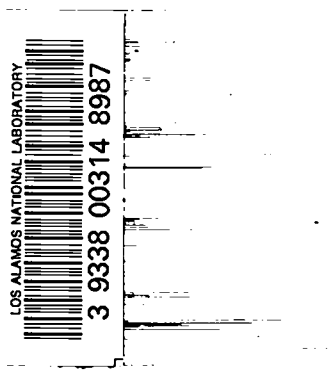
LA-3720

LOS ALAMOS SCIENTIFIC LABORATORY
of the
University of California
LOS ALAMOS • NEW MEXICO

FORTRAN SIN

A One-Dimensional Hydrodynamic Code for Problems Which Include Chemical Reactions, Elastic-Plastic Flow, Spalling, and Phase Transitions



UNITED STATES
ATOMIC ENERGY COMMISSION
CONTRACT W-7405-ENG. 36

This report expresses the opinions of the author or authors and does not necessarily reflect the opinions or views of the Los Alamos Scientific Laboratory.

Printed in the United States of America. Available from Clearinghouse for Federal Scientific and Technical Information National Bureau of Standards, U. S. Department of Commerce Springfield, Virginia 22151

Price: Printed Copy $3.00; Microfiche $0.65

LA-3720
UC-32, MATHEMATICS
AND COMPUTERS
TID-4500

LOS ALAMOS SCIENTIFIC LABORATORY
of the
University of California
LOS ALAMOS • NEW MEXICO

Report written: May 20, 1967
Report distributed: November 17, 1967

FORTRAN SIN

A One-Dimensional Hydrodynamic Code for Problems Which Include Chemical Reactions, Elastic-Plastic Flow, Spalling, and Phase Transitions

by

Charles L. Mader
William R. Gage

CONTENTS

FORTRAN SIN: A ONE-DIMENSIONAL HYDRODYNAMIC CODE FOR PROBLEMS WHICH INCLUDE CHEMICAL REACTIONS, ELASTIC-PLASTIC FLOW, SPALLING, AND PHASE TRANSITIONS

by

Charles L. Mader and William R. Gage

ABSTRACT

This report describes a CDC 6600 computer FORTRAN code for computing one-dimensional hydrodynamic problems in slab, cylindrical, or spherical geometry using realistic equations of state. Features available in the code include chemical reaction using an Arrhenius rate law, the C-J volume burn, or, for slabs, a gamma-law Taylor wave; elastic-plastic flow using the Hooke's law-Von Mises yield model; and spalling using the Whiteman and Skidmore model of the tensile stress at spalling as a linear function of the square root of the stress rate. The HOM equation of state is used to compute the equation of state for detonation products, undecomposed explosives, mixtures of the two, and condensed components which may have an instantaneous phase change.

I. INTRODUCTION

The SIN technique for solving reactive one-dimensional problems has been a useful research tool for almost ten years. The first version was written in machine language for the IBM 7090.[1] The second version was written in machine language for the IBM 7030 (STRETCH) and is called STRETCH SIN.[2] Because of its detailed treatment of the equation of state and its ability to compute the detailed structure of complicated reactive and nonreactive hydrodynamic problems, SIN has been used to study other interesting engineering and research problems. The new numerical techniques which resulted from these studies have been included in this new version of SIN, written in FORTRAN IV language for the CDC 6600 and called FORTRAN SIN.

The code includes the effects of heat conduction, viscosity, elastic-plastic flow, spalling, and reactive flow. As presented in this report, the code can handle 10 components and 999 mesh points; however, these limits can be increased several times without exceeding the core memory of the CDC 6600 or IBM 7030. The boundary conditions may be continuous, a free surface, a piston with a constant, a linearly varying velocity, or, for slabs, a reactive steady-state velocity.[3] The PIC or Landshoff forms for the artificial viscosity may be used, and a form for real viscosity in slab geometry is available.

The HOM equation of state is used. One may compute the equation of state of condensed explosives, detonation products, and mixtures of the two. The equation of state for the condensed component may include an instantaneous phase change such as that described for iron.[4]

The code includes the Hooke's law-Von Mises model for elastic-plastic flow as described for aluminum.[5] It also includes spalling, using the Whiteman and Skidmore model of the tensile stress at spalling as a linear function of the square root of the tensile stress.[6,7] It permits chemical reaction using an Arrhenius rate law, the C-J volume burn, or, for

slabs, a gamma-law Taylor wave.

The code includes features permitting microfilm listings and graphs of pressure, temperature, mass fraction, density, and particle velocity vs. Eulerian radius. Tape dumps are taken at input intervals of the cycle number, and the problem may be restarted from any desired tape dump. Problems with pistons may use the variable number of cells feature for a considerable reduction of running time.

The SIN code has been written not as a special purpose production code but as a general purpose research and engineering tool. It will continue to change (hopefully for the better) as new developments in the numerical description of real materials under high impulse become available.

This report describes the numerical methods of SIN in complete detail. It also presents sufficient details to enable a coder to follow and change the code. The latter information is not of interest to the casual reader.

II. THE FLOW EQUATIONS

The Nomenclature

D_{CJ}	the C-J detonation velocity
E	total energy (Mbar-cc/g)
E*	activation energy
I	internal energy (Mbar-cc/g)
j	net point of the Lagrangian mesh
K	a constant of about 2 or the coefficient of viscosity
M	mass
n	time cycle
P	pressure (Mbar)
q	artificial viscosity
R	Eulerian radius
r	Lagrangian radius
Sx	elastic stress deviator in x or r direction
Sz	elastic stress deviator in z direction
T	temperature (^{o}K)
U	particle velocity (cm/μsec)
V	specific volume (cc/g)
V_{CJ}	the C-J volume of the detonation products
W	mass fraction of undecomposed explosive
Y_o	yield strength
Z	frequency factor
α	=1 for slabs, =2 for cylinders, =3 for spheres
γ	γ-law gas constant since ln P = A + γ ln V, γ = B in HOM equation of state subroutine.
Δt	time (μsec)
λ	thermal conductivity coefficient
μ	shear modulus
ρ	= 1/V
ρ_o	$1/V_o$ where V_o is initial specific volume
$(\rho_o)_i$	initial density for the ith component (g/cc)

The Differential Equations

The Lagrangian conservation equations in one dimension for slabs, cylinders, and spheres are:

$$\frac{\partial U}{\partial t} = -R^{\alpha-1}\frac{\partial \sigma}{\partial M} - (\alpha-1)\frac{V\varphi}{R}$$ Conservation of momentum,

$$V = R^{\alpha-1}\frac{\partial R}{\partial M}$$ Conservation of mass,

and

$$\frac{\partial E}{\partial t} = \frac{-\partial \sigma U R^{\alpha-1}}{\partial M} + \lambda\frac{\partial}{\partial M}\left(R^{\alpha-1}\frac{\partial T}{\partial R}\right)$$ Conservation of energy with heat conduction term,

where $E = I + 0.5U^2$.

$$\frac{\partial R}{\partial t} = U,$$

and

$$dM = \rho_o r^{\alpha-1}\,dr = \rho R^{\alpha-1}\,dR,$$

where dM is element of mass per unit of solid angle (α=2 or 3) or of surface (α=1). σ is the sum of the viscous pressure, the equation of state pressure, and the stress deviator, Sx. $\varphi = 2Sx + Sz$, which for spherical geometry is 3/2 Sx since Sz = -1/2 Sx.

$$\frac{\partial Sx}{\partial t} = 2\mu\left(-\frac{\partial U}{\partial R} + \frac{1}{3V}\frac{\partial V}{\partial t}\right).$$

$$\frac{\partial Sz}{\partial t} = +\frac{2}{3}\left(\frac{\mu}{V}\frac{\partial V}{\partial t}\right).$$

In this report we use the convention that the stress deviators have the same sign as pressure; that is, positive in compression and negative in tension. This convention is the reverse of that used by most workers in the field.

The Difference Equations

The difference equations presented below are discussed in References 1 and 5. The pressure, temperature, energy, specific volume, and mass fraction of the cells are considered to be located at the centers of mass of the elements, and the particle velocity is considered to be located at the boundary between the cells.

A. The Initial Condiitons

$$R_{j+\frac{1}{2}} = \sum_{i=1}^{j} (\Delta R)_i,$$

where j = 1, 2, . . . , and the ith component is located between j and j+k where k is the number of cells for each component.

$$M_j = (\rho_o)_i \left[\left(R_{j-\frac{1}{2}} + R_{j+\frac{1}{2}}\right)^{1/2}\right]^{\alpha-1} (\Delta R)_i.$$

B. The Conservation Equations for Time Advancement

1. Subroutine VELOC (NMIN,NMAX)

$$U_{j+\frac{1}{2}}^{n+\frac{1}{2}} = U_{j+\frac{1}{2}}^{n-\frac{1}{2}} + \frac{(\Delta t)\left(R_{j+\frac{1}{2}}^{n}\right)^{\alpha-1}}{0.5(M_j + M_{j+1})} \left[\left(P_j^n - P_{j+1}^n\right) + \left(q_j^n - q_{j+1}^n\right)\right] - \frac{(\alpha-1)}{2} \frac{(\varphi)\left(V_j^n + V_{j+1}^n\right)(\Delta t)}{R_{j+\frac{1}{2}}^{n}},$$

where

$$\varphi = \frac{3\left(Sx_j^n + Sx_{j+1}^n\right)}{4} \quad \text{for } \alpha = 3,$$

and

$$\varphi = \frac{1}{2}\left[2\left(Sx_j^n + Sx_{j+1}^n\right) + \left(Sz_j^n + Sz_{j+1}^n\right)\right]$$

for $\alpha = 2$.

2. Subroutine RADIUS (NMIN,NMAX)

$$R_{j+\frac{1}{2}}^{n+1} = R_{j+\frac{1}{2}}^{n} + U_{j+\frac{1}{2}}^{n+\frac{1}{2}} (\Delta t). \qquad (1)$$

3. Subroutine VOLUM (NMIN,NMAX)

$$V_j^{n+1} = \left(\frac{R_{j-\frac{1}{2}}^{n+1} + R_{j+\frac{1}{2}}^{n+1}}{2}\right)^{\alpha-1} \left(R_{j+\frac{1}{2}}^{n+1} - R_{j-\frac{1}{2}}^{n+1}\right).$$

4. Subroutine ENERGY (NMIN,NMAX)

$$I_j^{n+1} = I_j^n + \frac{(\Delta t)}{M_j}\left\{\left[\frac{M_j P_{j-1}^n + M_{j-1} P_j^n}{M_j + M_{j-1}} + 0.5(q_j^n + q_{j-1}^n)\right] U_{j-\frac{1}{2}}^{n+\frac{1}{2}} \left(R_{j-\frac{1}{2}}^{n+1}\right)^{\alpha-1} - \left[\frac{M_{j+1} P_j^n + M_j P_{j+1}^n}{M_j + M_{j+1}} + 0.5(q_j^n + q_{j+1}^n)\right] U_{j+\frac{1}{2}}^{n+\frac{1}{2}} \left(R_{j+\frac{1}{2}}^{n+1}\right)^{\alpha-1}\right\} + \frac{1}{8}\left[\left(U_{j+\frac{1}{2}}^{n-\frac{1}{2}} + U_{j-\frac{1}{2}}^{n-\frac{1}{2}}\right)^2 - \left(U_{j+\frac{1}{2}}^{n+\frac{1}{2}} + U_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right)^2\right].$$

5. Subroutine HETCON (NMIN,NMAX)

$$I_j^{n+1} = I_j^{n+1} + \frac{\lambda(\Delta t)}{M_j}\left[\frac{\left(R_{j+\frac{1}{2}}^{n+1}\right)^{\alpha-1}\left(T_{j+1}^n - T_j^n\right)}{\frac{1}{2}\left(R_{j+1\frac{1}{2}}^{n+1} - R_{j-\frac{1}{2}}^{n+1}\right)} - \frac{\left(R_{j-\frac{1}{2}}^{n+1}\right)^{\alpha-1}\left(T_j^n - T_{j-1}^n\right)}{\frac{1}{2}\left(R_{j+\frac{1}{2}}^{n+1} - R_{j-1\frac{1}{2}}^{n+1}\right)}\right].$$

C. Viscosity - Subroutine VISCOS (NMIN,NMAX)

1. PIC Form

$$q_j^{n+1} = \frac{K}{V_j^{n+1}} (0.5)\left(U_{j-\frac{1}{2}}^{n+\frac{1}{2}} + U_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right)\left(U_{j-\frac{1}{2}}^{n+\frac{1}{2}} - U_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right),$$

if $\left(U_{j-\frac{1}{2}}^{n+\frac{1}{2}} - U_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right)$ is positive; otherwise, $q_j^{n+1} = 0$. The absolute value of q is used.

2. Landshoff Form

$$q_j^{n+1} = \frac{K}{V_j^{n+1}}\left(U_{j-\frac{1}{2}}^{n+\frac{1}{2}} - U_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right).$$

Restrictive conditions are the same as for the PIC form.

3. "Real" Form

$$q_j^{n+1} = 1.333 \frac{K}{V_j^{n+1}} \frac{\left(U_{j-\frac{1}{2}}^{n+\frac{1}{2}} - U_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right)}{M_j},$$

where K is the "coefficient of viscosity," and this form is appropriate only for slabs.

D. Burn Techniques - Subroutine BURN (NMIN,NMAX) for 1 and 2

1. Arrhenius Burn

$$W_j^{n+1} = W_j^n - \Delta t Z W_j^n e^{-E^*/R_g T_j^n}$$

where $1 \geq W \geq 0$.

2. C-J Volume Burn

Assumes that W varies linearly with V from V_o to V_{CJ}.

$$W_j^{n+1} = 1 - \frac{V_o - V_j^{n+1}}{V_o - V_{CJ}}$$

where

$$1 \geq W_j^{n+1} \geq 0,$$

and

$$W_j^n \geq W_j^{n+1} .$$

$$P_j^{n+1} = \left(1 - W_j^{n+1}\right)(P^1)$$

where P^1 is pressure of detonation products at V, I, and W = 0, if $W_j^{n+1} < 0.99$ (WMAX); otherwise, $P_j^{n+1} = P_o$. The P_j^{n+1} calculation is actually performed in the subroutine EQST.

3. Gamma-Law Taylor Wave Burn

The explosive is burned before the first time interval by assuming it to be a gamma-law explosive that has been detonated with a rear boundary of constant velocity. Coded as part of initial setup.

Knowing ρ_o, γ, D_{CJ}, we compute

$$U_{CJ} = \frac{D_{CJ}}{\gamma + 1}$$

$$P_{CJ} = \frac{\rho_o D_{CJ}^2}{\gamma + 1}$$

$$V_{CJ} = \left(\frac{\gamma}{\gamma + 1}\right)\left(\frac{1}{\rho_o}\right)$$

$$C_{CJ} = D_{CJ} - U_{CJ}$$

$$k = \frac{\gamma - 1}{\gamma + 1}$$

$$\ell = U_{CJ} - \frac{2}{\gamma - 1} C_{CJ}$$

and

$$t = \frac{R_n - R_o}{D_{CJ}}$$

where $R_n - R_o$ is thickness of explosive.

For each cell with a radius of $R_{j-\frac{1}{2}}$, we compute

$$Y = \left(R_n - R_{j-\frac{1}{2}}\right)/t$$

$$U_{j+\frac{1}{2}} = \frac{2}{\gamma + 1} Y + k\ell$$

$$C_j = \frac{\gamma - 1}{2}\left(U_{j+\frac{1}{2}} - \ell\right)$$

$$P_j = P_{CJ}\left(\frac{C_j}{C_{CJ}}\right)^{2\gamma/\gamma-1}$$

$$V_j = \left[\frac{[P_{CJ}(V_{CJ})^\gamma]}{P_j}\right]^{1/\gamma}$$

$$I_j = \frac{P_j V_j}{\gamma - 1} - \frac{P_{CJ} V_{CJ}}{\gamma - 1} + \frac{P_{CJ}}{2}(V_o - V_{CJ})$$

and

$$R_{j+\frac{1}{2}} = R_{j-\frac{1}{2}} + (M_j)(V_j).$$

With the new $R_{j+\frac{1}{2}}$ set equal to $R_{j-\frac{1}{2}}$, we return to compute Y, etc. When $U_{j+\frac{1}{2}}$ is equal to the applied piston velocity, the rest of the explosive cells are made identical to the last cell calculated.

E. The Elastic Stress Deviators - Subroutine STRESS (NMIN,NMAX)

$$Sx_j^{n+1} = Sx_j^n + 2\mu\left[-\left(\frac{U_{j+\frac{1}{2}}^{n+\frac{1}{2}} - U_{j-\frac{1}{2}}^{n+\frac{1}{2}}}{R_{j+\frac{1}{2}}^{n+1} - R_{j-\frac{1}{2}}^{n+1}}\right)(\Delta t) + \frac{2}{3}\left(\frac{V_j^{n+1} - V_j^n}{V_j^{n+1} + V_j^n}\right)\right].$$

$$Sz_j^{n+1} = Sz_j^n + \frac{4}{3}\mu\left(\frac{V_j^{n+1} - V_j^n}{V_j^{n+1} + V_j^n}\right).$$

F. The Equation of State and Yield Calculation - Subroutine EQST (NMIN,NMAX)

Enter HOM Subroutine with V_j^{n+1}, I_j^{n+1}, W_j^{n+1} to find P_j^{n+1}.

If $P_j^{n+1} > P_o$,

then

$$(P_{hyd})_j^{n+1} = (P_{hydrostat})_j^{n+1} = P_j^{n+1} - (2/3\ Y_o)\left(\frac{P_j^{n+1}}{0.050}\right)$$

where

$$\frac{P_j^{n+1}}{0.05} \leq 1.0.$$

The 0.050 is input PLAP. If $P_j^{n+1} < P_o$, then

$$(P_{hyd})_j^{n+1} = P_j^{n+1}.$$

For $\alpha = 1$ or 3 if

$$|Sx_j^{n+1}| \geq |2/3\ Y_o|,\ P_j^{n+1} = (P_{hyd})_j^{n+1} \pm 2/3\ Y_o$$

where the sign of $(2/3\ Y_o)$ is identical to the sign of Sx_j^{n+1}. Or if

$$|S_j^{n+1}| < |2/3\ Y_o|,\ P_j^{n+1} = (P_{hyd})_j^{n+1} + Sx_j^{n+1}.$$

For $\alpha = 2$

$$f = 2\left[(Sx_j^{n+1})^2 + Sx_j^{n+1}\ Sz_j^{n+1} + (Sz_j^{n+1})^2\right].$$

Then if

$$f < 2/3\ (Y_o)^2,\ P_j^{n+1} = (P_{hyd})_j^{n+1} + Sx_j^{n+1},$$

and

$$Sz_j^{n+1} = Sz_j^{n+1}.$$

Or if

$$f > 2/3\ (Y_o)^2,\ P_j^{n+1} = (P_{hyd})_j^{n+1} + \left(\sqrt{\frac{2/3\ (Y_o)^2}{f}}\right)(Sx_j^{n+1}),$$

and

$$Sz_j^{n+1} = (Sz_j^{n+1})\left(\sqrt{\frac{2/3\ (Y_o)^2}{f}}\right).$$

G. Total Energy of Components Coded as Part of Subroutine TIME

$$\text{Kinetic energy} = \sum_1^j \frac{\xi\ M_j \left(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right)^2}{(2)(0.9992768)}.$$

$$\text{Internal energy} = \sum_1^j \xi\ M_j \left(I_j^{n+1}\right),$$

where $\xi = 4\pi$ for spheres, 2π for cylinders, and 1 for slabs; and the units of energy are Mbar-cc/g.

H. The Boundary Conditions

1. An Applied Piston on Right Boundary

$U_{j+\frac{1}{2}}^n = U_{piston}$ where outside boundary is $j+\frac{1}{2}$, if $W_{j-3}^n > 0.5$; otherwise = final U_{piston}, or $U_{piston} = A + (B)(Time)$. For both,

$$P_{j+1}^n = P_j^n, \text{ and } Q_{j+1}^n = Q_j^n.$$

2. An Applied Piston on Left Boundary

$U_{+\frac{1}{2}}^n = U_{+\frac{1}{2}}^{n+1} = U_{piston}$ where boundary is at $j = \frac{1}{2}$ if $W_{j+3}^n > 0.5$; otherwise = final U_{piston}, or $U_{piston} = A + (B)(Time)$. For both,

$$P_o^n = P_1^n, \text{ and } q_o^n = q_1^n.$$

$R_{\frac{1}{2}}$ is computed by Equation (1).

3. A Steady-State Reaction Zone Piston on Left Boundary

The steady-state reaction zone piston is computed by iteration for a given detonation velocity by using the amount of reaction that occurred in a cell near the piston to determine the proper particle velocity of the piston.

For a W, iterate on V using linear feedback by calculating

$$I = I_o + \tfrac{1}{2}(\rho_o)^2(D_{CJ})^2(V_o - V)^2$$

and

$$P_r = P_o + (\rho_o)^2(D_{CJ})^2(V_o - V)$$

where P_r is Rayleigh line pressure.

Using the HOM equation of state, calculate P for V, I, and W. Continue iteration until $P_r - P \leq 1 \times 10^{-5}$. Calculate $U = \sqrt{(P - P_o)(V_o - V)}$ and assign it to final U_{piston}.

4. A Right Free-Surface Boundary

$$P_{j+1}^n = -P_j^n \text{ where outside boundary is } j+\tfrac{1}{2}.$$

$$U_{j+1}^n = U_j^n.$$

5. A Left Free-Surface Boundary

$P_o^n = -P_1^n$ where boundary is at $j = \frac{1}{2}$.

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

$q_o^n = q_1^n$.

6. A Right Continuum Boundary

$P_{j+1}^n = P_j^n$ where outside boundary is $j+\frac{1}{2}$.

7. A Left Continuum Boundary

$P_o^n = P_1^n$ where boundary is at $j = \frac{1}{2}$.

$q_o^n = q_1^n$.

$U_{\frac{1}{2}}^n = U_{1\frac{1}{2}}^n$ unless $\alpha = 2$ or 3 when $U_{\frac{1}{2}}^n = -U_{1\frac{1}{2}}^n$.

III. THE HOM EQUATION OF STATE SUBROUTINE

HOM is a FORTRAN subroutine which calculates the pressure and temperature given the internal energy, specific volume, and mass function of the solid for solids, gases, and mixtures.

The Nomenclature

C,S	coefficients to a linear fit of U_s and U_p
C1,S1	second set of coefficients to a linear fit of U_s and U_p
C_V	heat capacity of condensed component (cal/g/deg)
C_V'	heat capacity of gaseous component (cal/g/deg)
I	total internal energy (Mbar-cc/g)
P	pressure (Mbar)
SPA	spalling constant to relate spall pressure and tension rate
SPALL P	interface spalling pressure
T	temperature (^oK)
USP	ultimate spalling pressure
U_p	particle velocity
U_s	shock velocity
V	total volume (cc/g)
V_o	initial volume of condensed component (cc/g)
W	mass fraction of undecomposed explosive

Subscripts

g	gaseous component
H	Hugoniot
i	isentrope
s	condensed component

The Method

A. Condensed Components

(The mass fraction, W, is 1; the internal energy, I, is I_s; and the specific volume, V, is V_s). For volumes less than V_o, the experimental Hugoniot data are expressed as a linear fit of the shock and particle velocities. The Hugoniot temperatures are computed using the code described in Appendix C.

$$U_s = C + SU_p.$$

$$P_H = \frac{C^2(V_o - V_s)}{[V_o - S(V_o - V_s)]^2}.$$

$$\ln T_H = F_s + G_s \ln V_s + H_s(\ln V_s)^2 + I_s(\ln V_s)^3 + J_s(\ln V_s)^4.$$

$$I_H = \frac{1}{2} P_H(V_o - V_s).$$

$$P_s = \frac{\gamma_s}{V_s}(I_s - I_H) + P_H, \text{ where } \gamma_s = V\left(\frac{\partial P}{\partial E}\right)_V. \quad (2)$$

$$T_s = T_H + \frac{(I_s - I_H)(23{,}890)}{C_V}. \quad (3)$$

Two sets of C and S coefficients may be given. For $V_s <$ MINV, the fit $U_s = C1 + S1(U_p)$ is used with the corresponding changes to the above equations. Between MINV and VSW, the volume is set equal to MINV, and $U_s = C1 + S1(U_p)$ is used. For volumes greater than V_o, we use the Grüneisen equation of state and the P = 0 line as the standard curve.

$$P_s = \left[I_s - \frac{C_V}{(3)(23890)(\alpha)}\left(\frac{V_s}{V_o} - 1\right)\right]\frac{\gamma_s}{V_s}.$$

$$T_s = \frac{(I_s)(23{,}890)}{C_V} + T_o.$$

The spalling option is not used if SPA < 0.0001. If $P_s \le$ USP, set $P_s =$ SPALL P and set spall indicator. If $P_s \le \text{SPA}\sqrt{\Delta P/\Delta X}$ ($\Delta P/\Delta X$ is the tension

rate), and $P_s \leq$ SPMIN (5×10^{-3}), set P_s = SPALL P and set spall indicator. Do not spall if neither of the above conditions are satisfied.

B. Gas Components

(Mass fraction, W, is 0; the internal energy, I, is I_g; and the specific volume, V, is V_g). The pressure, volume, temperature, and energy values of the detonation products are computed using FORTRAN BKW[8] and fitted by a method of least squares to Eq. (4) through (6). A gamma-law gas may also be fit to these equations as a special case. A code to perform this is described in Appendix C.

$$\ln P_i = A + B\ln V_g + C(\ln V_g)^2 + D(\ln V_g)^3 + E(\ln V_g)^4 . \tag{4}$$

$$\ln I_i = K + L\ln P_i + M(\ln P_i)^2 + N(\ln P_i)^3 + O(\ln P_i)^4 . \tag{5}$$

$I_i = I_i - Z$ (where Z is a constant used to change the standard state to be consistent with the solid explosive standard state, and if the states are the same is used to keep I positive when making a fit).

$$\ln T_i = Q + R\ln V_g + S(\ln V_g)^2 + T(\ln V_g)^3 + U(\ln V_g)^4 . \tag{6}$$

$$-\frac{1}{3} = R + 2S\ln V_g + 3T(\ln V_g)^2 + 4U(\ln V_g)^3 .$$

$$P = \left(\frac{1}{\partial V_i}\right)(I_g - I_i) + P_i . \tag{7}$$

$$T = T_i + \frac{(I_g - I_i)(23{,}890)}{C_V'} \tag{8}$$

C. Mixture of Condensed and Gaseous Components

($0 < W < 1$)

$V = WV_s + (1 - W)V_g .$

$I = WI_s + (1 - W)I_g .$

$P = P_g = P_s .$

$T = T_g = T_s .$

Multiplying Eq. (3) by (W/C_V) and Eq. (8) by $(1 - W)/C_V$ and adding, we get, after substituting T for T_s and T_g and I for $WI_s + (1 + W)I_s$,

$$T = \frac{23{,}890}{C_V W + C_V'(1 - W)}\left\{I - \left[WI_H - I_i(1 - W)\right] + \frac{1}{23{,}890}\left[T_H C_V W + T_i C_V'(1 - W)\right]\right\} . \tag{9}$$

Equating Eq. (2) and (7) and substituting from (9), we get

$$P_H - P_i + \left(\frac{\gamma_s C_V}{V_s} - \frac{C_V'}{\partial V_g}\right)\left(\frac{1}{C_V W + C_V'(1 - W)}\left\{I - \left[WI_H + I_i(1 - W)\right] + \frac{1}{23{,}890}\left[T_H C_V W + T_i C_V'(1 - W)\right]\right\}\right) - \frac{1}{23{,}890}\left(\frac{\gamma_s C_V T_H}{V_s} - \frac{C_V' T_i}{\partial V_g}\right) = 0. \tag{10}$$

Knowing V, I, and W, one may use the linear feedback to iterate on either V_s or V_g until Eq. (10) is satisfied.

For $V < V_o$, we iterate on V_s with an initial guess of $V_s = V_o$ and a ratio to get the second guess of 0.999. For $V \geq V_o$, we iterate on V_g with an initial guess of $V_g = (V - 0.9\, V_o W)/(1 - W)$ and a ratio to get the second guess of 1.002.

If the iteration goes out of the physical region ($V_g \leq 0$ or $V_s \leq 0$), that point is replaced by $V_s = V_g = V$. Then knowing V_s and V_g, we calculate P and T.

The Calling Sequence

Call HOM (V, S, G, IND)

V, S, and G are dimensioned arrays of size 5, 23, and 17 numbers, respectively.

V(1)	specific volume V
V(2)	internal energy I
V(3)	mass fraction W
V(4)	$-\left\|\frac{\Delta P}{\Delta x}\right\|$ input; pressure P output
V(5)	temperature T output
S(1)	C
S(2)	S
S(3)	VSW
S(4)	C1
S(5)	S1
S(6)	F
S(7)	G
S(8)	H
S(9)	I

S(10)	J
S(11)	Y_s
S(12)	C_V
S(13)	V_o
S(14)	α
S(15)	SPA
S(16)	USP
S(17)	T_o
S(18)	P_o
S(22)	SPALL P
S(23)	MINV
G(1)	A
G(2)	B
G(3)	C
G(4)	D
G(5)	E
G(6)	K
G(7)	L
G(8)	M
G(9)	N
G(10)	O
G(11)	Q
G(12)	R
G(13)	S
G(14)	T
G(15)	U
G(16)	C_V'
G(17)	Z

Ind set to 0 for normal exit, to -1 for iteration error in mixture calculations, and to +1 for a spalled solid.

APPENDIX A
FORTRAN SIN INPUT

The following is a description of the input to FORTRAN SIN.

A few rules for setting up a problem are:

1. The steady-state reaction zone piston is suitable only for slab geometry and only at the left boundary.

2. The gamma-law Taylor wave option assumes that the user is using a gamma-law HOM equation of state. It is suitable only for slab geometry and assumes that the explosive has been burned from right to left. The option assumes that the user is using a right-boundary, initial-final velocity piston and uses the final velocity as the lowest particle velocity permitted in the detonation products Taylor wave.

3. The real viscosity option is suitable only for slab geometry.

4. The spalling option is suitable only for problems with approximately constant tension gradients and many mesh points.

5. The thermal conductivity option assumes that the system will have uniform thermal conductivity.

6. To restart from tape, one needs only to put the last tape dump number in column 1-5 of the first card. This tape dump number is printed on the listings. When the number in column 1-5 of the first card is greater than 10, the code searches through the tape dumps for a dump at a cycle greater than or equal to the number on the input card. When the proper dump is found, all the variables are read and the calculation proceeds.

1st card

Col.	Format	
1-5	I5	Number of components or, if > 10, the tape restart cycle number
6-10	I5	Total number of space increments
11-15	I5	Not used
16-20	I5	Not 0 for 4020 output
21-25	I5	Not 0 for graph of pressure vs. radius
26-30	I5	Not 0 for graph of temperature vs. radius
31-35	I5	Not 0 for graph at mass fraction vs. radius
36-40	I5	Not 0 for graph of volume vs. radius
41-45	I5	Not 0 for graph of particle velocity vs. radius
46-50	I5	Not 0 for inclusion of heat conduction
51-55	I5	Left boundary indicator 0 for continuum 1 for free surface 2 for initial velocity-final velocity piston 3 for A + BT velocity piston 4 for steady-state reaction zone piston
56-60	I5	Right boundary indicator

		0 for continuum
		1 for free surface
		2 for initial velocity-final velocity piston
		3 for A + BT velocity piston

2nd card

Col.	Format	
1-72	12A6	72 columns of alphabetic label for problem

3rd card

Col.	Format	
1-18	E18.11	Alpha - 1. for slab geometry 2. for cylindrical geometry 3. for spherical geometry
19-36	E18.11	Print every this many cycles
37-54	E18.11	Graph every this many cycles
55-72	E18.11	Tape dump every this many cycles

4th card

Col.	Format	
		Left boundary card - If a left boundary initial-final piston is requested, this card is:
1-18	E18.11	Initial piston velocity
19-36	E18.11	Final piston velocity
37-41	I5	Number of cells from left initially in calculation; if 0, all cells are used.
42-46	I5	Add on a cell every this many cycles.
		If a left-boundary A + BT piston is requested, this card is:
1-18	E18.11	A
19-36	E18.11	B
37-41	I5	Same as initial-final piston
42-46	I5	Same as initial-final piston
		If a left steady-state piston is requested, this card is:
1-18	E18.11	Detonation velocity
19-36	E18.11	Spike volume
37-41	I5	Same as initial-final piston
42-46	I5	Same as initial-final piston
		If a piston is not requested, no card appears for the left boundary.

Next card

Col.	Format	
1-18	E18.11	Right boundary card -
etc.	etc.	same as left boundary card. Again if not a piston, there is no card. A right steady-state reaction zone piston is not permitted. A right boundary initial-final velocity piston is required if a gamma-law Taylor wave burn is requested.

Next card

Col.	Format	
		If heat conductivity is included, this card contains:
1-18	E18.11	Heat conduction constant - same for all components.

The following cards are present for each component.

1st comp. card

Col.	Format	
1-12	2A6	Alphabetic name of component

2nd comp. card

Col.	Format	
1-5	I5	Number of space increments for this component
6-10	I5	Not 0 for gas or explosive
11-15	I5	0 for Arrhenius burn, 1 for C-J volume burn, 2 for gamma-law Taylor wave
16-20	I5	0 for PIC viscosity, 1 for Landshoff viscosity, 2 for real viscosity

3rd comp. card

Col.	Format	
1-18	E18.11	Space increment (cm)
19-36	E18.11	Time increment (μsec)
37-54	E18.11	Viscosity factor
55-72	E18.11	Initial density (g/cc)

4th comp. card

Col.	Format	
1-18	E18.11	Initial pressure (Mbar)
19-36	E18.11	Initial temperature (^{o}K)
37-54	E18.11	Initial internal energy (Mbar-cc/g)
55-72	E18.11	Initial velocity (cm/μsec)

5th-10th comp. card

Six cards of solid parameters for equation of state and elastic-plastic calculations. The variables are four per card with each card of the format 4E18.11. They are, in order: C, S, VSW, C1, S1, F, G, H, I, J, Y_s, C_V, V_o, α, SPA, USP, T_o, P_o, 2/3 Y_o, μ, PLAP, SPALL P, and MINV.

11th comp. card

Col.	Format	
1-18	E18.11	Initial mass fraction of the solid
19-36	E18.11	Frequency factor for Arrhenius burn
37-54	E18.11	Activation energy for Arrhenius burn
55-72	E18.11	Volume for C-J burn or C-J detonation velocity for a gamma-law Taylor wave

For a pure solid there are no more cards.

For a gas or explosive we have,

12th-16th comp. cards

Five cards of gas equation of state parameters. The variables are four to a card with each card of the format 4E18.11. They are, in order: A, B, C, D, E, K, L, M, N, O, Q, R, S, T, U, C_V', and Z.

APPENDIX B

THE FORTRAN SIN CODE

In this appendix we list the names or symbols of the variables in SIN both as used in the FORTRAN code and in this report. The listing of the FORTRAN code as it existed at the time of preparation of this report is also presented. Errors in this version of the code will be corrected as they become known to the authors. Anyone wishing to actually copy any part of this code should contact the authors for the latest version.

Variables Used in FORTRAN SIN

Code Name	Report Symbol
CR - cell radius	$R_{j-\frac{1}{2}}^{n+1}$
CRO - previous cell radius	$R_{j-\frac{1}{2}}^{n}$
CU - cell velocity	$U_{j-\frac{1}{2}}^{n+\frac{1}{2}}$
CUO - previous cell velocity	$U_{j-\frac{1}{2}}^{n-\frac{1}{2}}$
CV - cell volume	V_j^{n+1}
CVO - previous cell volume	V_j^n
CM - cell mass	M_j
CQ - cell viscosity	Q_j^n
ICF - cell flag word	
$XXYY_i$ - YY component index - XX spall flag 0 - no spall 1 - spalled cell 2 - interface cell	
CW - cell mass fraction	W_j^{n+1}
CWO - previous cell mass fraction	W_j^n
CT - cell temperature	T_j^n
CP - cell pressure	P_j^n
CS - cell stress deviator in x or r direction	Sx_j^n
CSZ - cell stress deviator in z direction	Sz_j^n
CI - cell internal energy	I_j^{n+1}
CIO - previous cell internal energy	I_j^n

LABEL - 12 words containing 72 columns of alphabetic information - the problem label - format 12A6

TTIME - total time elapsed in problem

DELT - time increment for this cycle

ICYCL - cycle number

PCNT - count for printing

GCNT - count for graphing

DCNT - count for dumping

PINC - print every PINC cycles

GINC - graph every GINC cycles

DINC - dump every DINC cycles

IALPH - 0 for slab, 1 for cylinder, 2 for sphere

HEATC - heat conductivity constant

GASW - set W to 0 if less than GASW

NCL - total number of cells in calculation

PTMIN - the pressure of one cell of a component must be greater than PTMIN (1. x 10^{-5}) to use that component's delta t.

IPNT - unused

IMC - not 0 for microfilm output

IPR - not 0 for a pressure vs. radius graph

ITR - not 0 for a temperature vs. radius graph

IWR - not 0 for a mass fraction vs. radius graph

IIR - not 0 for an internal energy vs. radius graph

IUR - not 0 for a particle velocity vs. radius graph

IHC - not 0 to include heat conduction

ILB - left boundary indicator

At Read-In	During Execution	
0	1	free surface
1	2	continuum
2	3	initial-final piston
3	4	A + BT piston
4	5	steady-state reaction zone piston

IRB - right boundary indicator (same as left except no reaction zone piston)

WSW - W to switch from initial to final for that piston

WMAX - for $W > WMAX$ (0.99) in C-J burn, $P = P_o$

SMIN - change in stress set to 0 if less than SMIN (1.0×10^{-9})

ALB, BLB - left boundary piston constants: either initial and final velocities or A and B in A + BT

ARB, BRB - right boundary piston constants (same as left)

NLI - number of cells initially in calculation from left. If 0, all cells are used.

NLINC - add on a cell from left every NLINC cycles

NLCNT - count for adding on cells from left

NRI, NRINC, NRCNT - same as above, only from right. Note: These numbers are used only if there is a piston at their respective boundaries. If there is no piston specified, all cells are used.

NLH - number of cells from the left in calculation at the time of interest

NRL - index of lowest cell in calculation from right end

DX - at input-space increment for each component-during calculation - space increment at right-most component

E - activation energy for Arrhenius buen

GAS - gas parameters

SOL - solid parameters

VCJ - C-J burn volume

VFACT - viscosity factor

Z - frequency factor for Arrhenius burn

NAM - alphabetic names of components (2 words per component)

DTIME - delta t for each component

NINC - number of space increments for each component

NCOM - number of components

IEXP - not 0 for explosive or gas

IBRN - 0 for Arrhenius burn, 1 for C-J volume burn, 2 for gamma-law Taylor wave

IVIS - 0 for PIC viscosity, 1 for Landshoff form, 2 for real form

ALPH - 1 for slab, 2 for cylinder, 3 for sphere

RHOO - initial density, only for the component being read

PO - initial pressure, only for the component being read

TO - initial temperature, only for the component being read

EO - initial internal energy, only for the component being read

UO - initial particle velocity, only for the component being read

WO - initial mass fraction, only for the component being read

NINC - number of space increments for this component

RFRNT - r coordinate value of the nearest spalled or interface cell of the same component with a lower index

EQ - five parameters used to call HOM. EQ(1) - volume, EQ(2) - energy, EQ(3) - mass fraction, EQ(4) - $|\Delta P/\Delta X|$ for spall input or pressure output, EQ(5) - temperature output

INDH - HOM indicator -1 error, 0 normal, +1 spalled

X - elastic-plastic ratio of P/PLAP, not greater than 1

DELS - change in stress deviator

TKE - total kinetic energy of each component

TIE - total internal energy of each component

LIN - line counter for printing

KIND - the index of cell on right-hand boundary

DCJ - C-J detonation velocity for gamma-law Taylor wave burn

DRT - total delta R for gamma-law Taylor wave burn

DRF - first R for gamma-law Taylor wave burn

DRL - last R for gamma-law Taylor wave burn

UCJ, PCJ, VCR, CCJ - C-J values for gamma-law Taylor wave burn

COK - $(\gamma - 1)/(\gamma + 1)$ for gamma-law Taylor wave burn

COL - UCJ - $\frac{2c}{\gamma - 1}$ for gamma-law Taylor wave burn

RXZDV - steady-state reaction zone piston detonation velocity

RXZVS - steady-state reaction zone piston spike volume guess

Listing of the FORTRAN Code

```
      PROGRAM FSIN (INPUT,OUTPUT,FILM,TAPE12=FILM,TAPE1)
C     CR        CELL RADIUS   J-1/2
C     CU        CELL VELOCITY  J-1/2
C     CV        CELL VOLUME  J
C     CM        CELL MASS  J
C     CQ        CELL VISCOSITY  J
C     ICF       CELL FLAG WORD  J
C     CW        CELL MASS FRACTION  J
C     CT        CELL TEMPERATURE  J
C     CP        CELL PRESSURE  J
C     CS        CELL STRESS DEVIATOR  J
C     CSZ       CELL STRESS DEVIATOR IN Z DIRECTION    J
C     CI        CELL INTERNAL ENERGY  J
000003      COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
     1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
     2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000003      COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
     1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,
     2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
     3NLCNT,NRCNT,NLH,NRL,DX,KIND,RXZDV,RXZVS,FPO,FEO,RXZ
000003      COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SDL(23,10),VCJ(10),
     1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
     2NCOM
000003      DIMENSION RXZ(10),EQ(5)
C     WSW    W TO SWITCH PISTON VELOCITY FROM INITIAL TO FINAL
000003      DATA WSW /0.02/
000003      DATA NRI /0/
000003      DATA NLI /0/
000003      DATA NLH /1/
C     COUNTS ARE SET LARGE SO CELLS ARENT ADDED ON IF NOT REQUESTED
000003      DATA NRCNT/100000/
000003      DATA NLCNT/100000/
C     READ INPUT DATA
C     NCOM      NUMBER OF COMPONENTS OR RESTART CYCLE NUMBER
C     NCL       TOTAL NUMBER OF CELLS IN THE CALCULATION
C     IPNT       UNUSED
C     IMC       NOT 0 FOR MICROFILM OUTPUT
C     IPR      NOT 0 TO GIVE A GRAPH OF PRESSURE
C     ITR      NOT 0 TO GIVE A GRAPH OF TEMPERATURE
C     IWR      NOT 0 TO GIVE A GRAPH OF MASS FRACTION
C     IIR      NOT 0 TO GIVE A GRAPH OF  VOLUME
C     IUR      NOT 0 TO GIVE A GRAPH OF PARTICLE VELOCITY
C     IHC       NOT 0 FOR CALCULATION OF HEAT CONDUCTION IN ENERGY
C     ILB       LEFT HAND BOUNDARY INDICATOR
C               = 0 FOR CONTINUUM
C               = 1 FOR FREE SURFACE
C               = 2 FOR INITIAL-FINAL PISTON
C               = 3 FOR A+BT PISTON
C               = 4  FOR STEADY STATE REACTION ZONE SLAB PISTON
C     IRB       RIGHT HAND BOUNDARY INDICATOR
C               = 0 FOR CONTINUUM
C               = 1 FOR FREE SURFACE
C               = 2 FOR INITIAL-FINAL PISTON
C               = 3 FOR A+BT PISTON
000003      READ 901,NCOM,NCL,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,IRB
C     SEE IF TAPE RESTART
000037      IF (NCOM.GT.10) GO TO 500
C     LABEL     72 COLUMNS OF ALPHABETIC COMMENT
000043      READ 900,LABEL
C     ALPH = 1.  FOR SLAB
C          = 2. FOR CYLINDER
```

```
C           = 3. FOR SPHERE
C      PINC      PRINT EVERY PINC CYCLES
C      GINC      GRAPH EVERY GINC CYCLES
C      DINC      DUMP EVERY DINC CYCLES
000050         READ 902,ALPH,PINC,GINC,DINC
000064         IALPH=ALPH-1.
C      READ IN PISTON QUANTITIES
C      ALB,BLB   PISTON CONSTANTS FOR LEFT PISTON
C      NLI       NUMBER OF CELLS INITIALLY IN CALCULATION FROM LEFT
C      NLINC     ADD ON A CELL FROM LEFT EVERY NLINC CYCLES
C      RXZDV     STEADY STATE REACTION ZONE PISTON DETONATION VELOCITY
C      RXZVS     STEADY STATE REACTION ZONE PISTON GUESSED SPIKE VOLUME
000067         IF (ILB.EQ.2.OR.ILB.EQ.3) READ 903,ALB,BLB,NLI,NLINC
000111         IF(ILB.EQ.4) READ 903,RXZDV,RXZVS,NLI,NLINC
C      NRI       NUMBER OF CELLS INITIALLY IN CALCULATION FROM RIGHT
C      NRINC     ADD ON A CELL FROM RIGHT EVERY NRINC CYCLES
C      ARB,BRB   PISTON CONSTANTS FOR RIGHT PISTON
000127         IF (IRB.EQ.2.OR.IRB.EQ.3) READ 903,ARB,BRB,NRI,NRINC
C      READ IN HEAT CONSTANT
000152         IF (IHC.NE.0) READ 902,HEATC
000161         K=2
000162         CRO(2)=0.
000163         CR(2)=0.
C      IALPH = 0 FOR SLAB
C              1 FOR CYLINDER
C              2 FOR SPHERE
000164         IF (IALPH.EQ.0) PRINT 904,LABEL
000172         IF (IALPH.EQ.1) PRINT 905,LABEL
000202         IF (IALPH.EQ.2) PRINT 906,LABEL
000212         PRINT 907,NCOM
000220         PRINT 908,NCL
000226         IF (ILB.EQ.0) PRINT 909
000233         IF (ILB.EQ.1) PRINT 910
000241         IF (ILB.EQ.2) PRINT 911,ALB,BLB,WSW,NLI,NLINC
000261         IF (ILB.EQ.3) PRINT 912,ALB,BLB,NLI,NLINC
000277         IF(ILB.EQ.4) PRINT930,RXZDV,RXZVS,NLI,NLINC
000315         IF (IRB.EQ.0) PRINT 913
000322         IF (IRB.EQ.1) PRINT 914
000330         IF (IRB.EQ.2) PRINT 915,ARB,BRB,WSW,NRI,NRINC
000350         IF (IRB.EQ.3) PRINT 916,ARB,BRB,NRI,NRINC
000366         PRINT 917
C      DO 4020 PRINTING IF REQUESTED
000372         IF (IMC.EQ.0) GO TO 1
C      ************************************************************
C                4020 OUTPUT BELOW
C      ************************************************************
000373         IF (IALPH.EQ.0) WRITE (12,904) LABEL
000402         IF (IALPH.EQ.1) WRITE (12,905) LABEL
000412         IF (IALPH.EQ.2) WRITE (12,906) LABEL
000422         WRITE (12,907) NCOM
000430         WRITE (12,908) NCL
000436         IF (ILB.EQ.0) WRITE (12,909)
000443         IF (ILB.EQ.1) WRITE (12,910)
000451         IF (ILB.EQ.2) WRITE (12,911) ALB,BLB,WSW,NLI,NLINC
000471         IF (ILB.EQ.3) WRITE (12,912) ALB,BLB,NLI,NLINC
000507         IF(ILB.EQ.4) WRITE(12,930) RXZDV,RXZVS,NLI,NLINC
000525         IF (IRB.EQ.0) WRITE (12,913)
000532         IF (IRB.EQ.1) WRITE (12,914)
000540         IF (IRB.EQ.2) WRITE (12,915) ARB,BRB,WSW,NRI,NRINC
000560         IF (IRB.EQ.3) WRITE (12,916) ARB,BRB,NRI,NRINC
000576         WRITE (12,917)
C      ************************************************************
```

```
C                4020 OUTPUT ABOVE
C     ******************************************************************
C     READ DATA AND SET UP MESH FOR EACH COMPONENT
000602     1 DO 10 I=1,NCOM
C       NAM       NAME OF COMPONENT
000604       READ 900,NAM(1,I),NAM(2,I)
C       NINC      NUMBER OF SPACE INCREMENTS FOR THIS COMPONENT
C       IEXP      NOT 0 FOR GAS OR EXPLOSIVE
C       IBRN      = 0 FOR ARRHENIUS BURN
C                 = 1 FOR CJ VOLUME BURN
C                 = 2 FOR GAMMA LAW TAYLOR WAVE
C       IVIS      = 0 FOR PIC VISCOSITY
C                 = 1 FOR LANDSHOFF VISCOSITY
C                 = 2 FOR REAL VISCOSITY
000617       READ 901,NINC,IEXP(I),IBRN(I),IVIS(I)
C       DX        SPACE INCREMENT IN CENTIMETERS
C        DTIME    TIME INCREMENT IN MICROSECONDS
C       VFACT     VISCOSITY FACTOR
C       RHO0      INITIAL DENSITY
000633       READ 902,DX,DTIME(I),VFACT(I),RHO0
C       P0        INITIAL PRESSURE
C       T0        INITIAL TEMPERATURE
C       E0        INITIAL INTERNAL ENERGY
C       U0        INITIAL VELOCITY
C       SOL(J,I) SOLID EQUATION OF STATE AND ELASTIC-PLASTIC CONSTANTS
000647       READ 902,P0,T0,E0,U0
000663       IF(I.GT.1) GO TO 11
000667       FP0 = P0
000670       FE0 = E0
000671    11 READ 902,(SOL(J,I),J=1,23)
C       W0        INITIAL MASS FRACTION
C       Z         FREQUENCY FACTOR FOR ARRHENIUS BURN
C       E         ACTIVATION ENERGY FOR ARRHENIUS BURN
C       VCJ       VOLUME FOR CJ BURN  OR DET VEL FOR GAMMA LAW TAYLOR WAVE
000705       READ 902,W0,Z(I),E(I),VCJ(I)
000721       PRINT 918,NAM(1,I),NAM(2,I),NINC,DX,DTIME(I)
000743       IF (IEXP(I).EQ.0) GO TO 2
000745       READ 902,(GAS(J,I),J=1,17)
000760       PRINT 919
000764       IF (IBRN(I).EQ.0) PRINT 921,E(I),Z(I)
000775       IF (IBRN(I).EQ.1) PRINT 922,VCJ(I),GASW
001007       IF (IBRN(I).EQ.2) PRINT 929,VCJ(I)
001017     2 IF (IVIS(I).EQ.0) PRINT 923,VFACT(I)
001026       IF (IVIS(I).EQ.1) PRINT 924,VFACT(I)
001036       IF (IVIS(I).EQ.2) PRINT 925,VFACT(I)
001046       PRINT 926,RHO0,P0,T0,E0,U0,W0
001066       PRINT 927,(SOL(J,I),J=1,23)
001102       IF (IEXP(I).NE.0) PRINT 928,(GAS(J,I),J=1,17)
001117       IF (IMC.EQ.0) GO TO 4
C     ******************************************************************
C                4020 OUTPUT BELOW
C     ******************************************************************
001120       WRITE (12,918) NAM(1,I),NAM(2,I),NINC,DX,DTIME(I)
001142       IF (IEXP(I).EQ.0) GO TO 3
001144       WRITE (12,919)
001147       IF (IBRN(I).EQ.0) WRITE (12,921) E(I),Z(I)
001160       IF (IBRN(I).EQ.1) WRITE (12,922) VCJ(I),GASW
001172       IF (IBRN(I).EQ.2) WRITE (12,929) VCJ(I)
001202     3 IF (IVIS(I).EQ.0) WRITE (12,923) VFACT(I)
001211       IF (IVIS(I).EQ.1) WRITE (12,924) VFACT(I)
001221       IF (IVIS(I).EQ.2) WRITE (12,925) VFACT(I)
001231       WRITE (12,926) RHO0,P0,T0,E0,U0,W0
```

```
001251          WRITE (12,927) (SOL(J,I),J=1,23)
001265          IF (IEXP(I).NE.0) WRITE (12,928) (GAS(J,I),J=1,17)
          C     ************************************************************
          C                  4020 OUTPUT ABOVE
          C     ************************************************************
          C     FILL IN CELL VALUES FOR THIS COMPONENT
001302          DO 9 J=1,NINC
001304        4 IF(IBRN(I).EQ.2) GO TO 12
001307          CRO(K+1)=CRO(K)+DX
001312          CR(K+1)=CRO(K+1)
001313          CUO(K)=U0
001315          CU(K)=CUO(K)
001316          CP(K)=P0
001320          CT(K)=T0
001321          CWO(K)=W0
001323          CW(K)=CWO(K)
001324          CSZ(K) = 0.
001325          CS(K) = 0.
001326          CIO(K)=E0
001330          CI(K)=CIO(K)
001331          CM(K)=RHO0*DX*((CRO(K)+CRO(K+1))*.5)**IALPH
001342          CVO(K)=1./RHO0
001344          CV(K)=CVO(K)
001345          ICF(K)=I
001347          CQ(K)=0.
001350          K=K+1
001351        9 CONTINUE
          C     SET SPALL FLAG FOR INTERFACE BETWEEN COMPONENTS
001354       17 IF (IEXP(I).NE.0) GO TO 18
001356          ICF(K-NINC)=ICF(K-NINC)+128
001361       18 NOINC(I) = NINC
001363       10 CONTINUE
          C     SET RIGHT AND LEFT BOUNDARY VALUES
001366          CM(1)=CM(2)
001367          CM(K)=RHO0*DX*(CRO(K)+DX*.5)**IALPH
001400          CQ(K)=0.
001401          KIND=K
001402          CRO(1)=-CRO(3)
001404          CRO(K+2)=CRO(K+1)+DX
001406          CI(1)=0.
001406          CS(1)=0.
001407          CSZ(1)=0.
001410          CV(1)=0.
001411          CV(K)=0.
001412          CS(K)=0.
001413          CSZ(K)=0.
001414          CI(K)=0.
001415          ILB=ILB+1
001416          IRB=IRB+1
001417          TTIME=0.
001420          DELT=0.
          C     FIND MAXIMUN DELTA T FOR FIRST CYCLE
001421          DO 20 I=1,NCOM
001422          IF (DELT.LT.DTIME(I)) DELT=DTIME(I)
001426       20 CONTINUE
          C     SET UP NUMBER OF CELLS IN CALCULATION
001431          NRL=NCL+2
          C     SEE IF THERE IS A LEFT PISTON
001433          IF(ILB.LT.2) GO TO 24
          C     GO TO 22   IF CELL ADDING ON NOT REQUESTED
001434          IF (NLI.EQ.0) GO TO 22
          C     SET NUMBER OF CELLS FROM LEFT AND COUNT FOR ADDING ON CELLS
```

```
001435          NLH=NLI+1
001437          NLCNT=NLINC
                C     SEE IF THERE IS A RIGHT PISTON
001440          IF(IRB.LT.2) GO TO 90
                C     ARE CELLS TO BE ADDED ON FROM RIGHT
001442          IF (NRI.EQ.0) GO TO 90
                C     SET INITIAL NUMBER OF CELLS FROM RIGHT AND COUNT FOR ADDING ON
001443       21 NRL=NCL+1-NRI
001446          NRCNT=NRINC
                C     CHECK TO SEE IF ALL CELLS IN CALCULATION
001447          IF (NRL.GT.NLH+1) GO TO 90
                C     USE ALL CELLS IN CALCULATION
001453       22 NLH=NCL+1
001455          NLI=1
001456          NRI=0
001457          GO TO 90
                C     SEE IF THERE IS A RIGHT PISTON
001460       24 IF(IRB.LT.2) GO TO 22
                C     ARE CELLS TO BE ADDED ON FROM RIGHT
001462          IF (NRI.EQ.0) GO TO 22
001463          NLH = 1
001464          GO TO 21
                C     SET LEFT BOUNDARY VALUES
001465       90 GO TO (100,110,120,130,140),ILB
                C     LEFT BOUNDARY CONTINUUM
001476      100 CP(1)=CP(2)
001500          CR(1)=CR(2)-CR(3)
001502          CQ(1)=CQ(2)
001503          CU(1)=-CU(2)
001505          IF (IALPH.EQ.0) CU(1)=CU(2)
001507          CU0(1)=CU(1)
001511          CT(1)=CT(2)
001512          GO TO 190
                C     LEFT BOUNDARY FREE SURFACE
001513      110 CP(1)=-CP(2)
001515          CR(1)=CR(2)-CR(3)
001517          CQ(1)=CQ(2)
001520          CU(1)=-CU(2)
001522          CU0(1)=CU(1)
001523          CT(1)=CT(2)
001524          GO TO 190
                C     LEFT BOUNDARY PISTON - INITIAL,FINAL
001525      120 CQ(1)=CQ(2)
001527          CR(1)=CR(2)-CR(3)
001531          CP(1)=CP(2)
001532          CT(1)=CT(2)
001534          IF (CW(4).LT.WSW) GO TO 125
001536          CU(1)=ALB
001537          CU0(1)=CU(1)
001540          CU(2)=CU(1)
001541          CU0(2)=CU0(1)
001542          GO TO 190
001543      125 CU(1)=BLB
001545          CU0(1)=CU(1)
001546          CU(2)=CU(1)
001547          CU0(2)=CU0(1)
001550          GO TO 190
                C     LEFT BOUNDARY PISTON - A+BT
001551      130 CQ(1)=CQ(2)
001553          CR(1)=CR(2)-CR(3)
001555          CP(1)=CP(2)
001556          CT(1)=CT(I)
```

```
001560          CU(1)=ALB+BLB*TTIME
001563          CUO(1)=CU(1)
001564          CU(2)=CU(1)
001565          CUO(2)=CUO(1)
          C     SET RIGHT BOUNDARY VALUES
001567        190 GO TO (200,210,220,230),IRB
          C     RIGHT BOUNDARY CONTINUUM
001577        200 CT(K)=CT(K-1)
001601          CP(K)=CP(K-1)
001603          CU(K)=CU(K-1)
001604          CUO(K)=CU(K)
001606          GO TO 290
          C     RIGHT BOUNDARY FREE SURFACE
001606        210 CT(K)=CT(K-1)
001610          CP(K)=-CP(K-1)
001612          CU(K)=CU(K-1)
001613          CUO(K)=CU(K)
001615          GO TO 290
          C     RIGHT BOUNDARY PISTON - INITIAL,FINAL
001615        220 CT(K)=CT(K-1)
001617          CP(K)=CP(K-1)
001621          IF (CW(K-3).LT.WSW) GO TO 225
001623          CU(K)=ARB
001624          CUO(K)=ARB
001625          GO TO 290
001626        225 CU(K)=BRB
001630          CUO(K)=BRB
001631          GO TO 290
          C     RIGHT BOUNDARY PISTON - A+BT
001632        230 CT(K)=CT(K-1)
001634          CP(K)=CP(K-1)
001636          CU(K)=ARB+BRB*TTIME
001641          CUO(K)=CU(K)
001643        290 CONTINUE
001643          IF (NLI.NE.0) CALL VELOC (2,NLH)
001646          IF (NRI.NE.0) CALL VELOC (NRL,NCL+1)
001653          IF (NLI.NE.0) CALL RADIUS (2,NLH+1)
001660          IF (NRI.NE.0) CALL RADIUS (NRL,NCL+2)
          C     SET RADIUS FOR CELL BEYOND RIGHT BOUNDARY
001665          CR(K+2)=CR(K+1)+DX
001670          IF (NLI.NE.0) CALL VOLUM (2,NLH)
001673          IF (NRI.NE.0) CALL VOLUM (NRL,NCL+1)
001700          IF (NLI.NE.0) CALL ENERGY (2,NLH)
001703          IF (NRI.NE.0) CALL ENERGY (NRL,NCL+1)
001710          IF (IHC.EQ.0) GO TO 300
001711          IF (NLI.NE.0) CALL HETCON (2,NLH)
001714          IF (NRI.NE.0) CALL HETCON (NRL,NCL+1)
001721        300 IF (NLI.NE.0) CALL VISCOS (2,NLH)
001724          IF (NRI.NE.0) CALL VISCOS (NRL,NCL+1)
001731          IF (NLI.NE.0) CALL BURN (2,NLH)
001734          IF (NRI.NE.0) CALL BURN (NRL,NCL+1)
          C     SET W NEAR AXIS
001741          CW(2)=CW(4)
001743          CW(3)=CW(4)
001744          IF (NLI.NE.0) CALL STRESS (2,NLH)
001747          IF (NRI.NE.0) CALL STRESS (NRL,NCL+1)
001754          IF (NLI.NE.0) CALL EQST (2,NLH)
001757          IF (NRI.NE.0) CALL EQST (NRL,NCL+1)
001764          IF (NLI.NE.0) CALL UPDATE (1,NLH+2)
001771          IF (NRI.NE.0) CALL UPDATE (NRL,NCL+3)
001776          CALL TIME
001777          GO TO 90
```

```
      C     RESTART FROM TAPE
002000      500 IRST=NCOM
      C     SEARCH FOR DUMP WITH PROPER CYCLE NUMBER
002002      501 READ (1) CRO,CUO,CWO,CSZ,CIO,CVO,ICF,CM,CQ,CT,CP,LABEL,DELT,HEATC,
             1IALPH,TTIME,PINC,GINC,DINC,PCNT,GCNT,OCNT,E,GAS,IBRN,SOL,VCJ,
             2VFACT,Z,NAM,DTIME,NOINC,NCOM,ICYCL,NCL,IPNT,IMC,IPR,ITR,IWR,IIR,
             3IUR,IHC,ILB,IRB,ALB,BLB,ARB,BRB,WSW,NLI,NRI,NLINC,NRINC,NLCNT,
             4NRCNT,NLH,NRL,DX,IVIS,IEXP,KIND,CR,CI,CW,CV,CU,CS,RXZDV,RXZVS,
             5RXZ,FPO,FEO
002227        K=KIND
002231        IF (IRST.GT.ICYCL) GO TO 501
002234        GO TO 90
      C     THE SLAB GAMMA LAW TAYLOR WAVE  ** ASSUMES DETONATION WAVE MOVED
      C     FROM RIGHT TO LEFT AND HAS BRB FINAL PARTICLE VELOCITY
002234       12 L = K
002236          DO 13 J=1,NINC
002237          CR(L+1) = CR(L) + DX
002242          CT(L)=T0
002244          CWO(L)=0.
002245          CW(L)=0.
002246          CSZ(L)=0.
002247          CS(L)=0.
002250          CM(L)=RHO0*DX
002252          CQ(L)=0.
002253          ICF(L)=I
002254          L=L+1
002256       13 CONTINUE
002260          DCJ=VCJ(I)
002261          DRT=CR(L)-CR(K)
002264          DRF=CR(K)
002266          DRL=CR(L)
002267          UCJ=DCJ/(-GAS(2,I)+1.)
002274          PCJ=(RHO0)*(DCJ*DCJ)/(-GAS(2,I)+1.)
002300          VCJR=(-GAS(2,I))/((-GAS(2,I)+1.)*RHO0)
002305          CCJ=DCJ-UCJ
002307          COK=(-GAS(2,I)-1.)/(-GAS(2,I)+1.)
002313          COL=UCJ-((2./(-GAS(2,I)-1.))*CCJ)
002320          COT=DRT/DCJ
002322          DO 14 J=1,NINC
002324          COY=(DRL-CR(K))/COT
002330          CU(K)=(2.*COY)/(-GAS(2,I)+1.)+COK*COL
002341          CP(K)=PCJ*((((-GAS(2,I)-1.)*0.5)*(CU(K)-COL))/CCJ)**(-GAS(2,I)*2./
             1(-GAS(2,I)-1.))
002361          CV(K)=(PCJ*(VCJR**(-GAS(2,I))))/CP(K))**(1./(-GAS(2,I)))
002376          CVO(K)=CV(K)
002400          CI(K)= (CP(K)*CV(K))/(-GAS(2,I)-1.)-(PCJ*VCJR)/(-GAS(2,I)-1.)
             1+ 0.5*PCJ*(1./RHO0-VCJR)
002417          CIO(K)=CI(K)
002421          DRP=CM(K)*CV(K)
002423          CR(K+1)=CR(K)+DRP
002425          CRO(K+1)=CR(K+1)
002426          CU(K)=-CU(K)
002430          CUO(K)=CU(K)
002431          IF(CU(K).GT.BRB) GO TO 15
002435          K=K+1
002436       14 CONTINUE
002440          GO TO 17
002441       15 L=K
002443          DO 16  N=J,NINC
002445          CU(K)=CU(L)
002450          CUO(K)=CU(L)
002451          CP(K)=CP(L)
```

```
002453          CV(K)=CV(L)
002454          CVO(K)=CV(L)
002456          CI(K)=CI(L)
002457          CIO(K)=CI(L)
002461          CR(K+1)=CR(K)+DRP
002463          CRO(K+1)=CR(K+1)
002465          K=K+1
002466       16 CONTINUE
002470          GO TO 17
                C      REACTION ZONE PISTON
                C      FOR SLABS AT LEFT BOUNDARY ONLY
002471          DATA RXZ(1)/ 0./
002471          DATA RXZ(2) /1.001/
002471          DATA RXZ(3) /+1.0E-5/
002471          DATA RXZ(10)/ 0./
002471       140 IF(RXZ.EQ.0.) RXZ=RXZVS
002473          IF(CW(4).LT.GASW) GO TO 120
002475       146 CALL LFB (RZV,RZF,RXZ)
002500          IF(RXZ(10)) 141,142,143
002502       141 PRINT 931,RXZ
002510          PCNT=1.
002512          GCNT=100.
002513          ICYCL=ICYCL-1
002515          TTIME=TTIME-DELT
002517          CALL TIME
002520          STOP
002522       931 FORMAT (33H1 REACTION ZONE PISTON LFB ERROR , 5E18.11)
002522       143 RZI=FEO    + 0.5/( SOL(13,I)*SOL(13,I))*RXZDV*RXZDV*((SOL(13,I)-
                   1RZV)**2)
002535          EQ(1)=RZV
002536          EQ(2)=RZI
002537       160 EQ(3)=CW(4)
002541          CALL HOM (EQ,SOL(1,I),GAS(1,I),INDH)
002550          IF(INDH) 144,145,144
002551       144 PRINT 932,I,EQ
002561       932 FORMAT (33H1  HOM LFB ERROR FOR RXZ PISTON  ,I5,5E18.11)
002561          GO TO 141
002562       145 RXP=1./(SOL(13,I)*SOL(13,I))*RXZDV*RXZDV*(SOL(13,I)-RZV)
002571          RZF=EQ(4)-RXP+ FPO
002574          GO TO 146
002575       142 ALB=SQRT((EQ(4) -FPO)*(SOL(13,I)-RZV))
002606          BLB=ALB
002607          GO TO 120
002607       900 FORMAT (12A6)
002607       901 FORMAT (12I5)
002607       902 FORMAT (4E18.11)
002607       903 FORMAT (2E18.11,2I5)
002607       904 FORMAT (80H1  A SIN ONE DIMENSIONAL REACTIVE HYDRODYNAMIC CALCULAT
                   1ION IN SLAB GEOMETRY FOR ,/,2X,12A6)
002607       905 FORMAT (87H1  A SIN ONE DIMENSIONAL REACTIVE HYDRODYNAMIC CALCULAT
                   1ION IN CYLINDRICAL GEOMETRY FOR ,/,2X,12A6)
002607       906 FORMAT (85H1  A SIN ONE DIMENSIONAL REACTIVE HYDRODYNAMIC CALCULAT
                   1ION IN SPHERICAL GEOMETRY FOR ,/,2X,12A6)
002607       907 FORMAT (//,32H   THE NUMBER OF COMPONENTS IS  ,I5)
002607       908 FORMAT (//,44H   THE TOTAL NUMBER OF SPACE INCREMENTS IS  ,I5)
002607       909 FORMAT (//,35H   THE LEFT BOUNDARY IS A CONTINUUM)
002607       910 FORMAT (//,38H   THE LEFT BOUNDARY IS A FREE SURFACE)
002607       911 FORMAT (//,59H   THE LEFT BOUNDARY IS A PISTON WITH INITIAL VELOCI
                   1TY OF  ,E10.3,24H AND FINAL VELOCITY OF  ,E10.3,/,10X,23HWHEN W+3
                   2IS LESS THAN  ,E10.3, 6H WITH ,I5.44H CELLS INITIALLY AND ADDING O
                   3N A CELL EVERY ,I5, 7H CYCLES)
002607       912 FORMAT (//,49H   THE LEFT BOUNDARY IS A PISTON WITH VELOCITY = ,
```

```
     1E10.3,1H+,E10.3,6H(TIME),/,10X, 5HWITH ,I5,44H CELLS INITIALLY AND
     2 ADDING ON A CELL EVERY ,I5, 7H CYCLES)
002607     913 FORMAT (//,36H    THE RIGHT BOUNDARY IS A CONTINUUM)
002607     914 FORMAT (//,39H    THE RIGHT BOUNDARY IS A FREE SURFACE)
002607     915 FORMAT (//,60H    THE RIGHT BOUNDARY IS A PISTON WITH INITIAL VELOC
     1ITY OF  ,1PE10.3,24H AND FINAL VELOCITY OF  ,1PE10.3,/,10X,23HWHEN
     2 W+3 IS LESS THAN  ,1PE10.3, 6H WITH ,I5,44H CELLS INITIALLY AND A
     3DDING ON A CELL EVERY ,I5, 7H CYCLES)
002607     916 FORMAT (//,50H    THE RIGHT BOUNDARY IS A PISTON WITH VELOCITY = ,
     11PE10.3,1H+,1PE10.3,6H(TIME),/,10X, 5HWITH ,I5,44H CELLS INITIALLY
     2 AND ADDING ON A CELL EVERY ,I5, 7H CYCLES)
002607     917 FORMAT (////,30X,33HTHE PARAMETERS FOR EACH COMPONENT)
002607     918 FORMAT (///,21H * THE COMPONENT IS  ,2A6,//,11H    AND HAS ,I5,
     122H SPACE INCREMENTS OF  ,1PE10.3,34H CM EACH AND A TIME INCREMENT
     2 OF  ,1PE10.3,13H MICROSECONDS)
002607     919 FORMAT (//,80H    IT IS TO BE CONSIDERED AS AN EXPLOSIVE AND WILL B
     1E BURNED BY THE TECHNIQUE OF)
002607     921 FORMAT (//,53H    THE ARRHENIUS RATE LAW WITH ACTIVATION ENERGY OF
     1 ,1PE10.3,46H CALORIES PER MOLE AND A FREQUENCY FACTOR OF  ,/,3X,
     21PE10.3,16H PER MICROSECOND)
002607     922 FORMAT (//,45H    THE CJ VOLUME BURN WITH A BURN VOLUME OF  ,1PE10.
     13,17H AND A MAX W OF  ,1PE10.3)
002607     923 FORMAT (//,68H    THE ARTIFICIAL PIC TYPE OF VISCOSITY IS USED WITH
     1 A CONSTANT OF  ,1PE10.3)
002607     924 FORMAT (//,74H    THE ARTIFICIAL LANDSHOFF TYPE OF VISCOSITY IS USE
     1D WITH A CONSTANT OF  ,1PE10.3)
002607     925 FORMAT (//,76H    THE TRUE FORM OF VISCOSITY IS USED WITH THE COEFF
     1ICIENT OF VISCOSITY OF  ,1PE10.3)
002607     926 FORMAT (//,43H    THE INITIAL CONDITIONS ARE A DENSITY OF ,1PE10.3,
     123H GRAMS/CC, PRESSURE OF ,1PE10.3,26H MEGABARS, TEMPERATURE OF ,/
     2,2X,1PE10.3,37H DEGREES KELVIN, INTERNAL ENERGY OF  ,1PE10.3,33H M
     3B-CC/GM, PARTICLE VELOCITY OF  ,1PE10.3,16H CM/MICROSECOND,,/,37H
     4  AND MASS FRACTION OF THE SOLID OF  ,1PE10.3)
002607     927 FORMAT (//129H    THE HOM EQUATION OF STATE PARAMETERS FOR THE SOLI
     1D C,S,VS,C1,S1,FS,GS,HS,IS,JS,GG,CV,V0,ALPHA,SPA,USP,T0,P0,Y0,MU,P
     2LAP,SPP,VS2,/,(6(2X,1PE18.11)))
002607     928 FORMAT (//108H    THE HOM EQUATION OF STATE PARAMETERS FOR THE DETO
     1NATION PRODUCTS   A,B,C,D,E, K,L,M,N,O, Q,R,S,T,U,CVG,Z,/,(6(2X,
     21PE18.11)))
002607     929 FORMAT(//,57H    A GAMMA LAW TAYLOR WAVE WITH DETONATION VELOCITY
     1OF  ,1PE10.3)
002607     930 FORMAT (//,77H    THE LEFT BOUNDARY IS A STEADY STATE PISTON WITH A
     1 DETONATION VELOCITY OF  ,1PE10.3,17H SPIKE VOLUME OF ,1PE10.3,/,
     26H WITH ,I5,44H CELLS INITIALLY AND ADDING ON A CELL EVERY ,I5,
     3 7H CYCLES)
002607          END
```

```
                SUBROUTINE BURN (NMIN,NMAX)
          C
          C     BURN PERFORMS THE CHEMICAL REACTION ON CELLS WITH INDICES
          C     NMIN TO NMAX
          C
000004          COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
               1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
               2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000004          COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
               1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,
               2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
               3NLCNT,NRCNT,NLH,NRL,DX,KIND,RXZDV,RXZVS,FPO,FEO,RXZ
000004          COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SOL(23,10),VCJ(10),
               1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
               2NCOM
000004          DATA WMAX /.99/
          C     SET W TO 0 IF LESS THAN GASW
000004          DATA GASW /*0.02/
000004          DO 10 I=NMIN,NMAX
          C     SKIP IF ALL BURNED
000005          IF (CW(I).EQ.0.) GO TO 10
          C     J IS INDEX OF COMPONENT
000007          KSP=ICF(I)/64
000012          J=ICF(I)-64*KSP
          C     CHECK FOR EXPLOSIVE OR GAS
000016          IF (IEXP(J).EQ.0) GO TO 10
          C     DETERMINE TYPE OF BURN
000020          IF (IBRN(J).NE.0) GO TO 1
          C     ARRHENIUS BURN
          C     IF E,Z, OR T LESS THAN 0.0001 DO NOT BURN
000021          IF(E(J).LT.0.0001) GO TO 2
000024          IF(Z(J).LT.0.0001) GO TO 2
000026          IF(CT(I).LT.0.0001) GO TO 2
000031          CW(I)=CWO(I)*(1.-DELT*Z(J)*EXP(-E(J)/(1.9865*CT(I))))
000050          GO TO 2
          C     CJ VOLUME BURN
000050        1 CW(I)=1.-(SOL(13,J)-CV(I))/(SOL(13,J)-VCJ(J))
000064          IF (CW(I).GT.CWO(I)) CW(I)=CWO(I)
000072        2 IF (CW(I).LT.GASW) CW(I)=0.
000077       10 CONTINUE
000102          RETURN
000102          END
```

```
                SUBROUTINE ENERGY (NMIN,NMAX)
              C
              C     ENERGY COMPUTES THE NEW INTERNAL ENERGY FOR CELLS WITH INDICES
              C     NMIN TO NMAX
              C
000004          COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
                1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
                2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000004          COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
                1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,
                2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
                3NLCNT,NRCNT,NLH,NRL,DX,KIND,RXZDV,RXZVS,FPO,FEO,RXZ
000004          COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SOL(23,10),VCJ(10),
                1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
                2NCOM
000004          IF (IALPH.EQ.0) GO TO 15
000005          DO 10 I=NMIN,NMAX
000006          CI(I)=CIO(I)+DELT/CM(I)*(((CM(I)*CP(I-1)+CM(I-1)*CP(I))/
                1(CM(I)+CM(I-1))+.5*(CQ(I)+CQ(I-1)))*CU(I)*CR(I)**IALPH-((CM(I+1)*
                2CP(I)+CM(I)*CP(I+1))/(CM(I)+CM(I+1))+.5*(CQ(I)+CQ(I+1)))*CU(I+1)
                3*CR(I+1)**IALPH)+((CUO(I+1)+CUO(I))**2-(CU(I+1)+CU(I))**2)*.125
000077       10 CONTINUE
000102          RETURN
000102       15 DO 20 I=NMIN,NMAX
000104          CI(I)=CIO(I)+DELT/CM(I)*(((CM(I)*CP(I-1)+CM(I-1)*CP(I))/
                1(CM(I)+CM(I-1))+.5*(CQ(I)+CQ(I-1)))*CU(I)-((CM(I+1)*
                2CP(I)+CM(I)*CP(I+1))/(CM(I)+CM(I+1))+.5*(CQ(I)+CQ(I+1)))*CU(I+1))
                3+((CUO(I+1)+CUO(I))**2-(CU(I+1)+CU(I))**2)*.125
000162       20 CONTINUE
000164          RETURN
000164          END
```

```
                SUBROUTINE EQST (NMIN,NMAX)
          C
          C     EQST DOES THE ELASTIC PLASTIC SPALLING EQUATION OF STATE
          C     CALCULATION FOR CELLS WITH INDICES NMIN TO NMAX
          C
000004            COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
                 1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
                 2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000004            COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
                 1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,
                 2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
                 3NLCNT,NRCNT,NLH,NRL,DX,KIND,RXZDV,RXZVS,FPO,FEO,RXZ
000004            COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SOL(23,10),VCJ(10),
                 1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
                 2NCOM
000004            DIMENSION EQ(5)
          C     PSMIN - MINIMUN ELASTIC-PLASTIC PRESSURE
000004            DATA PSMIN /1.E-4/
000004            RFRNT=0.
000005            DO 20 I=NMIN,NMAX
          C     KSP = 0 FOR NON SPALLED CELL
          C         = 1 FOR SPALLED CELL
          C         = 2 FOR INTERFACE CELL
000006            KSP=ICF(I)/64
          C     J IS INDEX OF COMPONENT
000012            J=ICF(I)-64*KSP
000015            IF (KSP.EQ.2) RFRNT=CR(I)
000021            IF (CV(I).LT.SOL(13,J)) GO TO 1
000027            IF (KSP.NE.0) GO TO 11
000030          1 EQ(1)=CV(I)
000033            EQ(2)=CI(I)
          C     CHECK FOR CJ EQUATION OF STATE
000035            IF (IBRN(J).NE.0) GO TO 9
000037            EQ(3)=CW(I)
000041            EQ(4)=0.
000042            IF (CV(I).GT.SOL(13,J)) EQ(4)=(CP(I))/ABS(CR(I)-RFRNT)
000056            CALL HOM (EQ,SOL(1,J),GAS(1,J),INDH)
000072            IF (INDH) 2,4,3
          C     ERROR IN HOM
000073          2 PRINT 900,I,EQ
          C     MAKE FAKE CALL TO TIME TO GET A LAST PRINT
000105            PCNT=1.
000106            GCNT=100.
000107            DCNT=100.
000110            ICYCL=ICYCL-1
000111            TTIME=TTIME-DELT
000113            CALL TIME
000117            STOP
000121        900 FORMAT (16H1   HOM ERROR   ,I5,5E18.11)
          C     CELL SPALLED
000121          3 IF (ICF(I).LT.64) ICF(I)=ICF(I)+64
000127            RFRNT=CR(I)
000132            CP(I)=EQ(4)
000134            GO TO 11
          C     REGULAR HOM EXIT
000134          4 CP(I)=EQ(4)
000137            IF (ABS(EQ(4)).LT.SOL(18,J)) CP(I)=SOL(18,J)
000151            CT(I)=EQ(5)
          C     IF AN EXPLOSIVE OR GAS, SKIP ELASTIC-PLASTIC
000154            IF (IEXP(J).NE.0) GO TO 20
          C     IF YO OR MU LESS THAN 0.0001 SKIP ELASTIC CALCULATION
```

```
                SUBROUTINE HETCON (NMIN,NMAX)
              C
              C     HETCON COMPUTES THE ADDITION TO THE INTERNAL ENERGY DUE TO
              C     HEAT CONDUCTION FOR CELLS WITH INDICES NMIN TO NMAX
              C
000004                COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
                     1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
                     2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000004                COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
                     1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,
                     2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
                     3NLCNT,NRCNT,NLH,NRL,DX,KIND,RXZDV,RXZVS,FPO,FEO,RXZ
000004                COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SOL(23,10),VCJ(10),
                     1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
                     2NCOM
000004                IF (IALPH.EQ.0) GO TO 15
000005                DO 10 I=NMIN,NMAX
000006           10 CI(I)=CI(I)+2.*HEATC*DELT/CM(I)*((CR(I+1)**IALPH)*(CT(I+1)-CT(I))
                    1/(CR(I+2)-CR(I))-(CR(I)**IALPH)*(CT(I)-CT(I-1))/(CR(I+1)-CR(I-1)))
000047                RETURN
000047           15 DO 20 I=NMIN,NMAX
000051           20 CI(I)=CI(I)+2.*HEATC*DELT/CM(I)*((CT(I+1)-CT(I))
                    1/(CR(I+2)-CR(I))-(CT(I)-CT(I-1))/(CR(I+1)-CR(I-1)))
000077                RETURN
000077                END
```

```
                SUBROUTINE RADIUS (NMIN,NMAX)
          C
          C     RADIUS COMPUTES THE CHANGES IN THE R COORDINATE IN A TIME STEP
          C     FOR CELLS WITH INDICES NMIN TO NMAX
          C
000004            COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
                 1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
                 2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000004            COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
                 1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,
                 2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
                 3NLCNT,NRCNT,NLH,NRL,DX,KIND,RXZDV,RXZVS,FPO,FEO,RXZ
000004            COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SOL(23,10),VCJ(10),
                 1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
                 2NCOM
000004            DO 10 I=NMIN,NMAX
000005          10 CR(I)=CRO(I)+CU(I)*DELT
000014            RETURN
000014            END
```

```
                SUBROUTINE STRESS (NMIN,NMAX)
          C
          C     STRESS CALCULATES THE CHANGE IN THE STRESS DEVIATOR FOR CELLS
          C     WITH INDICES NMIN TO NMAX
          C
000004          COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
                1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
                2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000004          COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
                1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,
                2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
                3NLCNT,NRCNT,NLH,NRL,DX,KIND
000004          COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SOL(23,10),VCJ(10),
                1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
                2NCOM
          C     CHANGE IN S MUST BE GREATER THAN SMIN BEFORE AFFECTING ANYTHING
000004          DATA SMIN /1.E-9/
000004          DO 10 I=NMIN,NMAX
000005          KSP=ICF(I)/64
          C     J IS INDEX OF COMPONENT
000011          J=ICF(I)-64*KSP
          C     IF YO OR MU LESS THAN 0.0001 SKIP ELASTIC CALCULATION
000014          IF(SOL(19,J).LT.0.0001)  GO TO 10
000021          IF(SOL(20,J).LT.0.0001)  GO TO 10
000024          DELS=2.*SOL(20,J)*((CU(I)-CU(I+1))*DELT/(CR(I+1)-CR(I))
                1+.6666666667*(CV(I)-CVO(I))/(CV(I)+CVO(I)))
000047          IF (ABS(DELS).LT.SMIN) DELS=0.
000053          CS(I)=CS(I) + DELS
000057          IF (IALPH.EQ.1) GO TO 20
000061          GO TO 10
          C     CALCULATE CSZ
000061       20 DELS = SOL(20,J)*1.33333334*(CV(I)-CVO(I))/(CV(I)+CVO(I))
000074          IF(ABS(DELS).LT.SMIN) DELS = 0.
000100          CSZ(I) = CSZ(I) + DELS
000104       10 CONTINUE
000107          RETURN
000107          END
```

```
                SUBROUTINE TIME
          C
          C     TIME INCREMENTS THE TIME FOR EACH CYCLE AND DOES THE
          C     PRINTING AND GRAPHING WHEN NECESSARY
          C
000002          COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
               1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
               2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000002          COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
               1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,
               2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
               3NLCNT,NRCNT,NLH,NRL,DX,KINU,RXZDV,RXZVS,FPO,FEO,RXZ
000002          COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SOL(23,10),VCJ(10),
               1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
               2NCOM
000002          DIMENSION ICHAR (2),TKE(10),TIE(10)
          C     PLOT CHARACTERS        1.  .       2.  X
000002          DATA ICHAR(1) /052/
000002          DATA ICHAR(2) /067/
          C     THE PRESSURE OF ONE CELL OF A COMPONENT MUST BE GREATER THAN
          C     PTMIN TO USE THAT COMPONENTS DELTA T
000002          DATA PTMIN /1.E-05/
000002          DATA ICYCL /0/
000002          DATA PCNT /1./
000002          DATA GCNT /1./
000002          DATA DCNT /100./
          C     INCREMENT TIME AND CYCLE NUMBER
000002          ICYCL=ICYCL+1
000004          TTIME=TTIME+DELT
000006          NRCNT=NRCNT-1
000007          NLCNT=NLCNT-1
          C     SEE IF WE NEED TO ADD ON CELLS FROM LEFT
000010          IF (NLCNT.GT.0) GO TO 4
          C     YES    ADD ON A CELL AND SET COUNT
000012          NLH=NLH+1
000013          NLCNT=NLINC
          C     SEE IF WE NEED TO ADD ON CELLS FROM RIGHT
000014        4 IF (NRCNT.GT.0) GO TO 5
          C     YES    ADD ON A CELL AND SET COUNT
000017          NRL=NRL-1
000020          NRCNT=NRINC
          C     CHECK IF ALL CELLS ARE IN THE CALCULATION
000021        5 IF (NRL.GT.NLH+1) GO TO 6
          C     YES   SET NLI,NRI,NLH, AND COUNTS ACCORDINGLY
000026          NLH=NCL+1
000027          NLI=1
000030          NRI=0
000031          NLCNT=100000
000032          NRCNT=100000
000033          GO TO 7
          C     CHECK IF ALL CELLS ARE IN THE CALCULATION
000033        6 IF (NLH.LT.NCL) GO TO 7
          C     YES   SET NLI,NRI,NLH, AND COUNTS ACCORDINGLY
000035          NRCNT=100000
000037          NLCNT=100000
000040          NLH=NCL+1
000041          NRI=0
000042          NLI=1
          C     DETERMINE DELT FOR NEXT CYCLE
000043        7 K=1
          C     SAVE DELTA T FOR PRINT OUT
```

```
000044          SVDEL=DELT
000046          DO 10 I=1,NCOM
000047          N=NOINC(I)
000051          DO 8 J=1,N
000053          IF (CP(K+J).GT.PTMIN) GO TO 9
000060        8 CONTINUE
000062          GO TO 10
000062        9 IF (DELT.GT.DTIME(I)) DELT=DTIME(I)
000067       10 K=K+N
                C     DECREMENT COUNTS
000073          PCNT=PCNT-1.
000075          GCNT=GCNT-1.
000077          DCNT=DCNT-1.
000100          IF (PCNT.GT.0.) GO TO 20
000102          PCNT=PINC
000103          PRINT 900,ICYCL,TTIME,LABEL
000114          PRINT 906,SVDEL,NLH,NRL
000126          LIN=56
000127          K=1
000130          DO 15 I=1,NCOM
000132          N=NOINC(I)
000134          TKE(I)=0.
000136          TIE(I)=0.
000137          L=K
000141          DO 11 J=1,N
000142          TKE(I)=TKE(I)+CM(L)*CU(L+1)*CU(L+1)
000150          TIE(I)=TIE(I)+CM(L)*CI(L)
000154          L=L+1
000155       11 CONTINUE
000160          TKE(I)=TKE(I)*.50036186
000163          IF (LIN.LT.6) GO TO 12
000165          PRINT 901 ,NAM(1,I),NAM(2,I),TKE(I),TIE(I)
000206          LIN=LIN-5
000210          GO TO 13
000210       12 PRINT 902,NAM(1,I),NAM(2,I),TKE(I),TIE(I)
000232          PRINT 903
000236          LIN=55
000237       13 IF (I.EQ.1) PRINT 903
000245          IF (I.EQ.1) N=N+1
000250          IF (I.EQ.NCOM) N=N+1
000253          EINO=-1.
000255          DO 14 J=1,N
000256          IF (CI(K).EQ.EINO) GO TO 14
000261          IFLG=ICF(K)/64
000264          IF (LIN.EQ.0) PRINT 904
000271          IF (LIN.EQ.0) LIN=58
000273          PRINT 905,K,CR(K),CU(K),CV(K),CI(K),CP(K),CT(K),CW(K),CQ(K),CM(K),
                1CS(K),CSZ(K),IFLG
000344          LIN=LIN-1
000346          EINO=CI(K)
000350       14 K=K+1
000354       15 CONTINUE
000357          IF (IMC.EQ.0) GO TO 20
                C     ************************************************************
                C               4020 OUTPUT BELOW
                C     ************************************************************
000360          WRITE (12,900) ICYCL,TTIME,LABEL
000371          WRITE (12,906) SVDEL,NLH,NRL
000403          LIN=56
000404          K=1
000405          DO 19 I=1,NCOM
000407          EINO=-1.
```

```
000411          N=NOINC(I)
000413          IF (LIN.LT.6) GO TO 16
000415          WRITE (12,901) NAM(1,I),NAM(2,I),TKE(I),TIE(I)
000436          LIN=LIN-5
000440          GO TO 17
000440       16 WRITE (12,902) NAM(1,I),NAM(2,I),TKE(I),TIE(I)
000462          WRITE (12,903)
000466          LIN=55
000467       17 IF (I.EQ.1) WRITE (12,903)
000475          IF (I.EQ.1) N=N+1
000500          IF (I.EQ.NCOM) N=N+1
000503          DO 18 J=1,N
000505          IF (CI(K).EQ.EINO) GO TO 18
000510          IFLG=ICF(K)/64
000513          IF (LIN.EQ.0) WRITE (12,904)
000520          IF (LIN.EQ.0) LIN=58
000522          WRITE (12,905) K,CR(K),CU(K),CV(K),CI(K),CP(K),CT(K),CW(K),CQ(K),
               1CM(K),CS(K),CSZ(K),IFLG
000573          LIN=LIN-1
000575          EINO=CI(K)
000577       18 K=K+1
000603       19 CONTINUE
            C     ********************************************************************
            C                  4020 OUTPUT ABOVE
            C     ********************************************************************
000606       20 IF (GCNT.GT.0.) GO TO 50
000611          GCNT=GINC
000612          IF (IMC.EQ.0) GO TO 50
            C     ********************************************************************
            C                  4020 OUTPUT BELOW
            C     ********************************************************************
000613          IF (IPR.EQ.0) GO TO 32
            C     GRAPH PRESSURE VS RADIUS
000614          CALL ADV(1)
000615          CALL DGA (123,1023,0,900.0.,CR(NCL+1),.90,-.1)
000625          CALL DLNLN (10,10)
000627          CALL SLLIN (10,2)
000631          CALL SBLIN (10,3)
000633          J=1
000634          K=2
000635          DO 31 I=1,NCOM
000637          N=NOINC(I)
000641          CALL PLOT (N,CR(K),1,CP(K),1,ICHAR(J),1)
000652          J=3-J
000654          K=K+N
000655       31 CONTINUE
000660          CALL LINCNT (60)
000661          WRITE (12,910) LABEL
000667          WRITE (12,911) TTIME,ICYCL
000677       32 IF (ITR.EQ.0) GO TO 34
            C     GRAPH TEMPERATURE VS RADIUS
000700          CALL ADV(1)
000702          CALL DGA (123,1023,0,900.0.,CR(NCL+1),5000.,0.)
000712          CALL DLNLN (10,10)
000714          CALL SLLIN (10,0)
000716          CALL SBLIN (10,3)
000720          J=1
000721          K=2
000722          DO 33 I=1,NCOM
000724          N=NOINC(I)
000726          CALL PLOT (N,CR(K),1,CT(K),1,ICHAR(J),1)
000737          J=3-J
```

```
000741          K=K+N
000742       33 CONTINUE
000745          CALL LINCNT (60)
000746          WRITE (12,910) LABEL
000754          WRITE (12,912) TTIME,ICYCL
000764       34 IF (IWR.EQ.0) GO TO 36
             C     GRAPH MASS FRACTION VS RADIUS
000765          CALL ADV(1)
000767          CALL DGA (123,1023,0,900,0.,CR(NCL+1),1.,0.)
000777          CALL DLNLN (10,10)
001001          CALL SLLIN (10,2)
001003          CALL SBLIN (10,3)
001005          J=1
001006          K=2
001007          DO 35 I=1,NCOM
001011          N=NOINC(I)
001013          CALL PLOT (N,CR(K),1,CW(K),1,ICHAR(J),1)
001024          J=3-J
001026          K=K+N
001027       35 CONTINUE
001032          CALL LINCNT (60)
001033          WRITE (12,910) LABEL
001041          WRITE (12,913) TTIME,ICYCL
001051       36 IF (IIR.EQ.0) GO TO 38
             C     GRAPH VOLUME VS RADIUS
001052          CALL ADV(1)
001054          CALL DGA (123,1023,0,900,0.,CR(NCL+1),2.,0.)
001064          CALL DLNLN (10,10)
001066          CALL SLLIN (10,3)
001070          CALL SBLIN (10,3)
001072          J=1
001073          K=2
001074          DO 37 I=1,NCOM
001076          N=NOINC(I)
001100          CALL PLOT (N,CR(K),1,CV(K),1,ICHAR(J),1)
001111          J=3-J
001113          K=K+N
001114       37 CONTINUE
001117          CALL LINCNT (60)
001120          WRITE (12,910) LABEL
001126          WRITE (12,914) TTIME,ICYCL
001136       38 IF (IUR.EQ.0) GO TO 50
             C     GRAPH VELOCITY VS RADIUS
001137          CALL ADV(1)
001141          CALL DGA (123,1023,0,900,0.,CR(NCL+1),1.0,-1.0)
001151          CALL DLNLN (10,10)
001153          CALL SLLIN (10,2)
001155          CALL SBLIN (10,3)
001157          J=1
001160          K=2
001161          DO 39 I=1,NCOM
001163          N=NOINC(I)
001165          CALL PLOT (N,CR(K),1,CU(K),1,ICHAR(J),1)
001176          J=3-J
001200          K=K+N
001201       39 CONTINUE
001204          CALL LINCNT (60)
001205          WRITE (12,910) LABEL
001213          WRITE (12,915) TTIME,ICYCL
             C     ************************************************************
             C               4020 OUTPUT ABOVE
             C     ************************************************************
```

```
001223       50 IF (DCNT.GT.0.) RETURN
001225          DCNT=DINC
001227          WRITE (1)CRO,CUO,CWO,CSZ,CIO,CVO,ICF,CM,CQ,CT,CP,LABEL,DELT,HEATC,
                1IALPH,TTIME,PINC,GINC,DINC,PCNT,GCNT,DCNT,E,GAS,IBRN,SOL,VCJ,
                2VFACT,Z,NAM,DTIME,NOINC,NCOM,ICYCL,NCL,IPNT,IMC,IPR,ITR,IWR,IIR,
                3IUR,IHC,ILB,IRB,ALB,BLB,ARB,BRB,WSW,NLI,NHI,NLINC,NRINC,NLCNT,
                4NRCNT,NLH,NRL,DX,IVIS,IEXP,KIND,CR,CI,CW,CV,CU,CS,RXZDV,RXZVS,
                5RXZ,FPO,FEO
001454          PRINT 916,ICYCL
001462          RETURN
001463      900 FORMAT (12H1  AT CYCLE ,I5,13H THE TIME IS ,1PE11.4,18H MICROSECON
                1DS FOR ,12A6)
001463      901 FORMAT (//,20H   THE COMPONENT IS ,2A6,26H WITH A KINETIC ENERGY O
                1F ,1PE11.4,27H AND AN INTERNAL ENERGY OF ,1PE11.4,//)
001463      902 FORMAT    (20H1  THE COMPONENT IS ,2A6,26H WITH A KINETIC ENERGY O
                1F ,1PE11.4,27H AND AN INTERNAL ENERGY OF ,1PE11.4,//)
001463      903 FORMAT (129H CELL    RADIUS       VEL          VOLUME       ENERGY
                1  PRESS        TEMP         MASS F       VISC         MASS       X,Z ST
                2RESS FLAG)
001463      904 FORMAT (129H1CELL    RADIUS       VEL          VOLUME       ENERGY
                1  PRESS        TEMP         MASS F       VISC         MASS       X,Z ST
                2RESS FLAG)
001463      905 FORMAT(I5,9(X,1PE11.4),X,2(3PF6.3),X,02)
001463      906 FORMAT (14H   DELTA T IS ,1PE10.3,25H WITH LEFT CELL INDEX OF ,I5,
                125H AND RIGHT CELL INDEX OF ,I5)
001463      910 FORMAT (10X,12A6)
001463      911 FORMAT (//,10X,38HPRESSURE (MEGABARS) VS RADIUS (CM) AT ,1PE10.3,
                131H MICROSECONDS AND CYCLE NUMBER ,I5)
001463      912 FORMAT (//,10X,43HTEMPERATURE (DEG KELVIN) VS RADIUS (CM) AT ,
                11PE10.3,31H MICROSECONDS AND CYCLE NUMBER ,I5)
001463      913 FORMAT (//,10X,41HMASS FRACTION OF SOLID VS RADIUS (CM) AT ,1PE10.
                13,31H MICROSECONDS AND CYCLE NUMBER ,I5)
001463      914 FORMAT (//,10X,45H SPECIFIC VOLUME   (CC/GM) VS RADIUS (CM) AT ,
                11PE10.3,31H MICROSECONDS AND CYCLE NUMBER ,I5)
001463      915 FORMAT (//,10X,46HPARTICLE VELOCITY (CM/MSEC) VS RADIUS (CM) AT ,
                11PE10.3,31H MICROSECONDS AND CYCLE NUMBER ,I5)
001463      916 FORMAT (///,10X,19HTAPE DUMP AT CYCLE ,I5)
001463          END
```

```
                SUBROUTINE UPDATE (NMIN,NMAX)
              C
              C     UPDATE MOVES THE NEW QUANTIES TO THE OLD TIME STEP QUANTIES
              C
000004          COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
               1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
               2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000004          COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
               1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,
               2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
               3NLCNT,NRCNT,NLH,NRL,DX,KIND,RXZDV,RXZVS,FPO,FEO,RXZ
000004          COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SOL(23,10),VCJ(10),
               1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
               2NCOM
000004          DO 10 I=NMIN,NMAX
000005          CRO(I)=CR(I)
000010          CUO(I)=CU(I)
000012          CVO(I)=CV(I)
000015          CWO(I)=CW(I)
000017       10 CIO(I)=CI(I)
000024          RETURN
000024          END
```

```
                SUBROUTINE VELOC (NMIN,NMAX)
          C
          C     VELOC CALCULATES THE NEW VELOCITY AT THE NEXT TIME STEP FOR
          C     CELLS WITH INDICES FROM NMIN TO NMAX
          C
000004            COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
                 1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
                 2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000004            COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
                 1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,IL8,
                 2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
                 3NLCNT,NRCNT,NLH,NRL,DX,KIND,RXZDV,RXZVS,FPO,FEO,RXZ
000004            COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SOL(23,10),VCJ(10),
                 1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
                 2NCOM
000004            IF (IALPH.EQ.0) GO TO 15
000005            DO 10 I=NMIN,NMAX
          C       J IS INDEX OF COMPONENT
000006            KSP=ICF(I)/64
000012            J = ICF(I)-64*KSP
000015            CU(I)=CUO(I)+2.*DELT*(CRO(I)**IALPH)/(CM(I-1)+CM(I))*(CP(I-1)
                 1-CP(I)+CQ(I-1)-CQ(I))
          C       IF YO OR MU LESS THAN 0.0001 SKIP ELASTIC CALCULATION
000040            IF(SOL(19,J).LT.0.0001) GO TO 10
000045            IF(SOL(20,J).LT.0.0001)  GO TO 10
000050            IF (CRO(I).EQ.0.) GO TO 10
000052            IF (IALPH.EQ.1) GO TO 50
          C       VELOCITY CALCULATION FOR ELASTIC SPHERE
000054            CU(I)=CU(I)-0.75*(CS(I)+CS(I-1))*(CVO(I)+CVO(I-1))*DELT/CRO(I)
000067            GO TO 10
          C       VELOCITY CALCULATION FOR ELASTIC CYLINDER
000067         50 CU(I)=CU(I)-(CS(I)+CS(I-1)+0.5*(CSZ(I)+CSZ(I-1)))*0.5*(CVO(I)+
                 1CVO(I-1))*DELT/CRO(I)
000107         10 CONTINUE
000112            RETURN
000112         15 DO 20 I=NMIN,NMAX
000114         20 CU(I)=CUO(I)+2.*DELT/(CM(I-1)+CM(I))*(CP(I-1)
                 1-CP(I)+CQ(I-1)-CQ(I))
000135            RETURN
000135            END
```

```
                SUBROUTINE VISCOS (NMIN,NMAX)
              C
              C     VISCOS COMPUTES THE VISCOSITY IN EITHER THE PIC, LANDSHOFF, OR
              C     REAL FORM FOR CELLS WITH INDICES NMIN TO NMAX
              C
000004                COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
                     1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
                     2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000004                COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
                     1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,
                     2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
                     3NLCNT,NRCNT,NLH,NRL,DX,KIND,RXZDV,RXZVS,FPO,FEO,RXZ
000004                COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SOL(23,10),VCJ(10),
                     1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
                     2NCOM
000004                DO 10 I=NMIN,NMAX
000005                UT=CU(I)-CU(I+1)
              C       J IS INDEX OF COMPONENT
000010                KSP=ICF(I)/64
000013                J=ICF(I)-64*KSP
              C       CHECK FOR REAL FORM
000017                IF (IVIS(J).EQ.2) GO TO 2
000022                CQ(I)=0.
000024                IF (UT.LE.0.) GO TO 10
              C       CHECK FOR LANDSHOFF FORM
000025                IF (IVIS(J).EQ.1) GO TO 1
              C       PIC FORM
000030                CQ(I)=ABS(VFACT(J)*.5*(CU(I)+CU(I+1))*UT/CV(I))
000042                GO TO 10
              C       LANDSHOFF FORM
000042              1 CQ(I)=VFACT(J)*UT/CV(I)
000050                GO TO 10
              C       REAL FORM
000050              2 CQ(I)=1.3333*VFACT(J)*UT/(CV(I)*CM(I))
000061             10 CONTINUE
000064                RETURN
000064                END
```

```
                SUBROUTINE VOLUM (NMIN,NMAX)
          C
          C     VOLUM FINDS THE SPECIFIC VOLUME OF CELLS WITH INDICES NMIN TO NMAX
          C
000004          COMMON CR(999),CRO(999),CU(999),CUO(999),CV(999),CVO(999),
               1CM(999),CQ(999),ICF(999),CW(999),CWO(999),CT(999),CP(999),
               2CS(999),CSZ(999),CI(999),CIO(999),LABEL(12)
000004          COMMON /MISC/ TTIME,DELT,ICYCL,PCNT,GCNT,DCNT,DINC,PINC,GINC,
               1IALPH,HEATC,GASW,NCL,PTMIN,IPNT,IMC,IPR,ITR,IWR,IIR,IUR,IHC,ILB,
               2IRB,WSW,PSMIN,WMAX,SMIN,ALB,BLB,ARB,BRB,NLI,NRI,NLINC,NRINC,
               3NLCNT,NRCNT,NLH,NRL,DX,KIND,RXZDV,RXZVS,FPO,FEO,RXZ
000004          COMMON /MATER/ E(10),GAS(17,10),IBRN(10),SOL(23,10),VCJ(10),
               1VFACT(10),Z(10),NAM(2,10),DTIME(10),NOINC(10),IVIS(10),IEXP(10),
               2NCOM
000004          DO 10 I=NMIN,NMAX
000005       10 CV(I)=((CR(I+1)+CR(I))*.5)**IALPH*(CR(I+1)-CR(I))/CM(I)
000024          RETURN
000024          END
```

```
      SUBROUTINE HOM (V,S,G,IND)
C
C     HOM CALCULATES THE EQUATION OF STATE FOR A SOLID, GAS, OR
C     SOLID-GAS MIXTURE.
C
C     THE PARAMETERS ARE
C     V    AN ARRAY OF DIMENSION 5
C          V(1)     SPECIFIC VOLUME       INPUT
C          V(2)     INTERNAL ENERGY       INPUT
C          V(3)     MASS FRACTION         INPUT
C          V(4)     -ABS(DP/DX)           INPUT
C          IF V(4) IS .GE.0  NO SPALLING CALCULATION IS DONE FOR A PURE
C          SOLID
C          V(4)     PRESSURE              OUTPUT
C          V(5)     TEMPERATURE           OUTPUT
C     S    AN ARRAY OF DIMENSION 18 CONTAINING THE PARAMETERS FOR THE
C          SOLID EQUATION OF STATE
C          S(1)     C
C          S(2)     S
C          S(3)     VSW   VOLUME TO SWITCH TO SECOND US,UP FIT
C          S(4)     C1
C          S(5)     S1
C          S(6)     F
C          S(7)     G
C          S(8)     H
C          S(9)     I
C          S(10)    J
C          S(11)    GAMMA
C          S(12)    CV
C          S(13)    V0   INITIAL VOLUME
C          S(14)    ALPHA
C          S(15)    SPALL A
C          S(16)    ULTIMATE SPALL PRESSURE
C          S(17)    T0    INITIAL TEMPERATURE
C          S(18)    P0     INITIAL PRESSURE
C     S(22) IS SPALL INTERFACE PRESSURE
C     S(23) IS MIN V FOR TWO PHASE FE EQUATION OF STATE
C     G    AN ARRAY OF DIMENSION 17 CONTAINING THE PARAMETERS FOR THE
C          GAS EQUATION OF STATE
C          G(1)     A
C          G(2)     B
C          G(3)     C
C          G(4)     D
C          G(5)     E
C          G(6)     K
C          G(7)     L
C          G(8)     M
C          G(9)     N
C          G(10)    O
C          G(11)    Q
C          G(12)    R
C          G(13)    S
C          G(14)    T
C          G(15)    U
C          G(16)    CV
C          G(17)    Z
C     IND    OUTPUT INDICATOR
C          SET TO 0 FOR NORMAL EXIT
C          SET TO 1 FOR SPALLED SOLID
C          SET TO -1 FOR HOM ERROR IN ITERATION
000006        DIMENSION V(5),S(23),G(17),VIT(10)
```

```
000006          DATA GASW /.01/
000006          DATA SOLW /.999/
000006          DATA SPMIN/5.0E-3/
000006          DATA VGSS /.9/
000006          DATA VIT(3) /1.E-5/
000006          DATA VIT(10) /0./
000006          IND=0
000006          IF (V(3).GT.SOLW) GO TO 10
000012          IF (V(3).LT.GASW) GO TO 110
000014          GO TO 210
                C  EQUATION OF STATE FOR SOLID ONLY
000014          10 IF (V(1).GT.S(13)) GO TO 50
                C     FOR TWO PHASE FE TYPE EQUATION OF STATE
000020          IF(V(1).GT.S(3)) GO TO 11
000023          IF(V(1).LT.S(23)) GO TO 45
000024          V(1)=S(23)
000026          GO TO 45
000026          11 C1=S(1)
000027          S1=S(2)
000031          20 VOMV=S(13)-V(1)
000033          HP=((C1/(S(13)-S1*VOMV))**2)*VOMV
000040          HE=HP*VOMV*.5
000042          V(4)=HP+(V(2)-HE)*S(11)/V(1)
                C     IF NO HEAT CAPACITY SKIP TEMP CALCULATION
000047          IF (S(12)) 21,22,21
000050          21 ALNV=ALOG(V(1))
000055          V(5)=(V(2)-HE)*23890./S(12)+EXP(S(6)+ALNV*(S(7)+ALNV*(S(8)+ALNV*
                1(S(9)+ALNV*S(10)))))
000077          22 RETURN
                C     SWITCH TO SECOND US.UP FIT
000100          45 C1=S(4)
000102          S1=S(5)
000103          GO TO 20
                C     SPALLING SOLID EQUATION OF STATE
000104          50 DPDX=V(4)
                C     IF ALPHA IS ZERO SET P=P0 AND DO NOT SPALL
000106          IF (S(14)) 51,51,52
000107          51 V(4)=S(18)
000111          V(5)=S(17)
000112          RETURN
000113          52 V(4)=(S(11)*(V(2)+(1.-V(1)/S(13))*S(12)*1.3952839+E-5/S(14)))/V(1)
000125          V(5)=V(2)*23890./S(12)+S(17)
000131          IF (DPDX.GE.0.) RETURN
                C     IF SPA LESS THAN / 0.0001 DO NOT SPALL
000134          IF (S(15) .LT.0.0001)    RETURN
000137          T=S(15)*SQRT(-DPDX)
000147          SPLP=-T
000150          IF (T.GT.S(16)) SPLP=-S(16)
000154          IF (T.LT.SPMIN) SPLP=-SPMIN
000160          IF (V(4).GT.SPLP) RETURN
000163          V(4)=S(22)
                C     SET IND FOR SPALLED SOLID
000165          IND=1
000166          RETURN
                C  EQUATION OF STATE FOR GAS ONLY
000166          110 ALNV=ALOG(V(1))
000173          ALNPI=G(1)+ALNV*(G(2)+ALNV*(G(3)+ALNV*(G(4)+ALNV*G(5))))
000203          ALNII=G(6)+ALNPI*(G(7)+ALNPI*(G(8)+ALNPI*(G(9)+ALNPI*G(10))))
000214          ALNTI=G(11)+ALNV*(G(12)+ALNV*(G(13)+ALNV*(G(14)+ALNV*G(15))))
000225          EI=EXP(ALNII)-G(17)
000233          V(4)=EXP(ALNPI)+(EI-V(2))/V(1)*(G(12)+ALNV*(G(13)+G(13)+ALNV*(
                13.*G(14)+ALNV*4.*G(15))))
```

```
000257          V(5)=EXP(ALNTI)+(V(2)-EI)*23890./G(16)
000272          RETURN
                C  EQUATION OF STATE FOR MIXTURE OF SOLID AND GAS
000272        210 OMW=1.-V(3)
000274          OMWR=1./OMW
000276          IF (V(1).LT.S(13)) GO TO 230
000300          WR=1./V(3)
000302          VIT(1)=(V(1)-V(3)*S(13)*VGSS)*OMWR
000307          VIT(2)=1.002
                C       IBR=1 FOR ITERATION ON VG
000310          IBR=1
000311        215 CALL LFB (X,F,VIT)
000321          IF (VIT(10)) 900,260,220
000323        220 IF (X.LE.0.) GO TO 225
000325          VG=X
000326          VS=(V(1)-OMW*VG)*WR
000332          IF (VS.LE.0.) GO TO 225
000333          IF (VS.GT.S(13)) VS=S(13)
000337          GO TO 250
                C       SET VS=VG=VOLUME WHEN GET IN TROUBLE
000340        225 VS=V(1)
000341          VG=V(1)
000342          X=V(1)
000343          GO TO 250
000344        230 VIT(1)=V(1)
000345          VIT(2)=.999
                C       IBR=2 FOR ITERATION ON VS
000347          IBR=2
000350        235 CALL LFB (X,F,VIT)
000360          IF (VIT(10)) 900,260,240
000362        240 IF (X.LE.0.) GO TO 225
000364          IF (X.GT.S(13)) X=S(13)
000367          VS=X
000370          VG=(V(1)-V(3)*VS)*OMWR
000374          IF (VG.LE.0.) GO TO 225
                C       CALCULATE TEMPERATURE/PRESSURE DIFFERENCE FOR MIXTURE ITERATION
000375        250 VOMV=S(13)-VS
000377          HP=((S(1)/(S(13)-S(2)*VOMV))**2)*VOMV
000404          HE=HP*.5*VOMV
000406          ALNV=ALOG(VS)
000412          HT=EXP(S(6)+ALNV*(S(7)+ALNV*(S(8)+ALNV*(S(9)+ALNV*S(10)))))
000427          ALNV=ALOG(VG)
000434          ALNPI=G(1)+ALNV*(G(2)+ALNV*(G(3)+ALNV*(G(4)+ALNV*G(5))))
000444          EI=EXP(G(6)+ALNPI*(G(7)+ALNPI*(G(8)+ALNPI*(G(9)+ALNPI*G(10)))))-
                1G(17)
000463          PI=EXP(ALNPI)
000467          TI=EXP(G(11)+ALNV*(G(12)+ALNV*(G(13)+ALNV*(G(14)+ALNV*G(15)))))
000504          BETER=-(G(12)+ALNV*(G(13)+G(13)+ALNV*(3.*G(14)+4.*ALNV*G(15))))
000515          TEMP=-G(16)*BETER/VG
000520          TEMP1=S(11)*S(12)/VS
000523          F=-(HT*TEMP1+TI*TEMP)*4.18585182E-5
000527          TEMP=TEMP+TEMP1
000531          VSTO=(S(12)-G(16))*V(3)+G(16)
000534          F=((OMW*G(16)*TI+V(3)*S(12)*HT)*4.18585182E-5+(EI-HE)*V(3)-EI+
                1V(2))*TEMP/VSTO+F-PI+HP
000556          GO TO (215,235),IBR
                C       HAVE FOUND A SOLUTION FOR THE MIXTURE
                C       GET THE TEMPERATURE AND PRESSURE
000563        260 VARST=((((TI-HT)*G(16)*4.18585182E-5+V(2)*OMWR-EI)*S(12)+HE*G(16))
                1*OMW/VSTO)-HE
000600          V(4)=HP+VARST*S(11)/VS
000604          V(5)=HT+VARST*23890./S(12)
```

```
000610            RETURN
                  C     ERROR IN HOM ITERATION      SET IND TO -1
000610              900 IND=-1
000611            RETURN
000612            END
```

```
                SUBROUTINE LFB (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 RDDT
              C
              C      ERROR EXITS OCCUR FOR
              C              1. TOO MANY ITERATIONS, .GT.CNTMAX
              C              2. TWO SUCESSIVE XP S OR FP S ARE EQUAL
000005                DIMENSION TX(10)
000005                DATA CNTMAX /1000./
000005                IF (TX(10).LE.0.) GO TO 1
000006                TX(10)=TX(10)+1.
000011                IF (TX(10)-3.) 2,3,4
              C  ENTRY FIRST TIME THROUGH
000014             1  TX(10)=1.
000016                IF (TX(1).EQ.0) TX(1)=1.
000020                XP=TX(1)
              C  GO GET F(XP)
000021                RETURN
              C  ENTRY SECOND TIME THROUGH
000022             2  TX(9)=FP
000024                TX(8)=XP
000025                TX(5)=FP
000026                IF (ABS(FP).LT.TX(3)) GO TO 18
000030                XP=TX(1)*TX(2)
              C  GO GET F(XP)
000031                RETURN
              C  ENTRY THIRD TIME THROUGH
000032             3  TX(5)=FP
000034                TX(6)=XP
000035                TX(7)=FP
000036                IF (ABS(FP).LT.TX(3)) GO TO 18
000040                XP=TX(6)-TX(7)*(TX(6)-TX(8))/(TX(7)-TX(9))
              C  GO GET F(XP)
000047                RETURN
              C  ENTRY FOR FOURTH AND SUCEEDING TIMES THROUGH
000050             4  IF (TX(10).GT.CNTMAX) GO TO 99
000054                TX(4)=XP
000055                TX(5)=FP
000056                T=TX(4)-TX(6)
000060                IF (T.EQ.0.) GO TO 99
000061                IF (ABS(FP).LT.TX(3)) GO TO 18
000063                R=TX(5)-TX(7)
000065                IF (R.EQ.0.) GO TO 99
000066                XP=TX(4)-TX(5)*(T/R)
000072                IF (TX(5)*TX(7).LT.0.) GO TO 11
000074                IF (TX(5)*TX(9).GE.0.) GO TO 11
000076                IF (XP.GT.TX(4)) GO TO 6
000102                IF (XP.GT.TX(8)) GO TO 10
000105             8  XP=TX(4)-TX(5)*(TX(4)-TX(8))/(TX(5)-TX(9))
000114            10  TX(7)=TX(5)
000116                TX(6)=TX(4)
              C  GO GET F(XP)
000117                RETURN
000120             6  IF (XP.GT.TX(8)) GO TO 8
```

```
000124          GO TO 10
000124       11 TX(9)=TX(7)
000126          TX(8)=TX(6)
000127          GO TO 10
             C  HAVE FOUND A SOLUTION
000130       18 TX(10)=0.
000131          TX(1)=XP
000132          TX(4)=XP
000133          RETURN
             C  AN ERROR HAS OCCURED
             C  SET COUNT NEGATIVE AND EXIT
000134       99 TX(10)=-TX(10)
000136          RETURN
000136          END
```

APPENDIX C

THE GAMMA-LAW EQUATION OF STATE AND HUGONIOT TEMPERATURE PROGRAMS

The HOM equation of state used in FORTRAN SIN requires fits to the equation of state parameters for detonation products and condensed components. The FORTRAN BKW code[8] produces the coefficients to the required fits using the Becker-Kistiakowsky-Wilson equation of state to describe the isentrope of the detonation products.

In this appendix we describe a code which will produce the coefficients to the required fits using a gamma-law equation of state to describe the isentrope of the detonation products. We also describe a code which calculates the single-shock Hugoniot temperatures using the technique of Walsh and Christian[9] and produces the required fits for use in the FORTRAN SIN code.

The Gamma-Law Equation of State Program

A. The Nomenclature

CJ	Chapman-Jouguet
D	detonation velocity
P	pressure
U_p	particle velocity
V	volume
V_o	initial specific volume
Z	constant added to energies to make them positive
γ	gamma
ρ_o	initial density

B. The Equations

$$P_{CJ} V_{CJ}^{\gamma} = C$$

$$V_{CJ} = \frac{\gamma V_o}{(\gamma + 1)}$$

$$\ln P = \ln C + \gamma \ln V$$

$$I = \frac{PV}{\gamma - 1} - \frac{P_{CJ} V_{CJ}}{\gamma - 1} + \frac{P_{CJ}}{2}(V_o - V_{CJ}) + Z$$

$$U_p = \frac{P_{CJ}}{\alpha' \rho D}\left[1 + \alpha' - \left(\frac{P}{P_{CJ}}\right)^{\alpha'}\right] \quad \text{where } \alpha' = \frac{\gamma - 1}{2\gamma}$$

$$\alpha = \left[\frac{(\alpha + 1)}{\left(1 + \frac{d\ln D}{d\ln \rho}\right)} - 2\right]^{-1}$$

$$3 = \frac{1 + \alpha}{\gamma}$$

C. The Input to the Code

1st card

Col.	Format	
1-72	12A6	Alphabetic label for problem

2nd card

Col.	Format	
1-18	E18.11	γ
19-36	E18.11	P_{CJ} (Mbar)
37-54	E18.11	ρ_o (g/cc)

3rd card

Col.	Format	
1-18	E18.11	D
19-36	E18.11	$d(\ln D)/d(\ln \rho)$
37-54	E18.11	Z

D. The FORTRAN Code

```
                PROGRAM GLAW (INPUT,OUTPUT,FILM,TAPE12=FILM)
          C     GAMMA,PCJ,RHO,DETV,DDDR,Z ARE INPUT CONSTANTS
000003          DIMENSION P(100 ),E(100 ),V(100 ),ALGP(100 ),ALGV(100 ),ALNE(100 )
                1,ALNP(100 ),UP(100 ),FTE(100 ),ECOEF(5),ALNV(100 ),W(100 )
                2,DELY(100 ),SB(5),T(5),A(5,5),LABEL(12),ST(5)
000003          DATA PLFACT/+0.8/
000003          DATA PMAX  /+1.0/
000003          DATA PMIN  /+1.0E-4/
000003          DATA PUFACT/+1.15/
000003          DATA AM  /+0.43429448819/
000003          DATA ICHAR /044/
000003          DATA ICHAR2/063/
000003      500 READ 900,LABEL
000011          READ 901,GAMMA,PCJ,RHO
000023          READ 901,DETV,DDDR,Z
000035      900 FORMAT (12A6)
000035      901 FORMAT (4E18.11)
000035          FITB=-GAMMA
000037          ALPHA= 1.0/((GAMMA+1.)/(1.+DDDR) - 2.)
000045          BETA= (1+ALPHA)/GAMMA
000050          FITR= -1./BETA
000051          VCJ= GAMMA/((GAMMA+1.)*RHO)
000054          C = PCJ*(VCJ**GAMMA)
000060          ALNC= ALOG(C)
000062          ALPHP= (GAMMA -1.)/(2.*GAMMA)
000065          ENCT = -PCJ*VCJ/(GAMMA-1.) + 0.5*PCJ*(1./RHO -VCJ) + Z
000077          P(1)=PCJ
000100          I=1
000101          K=1
000102          DO 10 J = 1,100
000103       20 ALNP(I)=ALOG(P(I))
000107          ALGP(I)=ALNP(I)*AM
000111          ALNV(I)=-(ALNP(I)-ALNC)/ GAMMA
000115          ALGV(I)= ALNV(I)*AM
000117          V(I)=EXP(ALNV(I))
000122          E(I)= (P(I)*V(I))/(GAMMA-1.)+ENCT
000130          ALNE(I) = ALOG(E(I))
000134          IF(P(I).GT. PCJ) GO TO 12
000140          UP(I)= PCJ/(RHO* ALPHP*DETV)*(1.+ALPHP-(P(I)/PCJ)**ALPHP)
000152          P(I+1) =P(I)*PLFACT
000154          IF(P(I+1).LT. PMIN) GO TO 11
000156          I=I+1
000160          K=K+1
000161          GO TO 20
000161       11 P(I+1) = PCJ*PUFACT
000164          GO TO 13
000164       12 P(I+1)=P(I)*PUFACT
000167       13 IF(P(I+1).GT.PMAX) GO TO 30
000173          I=I+1
000174          GO TO 20
000174       10 CONTINUE
000176       30 CALL PFTS(I,4,0,SIGMA,ALNP,ALNE,W,FTE,DELY,ECOEF,SB,T,ST,A)
000214          PRINT 911,LABEL
000222          PRINT 902,GAMMA,PCJ,RHO
000234          PRINT 903,DETV,DDDR,Z
000246          PRINT 904,ALPHA,BETA,ENCT
000260          PRINT 905,FITR,C
000270          PRINT 906,ALNC,FITB
000300          PRINT 907,(ECOEF(I),I=1,5)
000312          PRINT 908
000316          NOH=I-K
```

```
000320          M= I
000321          DO 40 L = 1,NOH
000322          FTE(M)=EXP(FTE(M))
000326          PRINT 909,P(M),V(M),E(M),FTE(M)
000342          M=M-1
000344       40 CONTINUE
000346          DO 50 L = 1,K
000350          FTE(L)=EXP(FTE(L))
000354          PRINT 910,P(L),V(L),E(L),FTE(L),UP(L)
000372       50 CONTINUE
000375          WRITE (12,911) LABEL
000402          WRITE (12,902) GAMMA,PCJ,RHO
000414          WRITE (12,903) DETV,DDDR,Z
000426          WRITE (12,904) ALPHA,BETA,ENCT
000440          WRITE (12,905) FITR,C
000450          WRITE (12,906) ALNC,FITB
000460          WRITE (12,907) (ECOEF(I),I=1,5)
000472          WRITE (12,908)
000476          M=I
000500          DO 140 L = 1,NOH
000501          WRITE (12,909) P(M),V(M),E(M),FTE(M)
000514          M=M-1
000516      140 CONTINUE
000520          DO 150 L=1,K
000522          WRITE (12,910) P(L),V(L),E(L),FTE(L),UP(L)
000537      150 CONTINUE
            C     GRAPH LOG PRESSURE VS VOLUME
000542          CALL ADV(1)
000543          CALL DGA (123.1023.0.900.-1.,+2.,+0.,-4.)
000553          CALL DLGLG
000554          CALL SLLOG
000555          CALL SBLOG
000556          CALL PLOT (I.ALGV,1,ALGP,1,ICHAR,0)
000565          CALL PLOT (1.ALGV.1.ALGP,1,ICHAR2,0)
000574          CALL LINCNT (60)
000576          WRITE (12,900) LABEL
000604          WRITE (12,912)
            C     GRAPH  LOG PRESSURE  VS PARTICLE VELOCITY
000610          CALL ADV(1)
000612          CALL DGA(123.1023,0,900,0.,+1.,+0.,-4.)
000622          CALL DLNLG(10)
000624          CALL SLLOG
000625          CALL SBLIN(10,2)
000627          CALL PLOT (K.UP.1.ALGP,1,ICHAR,1)
000636          CALL LINCNT (60)
000640          WRITE (12,900) LABEL
000646          WRITE (12,913)
000652          GO TO 500
000653      911 FORMAT (27H1 GAMMA LAW CALCULATION FOR,12A6)
000653      902 FORMAT (/,10H GAMMA IS ,1PE18.11,17H C-J PRESSURE IS ,1PE18.11,
               112H DENSITY IS ,1PE18.11)
000653      903 FORMAT (/,24H DETONATION VELOCITY IS ,1PE18.11,18H DLND/DLN(RHO) I
               1S ,1PE18.11,31H Z-CONSTANT ADDED TO ENERGY IS ,1PE18.11)
000653      904 FORMAT (/,19H COMPUTED ALPHA IS ,1PE18.11, 9H BETA IS ,1PE18.11,
               123H ENERGY CONSTANT    IS ,1PE18.11)
000653      905 FORMAT (/,11H LNT = Q + ,1PE18.11,30H LNV      AND  PV TO GAMMA IS
               1 ,1PE18.11)
000653      906 FORMAT (/, 8H LN(P)= ,1PE18.11,3X,1PE18.11,5HLNV  )
000653      907 FORMAT (/,8H LN(E)= ,1PE18.11,3X,1PE18.11,5HLNP  ,1PE18.11,6HLNP*2
               1 ,1PE18.11,6HLNP*3 ,1PE18.11,5HLNP*4)
000653      908 FORMAT (//,112H    PRESSURE(MBARS)     VOLUME (CC/GM)       ENERGY
               1(MB-CC/GM)      FIT ENERGY          PARTICLE VELOCITY(CM/USEC))
```

```
000653      909 FORMAT(4(3X,1PE18.11))
000653      910 FORMAT(5(3X,1PE18.11))
000653      912 FORMAT (59H     THE PRESSURE-VOLUME ISENTROPE FOR GAMMA LAW ASSUMPT
                 1ION )
000653      913 FORMAT (46H     THE PRESSURE -PARTICLE VELOCITY ISENTROPE )
000653          END
```

```
      SUBROUTINE   PFTS(M,KM,IW,SIGMA,X,F2,W,Y,DELY,B,SB,T,ST,A)
C     THIS IS A SPECIAL VERSION OF LA-PFTS FOR A MAX OF 100 POINTS
C     AND A MAX OF FIFTH DEGREE FIT  NO PRINTING  OR LEGENDRE POLYNOMIALS
C     ALSO WILL COMPUTE TILL GET THE FIT AND REQUIRES NO OUTPUT SUBROUTINE
C     M = NO OF DATA POINTS
C     KM = DEGREE OF FIT  (MAX OF 5)
C     IW = 0 FOR NO  WEIGHTS AND EQUAL 1 FOR WEIGHTS
C     SIGMA       IS STANDARD DEVIATION COMPUTED
C     X = X DATA       INPUT ARRAY
C     F2 = F2 DATA ARRAY  INPUT
C     W = WEIGHT DATA ARRAY  INPUT    SET EQUAL TO 1 IF IW IS 0
C     Y  =  F COMPUTED FROM FIT USING X
C     DELY = DIFFERENCE IN COMPUTED AND INPUT F
C     B = COEFFICIENTS TO FIT
C     SB = ESTIMATE OF ERRORS IN COEFFICIENTS TO FIT
C     T = COEFFICIENTS TO ORTHOGONAL POLYNOMINALS
C     ST = ERRORS IN T
C     A = AREA USED BY CALCULATION
000017      DIMENSION   S(5),X(100),F2(100),ST(5),SB(5),F(100),PM(100),P(100)
     1 , B(5),DELY(100),W(100),A(5,5),T(5),Y(100)
000017      LL=0
000020    9 FM=0.0
000021      A(1,1)=1.0
000024      A(2,2)=1.0
000026      FBAR=0.0
000027      XBAR=0.0
000030      DO10I=1,M
000031      IF(IW)1009,1010,1009
000032 1010 W2=1.0
000034      W(I)=1.0
000037      GOTO1011
000040 1009 W2=SQRT(W(I))
000051 1011 FM=FM+W(I)
000055      F(I)=W2*F2(I)
000061      PM(I) = W2
000064      FBAR=FBAR+F(I)*PM(I)
000071   10 XBAR=XBAR+X(I)*PM(I)**2
000077      XBAR=XBAR/FM
000100      T(1)=FBAR/FM
000103      A(2,1)=-XBAR
000106      PXF=0.0
000107      PXP=0.0
000110      DO20I=1,M
000111      P(I)=(X(I)-XBAR)*PM(I)
000117      PXF=PXF+P(I)*F(I)
000124   20 PXP=PXP+P(I)*P(I)
000131      T(2)=PXF/PXP
000134      PMXPM=FM
000136      S(1)=PMXPM
000137      KM=KM+1
000140      B(1)=T(1)*A(1,1)+T(2)*A(2,1)
000147      B(2)=T(2)*A(2,2)
000154   60 DO190K=2,KM
000156      IF(K-2)40,165,65
000160   40 STOP
000162   65 XPXP=0.0
000163      XPXPM=0.0
000164      B(K)=0.0
000167      DO70J=1,M
000171      XP=X(J)*P(J)
000175      XPXP=XPXP+XP*P(J)
```

```
000200       70 XPXPM=XPXPM+XP*PM(J)
000205          ALPHA=XPXP/PXP
000207          BETA=XPXPM/PMXPM
000211          PPXF=0.0
000212          PPXPP=0.0
000213          DO90I=1,M
000214       80 PT=P(I)
000217       81 P(I)=X(I)*PT-ALPHA*PT-BETA*PM(I)
000230       82 PPXF=PPXF+P(I)*F(I)
000235       83 PPXPP=PPXPP+P(I)*P(I)
000241       90 PM(I)=PT
000246          T(K)=PPXF/PPXPP
000252          PMXPM=PXP
000254          PXP=PPXPP
000254          A(K,1)=-ALPHA*A(K-1,1)-BETA*A(K-2,1)
000264          A(K,K-1)=A(K-1,K-2)-A(K-1,K-1)*ALPHA
000276          A(K,K)=1.0
000302          IF(K-3)150,150,110
000304      110 K1=K-2
000306          DO120I=2,K1
000310      120 A(K,I)=A(K-1,I-1)-ALPHA*A(K-1,I)-BETA*A(K-2,I)
000342      150 DO160I=1,K
000344      160 B(I)=B(I)+T(K)*A(K,I)
000364      165 SIG2=0.0
000365          DO180I=1,M
000367          Y(I)=POLY (X(I),K,B)
000404      175 DELY(I)=Y(I)-F2(I)
000413      180 SIG2=SIG2+(DELY(I)**2)*W(I)
000423          SIG2=SIG2/FLOAT(M-K)
000426          SIGMA=SQRT(SIG2)
000433          S(K) = PXP
000437          DO499I=1,K
000441      499 ST(I)=SIGMA/SQRT(S(I))
000457          DO501I=1,K
000461          SB(I)=0.0
000464          DO500J=I,K
000465      500 SB(I)=SB(I)+(A(J,I)*ST(J))**2
000507      501 SB(I)=SQRT(SB(I))
000522      190 CONTINUE
000524      220 KM=KM-1
000526          RETURN
000526           END
```

The Hugoniot Temperature Program

A. The Nomenclature

C, S	coefficients to $U_s = C + S(U_p)$ used from P_o to switch pressure
C1, S1	coefficients to $U_S = C1 + S1(U_p)$ used from switch pressure to maximum pressure
C1, $C_2\rho_o$, $(C_3\rho_o)^2$	coefficients for electronic term of Cowan equation of state
C_V	heat capacity (cal/g-°C)
K	isothermal compressibility ($Mbar^{-1}$)
P	pressure
T	temperature (°K)
T_o	initial temperature (°K)
T_V	temperature (volts)
U_p	particle velocity
U_s	shock velocity
V_o	$1/\rho_o$
α	linear coefficient of expansion ($^oC^{-1}$)
ΔV	volume increment ($\sim 1 \times 10^{-4}$) where $V_o - V_{smallest}/\Delta V < 5000$
ρ	density (g/cc)
ρ_o	initial density (g/cc)

B. The Equations

The following equation is solved

$$T = T_o e^{b(V_o - V)} + \frac{(V_o - V)P}{2C_V} + \frac{e^{-bV}}{2C_V} \times \int_{V_o}^{V} Pe^{bV}[2 - b(V_o - V)]dV,$$

with

$$b = \frac{3\alpha}{K(C_V)}$$, using Simpson's rule.

The input volume increment is used to compute a table of volumes which are used to compute pressures from

$$P = \frac{C^2(V_o - V)}{[V_o - S(V_o - V)]^2} .$$

The shock and particle velocities are found from

$$U_s = \frac{C + \sqrt{C^2 + 4SPV_o}}{2}$$

and

$$U_p = \frac{U_s - C}{S} .$$

The Cowan equation of state is

$$P = A + B\rho + C\rho^2 + D\rho^3 + E\rho^4 + (A1 + A2\rho)T_V + \left[C_1 + \frac{(C_2\rho_o)}{\rho} + \frac{(C_3\rho_o)^2}{\rho^2}\right] T_V^2 ,$$

where

$$\frac{3\alpha}{K} = A1 + A2\rho_o, \text{ and } A2 = \frac{1.447404}{(\text{Atomic weight})} ,$$

which is the form used in FORTRAN BKW.[8]

C. The Input to the Code

1st card

Col.	Format	
1-72	12A6	Alphabetic label for problem

2nd card

Col.	Format	
1-18	E18.11	C
19-36	E18.11	S
37-54	E18.11	ρ_o
55-72	E18.11	K

3rd card

Col.	Format	
1-18	E18.11	α
19-36	E18.11	C_V
37-54	E18.11	T_o
55-72	E18.11	ΔV

4th card

Col.	Format	
1-18	E18.11	Smallest P for temperature fit
19-36	E18.11	Largest P for temperature fit

Note: If the two-phase feature is used, the temperature fit will include for the C1, S1 volumes only those temperatures which increase with decreasing volume.

5th card

Col.	Format	
1-18	E18.11	Maximum pressure to be calculated
19-36	E18.11	C1
37-54	E18.11	S1
55-72	E18.11	Switch pressure

6th card

Col.	Format	
1-18	E18.11	Atomic weight. If zero, no Cowan fit.
19-36	E18.11	C_1
37-54	E18.11	$C_2\rho_o$
55-72	E18.11	$(C_3\rho_o)^2$

D. The FORTRAN Code

```
      PROGRAM SEOS (INPUT,OUTPUT,FILM,TAPE12=FILM)
000003      DIMENSION  V(5011),P(5010),EP(5010),T(500) ,US(500) ,UP(500)
           1,TC(5),FITT(500),LABEL(12),TV(500),ALNT(500),ALNVT(500)
           2,TCALC(500),W(500),DELY(500),SB(5),TZ(5),A(5,5),PV(500),PP(500)
           3,CRHO(500),ZP(500),CCOEF(5),ST(5) ,TX(10)
000003      DATA DKV /+11605.6/
000003      DATA CALMB /23890./
000003      DATA ICHAR /055/
000003      DATA TX(1) /0./
000003      DATA TX(2) /1.01/
000003      DATA TX(3) /+1.0E-5/
000003      DATA TX(10) /0./
000003  366 READ 900,LABEL
000011      READ 901,C,S,RHO,AK
000025      READ 901,ALPHA,CP,TO,DELTV
000041      READ 901,TP1,TP2
000051      READ 901,AMAXP,C1,S1,SWP
000065      READ 901,ATWT,AC1,AC2,AC3
          C     C,S, C1,S1 COEFFICIENTS TO  US-UP FITS
          C     SWITCH FROM  C,S TO  C1,S1  AT SWP -THE SWITCH PRESSURE
          C     ALPHA = LINEAR COEFFICIENT OF EXPANSION
          C     CP    = HEAT CAPACITY IN  CAL/GM/DEGC
          C     AK    = ISOTHERMAL COMPRESSIBILITY  IN MBARS
          C     RHO   = DENSITY  GM/CC
          C     TO    =  INITIAL TEMPERATURE IN DEG KELVIN
          C     DETV  =  VOLUME INCREMENT  ABT 1 X 10-4
          C     AMAXP  = MAX PRESSURE
          C     ATWT,AC1,AC2,AC3  COWAN EQ COEFFICIENTS SKIP IF ZERO
000101      VO=1./RHO
000103      Y=C
000105      Z=S
000106      I = 2
000107      BCV=CP/CALMB
000111      B= 3.*ALPHA/(AK*BCV)
000114      V(2) = VO-DELTV
000116      SWV = 0.
000117      DO 10 J = 2,5000
000121   11 P(I)=(Y*Y*(VO-V(I)))/((VO-Z*(VO-V(I)))*(VO-Z*(VO-V(I))))
000132      IF(P(I).GT. AMAXP) GO TO 13
          C     FOR PHASE CHANGE USE CONSTANT PRESSURE
000135      IF (P(I).LT.SWP.AND.Y.EQ.C1) P(I) = SWP
000147      IF(P(I).LT. SWP)  GO TO 12
000152      Y=C1
000153      Z=S1
000155      IF (SWV.EQ.0.) SWV = V(I)
000157   12 EP(I)=P(I)*EXP(B*V(I))*(2.-B*(VO-V(I)))
000173      V(I+1) = V(I)-DELTV
000175      I = I+1
000176   10 CONTINUE
000200   13 K=2
000201      M = I + 10
000203      DO 14 L = I,M
000204      P(L) = 0.
000205   14 CONTINUE
000207      I = I -1
000211      IT=1
000212      EP(1) = 0.
000213      PP(1) = 0.
000214      PV(1)= VO
000215      T(1)= TO
000217      US(1) = C
```

```
000220          UP(1) = 0.
000221          Y=C
000222          Z=S
000223          SUM = 0.
000224          DO 20 J=1,I,10
000225       21 PP(K)=P(J+10)
000230          IF (PP(K).LT.+1.E-8) GO TO 30
000232          PV(K)=V(J+10)
000234          IF (PV(K).GT.SWV) GO TO 27
000237          Y=C1
000240          Z=S1
000241       27 SUM = SUM + (EP(J)+4.*EP(J+1)+2.*EP(J+2)+ 4.*EP(J+3
               1)+2.*EP(J+4)+4.*EP(J+5)+2.*EP(J+6)+4.*EP(J+7)+2.* EP(J+8)+
               24.*EP(J+9)+ EP(J+10))
000271          T(K)= T0*EXP(B*(V0-PV(K)))+ ((V0-PV(K))*PP(K))/(2.*BCV) -
               1((EXP(-B*PV(K)))/(2.*BCV))*SUM*DELTV/3.
000321       22 US(K)=(Y+ SQRT(Y*Y + 4.*PP(K)*V0*Z))*0.5
000334          UP(K)=(US(K) -Y)/Z
000337          IF (T(K).LT.T(K-1)) GO TO 23
000341          IF(PP(K).GT.TP1.AND.PP(K).LT.TP2) GO TO 24
000351          GO TO 23
000351       24 ALNVT(IT)=ALOG(PV(K))
000355          ALNT (IT)=ALOG(T(K))
000361          IT=IT+1
000363       23 K=K+1
000365       20 CONTINUE
000367       30 IT =IT-1
000371          K=K-1
000372          HIP = PP(K)
000374          HIV = V0
000375          HIT = T(K)
000377          HIUS = US(K)
000400          HIUP = UP(K)
000402          CALL PFTS (IT,4.0,SIGMA,ALNVT,ALNT,W,TCALC,DELY,TC,SB,TZ,ST,A)
000417          PRINT 902,LABEL
000425          PRINT 903,C,S,SWP
000437          IF (SWP.GT.AMAXP) GO TO 41
000443          PRINT 904,SWP,AMAXP,C1,S1
000456       41 PRINT 905,RHO
000464          PRINT 906,AK
000472          PRINT 907,ALPHA
000500          PRINT 908,T0
000506          PRINT 909,CP
000514          PRINT 910,DELTV
000522          PRINT 911,TP1,TP2
000532          PRINT 912, (TC(I),I=1,5)
000544          PRINT 913
000550          DO 42 I=1,K
000552          PRINT 914, PV(I),PP(I),T(I),US(I),UP(I)
000567       42 CONTINUE
000572          PRINT 915
000575          DO 43 I=1,IT
000577          ALNVT(I)= EXP(ALNVT(1))
000603          ALNT(I)= EXP(ALNT(I))
000610          TCALC(I)=EXP(TCALC(I))
000615          PRINT 914, ALNVT(I), ALNT(I),TCALC(I)
000627       43 CONTINUE
000632          WRITE(12,902) LABEL
000637          WRITE(12,903) C,S,SWP
000651          IF(SWP.GT.AMAXP) GO TO 51
000655          WRITE(12,904) SWP,AMAXP,C1,S1
000670       51 WRITE(12,905) RHO
```

```
000674          WRITE(12,906) AK
000704          WRITE(12,907) ALPHA
000712          WRITE(12,908) TO
000720          WRITE(12,909) CP
000726          WRITE(12,910) DELTV
000734          WRITE(12,911) TP1,TP2
000744          WRITE(12,912) (TC(I),I=1,5)
000756          WRITE(12,913)
000762          DO 52 I=1,K
000764          WRITE(12,914) PV(I),PP(I),T(I),US(I),UP(I)
001001       52 CONTINUE
001004          WRITE(12,915)
001007          DO 53 I=1,IT
001011          WRITE(12,914) ALNVT(I), ALNT(I),TCALC(I)
001022       53 CONTINUE
             C     PRESSURE -VOLUME PLOT
001025          CALL ADV(1)
001026          CALL DGA(123,1023,0,900,0,,HIV,HIP,0,)
001036          CALL DLNLN (20,20)
001040          CALL SLLIN (20,2)
001042          CALL SBLIN (20,2)
001044          CALL PLOT (K,PV,1,PP,1,ICHAR,1)
001053          CALL LINCNT (60)
001055          WRITE (12,923)
001061          WRITE (12,900) LABEL
             C     TEMPERATURE-VOLUME PLOT
001067          CALL ADV(1)
001071          CALL DLNLN (20,20)
001073          CALL DGA(123,1023,0,900,0,,HIV,HIT,0,)
001103          CALL SLLIN (20,1)
001105          CALL SBLIN (20,2)
001107          CALL PLOT (K,PV,1,T,1,ICHAR,1)
001116          CALL LINCNT (60)
001120          WRITE (12,924)
001124          WRITE (12,900) LABEL
             C     PRESSURE-PARTICLE VELOCITY PLOT
001132          CALL ADV(1)
001134          CALL DGA(123,1023,0,900,0,,1,0,1,5,0,)
001144          CALL DLNLN (20,20)
001146          CALL SLLIN (20,2)
001150          CALL SBLIN (20,2)
001152          CALL PLOT (K,UP,1,PP,1,ICHAR,1)
001161          CALL LINCNT (60)
001163          WRITE (12,925)
001167          WRITE (12,900) LABEL
             C     SHOCK-PARTICLE VELOCITY PLOT
001175          CALL ADV(1)
001177          CALL DGA(123,1023,0,900,0,,HIUP,HIUS,0,)
001207          CALL DLNLN (20,20)
001211          CALL SLLIN (20,2)
001213          CALL SBLIN (20,2)
001215          CALL PLOT (K,UP,1,US,1,ICHAR,1)
001224          CALL LINCNT (60)
001226          WRITE (12,926)
001232          WRITE (12,900) LABEL
             C     CALCULATE RESIDUAL TEMPERATURES
001240        600 TX(1) = V0
001242            DO 601 I = 1,K
001243        605 CALL LFB(VC,F,TX)
001246            IF (TX(10)) 602,603,604
001250        604 F = -VC + V0 + V0*3,*ALPHA*(T(I)*EXP(B*(PV(I)-VC))-TO)
001265            GO TO 605
```

```
      C     USE US FOR RESIDUAL VOLUME AND UP FOR RESIDUAL TEMPERATURE
001265   603  US(I) = VC
001267        UP(I)=T(I) *EXP(B*(PV(I)-VC))
001276   601  CONTINUE
001300        GO TO 700
001301   602  PRINT 930
001305        GO TO 701
001306   700  PRINT 931
001312        PRINT 900,LABEL
001320   701  PRINT 932
001324        DO 702 I = 1,K
001326        PRINT 914,PV(I),PP(I),T(I),US(I),UP(I)
001343   702  CONTINUE
001346        WRITE (12,931)
001351        WRITE (12,900) LABEL
001357        WRITE (12,932)
001363        DO 703 I=1,K
001365        WRITE (12,914) PV(I),PP(I),T(I),US(I),UP(I)
001402   703  CONTINUE
001405        IF(ATWT.LT.0.1) GO TO 366
      C     COWAN FIT
001407    100 DO 101 I=1,K
001411        TV(I)= T(I)/DKV
001413        CRHO(I)= 1./PV(I)
001415    101 CONTINUE
001417        A2=1.447404/ATWT
001421        A1=-A2*RHO +3.0*ALPHA*DKV/AK
001426        DO 102 I=1,K
001430        ZP(I)= PP(I)-(A1+A2*CRHO(I))*TV(I) -(AC1+AC2/CRHO(I)+AC3*AC3/
             1(CRHO(I)*CRHO(I)))*TV(I)*TV(I)
001446    102 CONTINUE
001450        CALL PFTS (K,4.0,SIGMA,CRHO,ZP,W,FITT,DELY,CCOEF,SB,TZ,ST,A)
001466        PRINT 917
001472        PRINT 918,ATWT
001500        PRINT 919, (CCOEF(I),I=1,5)
001512        PRINT 920, A1,A2
001522        PRINT 921, AC1,AC2,AC3
001534        PRINT 922
001540        DO 103 I=1,K
001542        PRINT 914 ,ZP(I),FITT(I)
001551    103 CONTINUE
001554        WRITE (12,917)
001557        WRITE (12,918) ATWT
001565        WRITE (12,919) (CCOFF(I),I=1,5)
001577        WRITE (12,920) A1,A2
001607        WRITE (12,921) AC1,AC2,AC3
001621        WRITE (12,922)
001625        DO 104  I=1,K
001627        WRITE (12,914) ZP(I),FITT(I)
001636    104 CONTINUE
001641        GO TO 366
001641    900 FORMAT (12A6)
001641    901 FORMAT (4E18.11)
001641    902  FORMAT (41H1 SOLID EQUATION OF STATE CALCULATION FOR,12A6)
001641    903 FORMAT (/, 6H US = ,1PE18.11,3H + ,1PE18.11,17H S   FROM  PO TO ,
             11PE18.11,10H MEGABARS )
001641    904 FORMAT (/, 6H FROM ,1PE18.11,13H MEGABARS TO ,1PE18.11,29H MEGABAR
             1S USED THE FIT  US = ,1PE18.11,3H + ,1PE18.11, 3H S )
001641    905 FORMAT (/,24H THE INITIAL DENSITY IS ,1PE18.11, 7H GM/CC )
001641    906 FORMAT (/,24H THE COMPRESSIBILITY IS ,1PE18.11)
001641    907 FORMAT (/,40H THE LINEAR COEFFICIENT OF EXPANSION IS ,1PE18.11)
001641    908 FORMAT (/,27H THE INITIAL TEMPERATURE IS,1PE18.11)
```

```
001641      909 FORMAT (/,22H THE HEAT CAPACITY IS ,1PE18.11)
001641      910 FORMAT (/,25H THE VOLUME INCREMENT IS ,1PE18.11)
001641      911 FORMAT (/,32H THE TEMPERATURE FIT IS BETWEEN ,1PE18.11, 5H AND ,
               11PE18.11,10H MEGABARS )
001641      912 FORMAT (/,8H LN(T)= ,1PE18.11,3X,1PE18.11,5HLNV  ,1PE18.11,6HLNV**2
               1 ,1PE18.11,6HLNV**3 ,1PE18.11,5HLNV**4)
001641      913 FORMAT (//,110H      VOLUME IN CC/GM        PRESSURE IN MEGABARS  TE
               1MPERATURE DEG K      SHOCK VELOCITY        PARTICLE VELOCITY)
001641      914 FORMAT (5(4X,1PE18.11))
001641      915 FORMAT (   66H1      VOLUME IN CC/GM        INPUT TEMPERATURE     FI
               1T TEMPERATURE )
001641      917 FORMAT (41H1 THE COWAN EQUATION OF STATE COEFFICIENTS )
001641      918 FORMAT (/,21H THE ATOMIC WEIGHT IS,1PE18.11)
001641      919 FORMAT (/,6H  P = ,1PE18.11,3X,1PE18.11,5HRHO  ,1PE18.11,6HRHO**2 ,
               11PE18.11,6HRHO**3 ,1PE18.11,6HRHO**4 )
001641      920 FORMAT (/,5H    ( ,1PE18.11,3X,1PE18.11,23H RHO ) TEMPERATURE     )
001641      921 FORMAT (/,5H    ( ,1PE18.11,3X,1PE18.11, 6H /RHO ,1PE18.11,24H/RHO**
               12 )*TEMPERATURE**2  )
001641      922 FORMAT (//,40H       PRESSURE CALC         FIT PRESSURE )
001641      923 FORMAT (    40H  PRESSURE-VOLUME  HUGONIOT  FOR          )
001641      924 FORMAT (    40H  TEMPERATURE- VOLUME HUGONIOT FOR        )
001641      925 FORMAT (    50H  PRESSURE -PARTICLE VELOCITY HUGONIOT FOR          )
001641      926 FORMAT (    50H  SHOCK VELOCITY-PARTICLE VELOCITY HUGONIOT FOR    )
001641     930  FORMAT(44H1 AN ERROR OCCURRED IN RESIDUAL TEMP CALC    )
001641     931  FORMAT(44H1 RESIDUAL TEMPERATURE AND DENSITIES FOR     )
001641      932 FORMAT (//,114H      HUGONIOT VOLUME        HUGONIOT PRESSURE     HU
               1GONIOT TEMPERATURE  RESIDUAL VOLUME        RESIDUAL TEMPERATURE )
001641          END
```

```
                SUBROUTINE LFB (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
000005          DIMENSION TX(10)
000005          DATA CNTMAX /1000./
000005          IF (TX(10).LE.0.) GO TO 1
000006          TX(10)=TX(10)+1.
000011          IF (TX(10)-3.) 2,3,4
              C ENTRY FIRST TIME THROUGH
000014        1 IF (TX(1).EQ.0.) TX(1)=1.
000016          TX(10)=1.
000020          XP=TX(1)
              C GO GET F(XP)
000021          RETURN
              C ENTRY SECOND TIME THROUGH
000021        2 TX(9)=FP
000023          TX(8)=XP
000024          TX(5)=FP
000025          IF (ABS(FP).LT.TX(3)) GO TO 18
000027          XP=TX(1)*TX(2)
              C GO GET F(XP)
000030          RETURN
              C ENTRY THIRD TIME THROUGH
000031        3 TX(5)=FP
000033          TX(6)=XP
000034          TX(7)=FP
000035          IF (ABS(FP).LT.TX(3)) GO TO 18
000037          XP=TX(6)-TX(7)*(TX(6)-TX(8))/(TX(7)-TX(9))
              C GO GET F(XP)
000046          RETURN
              C ENTRY FOR FOURTH AND SUCEEDING TIMES THROUGH
000047        4 IF (TX(10).GT.CNTMAX) GO TO 99
000053          TX(4)=XP
000054          TX(5)=FP
000055          T=TX(4)-TX(6)
000057          IF (T.EQ.0.) GO TO 99
000060          IF (ABS(FP).LT.TX(3)) GO TO 18
000062          R=TX(5)-TX(7)
000064          IF (R.EQ.0.) GO TO 99
000065          XP=TX(4)-TX(5)*(T/R)
000071          IF (TX(5)*TX(7).LT.0.) GO TO 11
000073          IF (TX(5)*TX(9).GE.0.) GO TO 11
000075          IF (XP.GT.TX(4)) GO TO 6
000101          IF (XP.GT.TX(8)) GO TO 10
000104        8 XP=TX(4)-TX(5)*(TX(4)-TX(8))/(TX(5)-TX(9))
000113       10 TX(7)=TX(5)
000115          TX(6)=TX(4)
              C GO GET F(XP)
000116          RETURN
000117        6 IF (XP.GT.TX(8)) GO TO 8
000123          GO TO 10
```

```
000123        11 TX(9)=TX(7)
000125           TX(8)=TX(6)
000126           GO TO 10
              C  HAVE FOUND A SOLUTION
000127        18 TX(10)=0.
000130           TX(1)=XP
000131           TX(4)=XP
000132           RETURN
              C  AN ERROR HAS OCCURED
              C  SET COUNT NEGATIVE AND EXIT
000133        99 TX(10)=-TX(10)
000134           RETURN
000135           END
```

```
                SUBROUTINE   PFTS(M,KM,IW,SIGMA,X,F2,W,Y,DELY,B,SB,T,ST,A)
          C     THIS IS A SPECIAL VERSION OF LA-PFTS FOR A MAX OF 500 POINTS
          C     AND A MAX OF FIFTH DEGREE FIT  NO PRINTING  OR LEGENDRE POLYNOMIALS
          C     ALSO WILL COMPUTE TILL GET THE FIT AND REQUIRES NO OUTPUT SUBROUTINE
          C     M = NO OF DATA POINTS
          C     KM = DEGREE OF FIT  (MAX OF 5)
          C     IW = 0 FOR NO  WEIGHTS AND EQUAL 1 FOR WEIGHTS
          C     SIGMA       IS STANDARD DEVIATION COMPUTED
          C     X = X DATA       INPUT ARRAY
          C     F2 = F2 DATA ARRAY  INPUT
          C     W = WEIGHT DATA ARRAY  INPUT    SET EQUAL TO 1 IF IW IS 0
          C     Y  =  F COMPUTED FROM FIT USING X
          C     DELY = DIFFERENCE IN COMPUTED AND INPUT F
          C     B = COEFFICIENTS TO FIT
          C     SB = ESTIMATE OF ERRORS IN COEFFICIENTS TO FIT
          C     T = COEFFICIENTS TO ORTHOGONAL POLYNOMINALS
          C     ST = ERRORS IN T
          C     A = AREA USED BY CALCULATION
000017            DIMENSION   S(5),X(500),F2(500),ST(5),SB(5),F(500),PM(500),P(500)
                 1 ,B(5),DELY(500),W(500),A(5,5),T(5),Y(500)
000017            LL=0
000020          9 FM=0.0
000021            A(1,1)=1.0
000024            A(2,2)=1.0
000026            FBAR=0.0
000027            XBAR=0.0
000030            DO10I=1,M
000031            IF(IW)1009,1010,1009
000032       1010 W2=1.0
000034            W(I)=1.0
000037            GOTO1011
000040       1009 W2=SQRT(W(I))
000051       1011 FM=FM+W(I)
000055            F(I)=W2*F2(I)
000061            PM(I) = W2
000064            FBAR=FBAR+F(I)*PM(I)
000071         10 XBAR=XBAR+X(I)*PM(I)**2
000077            XBAR=XBAR/FM
000100            T(1)=FBAR/FM
000103            A(2,1)=-XBAR
000106            PXF=0.0
000107            PXP=0.0
000110            DO20I=1,M
000111            P(I)=(X(I)-XBAR)*PM(I)
000117            PXF=PXF+P(I)*F(I)
000124         20 PXP=PXP+P(I)*P(I)
000131            T(2)=PXF/PXP
000134            PMXPM=FM
000136            S(1)=PMXPM
000137            KM=KM+1
000140            B(1)=T(1)*A(1,1)+T(2)*A(2,1)
000147            B(2)=T(2)*A(2,2)
000154         60 DO190K=2,KM
000156            IF(K-2)40,165,65
000160         40 STOP
000162         65 XPXP=0.0
000163            XPXPM=0.0
000164            B(K)=0.0
000167            DO70J=1,M
000171            XP=X(J)*P(J)
000175            XPXP=XPXP+XP*P(J)
```

```
000200      70 XPXPM=XPXPM+XP*PM(J)
000205         ALPHA=XPXP/PXP
000207         BETA=XPXPM/PMXPM
000211         PPXF=0.0
000212         PPXPP=0.0
000213         DO90I=1,M
000214      80 PT=P(I)
000217      81 P(I)=X(I)*PT-ALPHA*PT-BETA*PM(I)
000230      82 PPXF=PPXF+P(I)*F(I)
000235      83 PPXPP=PPXPP+P(I)*P(I)
000241      90 PM(I)=PT
000246         T(K)=PPXF/PPXPP
000252         PMXPM=PXP
000254         PXP=PPXPP
000254         A(K,1)=-ALPHA*A(K-1,1)-BETA*A(K-2,1)
000264         A(K,K-1)=A(K-1,K-2)-A(K-1,K-1)*ALPHA
000277         A(K,K)=1.0
000303         IF(K-3)150,150,110
000305     110 K1=K-2
000307         DO120I=2,K1
000311     120 A(K,I)=A(K-1,I-1)-ALPHA*A(K-1,I)-BETA*A(K-2,I)
000344     150 DO160I=1,K
000346     160 B(I)=B(I)+T(K)*A(K,I)
000366     165 SIG2=0.0
000367         DO180I=1,M
000371         Y(I)=POLY (X(I),K,B)
000406     175 DELY(I)=Y(I)-F2(I)
000415     180 SIG2=SIG2+(DELY(I)**2)*W(I)
000426         SIG2=SIG2/FLOAT(M-K)
000430         SIGMA=SQRT(SIG2)
000436         S(K) = PXP
000442         DO499I=1,K
000444     499 ST(I)=SIGMA/SQRT(S(I))
000462         DO501I=1,K
000464         SB(I)=0.0
000467         DO500J=I,K
000470     500 SB(I)=SB(I)+(A(J,I)*ST(J))**2
000512     501 SB(I)=SQRT(SB(I))
000525     190 CONTINUE
000527     220 KM=KM-1
000531         RETURN
000531          END
```

```
                FUNCTION POLY  (X,N,A)
000005          DIMENSIONA(2)
000005          Y=A(N)
000007          DO1I=2,N
000010          J=N-I+1
000012        1 Y=A(J)+Y*X
000020          POLY   =Y
000021          RETURN
000022          END
```

ACKNOWLEDGMENTS

It is a pleasure to acknowledge the assistance of and discussions through the years with Frank Harlow of T-3; George N. White, Jr., and P. J. Blewett of T-5; W. Fickett of GMX-10; and, especially, Louis C. Smith of GMX-2.

REFERENCES

1. Mader, Charles L., "The Hydrodynamic Hot Spot and Shock Initiation of Homogeneous Explosives," Los Alamos Scientific Laboratory Report LA-2703, 1962.
2. Mader, Charles L., "STRETCH SIN - A Code for Computing One-Dimensional Reactive Hydrodynamic Problems," Los Alamos Scientific Laboratory Report LADC-5795, 1963.
3. Mader, Charles L., "A Study of the One-Dimensional Time-Dependent Reaction Zone of Nitromethane and Liquid TNT," Los Alamos Scientific Laboratory Report LA-3297, 1965.
4. Mader, Charles L., "An Equation of State for Iron Assuming an Instantaneous Phase Change," Los Alamos Scientific Laboratory Report LA-3599, 1966.
5. Mader, Charles L., "One-Dimensional Elastic-Plastic Calculations for Aluminum," Los Alamos Scientific Laboratory Report LA-3678, 1967.
6. Mader, Charles L., "GMX-2 Spalling Calculations," Los Alamos Scientific Laboratory Report LADC-7692, 1965.
7. Breed, B. R., Mader, Charles L., and Venable, Douglas, J. Appl. Phys., 38, 3271 (1967).
8. Mader, Charles L., "FORTRAN BKW - A Code for Computing the Detonation Properties of Explosives," Los Alamos Scientific Laboratory Report LA-3704, 1967.
9. Walsh, John M., and Christian, Russell H., Phys. Rev., 97, 1554 (1955).