCX)
      DEL=SHIFT(DEL,30)
      DEL=ABS(T-DEL)
      IF(DEL.GT.DELS) GO TO 30
      IEX=J
      DELS=DEL
   30 CONTINUE
C . . INITIAL GUESS ON ENERGY
   40 EX=TBLS(LOC+3+NR+IEX)+TBLS(LOC+3+NR+NE+IRX)
C . . ITERATE USING NEWTONS METHOD
      K=0
      IFL=1
   50 K=K+1
      IF(K.EQ.50) GO TO 90
      CALL T4EOSA
      E=EX
      P=ZP(1)
      TTEST=ABS(T-ZT(1))-1.0E-05*(ABS(T)+1.0E-02)
      IF(TTEST.LT.0.) RETURN
      D=-ZT(3)
      IF(D.EQ.C.0) GO TO 90
      EX=EX-(T-ZT(1))/D
      GO TO 50
   90 IFL=0
      RETURN
      END
```

```
      SUBROUTINE INV301(DSTR,LOC,RO,LDS)
C-----------------------------------------------------------------------
C
C  SUBROLTINE:   INV301(DSTR,LOC,RO,LDS)
C
C  PURPOSE:      INVERT DATA STRING OF TYPE 301 TO TYPE 302.
C
C  ARGUMENTS:    DSTR (INPUT) - TABLE STORAGE ARRAY
C                LOC  (INPUT) - STARTING LOCATION OF DATA STRING
C                               IN DSTR
C                RO   (INPUT) - APPROXIMATE DENSITY OF SOLID
C                LDS (OUTPUT) - LENGTH OF NEW DATA STRING
C
C  REMARKS:      DSTR CAN BE DECLARED LCM ON THE CDC 7600.
C                THIS ROUTINE OVERWRITES LOCATIONS FOLLOWING THE
C                DATA STRING.  IT EXPANDS THE STRING BY NR WORDS,
C                WHERE NR IS THE NUMBER OF DENSITIES.  IT ALSO
C                USES 3*NT WORDS AS TEMPORARY STORAGE, WHERE NT
C                IS THE NUMBER OF TEMPERATURES.
C
C  EXTERNALS:    ISRCHK, RATFN1.
C
C  PROGRAMMER:   G. I. KERLEY, T-4.
C
C  DATE:         4 OCTOBER 1977
C
C-----------------------------------------------------------------------
      LEVEL 2,DSTR
      DIMENSION OSTR(1)
      COMMON/INTORD/IFN
      COMMON/RTBLK1/LOCX,NR,LOCY,KY,JX,NT,INT,ET,Z(2)
      INT=1
      IFNS=IFN
      IFN=0
      NR = DSTR(LOC)
      NT = DSTR(LOC+1)
      LOCT = 2+NR+LOC
      LCEC = LOCT+NT
      LOCP = LCEC+NR
      LOCE = LOCP+NR*NT
      LOCN = LOCE+NR*NT
      IMAX = 2*NR*NT
      DO 1 I=1,IMAX
 1    DSTR(LOCN-I) = DSTR(LOCN-I-NR)
      DO 2 I=1,NR
      JJ = LOCE+I-1
      Q = 1.E-12*ABS(DSTR(JJ))
      DSTP(LCEC+I-1) = DSTR(JJ)
      DSTR(JJ) = 0.
      DO 2 J=2,NT
      JJ = JJ+NR
      DSTR(JJ) = DSTR(JJ)-DSTR(LCEC+I-1)
      IF(DSTR(JJ)-DSTR(JJ-NR).LT.Q) DSTR(JJ)=DSTR(JJ-NR)+Q
 2    CONTINUE
      I = ISRCHK(RO,DSTR(LOC+3),NR-2,1,0)+1
      DO 3 J=1,NT
      DSTR(LOCN+J-1) = DSTR(LOCT+J-1)
 3    DSTR(LOCT+J-1) = DSTR(LOCE+I-1+NR*(J-1))
      DO 5 I=1,NR
      LOCX = LOCE+I-1
```

```
      DO 4 J=1,NT
      ET = OSTR(LOCT+J-1)
      JX = ISRCHK(ET,DSTR(LOCX+NR),NT-2,NR,0)+1
      LOCY = LOCP+I-1
      KY = NR
      CALL RATFN1
      DSTR(LOCN+NT+J-1) = Z(1)
      LOCY = LOCN
      KY = 1
      CALL RATFN1
    4 DSTR(LOCN+NT+NT+J-1) = Z(1)
      DO 5 J=1,NT
      DSTR(LOCP+I-1+NR*(J-1)) = DSTR(LOCN+NT+J-1)
    5 DSTR(LOCX+NR*(J-1)) = DSTR(LOCN+NT+NT+J-1)
      LDS = LOCN-LOC
      IFN=IFNS
      RETURN
      END
```

```
      SUBROUTINE T4EOSA
C-----------------------------------------------------------------------
C
C  SUBROUTINE   T4EOSA
C
C  PURPOSE      TO COMPUTE A DENSITY SCALED,ENERGY SHIFTED,
C               ECUATION OF STATE AUGMENTED BY A ANALYTIC
C               PRESSURE RAMP
C
C               CCMMON/SESIN/IR,IDT,R,E,IBR,IFL
C               COMMON/SESOUT/P(3),T(3)
C               IR    (INPUT)   REGION. NO.
C               IDT   (INPUT)   DATA TYPE CORRESPONDING TO ENERGY BASED
C                               (TYPE 302) SESAME TABLE
C               R     (INPUT)   DENSITY
C               E     (INPUT)   INTERNAL ENERGY
C               P(1)  (OUTPUT)  PRESSURE
C               P(2)  (OUTPUT)  DENSITY DERIVATIVE OF PRESSURE
C               P(3)  (OUTPUT)  ENERGY DERIVATIVE OF PRESSURE
C               T(1)  (OUTPUT)  TEMPERATURE
C               T(2)  (OUTPUT)  DENSITY DERIVATIVE OF TEMP
C               T(3)  (OUTPUT)  ENERGY DERIVATIVE OF TEMP
C               IBR   (INPUT)   =0 TO OUTPUT BOTH P AND T
C                               =1 TO OUTPUT P ONLY
C                               =2 TO OUTPUT T ONLY
C               IFL   (IN/OUT)  INPUT
C                               0=CHOOSE BETWEEN RAMP AND TABLES
C                               1=FORCE USE OF TABLES
C                               OUTPUT
C                               0=PRESSURE COMPUTED FROM RAMP
C                               1=PPESSURE COMPUTED FROM TABLES
C                               NOTE THAT THIS FLAG CAN BE USED
C                               TO SATISFY REVERSIBILITY CONDITIONS.
C
C  REMARKS      COMMON/EOSCOM/ MUST BE SUPPLIED BY THE USER
C               PROGRAM. NREG IS THE NUMBER OF REGIONS. THE ARRAYS
C               CCNTAIN VALUES FOR PARAMETERS IN EACH REGION,THEY ARE
C               SR(IR) - DENSITY SCALE FACTOR FOR REGION IR
C               ES(IR) - ENERGY SHIFT
C               RO(IR) - RHOO FOR MATERIAL
C               A1(IR) - RAMP PARAMETER IF 0 THEN NO RAMP
C               A2(IR) - RAMP PARAMETER
C               A3(IR) - RAMP PARAMETER
C               EM(IR) - MELT ENERGY
C
C  EXTERNALS    T4DATI
C
C  PROGRAMMER   J.ABDALLAH,JR.
C
C  DATE         14 JUNE 1979
C
C-----------------------------------------------------------------------
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VD(ML),PD
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRD(ML),
     +MAT(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)
      COMMON/EOSCOM/SR(ML),ES(ML),
```

```
     $                   A1(ML),A2(ML),A3(ML),EM(ML),IRV(ML)
      COMMON/SESIN/IR,IDT,R,E,IBR,IFL
      COMMON/SESOUT/P(3),T(3)
      DIMENSION RO(ML)
      EQUIVALENCE(RO,ROW)
C . . SCALE DENSITY
      RSAVE=R
      R=SR(IR)*R
C . . SHIFT AND SCALE ENERGY
      ESAVE=E
      E=(ES(IR)+E)/SR(IR)
C . . COMPUTE EOS FROM TABLES
      CALL T4DATI
C . . IF FORCED TABLES - RETURN
      IF(IFL.EQ.1) RETURN
      IF(IBR.NE.2) GO TO 5
C . . TEMP ONLY
      IFL=1
      RETURN
 5    IF(A1(IR).NE.0.0) GO TO 10
C . . NO RAMP INPUT RETURN
      IFL=1
      RETURN
 10   IF(ESAVE.LE.EM(IR)) GO TO 20
C . . ENERGY IS GREATER THAN THE MELT ENERGY - RETURN
      IFL=1
      RETURN
C . . COMPUTE RAMP PRESSURE
 20   COMP=RSAVE/RO(IR)
      P1=A1(IR)*(COMP-1.0)
      DPR1=A1(IR)/RO(IR)
      IF(A2(IR).LE.0.0) GO TO 25
      P2=A2(IR)*(COMP-A3(IR))
      DPR2=A2(IR)/RO(IR)
      IF(P1.LT.P2) GO TO 25
      P1=P2
      DPR1=DPR2
 25   IF(P1.LT.P(1)) GO TO 30
C . . RAMP USED FOR PRESSURE
      IFL=0
      P(1)=P1
      P(2)=DPR1
      P(3)=0.0
      RETURN
C . . PRESSURE FROM TABLES
 30   IFL=1
      RETURN
      END
```

```
      SUBROUTINE PERTC8(IR,TBLS,ZBAR,ABAR)
C
C     ROUTINE TO PERTURB A 301-LIKE EOS TABLE BY MEANS OF A GAUSSIAN
C     BUMP ON THE ISOTHERMS.  ONE MAY ALSO INCLUDE A HARDNESS
C     FACTOR FOR THE PRESSURE AND ENERGIES.
C
      LEVEL 2,TBLS
      PARAMETER (MCL=500,ML=21,NGC=19,MLGC=NGC*ML,MLDWDT=20*ML,
     +NUMV=10,MQL=((NUMV+1)/3+1)*MCL+100,NDW=20,NCF=8,
     +MXDUMP=30,NDX=2*MXDUMP+2,MTAB=1,NTAB=MTAB*3742
     +,NSM=4,NWPM=3728,NSD=NSM*NWPM+132,ML2=100)
      COMMON/INIT/DTO(ML),XMU(ML),YO(ML),XL(ML),XV(ML),NV(ML),VO(ML),PO
     +(ML),TO(ML),ROW(ML),JMIN(ML2),JMAX(ML2),IBRN(ML),PLAP(ML),DRO(ML),
     +MAI(ML),UO(ML),UT(ML),DTCF(ML),QO(ML),TMLT(ML),TMC(ML)
      COMMON/S2DIR/LCMX,NREG,LCFW(ML,1)
      COMMON/T4PERT/LPERT(5),ZETA(5),VLOW(5),VHI(5),HARD(5)
      DIMENSION RO(ML)
      EQUIVALENCE (RO,ROW)
      DIMENSION TBLS(1)
      RHOZRO = RO(IR)
      IF(RHOZRO.EQ.0.0) RHOZRO = TBLS(2)
C
C     DOES THIS EOS GET PERTURBED?
C
      IM = ABS(TBLS(1))
      DO 200 LI=1,5
      IF(IM.EQ.IABS(LPERT(LI))) GO TO 202
  200 CONTINUE
      GO TO 299
  202 CONTINUE
C
C     DO PERTURBATION -- PRELIMINARY STUFF
C
      VINLOW = 1.0/VLOW(LI)
      VINHI  = 1.0/VHI(LI)
      CEN = 0.5*(VINLOW + VINHI)
      SIG = 2.0/(VINLOW - VINHI)
      LCEP = TBLS(3)
      LTEP = TBLS(4)
      C1 = ZETA(LI)*ZBAR*(8.3144E-03)/ABAR
C
C     FIND THE COMPRESSION JUST LESS THAN CEN
C
      RHOCEN = RHOZRO*CEN
      DO 210 LCEN = 1,LCEP
      IF(TBLS(4+LCEN).GT.RHOCEN) GO TO 211
  210 CONTINUE
  211 LCEN = LCEN - 1
C
C     FIRST COMPUTE BIAS FOR PRESSURE AND ENERGY LOOKUP IN TBLS
C
      NPRSK = 4 + LCEP + LTEP
      NERSK = NPRSK + LCEP*LTEP
      ICEN = NPRSK + LCEN + 7*LCEP
C
C     ORIGINAL PRESSURE AT OR NEAR CENTER OF GAUSSIAN
C
      PCEN = TBLS(ICEN)
C
C     FIND THE REGION THAT NEEDS MODIFYING
```

```
C     LOWEST COMPRESSION INDEX = IPL
      RINHI = RHOZRO*VINHI
      DO 212 IPL=1,LCEP
      IF(TBLS(4+IPL).GT.RINHI) GO TO 214
  212 CONTINUE
  214 IPL = IPL - 1
C
C     HIGHEST COMPRESSION INDEX = IPH
C
      RINLOW = RHOZRO*VINLOW
      DO 215 IPH=IPL,LCEP
      IF(TBLS(4+IPH).GE.RINLOW) GO TO 216
  215 CONTINUE
  216 CONTINUE
C
C     TABLE REPLACEMENT WITH PERTURBED VALUES
C
      DO 230 JP=IPL,IPH
C     THIS IS THE COMPRESSION LOOP
      RHO = TBLS(4+JP)
      ETA = RHO/RHOZRO
      F = EXP(-(SIG*(CEN-ETA))**2)
      DO 231 JT=1,LTEP
C     THIS IS THE TEMPERATURE LOOP
      T = TBLS(4+LCEP+JT)
C     IDEAL GAS SHAPED BY A GAUSSIAN
      PHAT = F*C1*RHO*T
      MSKIP = JP + (JT-1)*LCEP
      IPX = NPRSK + MSKIP
      IEX = NERSK + MSKIP
      TBLS(IPX) = HARD(LI)*TBLS(IPX) + PHAT
      TBLS(IEX) = HARD(LI)*TBLS(IEX)
  231 CONTINUE
  230 CONTINUE
  299 CONTINUE
      RETURN
      END
```

000014 COMPLETE R 2 T14 U R 080501 12/01/80 09:51:07 BOX T14 SESAME LIST 1201R0951 0024

## V. SAMPLE PROBLEMS

To aid users in setting up problems in HYDROX, we have included a set of sample problems. Each of the problems is concerned with the use of HE in contact with a metal plate. The first two sample problems consist of an aluminum plate striking a piece of PBX-9404 treated in the Forest Fire HE model. As shown in the results, the 1-mm-thick plate drives the PBX-9404 to a full detonation, whereas the 0.5-mm driven system does not proceed to full detonation. The third problem is a 3-cm piece of PBX-9404 treated in the buildup HE model driving an aluminum plate. Spall layers are allowed to form in the aluminum and are evident in the distance-time plot. The last problem consists of 5 cm of Comp B in contact with an aluminum plate.

All plots were obtained by using the graphics code GAS, and for the distance-time plots the code OTGAS was used as an intermediate step. Further details on the graphics may be found in LASL Utility Routine LTSS-523.

### A. 1 mm of Aluminum Impacting PBX-9404 Using the Forest Fire Model

Input File DATA

```
P$INP NM=2,TEND=1.,NG=20,IALPH=1,RO=.5,
      NMAX=35,NADD=2,
      LABEL=27H FOREST FIRE 9404/AL DRIVER $
P$SU MAT=2,R2=.4,NCI=30,UO=-.1,ME=1 $
P$ESC NV=2,XV=.3 $
P$SU MAT=24,R2=0.,NCI=120 $
```

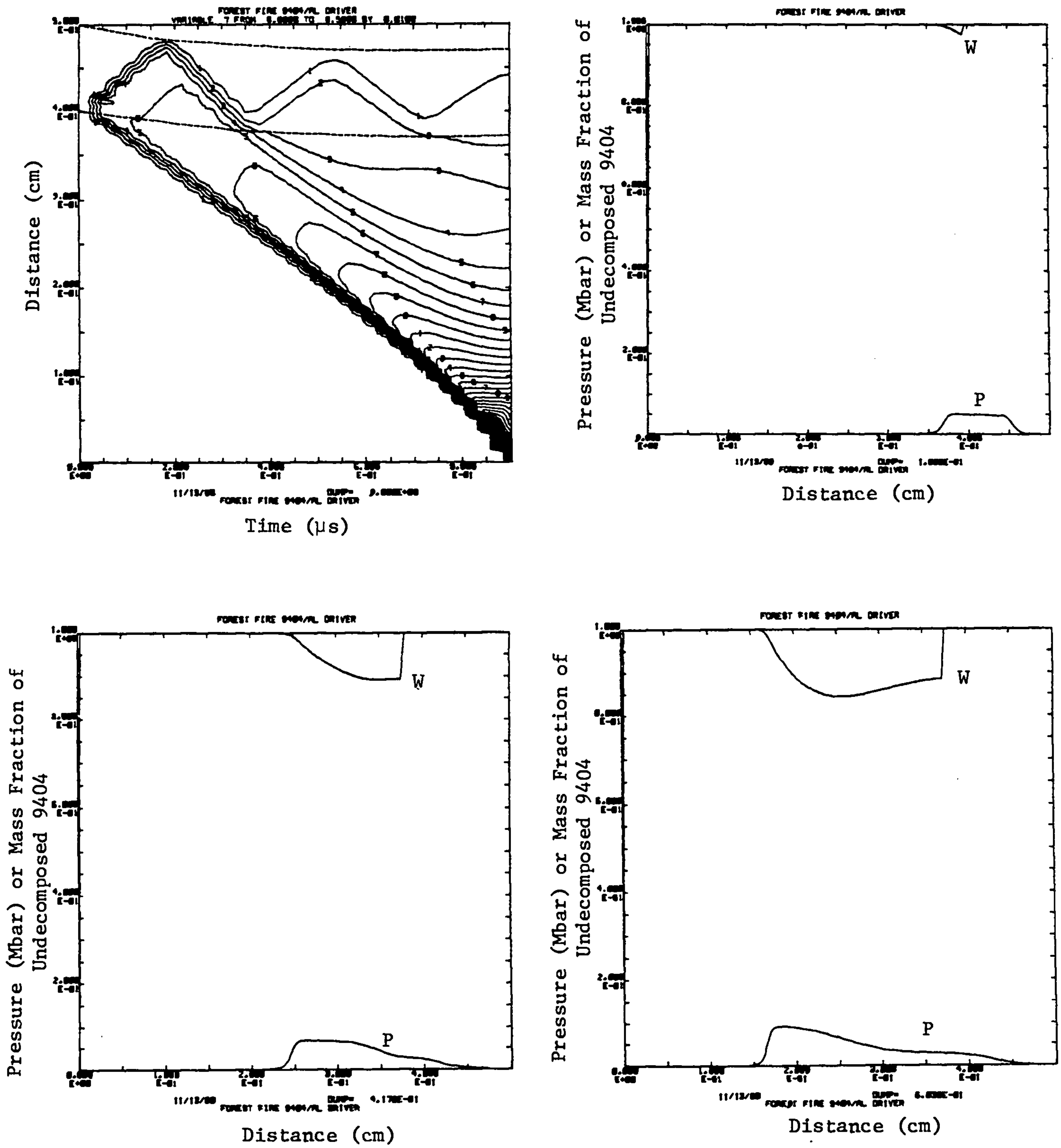



Fig. 9.
Pressure contours (0.010 Mbar).

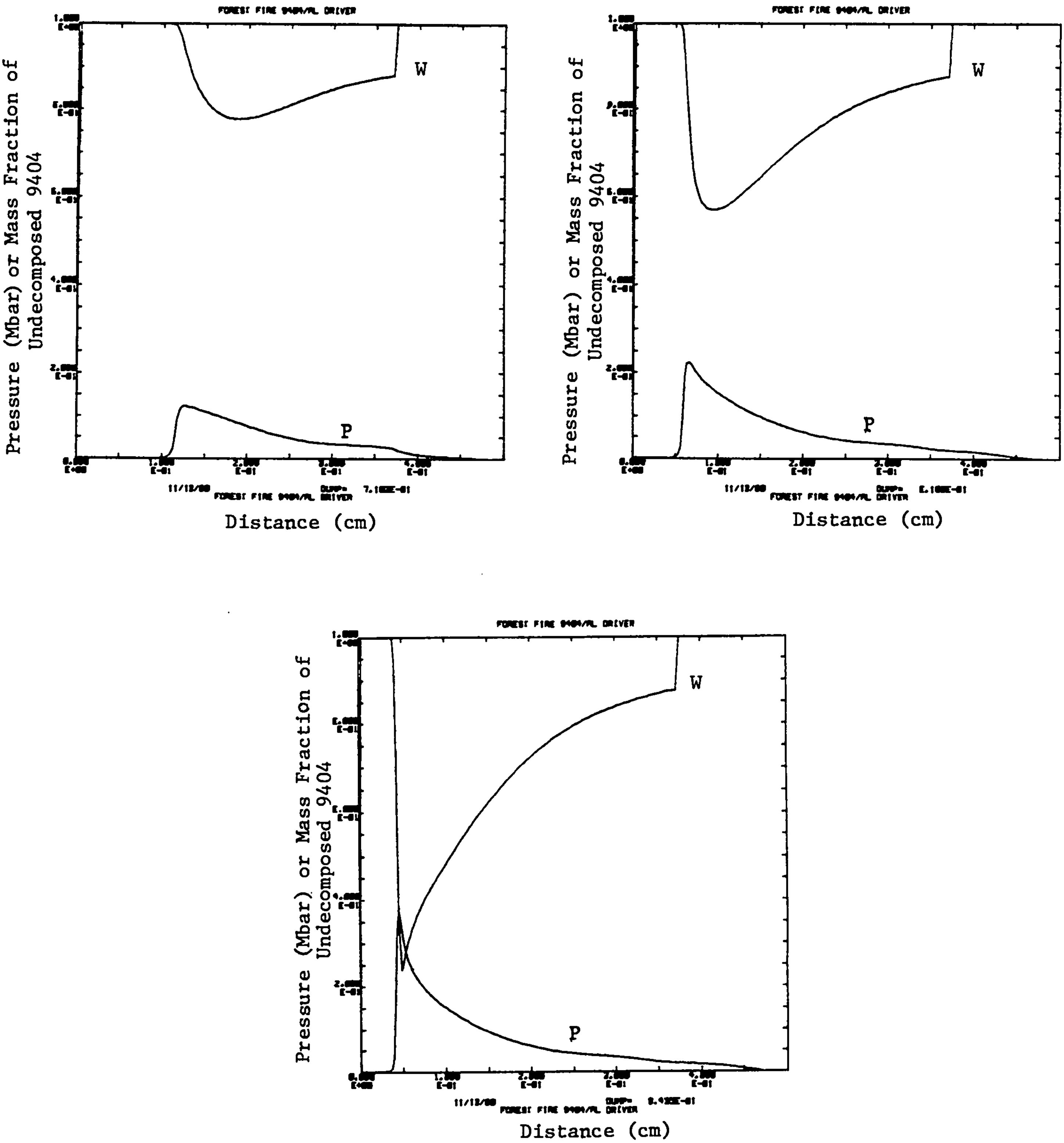

FOREST FIRE 9404/PL DRIVER
W
P
Pressure (Mbar) or Mass Fraction of Undecomposed 9404
Distance (cm)
FOREST FIRE 9404/PL DRIVER
W
P
Pressure (Mbar) or Mass Fraction of Undecomposed 9404
Distance (cm)
FOREST FIRE 9404/PL DRIVER
W
P
Pressure (Mbar) or Mass Fraction of Undecomposed 9404
Distance (cm)

Fig. 9. (cont)

## B. 0.5 mm of Al Impacting 9404 Using the Forest Fire Model

### Input File DATA

```
P$INP NM=2,TEND=1.,NG=20,IALPH=1,RO=.45,
      NMAX=35,NADD=2,
      LABEL=27H FOREST FIRE 9404/AL DRIVER $
P$SU MAT=2,R2=.4,NCI=15,UO=-.1,ME=1 $
P$ESC NV=2,XV=.3 $
P$SU MAT=24,R2=0.,NCI=120 $
```

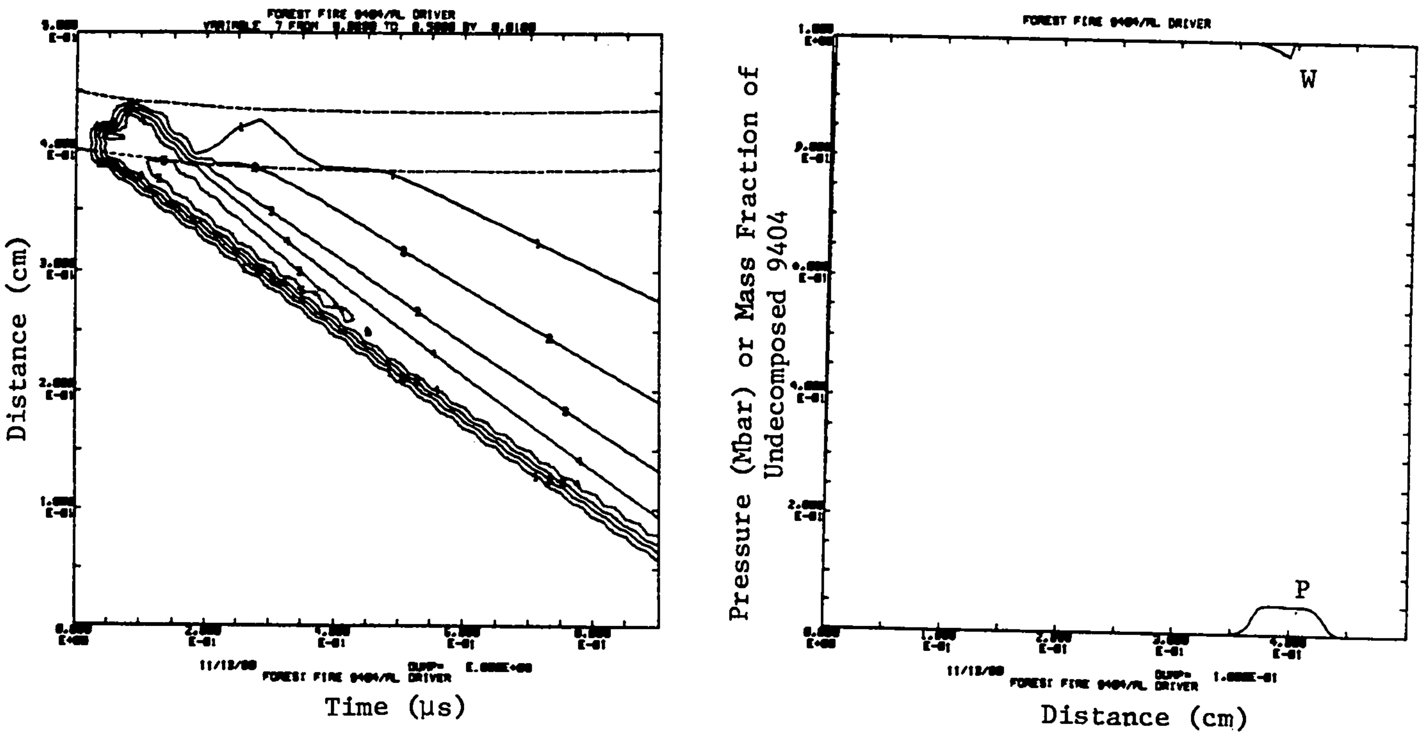



Fig. 10.
Pressure contours (0.010 Mbar)

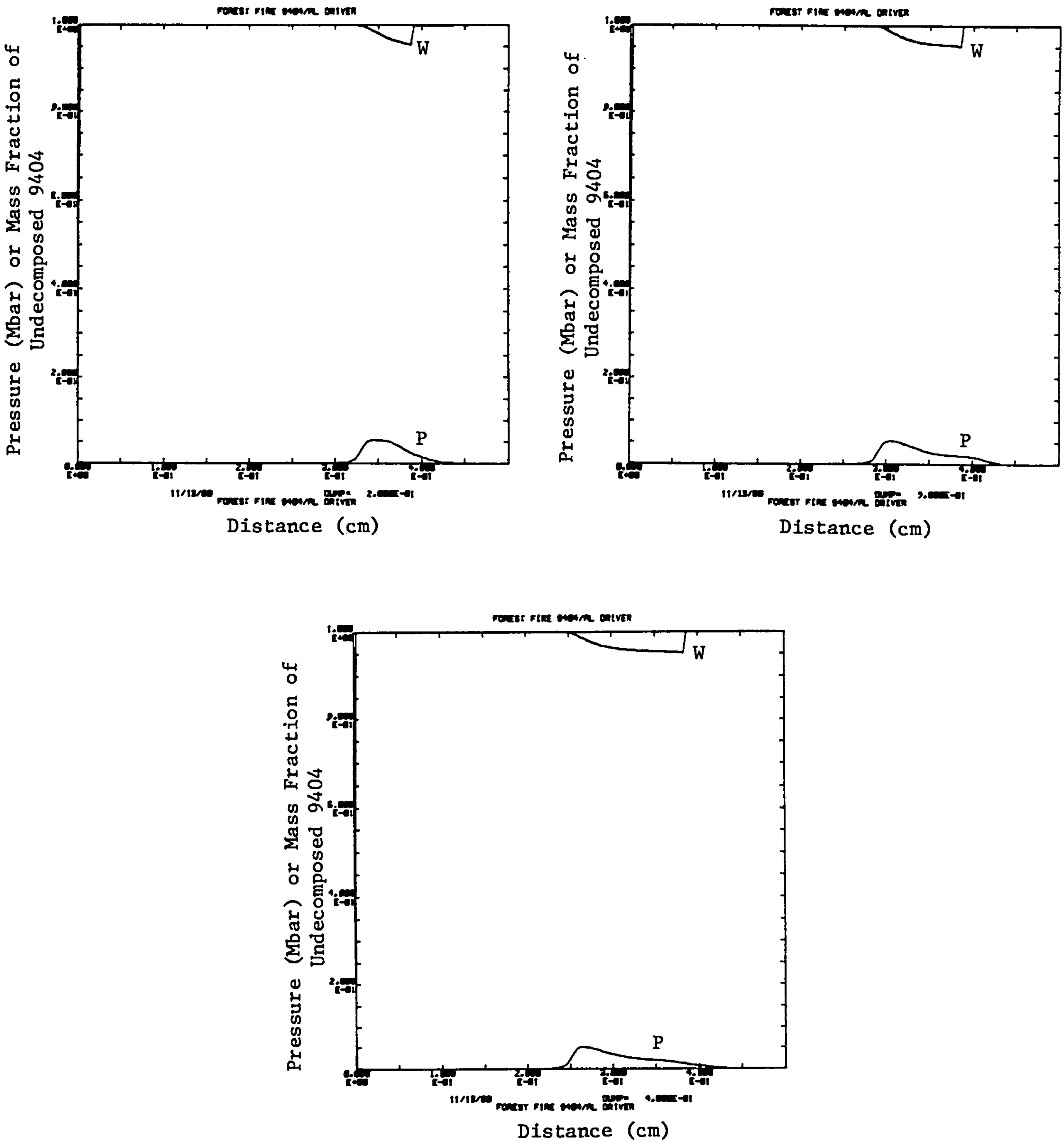

FOREST FIRE 9404/RL DRIVER
Pressure (Mbar) or Mass Fraction of Undecomposed 9404
W
P
Distance (cm)
FOREST FIRE 9404/RL DRIVER
Pressure (Mbar) or Mass Fraction of Undecomposed 9404
W
P
Distance (cm)
FOREST FIRE 9404/RL DRIVER
Pressure (Mbar) or Mass Fraction of Undecomposed 9404
W
P
Distance (cm)

Fig. 10. (cont)

## C. 3 cm of 9404 in the Buildup Model Pushing 1 cm of Al

### Input File DATA

```
P$INP NM=2,TEND=10.,NG=10,IALPH=1,RO=4.0,NDF=2,
     LABEL=17H BUILD UP 9404/AL $
P$SU IEOS=2,MAT=22,R2=1.,NCI=100,ME=1 $
P$ESC IBRN=3,BUD=.4,XV=3. $
P$BURN VCJ=.88,E=.187234 $
P$SU MAT=2,R2=0.,NCI=50,ME=1 $
P$ESC YO=.00367,XMU=.256,XV=3.5 $
```

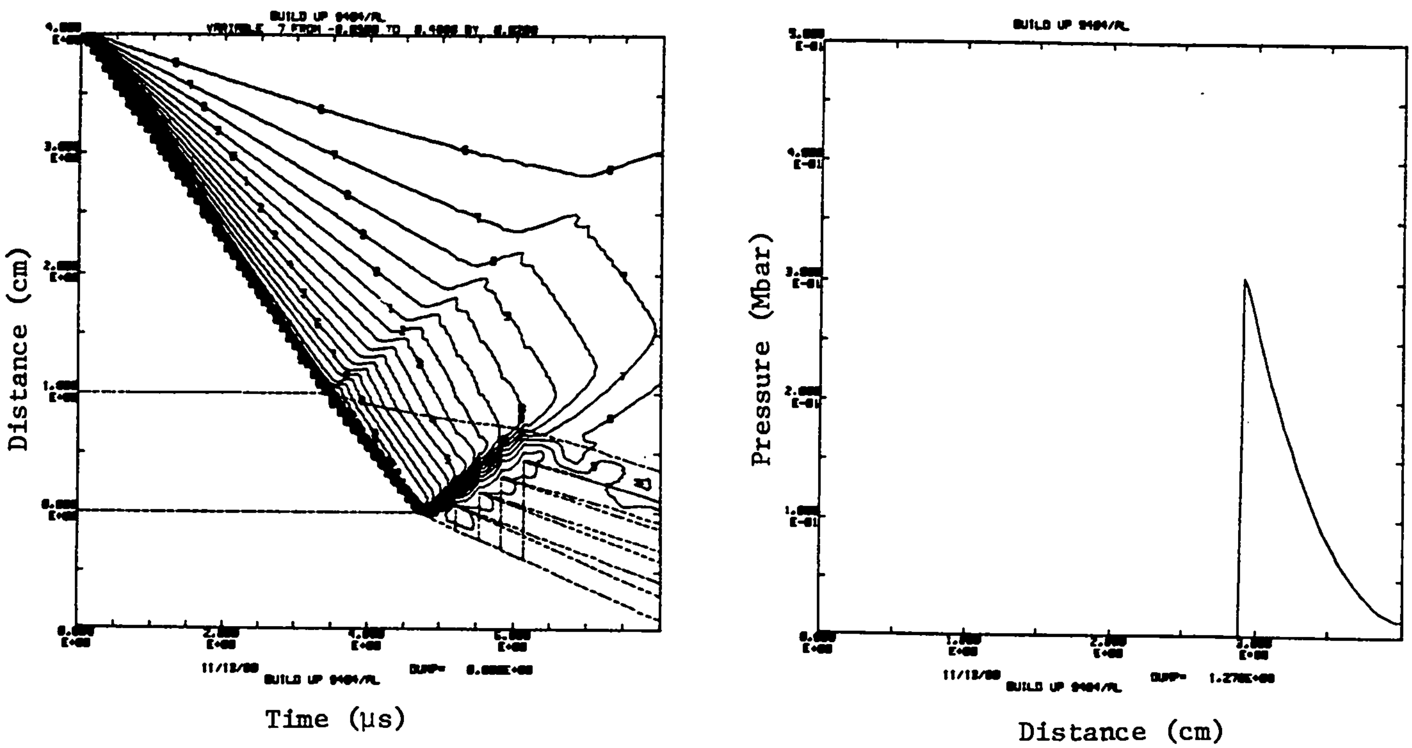



Fig. 11.
Pressure contours (0.020 Mbar) and Spall Layers.

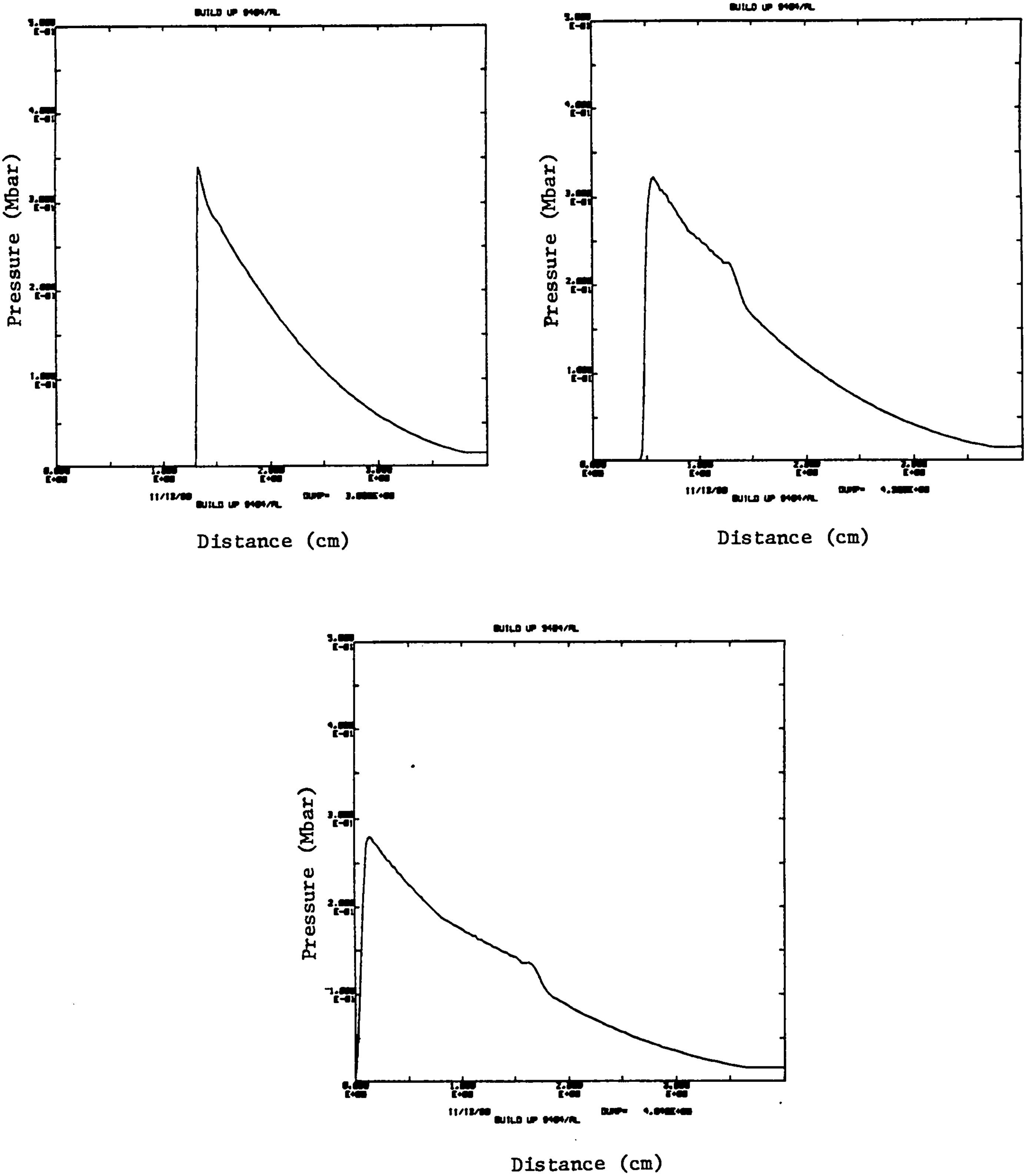
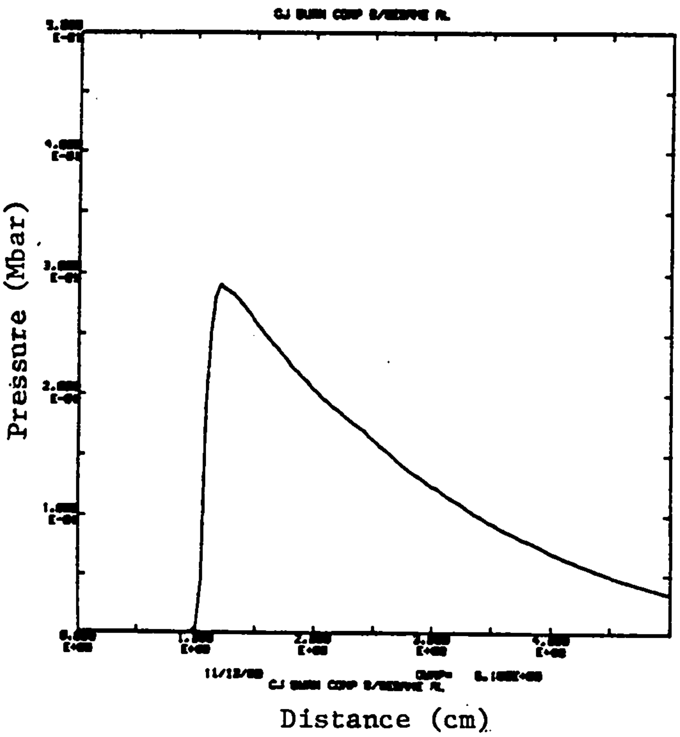

Pressure (Mbar)
Distance (cm)
Pressure (Mbar)
Distance (cm)
Pressure (Mbar)
Distance (cm)

Fig. 11. (cont)

D. 5 cm of Comp. B in the C-J Volume Burn Model Pushing 1 cm of Al with a SESAME Equation of State

Input File DATA

```
P$INP NM=2,RO=6.,IALPH=1,NG=20,TEND=8.,UI=-.213564,UF=.15,
      NMAX=10,NADD=2,
      LABEL=26H CJ BURN COMP B/SESAME AL $
P$SU MAT=19,R2=1.,NCI=100,ME=1 $
P$ESC IBRN=2,XV=2.5 $
P$BURN VCJ=.428513 $
P$SU IEOS=4,MAT=3710,NCI=25,R2=0.,ME=1 $
P$ESC XV=3. $
```



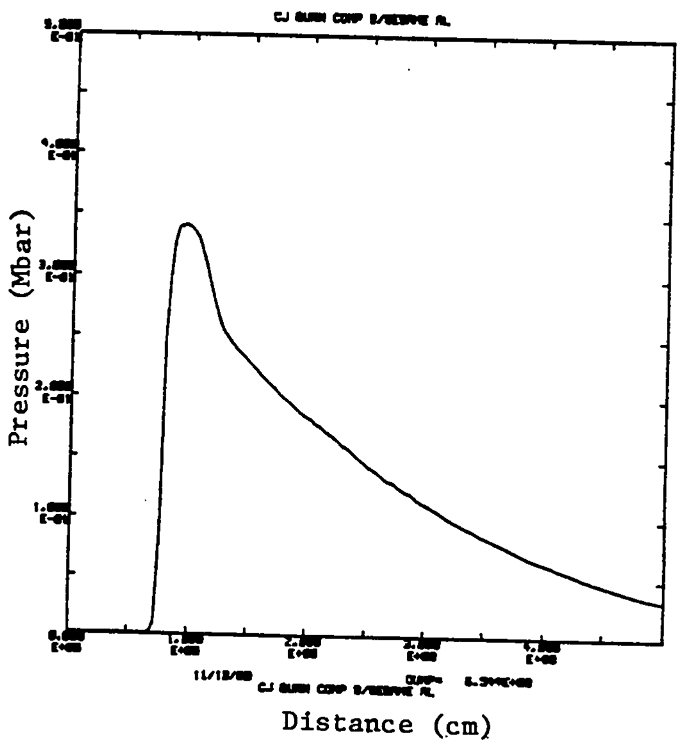




Fig. 12.

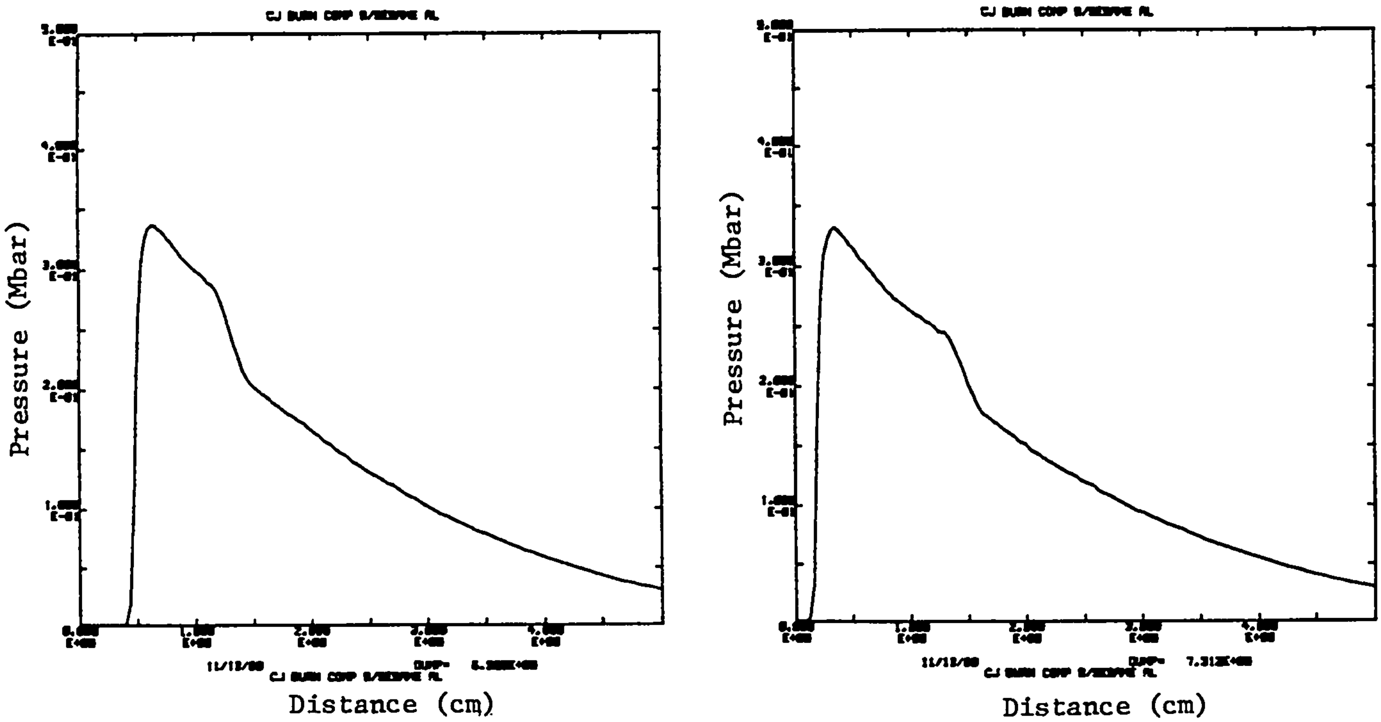



Fig. 12. (cont)

## VI. HMLB AND SESAME EQUATION-OF-STATE LIBRARIES

In this section we discuss the use of the library HMLB for the HOM EOS and the SESAME tabular EOS library. Since the use of the HOM EOS can require the specification of many constants, we have provided a library of typical materials for metals, gases, and reacting materials. Any particular EOS constant in HMLB may be changed in HYDROX on the ESC NAMELIST since the library values are read first, and then the values specified in ESC are changed. Unspecified variables in ESC are thus defaulted to the values in HMLB. In Sec. A we give a list of the materials on HMLB, a cross-reference table of HYDROX and SIN variable names, and a listing of all the HMLB constants. Further information on the HOM EOS may be found in Ref. 2 of Sec. I.

HYDROX includes the capability of accessing the SESAME tabular EOS library and uses special subroutines for reading the library and doing the required numerical interpolation. In Sec. B we have merely listed the materials that are currently available. Reference 4 of Sec. I contains additional information about the SESAME library.

### A. HMLB

HMLB is a library of constants for 29 different materials for use by the HOM equation of state in HYDROX. At LASL, HMLB may be obtained by the LTSS command:

```
MASS GET /HYDROX/HMLB
```

Before running HYDROX, HMLB must be obtained as a local file if the HOM EOS is used; IEOS=1 and MAT≠0 in the SU NAMELIST causes a search of HMLB for the material number specified by MAT.

## 1. Materials in the HMLB HOM EOS Library

```
 1  BE
 2  2024 AL
 3  NI
 4  COPPER
 5 STEEL
 6  TA
 7  AU
 8  LEAD
 9  U
10 301 POLYIMIDE
11  PLEXG
12  CH2
13  FOAM
14 AIR 1
15 AIR 2
16  BAHE
17 NQ ARRHENIUS
18 NQ FOREST FIRE
19  COMP B SHARP SHOCK
20  COMP B CJ(ARRH)
21 9404 CJ
22 9404 BUILD UP EOS
23 9404 GAMMA LAW
24 9404 FOREST FIRE
25 TATB/WAX FF F(P)
26 X0290 FF PCJ=.285
27 NQ/ESTANE 95/5 FF(T)
30 NQ FF RH ZERO ORDER
31  TNT
```

## 2. Cross-Reference Table Between HYDROX Variables and SIN Variables for HMLB

| HYDROX | SIN |
|---|---|
| MAT | NMAT |
| --- | IEXP |
| IBRN | IBRN |
| NV | IVIS |
| XV | VFACT |
| ROW | RHOØ |
| PØ | PØ |
| TØ | TØ |
| ZI | EØ |
| UØ | UØ |
| C1 | SOL(1) |

| HYDROX | SIN |
|---|---|
| S1 | SOL(2) |
| SWV | SOL(3) |
| C2 | SOL(4) |
| S2 | SOL(5) |
| FS | SOL(6) |
| GS | SOL(7) |
| HS | SOL(8) |
| SI | SOL(9) |
| SJ | SOL(10) |
| GAMMA | SOL(11) |
| CV | SOL(12) |
| VØ | SOL(13) |
| ALP | SOL(14) |
| SP | SOL(15) |
| USP | SOL(16) |
| TØ | SOL(17) |
| PØ | SOL(18) |
| YØ | SOL(19) |
| XMU | SOL(20) |
| PLAP | SOL(21) |
| --- | SOL(22) |
| VMN | SOL(23) |
| WØ | WØ |
| Z | Z |
| E | E |
| VCJ | VCJ |

| HYDROX | SIN | |
|---|---|---|
| --- | DCJ | |
| --- | BCJUP | |
| ND | NDWDT | |
| PCJ | BPCJ | |
| PM | AMINP | |
| DWDT | DWDT | |
| GC | GAS | |
| A | GAS(1) | When the Barnes EOS is used |
| BR | GAS(2) | |
| BA | GAS(3) | |
| VBØ | GAS(4) | |
| VBSW | GAS(5) | |
| BUA | BUA | |
| BUB | BUB | |
| BUMAX | BUMAX | |
| BUDV | BUDV | |

## 3. Listing of the EOS Constants in HMLB

```
$N
 NMAT    =          1,
 NN      =         -1,
$END
          *****  BE                    *****

$DAT
 IEXP    =          0,
 IBRN    =          0,
 IVIS    =          0,
 VFACT   =    2.0000000000000E+00,
 RHO0    =    1.8450000000000E+00,
 P0      =    1.0000000000000E-06,
 T0      =    3.0000000000000E+02,
 E0      =    0.                 ,
 U0      =    0.                 ,
 SOL     =    7.9750000000000E-01,    1.0910000000000E+00,
              1.0000000000000E-02,    0.                 ,
              0.                 ,    1.4949229523200E+01,
              3.4866986350100E+01,    3.9874001880200E+01,
              9.4841753554900E+00,   -1.9607277845000E+00,
              1.1800000000000E+00,    4.7400000000000E-01,
              5.4200542005400E-01,    1.2330000000000E-05,
              0.                 ,    0.                 ,
              3.0000000000000E+02,    1.0000000000000E-06,
              0.                 ,    0.                 ,
              0.                 ,    1.0000000000000E-06,
              0.                 ,
 W0      =    1.0000000000000E+00,
 Z       =    0.                 ,
 E       =    0.                 ,
 VCJ     =    0.                 ,
 DCJ     =    0.                 ,
 BCJUP   =    0.                 ,
 NDWDT   =          0,
 BPCJ    =    0.                 ,
 AMINP   =    0.                 ,
 DWDT    =    0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
 GAS     =    0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,
 BUA     =    0.                 ,
 BUB     =    0.                 ,
 BUMAX   =    0.                 ,
 BUDV    =    0.                 ,
$END
```

```
$N
 NMAT   =        2,
 NN     =       -1,
$END
          *****  2024 AL              *****


$DAT
 IEXP   =         0,
 IBRN   =         0,
 IVIS   =         0,
 VFACT  =   2.0000000000000E+00,
 RHOO   =   2.7850000000000E+00,
 PO     =   1.0000000000000E-06,
 TO     =   3.0000000000000E+02,
 EO     =   0.                 ,
 UO     =   0.                 ,
 SOL    =   5.3500000000000E-01,   1.3500000000000E+00,
            1.0000000000000E-01,   0.                 ,
            0.                 ,  -7.9611586687400E+01,
           -3.1753356163300E+02,  -4.3852537153300E+02,
           -2.6424824896000E+02,  -5.7973496473200E+01,
            1.7000000000000E+00,   2.2000000000000E-01,
            3.5906642728900E-01,   2.4000000000000E-05,
            7.1400000000000E-02,   5.0000000000000E-02,
            3.0000000000000E+02,   1.0000000000000E-06,
            0.                 ,   0.                 ,
            5.0000000000000E-02,   1.0000000000000E-06,
            0.                 ,
 WO     =   1.0000000000000E+00,
 Z      =   4.6000000000000E+03,
 E      =   9.5000000000000E+02,
 VCJ    =   0.                 ,
 DCJ    =   0.                 ,
 BCJUP  =   0.                 ,
 NDWDT  =         0,
 BPCJ   =   0.                 ,
 AMINP  =   0.                 ,
 DWDT   =   0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
 GAS    =   0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,
 BUA    =   0.                 ,
 BUB    =   0.                 ,
 BUMAX  =   0.                 ,
 BUDV   =   0.                 ,
$END
```

```
$N
 NMAT   =         3,
 NN     =        -1,
$END
        *****  NI                  *****

$DAT
 IEXP   =         0,
 IBRN   =         0,
 IVIS   =         0,
 VFACT  =    2.0000000000000E+00,
 RHOO   =    8.8600001000000E+00,
 PO     =    1.0000000000000E-06,
 TO     =    3.0000000000000E+02,
 EO     =    0.                 ,
 UO     =    0.                 ,
 SOL    =    4.6460000000000E-01,    1.4450000000000E+00,
             1.0000000000000E-02,    0.                 ,
             0.                 ,   -2.0354981593800E+03,
            -3.7098532585100E+03,   -2.5122620102000E+03,
            -7.5077048497800E+02,   -8.3468184762400E+01,
             1.8900000000000E+00,    1.0500000000000E-01,
             1.1286681715600E-01,    1.3000000000000E-05,
             0.                 ,    0.                 ,
             3.0000000000000E+02,    1.0000000000000E-06,
             0.                 ,    0.                 ,
             0.                 ,    1.0000000000000E-06,
             0.                 ,
 WO     =    1.0000000000000E+00,
 Z      =    0.                 ,
 E      =    0.                 ,
 VCJ    =    0.                 ,
 DCJ    =    0.                 ,
 BCJUP  =    0.                 ,
 NDWDT  =         0,
 BPCJ   =    0.                 ,
 AMINP  =    0.                 ,
 DWDT   =    0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
 GAS    =    0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,
 BUA    =    0.                 ,
 BUB    =    0.                 ,
 BUMAX  =    0.                 ,
 BUDV   =    0.                 ,
$END
```

```
$N
 NMAT    =          4,
 NN      =         -1,
$END
        *****  COPPER              *****


$DAT
 IEXP    =          0,
 IBRN    =          0,
 IVIS    =          0,
 VFACT   =    2.0000000000000E+00,
 RHOO    =    8.9030000000000E+00,
 PO      =    1.0000000000000E-06,
 TO      =    3.0000000000000E+02,
 EO      =    0.                 ,
 UO      =    0.                 ,
 SOL     =    3.9580000000000E-01,   1.4970000000000E+00,
              1.0000000000000E-03,   0.                 ,
              0.                 ,  -7.2269373246900E+02,
             -1.4234042687800E+03,  -1.0197498949400E+03,
             -3.1807912677400E+02,  -3.6471449881900E+01,
              2.0000000000000E+00,   9.3000000000000E-02,
              1.1232168931800E-01,   1.7670000000000E-05,
              1.0000000000000E-01,   1.2000000000000E-01,
              3.0000000000000E+02,   1.0000000000000E-06,
              3.0000000000000E-03,   4.6600000000000E-01,
              5.0000000000000E-02,   1.0000000000000E-06,
              0.                 ,
 WO      =    1.0000000000000E+00,
 Z       =    5.5700000000000E+03,
 F       =    1.3530000000000E+03,
 VCJ     =    0.                 ,
 DCJ     =    0.                 ,
 BCJUP   =    0.                 ,
 NDWDT   =          0,
 BPCJ    =    0.                 ,
 AMINP   =    0.                 ,
 DWDT    =    0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
 GAS     =    0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,   0.                 ,
              0.                 ,
 BUA     =    0.                 ,
 BUB     =    0.                 ,
 BUMAX   =    0.                 ,
 BUDV    =    0.                 ,
$END
```

```
$N
 NMAT   =          5,
 NN     =         -1,
$END
         ***** STEEL                    *****


$DAT
 IEXP   =          0,
 IBRN   =          0,
 IVIS   =          0,
 VFACT  =     2.0000000000000E+00,
 RHOO   =     7.9170000000000E+00,
 PO     =     1.0000000000000E-06,
 TO     =     3.0000000000000E+02,
 EO     =     0.                 ,
 UO     =     0.                 ,
 SOL    =     4.5800000000000E-01,     1.5100000000000E+00,
              1.0000000000000E-02,     0.                 ,
              0.                 ,    -3.8238258745300E+03,
             -7.0321195402400E+03,    -4.8267021389400E+03,
             -1.4667840211800E+03,    -1.6639161598300E+02,
              2.0000000000000E+00,     1.0700000000000E-01,
              1.2631047113800E-01,     1.1700000000000E-05,
              0.                 ,     0.                 ,
              3.0000000000000E-02,     1.0000000000000E-06,
              0.                 ,     0.                 ,
              0.                 ,     1.0000000000000E-06,
              0.                 ,
 WO     =     1.0000000000000E+00,
 Z      =     0.                 ,
 E      =     0.                 ,
 VCJ    =     0.                 ,
 DCJ    =     0.                 ,
 BCJUP  =     0.                 ,
 NDWDT  =          0,
 BPCJ   =     0.                 ,
 AMINP  =     0.                 ,
 DWDT   =     0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
 GAS    =     0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,     0.                 ,
              0.                 ,
 BUA    =     0.                 ,
 BUB    =     0.                 ,
 BUMAX  =     0.                 ,
 BUDV   =     0.                 ,
$END
```

```
$N
 NMAT   =          6,
 NN     =         -1,
$END
        *****  TA                    *****


$DATA
 IEXP   =          0,
 IBRN   =          0,
 IVIS   =          0,
 VFACT  =   2.0000000000000E+00,
 RHOO   =   1.6690000000000E+01,
 PO     =   1.0000000000000E-06,
 TO     =   3.0000000000000E+02,
 EO     =   0.                 ,
 UO     =   0.                 ,
 SOL    =   3.4140000000000E-01,   1.2010000000000E+00,
            1.0000000000000E-02,   0.                 ,
            0.                 ,  -2.7769236055300E+03,
           -4.0152531315500E+03,  -2.1587503349800E+03,
           -5.1221616258700E+02,  -4.5223258682200E+01,
            1.7000000000000E+00,   3.3000000000000E-02,
            5.9916117435600E-02,   1.6590000000000E-05,
            0.                 ,   0.                 ,
            3.0000000000000E+02,   1.0000000000000E-06,
            4.0000000000000E-03,   7.0500000000000E-01,
            5.0000000000000E-02,   0.                 ,
            0.                 ,
 WO     =   1.0000000000000E+00,
 Z      =   5.0000000000000E+03,
 E      =   3.2700000000000E+03,
 VCJ    =   0.                 ,
 DCJ    =   0.                 ,
 BCJUP  =   0.                 ,
 NDWDT  =          0,
 BPCJ   =   0.                 ,
 AMINP  =   0.                 ,
 DWDT   =   0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
 GAS    =   0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,
 BUA    =   0.                 ,
 BUB    =   0.                 ,
 BUMAX  =   0.                 ,
 BUDV   =   0.                 ,
$END
```

```
$N
 NMAT   =         7,
 NN     =        -1,
$END
         *****  AU                      *****


$DAT
 IEXP   =         0,
 IBRN   =         0,
 IVIS   =         0,
 VFACT  =    2.0000000000000E+00,
 RHOO   =    1.9240000000000E+01,
 PO     =    1.0000000000000E-06,
 TO     =    3.0000000000000E+02,
 EO     =    0.                 ,
 UO     =    0.                 ,
 SOL    =    3.0750000000000E-01,    1.5600000000000E+00,
             1.0000000000000E-02,    0.                 ,
             0.                 ,   -3.8649405250300E+03,
            -5.1941782941600E+03,   -2.6112896415500E+03,
            -5.8203419236900E+02,   -4.8455143294100E+01,
             2.0000000000000E+00,    3.1200000000000E-02,
             5.1975051970000E-02,    1.4200000000000E-05,
             0.                 ,    0.                 ,
             3.0000000000000E+02,    1.0000000000000E-06,
             0.                 ,    0.                 ,
             5.0000000000000E-02,    0.                 ,
             0.                 ,
 WO     =    1.0000000000000E+00,
 Z      =    5.0000000000000E+03,
 E      =    1.3360000000000E+03,
 VCJ    =    0.                 ,
 DCJ    =    0.                 ,
 RCJUP  =    0.                 ,
 NDWDT  =         0,
 BPCJ   =    0.                 ,
 AMINP  =    0.                 ,
 DWDT   =    0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
 GAS    =    0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,
 BUA    =    0.                 ,
 BUB    =    0.                 ,
 BUMAX  =    0.                 ,
 BUDV   =    0.                 ,
$END
```

```
$N
 NMAT   =        8,
 NN     =       -1,
$END
         *****  LEAD                 *****


$DAT
 IEXP   =        0,
 IBRN   =        0,
 IVIS   =        0,
 VFACT  =    2.0000000000000E+00,
 RHOO   =    1.1340000000000E+01,
 PO     =    1.0000000000000E-06,
 TO     =    3.0000000000000E+02,
 EO     =    0.                 ,
 UO     =    0.                 ,
 SOL    =    2.0280000000000E-01,    1.5170000000000E+00,
             1.0000000000000E-03,    0.                 ,
             0.                 ,    3.9729257221900E+02,
             4.4550742326900E+02,    1.6992751638400E+02,
             2.2953413952900E+01,    4.8665122596100E-01,
             2.0340000000000E+00,    3.0000000000000E-02,
             8.8183421516800E-02,    2.8370000000000E-05,
             1.0000000000000E-01,    2.0000000000000E-02,
             3.0000000000000E+02,    1.0000000000000E-06,
             0.                 ,    0.                 ,
             0.                 ,    1.0000000000000E-06,
             0.                 ,
 WO     =    1.0000000000000E+00,
 Z      =    3.5000000000000E+03,
 E      =    5.9900000000000E+02,
 VCJ    =    0.                 ,
 DCJ    =    0.                 ,
 BCJUP  =    0.                 ,
 NDWDT  =        0,
 BPCJ   =    0.                 ,
 AMINP  =    0.                 ,
 DWDT   =    0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
 GAS    =    0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,
 BUA    =    0.                 ,
 BUB    =    0.                 ,
 BUMAX  =    0.                 ,
 BUDV   =    0.                 ,
$END
```

```
$N
 NMAT   =          9,
 NN     =         -1,
$END
         *****  U                      *****

$DAT
 IEXP   =          0,
 IBRN   =          0,
 IVIS   =          0,
 VFACT  =     2.0000000000000E+00,
 RHOO   =     1.8910000000000E+01,
 PO     =     1.0000000000000E-06,
 TO     =     3.0000000000000E+02,
 EO     =     0.                 ,
 UO     =     0.                 ,
 SOL    =     2.5400000000000E-01,    1.5000000000000E+00,
              1.0000000000000E-04,    0.                 ,
              0.                 ,    8.1777341466300E+02,
              6.1009951139000E+02,    9.0904269245400E+01,
             -2.2037316277600E+01,   -4.8735636843100E+00,
              2.0000000000000E+00,    2.7600000000000E-02,
              5.2882072000000E-02,    1.1666000000000E-05,
              0.                 ,    0.                 ,
              3.0000000000000E+02,    1.0000000000000E-06,
              0.                 ,    0.                 ,
              0.                 ,    1.0000000000000E-06,
              0.                 ,
 WO     =     1.0000000000000E+00,
 Z      =     0.                 ,
 E      =     0.                 ,
 VCJ    =     0.                 ,
 DCJ    =     0.                 ,
 BCJUP  =     0.                 ,
 NDWOT  =          0,
 BPCJ   =     0.                 ,
 AMINP  =     0.                 ,
 DWOT   =     0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
 GAS    =     0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,
 BUA    =     0.                 ,
 BUB    =     0.                 ,
 BUMAX  =     0.                 ,
 BUDV   =     0.                 ,
$END
```

```
$N
 NMAT   =         10,
 NN     =         -1,
$END
        ***** 301 POLYIMIDE          *****


$DAT
 IEXP   =          0,
 IBRN   =          0,
 IVIS   =          2,
 VFACT  =     2.0000000000000E-03,
 RHOO   =     1.4140000000000E+00,
 PO     =     0.                 ,
 TO     =     3.0000000000000E+02,
 EO     =     0.                 ,
 UO     =     0.                 ,
 SOL    =     2.6500000000000E-01,    1.5100000000000E+00,
              4.4878000000000E-01,    4.9000000000000E-01,
              4.6860000000000E-01,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              1.0000000000000E+00,    2.5000000000000E-01,
              7.0721357850000E-01,    1.0000000000000E-05,
              0.                 ,    0.                 ,
              3.0000000000000E+02,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              4.4878000000000E-01,
 WO     =     1.0000000000000E+00,
 Z      =     0.                 ,
 E      =     0.                 ,
 VCJ    =     0.                 ,
 DCJ    =     0.                 ,
 BCJUP  =     0.                 ,
 NDWDT  =          0,
 BPCJ   =     0.                 ,
 AMINP  =     0.                 ,
 DWDT   =     0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
 GAS    =     0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,
 BUA    =     0.                 ,
 BUB    =     0.                 ,
 BUMAX  =     0.                 ,
 BUDV   =     0.                 ,
$END
```

```
$N
 NMAT   =        11,
 NN     =        -1,
$END
        *****  PLEXG                *****


$DAT
 IEXP   =          0,
 IBRN   =          0,
 IVIS   =          0,
 VFACT  =     2.0000000000000E+00,
 RHOO   =     1.1800000000000E+00,
 PO     =     1.0000000000000E-06,
 TO     =     3.0000000000000E+02,
 EO     =     0.                 ,
 UO     =     0.                 ,
 SOL    =     2.4320000000000E-01,    1.5785000000000E+00,
              1.0000000000000E-04,    0.                 ,
              0.                 ,    5.2938024350600E+00,
             -4.2495037136800E+00,   -1.5505557633200E+01,
             -3.0863807557200E+01,   -1.4670819373900E+01,
              1.0000000000000E+00,    3.5000000000000E-01,
              8.4745762700000E-01,    1.0000000000000E-04,
              1.0000000000000E+02,    1.0000000000000E+00,
              3.0000000000000E+02,    1.0000000000000E-06,
              0.                 ,    0.                 ,
              0.                 ,    1.0000000000000E-06,
              0.                 ,
 WO     =     1.0000000000000E+00,
 Z      =     0.                 ,
 E      =     0.                 ,
 VCJ    =     0.                 ,
 DCJ    =     0.                 ,
 BCJUP  =     0.                 ,
 NDWDT  =          0,
 BPCJ   =     0.                 ,
 AMINP  =     0.                 ,
 DWDT   =     0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
 GAS    =     0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,
 BUA    =     0.                 ,
 BUB    =     0.                 ,
 BUMAX  =     0.                 ,
 BUDV   =     0.                 ,
$END
```

```
$N
 NMAT    =         12,
 NN      =         -1,
$END
         *****  CH2                    *****

$DAT
 IEXP    =           0,
 IBRN    =           0,
 IVIS    =           0,
 VFACT   =     2.0000000000000E+00,
 RHOO    =     9.1500000000000E-01,
 PO      =     1.0000000000000E-06,
 TO      =     3.0000000000000E+02,
 EO      =     0.                 ,
 UO      =     0.                 ,
 SOL     =     2.9010000000000E-01,    1.4810000000000E+00,
               1.0000000000000E-02,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               5.0000000000000E-01,    5.0000000000000E-01,
               1.0928961750000E+00,    1.0000000000000E-04,
               0.                 ,    0.                 ,
               3.0000000000000E+02,    1.0000000000000E-06,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,
 WO      =     1.0000000000000E+00,
 Z       =     0.                 ,
 E       =     0.                 ,
 VCJ     =     0.                 ,
 DCJ     =     0.                 ,
 BCJUP   =     0.                 ,
 NDWDT   =           0,
 BPCJ    =     0.                 ,
 AMINP   =     0.                 ,
 DWDT    =     0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
 GAS     =     0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,
 BUA     =     0.                 ,
 BUB     =     0.                 ,
 BUMAX   =     0.                 ,
 BUDV    =     0.                 ,
$END
```

```
$N
 NMAT   =        13,
 NN     =        -1,
$END
        *****  FOAM                  *****


$DAT
 IEXP   =         1,
 IBRN   =         1,
 IVIS   =         0,
 VFACT  =    2.0000000000000E+00,
 RHOO   =    2.5600000000000E-01,
 PO     =    1.0000000000000E-06,
 TO     =    3.0000000000000E+02,
 EO     =    0.                 ,
 UO     =    0.                 ,
 SOL    =    1.0000000000000E-02,    1.4000000000000E+00,
             1.0000000000000E-03,    0.                 ,
             0.                 ,    5.7795005289400E+00,
             6.5738798524400E-01,    5.2908852170300E+00,
            -3.2818147067500E+01,    3.7389900213800E+01,
             1.0000000000000E-01,    5.0000000000000E-01,
             3.9062500000000E+00,    1.0000000000000E-05,
             0.                 ,    0.                 ,
             3.0000000000000E+02,    1.0000000000000E-06,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,
 WO     =    1.0000000000000E+00,
 Z      =    0.                 ,
 E      =    0.                 ,
 VCJ    =    2.5000000000000E+00,
 DCJ    =    2.5000000000000E+00,
 BCJUP  =    0.                 ,
 NDWDT  =          0,
 BPCJ   =    0.                 ,
 AMINP  =    0.                 ,
 DWDT   =    0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
 GAS    =   -2.3056290298900E+00,   -2.0327298176000E+00,
             4.1624706382900E-02,   -8.5046298250600E-03,
             1.3710841934300E-02,   -8.1341769155800E-01,
             3.5139377096100E-01,    3.4051522790100E-02,
             8.4954254906300E-04,   -2.6165006722800E-05,
             8.5358524401000E+00,   -4.8068713991300E-01,
            -1.0822141051100E-01,    6.6680172755100E-02,
             9.1399439066400E-03,    5.0000000000000E-01,
             1.0000000000000E-01,
 BUA    =    0.                 ,
 BUB    =    0.                 ,
 BUMAX  =    0.                 ,
 BUDV   =    0.                 ,
$END
```

```
$N
 NMAT   =        14,
 NN     =        -1,
$END
        ***** AIR 1                ***** 


$DAT
 IEXP   =         1,
 IBRN   =         0,
 IVIS   =         0,
 VFACT  =   1.0000000000000E-02,
 RHOO   =   2.2022200000000E-03,
 PO     =   1.1000000000000E-05,
 TO     =   3.0000000000000E+02,
 EO     =   3.0000000000000E-03,
 UO     =   0.                 ,
 SOL    =   0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            4.5408723000000E+02,  1.0000000000000E-05,
            0.                 ,  0.                 ,
            3.0000000000000E+02,  1.1000000000000E-05,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,
 WO     =   0.                 ,
 Z      =   0.                 ,
 E      =   0.                 ,
 VCJ    =   0.                 ,
 DCJ    =   0.                 ,
 BCJUP  =   0.                 ,
 NDWDT  =         0,
 BPCJ   =   0.                 ,
 AMINP  =   0.                 ,
 DWDT   =   0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
            0.                 ,  0.                 ,
 GAS    =  -6.4815597907600E-01, -2.3284494242200E+00,
            2.2211209300100E-01, -8.8771800708400E-03,
           -1.7838446631100E-03, -1.7128377158500E-01,
            3.5923780267100E-01,  2.4989330519100E-02,
            1.6963777637300E-03,  7.3978983872900E-05,
            8.7130933986300E+00, -4.8115967696200E-01,
            4.7902390066100E-02,  5.9890749038100E-03,
           -2.1837533877900E-03,  5.0000000000000E-01,
            1.0000000000000E-01,
 BUA    =   0.                 ,
 BUB    =   0.                 ,
 BUMAX  =   0.                 ,
 BUDV   =   0.                 ,
$END
```

```
$N
 NMAT   =        15,
 NN     =        -1,
$END
        ***** AIR 2                 *****


$DAT
 IEXP   =         1,
 IBRN   =         0,
 IVIS   =         1,
 VFACT  =    2.0000000000000E-01,
 RHO0   =    9.2462900000000E-04,
 P0     =    6.0000000000000E-07,
 T0     =    3.0000000000000E+02,
 E0     =    2.4668800000000E-03,
 U0     =    0.                 ,
 SOL    =    0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             9.2965314640000E+02,   0.                 ,
             0.                 ,   0.                 ,
             3.0000000000000E+02,   1.0000000000000E-06,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,
 W0     =    0.                 ,
 Z      =    0.                 ,
 E      =    0.                 ,
 VCJ    =    0.                 ,
 DCJ    =    0.                 ,
 BCJUP  =    0.                 ,
 NDWDT  =         0,
 BPCJ   =    0.                 ,
 AMINP  =    0.                 ,
 DWDT   =    0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
             0.                 ,   0.                 ,
 GAS    =   -4.5925651463400E+00,  -9.1684449125200E-01,
            -2.1288704554600E-01,   5.4146702954700E-02,
            -4.6651572500600E-03,  -1.6181507888800E+00,
             8.5966902857500E-02,   1.6688939090600E-03,
            -1.2702592247200E-04,  -3.4388375401000E-06,
             8.1411219425200E+00,   1.0099419080700E-01,
            -2.1837686705400E-01,   5.4901424967700E-02,
            -4.7035249074400E-03,   5.0000000000000E-01,
             1.0000000000000E-01,
 BUA    =    0.                 ,
 BUB    =    0.                 ,
 BUMAX  =    0.                 ,
 BUDV   =    0.                 ,
$END
```

```
$N
 NMAT    =          16,
 NN      =          -1,
$END
        *****   8AHE                  *****


$DAT
 IEXP    =           1,
 IBRN    =           3,
 IVIS    =           1,
 VFACT   =     2.0000000000000E+00,
 RHOO    =     2.6040000000000E+00,
 PO      =     1.0000000000000E-06,
 TO      =     3.0000000000000E+02,
 EO      =     0.                 ,
 UO      =     0.                 ,
 SOL     =     2.7150000000000E-01,    2.5760000000000E+00,
               1.0000000000000E-01,    0.                 ,
               0.                 ,   -8.6661849555200E+00,
              -5.8313782208900E+01,   -6.9716341085000E+01,
              -8.2009910278300E+00,    2.0719556900800E+01,
               6.7470000000000E-01,    4.0000000000000E-01,
               3.8402457700000E-01,    5.0000000000000E-05,
               0.                 ,    0.                 ,
               3.0000000000000E+02,    1.0000000000000E-06,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,
 WO      =     1.0000000000000E+00,
 Z       =     0.                 ,
 E       =     0.                 ,
 VCJ     =     4.8800000000000E-01,
 DCJ     =     4.8800000000000E-01,
 BCJUP   =     0.                 ,
 NDWDT   =          0,
 BPCJ    =     0.                 ,
 AMINP   =     0.                 ,
 DWDT    =     0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
 GAS     =    -6.2035295000000E+00,   -3.4936691000000E+00,
               0.                 ,    0.                 ,
               0.                 ,   -2.1397597000000E+00,
               4.3255033000000E-01,    8.5046586000000E-02,
               7.6787070000000E-03,    2.6461528000000E-04,
               0.                 ,   -1.4300000000000E+00,
               0.                 ,    0.                 ,
               0.                 ,    5.0000000000000E-01,
               6.0000000000000E-02,
 BUA     =     0.                 ,
 BUB     =     0.                 ,
 BUMAX   =     0.                 ,
 BUDV    =     0.                 ,
$END
```

```
$N
 NMAT   =        17,
 NN     =        -1,
$END
        ***** NO ARRHENIUS          *****

$DAT
 IEXP   =         1,
 IBRN   =         0,
 IVIS   =         1,
 VFACT  =    4.0000000000000E-01,
 RHOO   =    1.7140010000000E+00,
 PO     =    1.0000000000000E-06,
 TO     =    3.0000000000000E+02,
 EO     =    0.                 ,
 UO     =    0.                 ,
 SOL    =    2.3440000000000E-01,    1.7500000000000E+00,
             1.0000000000000E-06,    0.                 ,
             0.                 ,   -4.3375662728300E+00,
            -5.2901710589600E+01,   -1.0134141654400E+02,
            -8.1576457053800E+01,   -2.1688340014100E+01,
             5.0000000000000E-01,    1.0000000000000E+00,
             5.8343057000000E-01,    5.0000000000000E-05,
             0.                 ,    0.                 ,
             3.0000000000000E+02,    1.0000000000000E-06,
             0.                 ,    0.                 ,
             0.                 ,    1.0000000000000E-06,
             0.                 ,
 WO     =    1.0000000000000E+00,
 Z      =    0.                 ,
 E      =    4.5000000000000E+04,
 VCJ    =    0.                 ,
 DCJ    =    0.                 ,
 RCJUP  =    0.                 ,
 NDWDT  =         0,
 BPCJ   =    0.                 ,
 AMINP  =    0.                 ,
 DWDT   =    0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
 GAS    =   -3.7984107052400E+00,   -3.0340338334900E+00,
             1.3602389047600E-01,   -1.4409992462100E-01,
            -1.6149804591500E-01,   -1.6059840383600E+00,
             5.1283010050500E-01,    7.7617166547600E-02,
             3.8767269529200E-03,   -5.9008825467300E-05,
             7.2000000000000E+00,   -8.8963147931300E-01,
            -1.7250193817700E-01,   -1.3076342573300E-01,
            -1.5260936054800E-01,    5.0000000000000E-01,
             1.0000000000000E-01,
 BUA    =    0.                 ,
 BUB    =    0.                 ,
 BUMAX  =    0.                 ,
 BUDV   =    0.                 ,
$END
```

```
$N
 NMAT   =        18,
 NN     =        -1,
$END
         ***** NO FOREST FIRE        *****


$DAT
 IEXP   =         1,
 IBRN   =         4,
 IVIS   =         1,
 VFACT  =     4.0000000000000E-01,
 RHOO   =     1.6990000000000E+00,
 PO     =     1.0000000000000E-06,
 TO     =     3.0000000000000E+02,
 EO     =     0.                 ,
 UO     =     0.                 ,
 SOL    =     3.0000000000000E-01,    1.7950000000000E+00,
              0.                 ,    0.                 ,
              0.                 ,   -4.3375662728300E+00,
             -5.2981710589500E+01,   -1.0134141654400E+02,
             -8.1576457053800E+01,   -2.1688340014100E+01,
              1.5000000000000E+00,    1.0000000000000E+00,
              5.8858151000000E-01,    5.0000000000000E-05,
              0.                 ,    0.                 ,
              3.0000000000000E+02,    1.0000000000000E-06,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,
 WO     =     1.0000000000000E+00,
 Z      =     0.                 ,
 E      =     0.                 ,
 VCJ    =     0.                 ,
 DCJ    =     0.                 ,
 BCJUP  =     0.                 ,
 NDWDT  =        15,
 BPCJ   =     2.9000000000000E-01,
 AMINP  =     5.0000000000000E-02,
 DWDT   =     4.9202389622500E+13,   -1.1233545305500E+14,
              1.1697639473600E+14,   -7.3559180215000E+13,
              3.1173346662900E+13,   -9.4073157340500E+12,
              2.0822288928400E+12,   -3.4297744207800E+11,
              4.2189947999700E+10,   -3.8511339183400E+09,
              2.5629565712500E+08,   -1.2022334048800E+07,
              3.7304675705800E+05,   -6.6173053564300E+03,
              3.3511179937000E+01,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
 GAS    =    -3.7813223223600E+00,   -2.8669072289500E+00,
              4.5780376665900E-01,   -2.1393476004800E-01,
             -3.8759650139800E-01,   -1.5813814217000E+00,
              5.3353834411400E-01,    9.1038624589100E-02,
              7.3248274107000E-03,    2.3436649822400E-04,
              6.7027481505400E+00,   -6.6694075746100E-01,
              1.7172154149600E-01,   -2.3654570859100E-01,
             -3.8360574585200E-01,    5.0000000000000E-01,
              1.0000000000000E-01,
 BUA    =     0.                 ,
 BUB    =     0.                 ,
 BUMAX  =     0.                 ,
 BUDV   =     0.                 ,
$END
```

```
$N
 NMAT   =        19,
 NN     =        -1,
$END
          *****  COMP B SHARP SHOCK  *****

$DAT
 IEXP   =         1,
 IBRN   =         3,
 IVIS   =         0,
 VFACT  =   3.5000000000000E+00,
 RHOO   =   1.7130000000000E+00,
 PO     =   1.0000000000000E-06,
 TO     =   3.0000000000000E+02,
 EO     =   0.                 ,
 UO     =   0.                 ,
 SOL    =   2.4000000000000E-01,   2.5500000000000E+00,
            1.0000000000000E-03,   0.                 ,
            0.                 ,  -9.3035923780500E+00,
           -7.3193114947300E+01,  -1.1984909720700E+02,
           -6.5549768378000E+01,   8.8811032659100E-01,
            8.0000000000000E-01,   2.5900000000000E-01,
            5.8309037900000E-01,   5.0000000000000E-05,
            0.                 ,   0.                 ,
            3.0000000000000E+02,   1.0000000000000E-06,
            0.                 ,   0.                 ,
            0.                 ,   1.0000000000000E-06,
            0.                 ,
 WO     =   1.0000000000000E+00,
 Z      =   0.                 ,
 E      =   0.                 ,
 VCJ    =   8.0300000000000E-01,
 DCJ    =   8.0300000000000E-01,
 BCJUP  =   0.                 ,
 NDWDT  =         0,
 BPCJ   =   0.                 ,
 AMINP  =   0.                 ,
 DWDT   =   0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
 GAS    =  -3.5630947234000E+00,  -2.7600000000000E+00,
            0.                 ,   0.                 ,
            0.                 ,  -1.5607327324500E+00,
            5.1979454280200E-01,   6.7891598954500E-02,
            3.2585940953600E-03,   2.2708937422000E-05,
            0.                 ,  -5.5226424100000E-01,
            0.                 ,   0.                 ,
            0.                 ,   5.0000000000000E-01,
            1.0000000000000E-01,
 BUA    =   0.                 ,
 BUB    =   0.                 ,
 BUMAX  =   0.                 ,
 BUDV   =   0.                 ,
$END
```

```
$N
 NMAT    =        20,
 NN      =        -1,
$END
        *****  COMP B CJ(ARRH)     *****


$DAT
 IEXP    =         1,
 IBRN    =         1,
 IVIS    =         0,
 VFACT   =    3.5000000000000E+00,
 RHOO    =    1.7300000000000E+00,
 PO      =    1.0000000000000E-06,
 TO      =    3.0000000000000E+02,
 EO      =    0.                 ,
 UO      =    0.                 ,
 SOL     =    2.3100000000000E-01,    1.8300000000000E+00,
              1.0000000000000E-02,    0.                 ,
              0.                 ,   -8.6482257659700E+00,
             -7.6497948971400E+01,   -1.4880798590100E+02,
             -1.2260687862300E+02,   -3.4139045857900E+01,
              2.0000000000000E+00,    2.5000000000000E-01,
              5.7803468200000E-01,    5.0000000000000E-05,
              0.                 ,    0.                 ,
              3.0000000000000E+02,    1.0000000000000E-06,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,
 WO      =    1.0000000000000E+00,
 Z       =    1.0000000000000E+13,
 E       =    4.5000000000000E+04,
 VCJ     =    4.3311700000000E-01,
 DCJ     =    4.3311700000000E-01,
 RCJUP   =    0.                 ,
 NDWDT   =         0,
 RPCJ    =    0.                 ,
 AMINP   =    0.                 ,
 DWDT    =    0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
 GAS     =   -3.8101492210200E+00,   -2.9887450000000E+00,
              0.                 ,    0.                 ,
              0.                 ,   -1.5935241619500E+00,
              5.0113382815600E-01,    7.8549138663000E-02,
              5.3425796030900E-03,    1.2783948755700E-04,
              7.5081772908100E+00,   -4.5925899205700E-01,
              5.8670140664100E-02,    2.1435450837900E-02,
             -7.0630889677900E-03,    5.0000000000000E-01,
              1.0000000000000E-01,
 BUA     =    0.                 ,
 BUB     =    0.                 ,
 BUMAX   =    0.                 ,
 BUDV    =    0.                 ,
$END
```

```
$N
 NMAT    =        21,
 NN      =        -1,
$END
        ***** 9404 CJ              *****

$DAT
 IEXP    =          1,
 IBRN    =          1,
 IVIS    =          0,
 VFACT   =    1.5000000000000E+00,
 RHO0    =    1.8440000000000E+00,
 PO      =    1.0000000000000E-06,
 TO      =    3.0000000000000E+02,
 EO      =    0.                 ,
 UO      =    0.                 ,
 SOL     =    2.7150000000000E-01,    2.5760000000000E+00,
              1.0000000000000E-01,    0.                 ,
              0.                 ,   -8.6561849555200E+00,
             -5.8313782209900E+01,   -6.9716341085000E+01,
             -8.2009910278300E+00,    2.0719556900800E+01,
              6.7470000000000E-01,    4.0000000000000E-01,
              5.4229935000000E-01,    5.0000000000000E-05,
              0.                 ,    0.                 ,
              3.0000000000000E+02,    1.0000000000000E-06,
              0.                 ,    0.                 ,
              0.                 ,    1.0000000000000E-06,
              0.                 ,
 WO      =    1.0000000000000E+00,
 Z       =    0.                 ,
 E       =    0.                 ,
 VCJ     =    4.1318000000000E-01,
 DCJ     =    4.1318000000000E-01,
 BCJUP   =    0.                 ,
 NDWDT   =          0,
 BPCJ    =    0.                 ,
 AMINP   =    0.                 ,
 DWDT    =    0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
 GAS     =   -3.9071763513500E+00,   -3.2000000000000E+00,
              0.                 ,    0.                 ,
              0.                 ,   -1.6380521783900E+00,
              5.2133755898000E-01,    8.5757555184200E-02,
              6.2392162210300E-03,    1.6563329597600E-04,
              0.                 ,   -1.1085600000000E+00,
              0.                 ,    0.                 ,
              0.                 ,    5.0000000000000E-01,
              1.0000000000000E-01,
 BUA     =    0.                 ,
 BUB     =    0.                 ,
 BUMAX   =    0.                 ,
 BUDV    =    0.                 ,
$END
```

```
$N
 NMAT   =        22,
 NN     =        -1,
$END
        ***** 9404 BUILD UP EOS    *****


$DAT
 IEXP   =         5,
 IBRN   =         1,
 IVIS   =         0,
 VFACT  =   2.5000000000000E+00,
 RHOO   =   1.8440000000000E+00,
 PO     =   1.0000000000000E-06,
 TO     =   3.0000000000000E+02,
 EO     =   0.                 ,
 UO     =   0.                 ,
 SOL    =   2.7150000000000E-01,   2.5760000000000E+00,
            1.0000000000000E-01,   0.                 ,
            0.                 ,  -8.6661849555200E+00,
           -5.8313782208900E+01,  -6.9716341085000E+01,
           -8.2009910278300E+00,   2.0719556900800E+01,
            6.7470000000000E-01,   4.0000000000000E-01,
            5.4229935000000E-01,   5.0000000000000E-05,
            0.                 ,   0.                 ,
            3.0000000000000E+02,   1.0000000000000E-06,
            0.                 ,   0.                 ,
            0.                 ,   1.0000000000000E-06,
            0.                 ,
 WO     =   1.0000000000000E+00,
 Z      =   0.                 ,
 E      =   0.                 ,
 VCJ    =   4.1469950000000E-01,
 DCJ    =   4.1469950000000E-01,
 BCJUP  =   0.                 ,
 NDWDT  =        0,
 BPCJ   =   0.                 ,
 AMINP  =   0.                 ,
 DWDT   =   0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
            0.                 ,   0.                 ,
 GAS    =  -3.9513022262000E+00,  -3.2500000000000E+00,
            0.                 ,   0.                 ,
            0.                 ,  -1.6427052459600E+00,
            5.1809393382100E-01,   8.7301259389500E-02,
            6.5604492690900E-03,   1.8255953940800E-04,
            0.                 ,  -1.1669894169900E+00,
            0.                 ,   0.                 ,
            0.                 ,   5.0000000000000E-01,
            1.0000000000000E-01,
 BUA    =   2.6800000000000E+00,
 BUB    =   1.3900000000000E+00,
 BUMAX  =   3.7000000000000E+00,
 BUDV   =   8.8000000000000E-01,
$END
```

```
$N
 NMAT    =         23,
 NN      =         -1,
$END
         ***** 9404 GAMMA LAW        *****

$DAT
 IEXP    =          1,
 IBRN    =          2,
 IVIS    =          0,
 VFACT   =    2.5000000000000E+00,
 RHOO    =    1.8440000000000E+00,
 PO      =    1.0000000000000E-06,
 TO      =    3.0000000000000E+02,
 EO      =    0.                 ,
 UO      =    0.                 ,
 SOL     =    2.7150000000000E-01,    2.5760000000000E+00,
              1.0000000000000E-01,    0.                 ,
              0.                 ,   -8.6661849555200E+00,
             -5.8313782208900E+01,   -6.9715341085000E+01,
             -8.2009910278300E+00,    2.0719556900800E+01,
              6.7470000000000E-01,    4.0000000000000E-01,
              5.4229935000000E-01,    5.0000000000000E-05,
              0.                 ,    0.                 ,
              3.0000000000000E+02,    1.0000000000000E-06,
              0.                 ,    0.                 ,
              0.                 ,    1.0000000000000E-06,
              0.                 ,
 WO      =    1.0000000000000E+00,
 Z       =    0.                 ,
 E       =    0.                 ,
 VCJ     =    8.8000000000000E-01,
 DCJ     =    8.8000000000000E-01,
 BCJUP   =    0.                 ,
 NDWDT   =          0,
 BPCJ    =    0.                 ,
 AMINP   =    0.                 ,
 DWDT    =    0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
              0.                 ,    0.                 ,
 GAS     =   -3.9513022262000E+00,   -3.2500000000000E+00,
              0.                 ,    0.                 ,
              0.                 ,   -1.6427062459600E+00,
              5.1809393382100E-01,    8.7301259389500E-02,
              6.5604492690900E-03,    1.9255953940800E-04,
              0.                 ,   -1.1669884169900E+00,
              0.                 ,    0.                 ,
              0.                 ,    5.0000000000000E-01,
              1.0000000000000E-01,
 BUA     =    0.                 ,
 BUB     =    0.                 ,
 BUMAX   =    0.                 ,
 BUDV    =    0.                 ,
$END
```

```
$N
 NMAT   =        24,
 NN     =        -1,
$END
      *****  9404 FOREST FIRE      *****


$DAT
 IEXP   =         1,
 IBRN   =         4,
 IVIS   =         2,
 VFACT  =    2.0000000000000E-03,
 RHOO   =    1.8440000000000E+00,
 PO     =    1.0000000000000E-06,
 TO     =    3.0000000000000E+02,
 EO     =    0.                 ,
 UO     =    0.                 ,
 SOL    =    2.4230000000000E-01,     1.8830000000000E+00,
             1.0000000000000E-02,     0.                 ,
             0.                 ,    -9.0418722204200E+00,
            -7.1318525243500E+01,    -1.2520497936000E+02,
            -9.2042417760300E+01,    -2.2189382572700E+01,
             6.7500000000000E-01,     4.0000000000000E-01,
             5.4229934924100E-01,     5.0000000000000E-05,
             0.                 ,     0.                 ,
             3.0000000000000E+02,     1.0000000000000E-06,
             0.                 ,     0.                 ,
             0.                 ,     0.                 ,
             0.                 ,
 WO     =    1.0000000000000E+00,
 Z      =    0.                 ,
 E      =    0.                 ,
 VCJ    =    8.8800000000000E-01,
 DCJ    =    0.                 ,
 RCJUP  =    0.                 ,
 NDWDT  =        14,
 BPCJ   =    3.6300000000000E-01,
 AMINP  =    1.5000000000000E-02,
 DWDT   =    2.5277953727000E+10,    -6.9876089170000E+10,
             8.6704208069000E+10,    -6.3781135352000E+10,
             3.0950369616000E+10,    -1.0433258901000E+10,
             2.5068548091000E+09,    -4.3377143285000E+08,
             5.4017707404000E+07,    -4.7962436917000E+06,
             2.9889932207000E+05,    -1.2887959724000E+04,
             4.0524452315000E+02,    -8.3979132644000E+00,
             0.                 ,     0.                 ,
             0.                 ,     0.                 ,
             0.                 ,     0.                 ,
 GAS    =   -3.5390625996400E+00,    -2.5773759039300E+00,
             2.6007542333200E-01,     1.3908357850800E-02,
            -1.1396302407500E-02,    -1.6191304113300E+00,
             5.2151853419200E-01,     6.7750659410700E-02,
             4.2652426469100E-03,     1.0467999990200E-04,
             7.3642291979000E+00,    -4.9365822238900E-01,
             2.9235306096100E-02,     3.3027740221900E-02,
            -1.1453249820600E-02,     5.0000000000000E-01,
             1.0000000000000E-01,
 BUA    =    0.                 ,
 BUB    =    0.                 ,
 BUMAX  =    0.                 ,
 BUDV   =    0.                 ,
$END
```

```
$N
 NMAT   =        25,
 NN     =        -1,
$END
        ***** TATB/WAX FF F(P)      *****


$DAT
 IEXP   =         1,
 IBRN   =         4,
 IVIS   =         1,
 VFACT  =    4.0000000000000E-01,
 RHOO   =    1.7410000000000E+00,
 PO     =    1.0000000000000E-06,
 TO     =    3.0000000000000E+02,
 EO     =    0.                 ,
 UO     =    0.                 ,
 SOL    =    2.6200000000000E-01,    1.8460000000000E+00,
             1.0000000000000E-02,    0.                 ,
             0.                 ,   -1.0565561478500E+01,
            -8.8519912048700E+01,   -1.7580617026200E+02,
            -1.4821742233500E+02,   -4.2691241225500E+01,
             1.7000000000000E+00,    1.0000000000000E+00,
             5.7438253877100E-01,    5.0000000000000E-05,
             0.                 ,    0.                 ,
             3.0000000000000E+02,    1.0000000000000E-06,
             0.                 ,    0.                 ,
             0.                 ,    1.0000000000000E-06,
             0.                 ,
 WO     =    1.0000000000000E+00,
 Z      =    0.                 ,
 E      =    0.                 ,
 VCJ    =    0.                 ,
 DCJ    =    0.                 ,
 BCJUP  =    0.                 ,
 NDWDT  =        15,
 BPCJ   =    2.4200000000000E-01,
 AMINP  =    5.0000000000000E-02,
 DWDT   =    2.1074823834300E+15,   -4.0485613115500E+15,
             3.5604013860000E+15,   -1.8985456595000E+15,
             6.8538310052000E+14,   -1.7708730387300E+14,
             3.3749845396400E+13,   -4.8166840092100E+12,
             5.1695074669000E+11,   -4.1491881910200E+10,
             2.4497203379100E+09,   -1.0306137019700E+08,
             2.9147295142500E+06,   -4.9191520774900E+04,
             3.4692852517900E+02,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
 GAS    =   -3.8151151160600E+00,   -2.6820304590800E+00,
             2.6707281671500E-01,    1.1008177275600E-01,
            -4.2627029473100E-02,   -1.5324680893000E+00,
             5.6368029118800E-01,    1.1099121788600E-01,
             1.1097910232600E-02,    4.3719078163500E-04,
             7.2050716427100E+00,   -4.7824523434400E-01,
             6.0919683634800E-02,    7.4133402313400E-03,
            -5.0885279490900E-03,    5.0000000000000E-01,
             1.0000000000000E-01,
 BUA    =    0.                 ,
 BUB    =    0.                 ,
 BUMAX  =    0.                 ,
 BUDV   =    0.                 ,
$END
```

```
$N
 NMAT   =        26,
 NN     =        -1,
$END
         ***** X0290 FF PCJ=.285    *****


$DAT
 IEXP   =         1,
 IBRN   =         4,
 IVIS   =         0,
 VFACT  =    2.0000000000000E+00,
 RHOO   =    1.8940000000000E+00,
 PO     =    0.                 ,
 TO     =    0.                 ,
 EO     =    0.                 ,
 UO     =    0.                 ,
 SOL    =    2.4000000000000E-01,    2.0500000000000E+00,
             0.                 ,    0.                 ,
             0.                 ,   -2.3014168556000E+01,
            -1.3631901377800E+02,   -2.3506821666100E+02,
            -1.7104959098300E+02,   -4.2263550556900E+01,
             1.5000000000000E+00,    3.0000000000000E-01,
             5.2798310454100E-01,    5.0000000000000E-05,
             0.                 ,    0.                 ,
             3.0000000000000E+02,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,
 WO     =    1.0000000000000E+00,
 Z      =    0.                 ,
 E      =    0.                 ,
 VCJ    =    3.9423000000000E-01,
 DCJ    =    0.                 ,
 RCJUP  =    0.                 ,
 NDWDT  =        15,
 BPCJ   =    2.8500000000000E-01,
 AMINP  =    5.5000000000000E-02,
 DWDT   =    8.1425481008000E+13,   -1.9413888714000E+14,
             2.1097494446000E+14,   -1.3836158944000E+14,
             6.1110150327000E+13,   -1.9206772744000E+13,
             4.4248348854000E+12,   -7.5817519329000E+11,
             9.6978690159000E+10,   -9.2043893492000E+09,
             6.3734890585000E+08,   -3.1181988011000E+07,
             1.0170082035000E+06,   -1.9660891926000E+04,
             1.6223658470000E+02,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
 GAS    =   -3.8782854115900E+00,   -2.6903229723100E+00,
             2.2207418495100E-01,    7.4248212800000E-02,
            -3.4281943072700E-02,   -1.5888961537700E+00,
             5.3489544838500E-01,    9.4282425112400E-02,
             6.2564345992400E-03,    2.8935792258200E-04,
             7.0674029264900E+00,   -5.6700324443000E-01,
             5.1794158609500E-02,    9.8485394639500E-03,
            -1.0921841974800E-02,    5.0000000000000E-01,
             1.0000000000000E-01,
 BUA    =    0.                 ,
 BUB    =    0.                 ,
 BUMAX  =    0.                 ,
 BUDV   =    0.                 ,
$END
```

```
$N
 NMAT   =        27,
 NN     =        -1,
$END
        ***** NQ/ESTANE 95/5 FF(T) *****


$DAT
 IEXP   =         1,
 IBRN   =         4,
 IVIS   =         1,
 VFACT  =    2.5000000000000E+00,
 RHOO   =    1.6990000000000E+00,
 PO     =    0.                 ,
 TO     =    3.0000000000000E+02,
 EO     =    0.                 ,
 UO     =    0.                 ,
 SOL    =    3.0000000000000E-01,     1.7950000000000E+00,
             0.                 ,     0.                 ,
             0.                 ,    -4.3375662728300E+00,
            -5.2981710589500E+01,    -1.0134141654400E+02,
            -8.1576457053800E+01,    -2.1688340014100E+01,
             1.5000000000000E+00,     1.0000000000000E+00,
             5.8858151000000E-01,     5.0000000000000E-05,
             0.                 ,     0.                 ,
             3.0000000000000E+02,     1.0000000000000E-06,
             0.                 ,     0.                 ,
             0.                 ,     0.                 ,
             0.                 ,
 WO     =    1.0000000000000E+00,
 Z      =    0.                 ,
 E      =    0.                 ,
 VCJ    =    0.                 ,
 DCJ    =    0.                 ,
 BCJUP  =    0.                 ,
 NDWDT  =        15,
 BPCJ   =    1.3965931925800E+03,
 AMINP  =    3.9413000365800E+02,
 DWDT   =   -2.4754151108200E-37,     2.9718825932700E-33,
            -1.6403011725300E-29,     5.5162577515100E-26,
            -1.2626800047100E-22,     2.0810584469500E-19,
            -2.5467895398800E-16,     2.3512495264500E-13,
            -1.6458549154900E-10,     8.6951979083100E-08,
            -3.4143685421200E-05,     9.6694517066800E-03,
            -1.8688014733600E+00,     2.2101444421200E+02,
            -1.2115464430700E+04,     0.                 ,
             0.                 ,     0.                 ,
             0.                 ,     0.                 ,
 GAS    =   -3.7813223223600E+00,    -2.8659072289500E+00,
             4.5780376665900E-01,    -2.1393476004800E-01,
            -3.8759650139800E-01,    -1.5813814217000E+00,
             5.3353834411400E-01,     9.1039624589100E-02,
             7.3248274107000E-03,     2.3436649822400E-04,
             6.7027481505400E+00,    -6.6694075746100E-01,
             1.7172154149600E-01,    -2.3654570859100E-01,
            -3.8360574585200E-01,     5.0000000000000E-01,
             1.0000000000000E-01,
 BUA    =    0.                 ,
 BUB    =    0.                 ,
 BUMAX  =    0.                 ,
 BUDV   =    0.                 ,
$END
```

```
$N
 NMAT    =         30,
 NN      =         -1,
$END
        ***** NO FF RH ZERO ORDER  *****


$DAT
 IEXP    =         1,
 IBRN    =         4,
 IVIS    =         1,
 VFACT   =     2.5000000000000E+00,
 RHOO    =     1.6990000000000E+00,
 PO      =     0.                 ,
 TO      =     3.0000000000000E+02,
 EO      =     0.                 ,
 UO      =     0.                 ,
 SOL     =     3.0000000000000E-01,    1.7950000000000E+00,
               0.                 ,    0.                 ,
               0.                 ,   -4.3375662728300E+00,
              -5.2981710589500E+01,   -1.0134141654400E+02,
              -8.1576457053800E+01,   -2.1688340014100E+01,
               1.5000000000000E+00,    1.0000000000000E+00,
               5.8858151000000E-01,    5.0000000000000E-05,
               0.                 ,    0.                 ,
               3.0000000000000E+02,    1.0000000000000E-06,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,
 WO      =     1.0000000000000E+00,
 Z       =     0.                 ,
 E       =     0.                 ,
 VCJ     =     0.                 ,
 DCJ     =     0.                 ,
 BCJUP   =     0.                 ,
 NDWDT   =         15,
 BPCJ    =     2.8500000000000E-01,
 AMINP   =     9.0000000000000E-02,
 DWDT    =     1.0493710061000E+13,   -2.4403909027000E+13,
               2.5848703251000E+13,   -1.6508273785000E+13,
               7.0926386979000E+12,   -2.1655438308000E+12,
               4.8379711632000E+11,   -8.0196999294000E+10,
               9.8904508407000E+09,   -9.0036142748000E+08,
               5.9260988094000E+07,   -2.7058239629000E+06,
               7.8400413699000E+04,   -1.0435423528000E+03,
              -1.4056835920000E+01,    0.                 ,
               0.                 ,    0.                 ,
               0.                 ,    0.                 ,
 GAS     =    -3.7813223223600E+00,   -2.8669072289500E+00,
               4.5780376665900E-01,   -2.1393476004800E-01,
              -3.8759650139800E-01,   -1.5813814217000E+00,
               5.3353834411400E-01,    9.1038624589100E-02,
               7.3248274107000E-03,    2.3436649822400E-04,
               6.7027481505400E+00,   -6.6694075746100E-01,
               1.7172154149600E-01,   -2.3654570859100E-01,
              -3.8360574585200E-01,    5.0000000000000E-01,
               1.0000000000000E-01,
 BUA     =     0.                 ,
 BUB     =     0.                 ,
 BUMAX   =     0.                 ,
 BUDV    =     0.                 ,
 $END
```

```
$N
 NMAT   =          31,
 NN     =          -1,
$END
          *****  TNT                    *****


$DAT
 IEXP   =           1,
 IBRN   =           1,
 IVIS   =           0,
 VFACT  =    2.0000000000000E+00,
 RHOO   =    1.6400000000000E+00,
 PO     =    1.0000000000000E-06,
 TO     =    3.0000000000000E+02,
 EO     =    0.                 ,
 UO     =    0.                 ,
 SOL    =    3.0330000000000E-01,    1.3660000000000E+00,
             1.0000000000000E-03,    0.                 ,
             0.                 ,    1.1487924489200E+01,
             2.9423815334900E+01,    5.1407881602000E+01,
             3.6666803808000E+01,    1.0775309358200E+01,
             1.7300000000000E+00,    2.9300000000000E-01,
             6.0975609760000E-01,    5.0000000000000E-05,
             0.                 ,    0.                 ,
             3.0000000000000E+02,    1.0000000000000E-06,
             0.                 ,    0.                 ,
             0.                 ,    1.0000000000000E-06,
             0.                 ,
 WO     =    1.0000000000000E+00,
 Z      =    0.                 ,
 E      =    0.                 ,
 VCJ    =    4.6300000000000E-01,
 OCJ    =    4.6300000000000E-01,
 BCJUP  =    0.                 ,
 NOWDT  =           0,
 BPCJ   =    0.                 ,
 AMINP  =    0.                 ,
 DWDT   =    0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
             0.                 ,    0.                 ,
 GAS    =   -4.0927913585100E+00,   -3.1600000000000E+00,
             0.                 ,    0.                 ,
             0.                 ,   -1.6017487387500E+00,
             4.6320457097100E-01,    8.2694800461400E-02,
             6.7200941511000E-03,    2.0718222636500E-04,
             0.                 ,   -9.9000000000000E-01,
             0.                 ,    0.                 ,
             0.                 ,    5.0000000000000E-01,
             1.0000000000000E-01,
 BUA    =    0.                 ,
 BUB    =    0.                 ,
 BUMAX  =    0.                 ,
 BUDV   =    0.                 ,
$END
```

## B. Using the SESAME Option in HYDROX (by G. I. Kerley)

Specify a SESAME EOS in NAMELIST SU, by setting IEOS = 4 and MAT equal to the material number. In NAMELIST ESC, specify the parameters for spall, viscosity, and elastic-plastic flow, as with other EOS options. In order to use special options available with the SESAME tables, specify the parameters discussed below.

### 1. Initial State Calculation. In NAMELIST ESC,

ROW = initial density in $g/cm^3$

TO = initial temperature in K

ZI = initial internal energy ($Mbar\text{-}cm^3/g$)

If ROW is not input, the code will obtain this quantity from the SESAME library. An input value will override the SESAME number. Parameters TO and ZI default to zero.

If the user specifies TO $\neq$ 0 and if ZI = 0, the code will calculate ZI from ROW and TO. This feature is especially useful when the material is a gas, but it can be used for solids and liquids as well.

### 2. Density and Energy Scaling. In NAMELIST ESC,

SR = density scale factor (default = 1)

ES = energy shift in $Mbar\text{-}cm^3/g$ (default = 0)

Using these parameters, the EOS is scaled according to the following relations.

$$\begin{aligned} P(\rho,E) &= P_{TAB}(\rho_T,E_T) \quad , \\ \rho_T &= SR^* \rho \quad , \\ E_T &= (E + ES)/SR \quad , \end{aligned} \tag{1}$$

where P, $\rho$, and E are the pressure, density, and energy variables used by the code, and $P_{TAB}(\rho_T,E_T)$ is the tabular EOS.

The parameter SR is useful for treating isotopic mixtures. If $A_T$ is the atomic weight for the EOS table, an EOS for an atomic weight A is obtained by setting

$$SR = A_T/A \quad . \tag{2}$$

For example, set SR = 2 to scale the SESAME $D_2$ EOS, #5263, to $H_2$. Similarly, set SR = .80 to obtain an EOS for a 50:50 DT mixture.

The parameter ES can be used to change the energy zero of the table. It is intended for use primarily with the "ramp" option, discussed below.

3. Foams and Phase Transitions. In NAMELIST ESC,

A1, A2, A3 = ramp parameters (default = 0)

ES = energy shift in Mbar-cm$^3$/g (default = 0)

IRV = reversible/irreversible flag (default = 0)

EM = "melt" energy in Mbar-cm$^3$/g (default = 1000)

For treatment of foams and certain types of phase transitions, it is possible to modify the SESAME EOS by adding a "ramp" which describes the behavior of the material at low stress levels.

The material starts out in either a porous state or low-density phase. The EOS as a function of the density ρ is given by

$$P = A_1(\rho/\rho_0 - 1) \quad , \tag{3}$$

where $\rho_0$ is the initial density (ROW) and $A_1$ is the bulk modulus in Mbar. $A_1$ can be computed from

$$A_1 = 0.01\rho_0 C_0^2 \tag{4}$$

where $C_0$ is the bulk sound speed in km/s. If $A_1$ = 0, no ramp calculation is performed.

At some pressure $P_i$, the material will begin to "crush," or transform to the high-density phase which is described by the SESAME EOS. The EOS of the crush curve is

$$P = A_2(\rho/\rho_0 - A_3) \quad . \tag{5}$$

(If $A_2 = 0$, there is no crush region and Eq. (3) is continued on until it crosses the SESAME hydrostat.) $A_2$ is related to the transition pressure by

$$A_2 = \frac{P_i A_1}{P_i + A_1(1 - A_3)} \quad \text{(Mbar)}. \tag{6}$$

The transition pressure for foams is usually rather small (<1 kbar). For a phase transition, $P_i$ must be obtained from experiment. The parameter $A_3$ can be adjusted to give the correct slope of the crush curve. In the absence of data, the default value ($A_3$ = 0) should give acceptable results.

At some pressure $P_f$, the ramp crosses the SESAME hydrostat. At that point, the cell is said to be "crushed." Subsequently, the material may behave either reversibly (follow the ramp on expansion) or irreversibly (stay in the high-density phase on expansion). Foams are normally irreversible, but phase transitions may be either. The behavior is controlled by setting

$$\mathrm{IRV} = \begin{matrix} 1 & \text{reversible,} \\ 0 & \text{irreversible,} \end{matrix} \tag{7}$$

where the default is IRV = 0. In HYDROX, the material will also behave irreversibly if it melts; i.e., if the melt energy EM is exceeded. EM depends upon the path and must be treated carefully. The default value (EM = 1000) is sufficiently high that "melting" will not occur in most cases of interest.

For foams, the energy shift ES should be set to zero. For phase transitions, set

$$ES = -\Delta E \text{ (initial} \rightarrow \text{final)} \quad , \tag{8}$$

where $\Delta E$ is the energy required to transform the low-density phase to the high-density phase. Hence there are two cases. If the initial phase is stable, ES is negative. If the initial phase is metastable, ES is positive.

4. Interpolation Option. In NAMELIST INP, set

IFN = 0 rational function algorithm,

IFN = 1 bilinear algorithm,

where the default is IFN = 0. The rational function option is the more accurate interpolation scheme. The bilinear scheme is faster and is sufficiently accurate for some applications.

5. Table of SESAME Materials in SES2L

SHORT SUMMARY FOR VERSION 67 OF SESAME LIBRARY FILE SES2L
DATED 91280 FOR 45 MATERIALS

| NUMBER | MATERIAL | ZBAR | ABAR | RO | TABLES | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1540 | URANIUM | 92.0 | 238.0 | 19.0 | 101 | 201 | 301 | | |
| 2020 | BERYLLIUM | 4.0 | 9.0 | 1.8 | 101 | 102 | 201 | 301 | |
| 2021 | BERYLLIUM | 4.0 | 9.0 | 1.9 | 101 | 201 | 301 | | |
| 2140 | IRON | 26.0 | 55.9 | 7.9 | 101 | 201 | 301 | | |
| 2144 | IRON | 26.0 | 55.8 | 7.9 | 101 | 102 | 201 | 301 | |
| 2145 | IRON - REACTOR SAFET | 26.0 | 55.8 | 7.9 | 101 | 102 | 201 | 301 | 401 |
| 2200 | LITHIUM | 3.0 | 6.9 | .5 | 101 | 201 | 301 | | |
| 2445 | SODIUM | 11.0 | 23.0 | 1.0 | 101 | 102 | 201 | 301 | |
| 2446 | SODIUM | 11.0 | 23.0 | 1.0 | 101 | 102 | 201 | 301 | |
| 2448 | SODIUM | 11.0 | 23.0 | 1.0 | 101 | 201 | 301 | 401 | |
| 2700 | GOLD | 79.0 | 197.0 | 19.3 | 101 | 102 | 201 | 301 | |
| 2701 | GOLD | 79.0 | 197.0 | 19.3 | 101 | 201 | 301 | | |
| 2980 | MOLYBDENUM | 42.0 | 95.9 | 10.2 | 101 | 201 | 301 | | |
| 3100 | NICKEL | 28.0 | 58.7 | 8.9 | 101 | 201 | 301 | | |
| 3200 | LEAD | 82.0 | 207.2 | 11.3 | 101 | 102 | 201 | 301 | |
| 3330 | COPPER | 29.0 | 63.5 | 8.9 | 101 | 201 | 301 | | |
| 3541 | TUNGSTEN | 74.0 | 183.9 | 19.2 | 101 | 201 | 301 | | |
| 3710 | ALUMINUM | 13.0 | 27.0 | 2.7 | 101 | 201 | 301 | | |
| 3730 | PLATINUM | 78.0 | 195.1 | 21.4 | 101 | 201 | 301 | | |
| 4100 | BRASS | 29.8 | 65.3 | 8.5 | 101 | 102 | 201 | 301 | |
| 4270 | STAINLESS STEEL | 25.8 | 55.4 | 7.9 | 101 | 201 | 301 | | |
| 4271 | STEEL | 26.0 | 55.8 | 7.9 | 101 | 201 | 301 | | |
| 5170 | ARGON | 18.0 | 39.9 | 1.5 | 101 | 102 | 201 | 301 | |
| 5180 | KRYPTON | 36.0 | 83.8 | 2.5 | 101 | 102 | 201 | 301 | |
| 5263 | DEUTERIUM | 1.0 | 2.0 | .2 | 101 | 102 | 201 | 301 | 303 |
| 5410 | NEON | 10.0 | 20.2 | 1.4 | 101 | 201 | 301 | | |
| 5500 | METHANE | 2.0 | 3.2 | .5 | 101 | 102 | 201 | 301 | 401 |
| 5501 | METHANE | 2.0 | 3.2 | .5 | 101 | 102 | 201 | 301 | 401 |
| 5760 | HELIUM | 2.0 | 4.0 | .2 | 101 | 201 | 301 | | |
| 7081 | BORON CARBIDE - REAC | 5.2 | 10.4 | 2.5 | 101 | 201 | 301 | 401 | |
| 7111 | NEVADA ALLUVIUM | 9.4 | 19.8 | 2.4 | 101 | 201 | 301 | | |
| 7150 | WATER | 3.3 | 6.0 | 1.0 | 101 | 201 | 301 | | |
| 7151 | STEAM | 3.3 | 6.0 | 1.0 | 101 | 201 | 301 | | |
| 7170 | POLYETHYLENE | 2.7 | 4.7 | .9 | 101 | 201 | 301 | | |
| 7240 | LITHIUM DEUTERIDE | 2.0 | 4.0 | .8 | 101 | 201 | 301 | | |
| 7370 | LITHIUM HYDRIDE | 2.0 | 3.5 | .7 | 101 | 201 | 301 | | |
| 7380 | QUARTZ | 10.0 | 20.0 | 2.2 | 101 | 201 | 301 | | |
| 7390 | WESTERLY GRANITE | 10.3 | 20.7 | 2.6 | 101 | 201 | 301 | | |
| 7410 | ALUMINA | 10.0 | 20.4 | 4.0 | 101 | 201 | 301 | | |
| 7432 | URANIUM DIOXIDE | 36.0 | 90.0 | 11.0 | 101 | 201 | 301 | 401 | |
| 7520 | MICA | 6,9 | 13.5 | 2,7 | 101 | 201 | 301 | | |
| 7560 | POLYURETHANE | 3.8 | 7.0 | 1.3 | 101 | 201 | 301 | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7590 | POLYSTYRENE | 3.5 | 6.5 | 1.0 | 101 | 201 | 301 |
| 7830 | DIAMOND | 6.0 | 12.0 | 3.5 | 101 | 201 | 301 |
| 8180 | HIGH EXPLOSIVE | 5.6 | 11.0 | 1.8 | 101 | 201 | 301 |

NOTES.

TABLES 100-199 CONTAIN HOLLERITH DATA
TABLES 201 CONTAIN BASIC DATA
TABLES 301 CONTAIN TOTAL EOS DATA
TABLES 303 CONTAIN ION EOS DATA
TABLES 401 CONTAIN VAPORIZATION DATA



☆U.S. GOVERNMENT PRINTING OFFICE: 1981-0-777-022/37

Printed in the United States of America
Available from
National Technical Information Service
US Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Microfiche $3.50 (A01)

| Page Range | Domestic Price | NTIS Price Code | Page Range | Domestic Price | NTIS Price Code | Page Range | Domestic Price | NTIS Price Code | Page Range | Domestic Price | NTIS Price Code |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 001-025 | $ 5.00 | A02 | 151-175 | $11.00 | A08 | 301-325 | $17.00 | A14 | 451-475 | $23.00 | A20 |
| 026-050 | 6.00 | A03 | 176-200 | 12.00 | A09 | 326-350 | 18.00 | A15 | 476-500 | 24.00 | A21 |
| 051-075 | 7.00 | A04 | 201-225 | 13.00 | A10 | 351-375 | 19.00 | A16 | 501-525 | 25.00 | A22 |
| 076-100 | 8.00 | A05 | 226-250 | 14.00 | A11 | 376-400 | 20.00 | A17 | 526-550 | 26.00 | A23 |
| 101-125 | 9.00 | A06 | 251-275 | 15.00 | A12 | 401-425 | 21.00 | A18 | 551-575 | 27.00 | A24 |
| 126-150 | 10.00 | A07 | 276-300 | 16.00 | A13 | 426-450 | 22.00 | A19 | 576-600 | 28.00 | A25 |
| | | | | | | | | | 601-up | † | A99 |

†Add $1.00 for each additional 25-page increment or portion thereof from 601 pages up.