

SANDIA REPORT

SAND88-0831 • UC-32

Unlimited Release

Printed June 1988

RS-8232-2167436

cy1



8232-21/067436



0000001 -

A Tabular Equation of State Option for the WONDY Code

G. I. Kerley

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors or subcontractors.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND88-0831
Unlimited Release
Printed June 1988

A Tabular Equation of State Option for the WONDY Code

G. I. Kerley
Computational Physics and Mechanics Division II
Solid Dynamics Department

ABSTRACT

This report discusses a tabular equation of state (EOS) package that has been developed for the one-dimensional wave propagation code WONDY VII. The new option, STAT7, accesses EOS data from a library having the Sesame format. STAT7 also offers an optional two-state feature that can be used to set up a porous material model, a low pressure phase transition, or a metastable initial state (as in a reactive material). In addition, it includes the elastic-plastic work hardening model from STAT1. User input is similar to that for the other WONDY material models, and installation of the update set is easily accomplished using the UPDATER procedure. Three sample problems are discussed - electron beam deposition in an aluminum plate, an explosive detonation inside a brass shell, and shock vaporization of an aluminum plate.

Contents

1	Introduction	1
2	Implementation of the Tabular EOS	3
2.1	Structure of the Sesame Library	3
2.2	Interpolation Method	3
2.3	Solution of the Energy Balance Equation	4
2.4	Units	5
2.5	Storage Requirements and Computing Time	5
3	Input and Use of the Two-State Model	7
3.1	General	7
3.2	Definition of Input Variables	7
3.3	Analytic EOS	9
3.4	Crush Region	10
3.5	Types of Transitions	10
3.6	Scaling the Density and Energy	11
4	Examples	13
4.1	General Procedures	13
4.2	Electron Beam Deposition in Aluminum	13
4.3	Explosive Charge Inside a Brass Shell	16
4.4	Aluminum Shock Vaporization	19
5	Acknowledgements	22
6	References	23
	APPENDIX - EOS Updates for WONDY VII	25

1 Introduction

WONDY is a Lagrangian finite difference code for calculating wave propagation in one spatial dimension (planar, cylindrical, or spherical)^{1,2}. It provides a variety of initial and boundary conditions, rezoning features, and options for treating voids that make it suitable for a wide range of problems. As with all wavecodes, the accuracy and usefulness of its predictions depend upon the validity of the material constitutive models that are used. WONDY offers a number of options that are not available in many codes, such as variable yield strength, and internal state variable treatments of porosity and damage. It is also designed to be used as a research tool, allowing users to develop and install new material models.

However, WONDY provides only simple equations of state (EOS) as standard options. The principal option is the Mie-Grüneisen EOS, an approximation having serious limitations³. For example, it gives poor results for the expanded liquid and vapor regions, for certain multiply reshocked or adiabatically compressed states, for high temperatures where electronic excitations occur, and for many cases involving phase transitions or chemical reactions. A version of WONDY that uses the analytic EOS (ANEOS) of Thompson and Lauson⁴ is also available for special purpose calculations⁵.

The Sesame library was developed to meet the need for EOS data having a standard, general, tabular format that is suitable for use in wave propagation codes⁶⁻¹⁰. The Sesame tables can be used to represent EOS surfaces of arbitrary complexity. Therefore, extensive theoretical calculations can be made and saved, and it is possible to use sophisticated models that cannot be built into any analytical EOS option. The use of tables also offers a simple way to build in the capability for accessing the wide variety of EOS models that have been developed in recent years³.

However, the use of tabular EOS also has certain difficulties and limitations. For example, variations in density and other properties can occur because of porosity or differences in chemical composition. These variations are more easily included in an analytic model, where the input parameters can be varied, than in a tabular EOS which has been precomputed. Furthermore, many classes of materials do not exhibit the ideal equilibrium behavior that is normally assumed in most EOS theories. In particular,

models of reactive and other nonequilibrium phenomena normally require *two* EOS models, one for the initial state and one for the final state; the material is treated as a mixture of the two, using an internal state variable to keep track of the composition.

The STAT7 tabular EOS package described in this report has been designed to eliminate these common problems. We have adopted the philosophy that the tabular EOS represents the “final state”. An optional analytic EOS, with parameters defined by the user, can also be defined to represent an “initial state”. An internal state variable is also provided. This two-state model permits treatment of a wide variety of problems, including porosity, phase transitions, and material reactivity. The models used for material strength, fracture, and energy deposition are taken from the STAT1 subroutine in WONDY. Hence STAT7 is identical to the STAT1 model, except that the Mie-Grüneisen EOS has been replaced by the new option.

The methods used to implement use of the tabular EOS option in WONDY are described in Sec. 2. User input and details concerning the two-state feature are discussed in Sec. 3. Sample problems are given in Sec. 4. The STAT7 model is presently available as an update set for the Cray-CTSS version of the code, WONDY VII²; it is most easily installed using the CCL procedure UPDATER (from SLTLIB), as illustrated in Sec. 4.1. The update set is available on IFS (file WONSES, under node /E00024305/HYDR/WONDY/UPDATES). A listing of the update set is also given in the Appendix.

Tables of EOS data for use in calculations can be obtained from a variety of sources (Section 2.1). Unfortunately, many of these tables are not well documented and evaluated for general use. Therefore, the EOS file must be created for each specific problem until a permanent library of data can be established.

2 Implementation of the Tabular EOS

2.1 Structure of the Sesame Library

The structure of the Sesame library is given in Refs. 6-11, and we will discuss only a few points here. The Sesame library consists of one or more files, each file containing EOS data tables for one or more materials. These files can be either sequential or random; for use by WONDY and other Sandia codes, we have decided to employ the sequential structure, thereby eliminating calls to system-dependent I/O routines. The data on these files can be generated using codes such as PANDA¹¹ or ANEOS⁴, or it can be obtained from the Los Alamos library or from other laboratories. We will assist users in identifying appropriate sources of data and in creating the data files that are needed for their specific applications. If there is sufficient interest, an EOS library will be created, documented, and maintained on IFS, for general use at SNLA.

EOS data in the library are identified by *material number* and by *data type*. Two types of data records are required for the present purpose. In the main WONDY calculation, the EOS and energy balance equation are solved simultaneously (Section 2.3). The EOS is obtained from the *301 table*, which gives pressure and internal energy as functions of density and temperature. In addition, the initial density, energy, and pressure must be specified for each material in the calculation. Along with the EOS, the *201 table* gives a reference state that is used to define the initial conditions in cases where this information is not provided by the user.

In order to use the Sesame option, the user must first obtain the library file containing the data and name it "sesame". The material number for each plate designated STAT7 is provided as one the EOS constants (records 15-19), along with data for the initial state, strength, and fracture models. Details of the setup are given in Section 3.

2.2 Interpolation Method

Because the WONDY difference equations do not make use of the temperature variable, the 301 EOS tables are "inverted", making new tables of pressure and temperature as functions of density and internal energy (a Sesame 302 data type). The

setup is carried out in routine SESLIB (see the Appendix). The 201 and 301 tables are obtained from the library file, loaded into a local array with pointers to the data, and inverted (routine INV302). In the present updates, the temperature calculations are omitted to reduce computing time and storage requirements. If the temperature information is needed for plotting purposes or for a particular model, only minor changes to the update set are required.

The default method of interpolating on the EOS tables is the rational function algorithm described in Ref. 12. This method gives good values for derivatives of the function, including cases where the derivatives are discontinuous, as can occur in the vicinity of phase transitions. In problems where computing time becomes a limiting factor, it may be necessary to employ a faster but less accurate bilinear interpolation method. This option can be obtained by setting the parameter IFN=1 in COMMON block /RTBLK2/. (This change should be made in routine TB302.)

2.3 Solution of the Energy Balance Equation

In WONDY, the difference equation that is used to advance the energy in a given spatial zone from time t_n to time t_{n+1} can be written in the following form (see Eq. 2.46 of Ref. 1).

$$E_{n+1} = c_1 P(\rho_{n+1}, E_{n+1}) + c_2, \quad (1)$$

where

$$c_1 = 2(\rho_{n+1} - \rho_n) / (\rho_{n+1} + \rho_n)^2, \quad (2)$$

and

$$c_2 = E_n + c_1 [P(\rho_n, E_n) + 2q_{n+1/2}] + \Delta E^d + \Delta Q. \quad (3)$$

Here ΔE^d is the deviatoric (material strength) contribution to the energy, ΔQ is the energy deposition, and $q_{n+1/2}$ is the average artificial viscosity during the time step. These three terms are identical to those for STAT1; only the hydrodynamic relation $P(\rho, E)$ has been changed in STAT7.

Equation 1 must be solved for E_{n+1} , all other quantities being known. In STAT1, which uses an analytic form for $P(\rho, E)$, this equation is solved in closed form. In STAT7, we use a Newton-Raphson iteration that is designed to minimize the number

of times the EOS routines must be called. Let E_{n+1}^i be the estimate of E_{n+1} at the i -th iteration. A new estimate is computed from

$$E_{n+1}^{i+1} - E_{n+1}^i = \frac{c_1 P(\rho_{n+1}, E_{n+1}^i) + c_2 - E_{n+1}^i}{1 - c_1 (\partial P / \partial E)_\rho}, \quad (4)$$

where the derivative, evaluated at ρ_{n+1} and E_{n+1}^i , is supplied by the interpolation routine. The initial estimate of the energy is computed from the isentropic approximation,

$$E_{n+1}^1 = c_1 [P_n + C_s^2 (\rho_{n+1} - \rho_n)] + c_2, \quad (5)$$

where C_s is the sound speed from the previous time step t_n . The iteration is assumed to be converged if

$$|E_{n+1}^{i+1} - E_{n+1}^i| \leq 10^{-6} (|E_n| + 10^3 \text{ J/kg}). \quad (6)$$

An error message is printed on the output file in cases where the iteration does not converge. In test calculations, we found that the method required an average of about 1.3 calls to the EOS routines per zone per time step.

As discussed in Sec. 3.5, a history variable is used to keep track of the material state in certain models. To prevent an artificial change of state from occurring during the iteration, the value from the previous time step is always passed to the EOS routines, and the value returned by the routines is permanently saved at the end of the iteration.

2.4 Units

For consistency with typical usage of WONDY, the new EOS option uses mks units (length in m, time in s, density in kg/m^3 , stress in N/m^2 (Pa), and energy in J/kg).

2.5 Storage Requirements and Computing Time

The update set given in the Appendix allocates a 25000-word storage array for the Sesame tables. The amount of storage actually required for a given problem depends upon the number of plates that are designated STAT7 and upon the size of the EOS tables that are used. Sesame tables can vary in size; they typically range from 5000 to 7500 words but can be much larger. The EOS routine reads the entire table into

the storage array but overwrites the temperature table to save space. If the same EOS table is requested for more than one plate, the routines will load and use only one copy of that table.

An error message is written to the output file if the allocated storage is insufficient. The maximum array dimension MXTB can be changed easily, by modifying the PARAMETER statements in routines SESLIB, S2EOSI, S2INIT, and TB302.

Timing studies on the test problems described in Section 4 show that computing time increases by a factor of about five when STAT1 is replaced by STAT7 for all materials. (Of course, the tabular option can be used along with any of the other EOS options.) Most WONDY problems are relatively inexpensive, and a significant improvement in the EOS will be worth this extra cost in most cases. However, techniques for speeding up the tabular EOS computations have been developed elsewhere^{8,9}; use of these and other methods could be explored in future work.

3 Input and Use of the Two-State Model

3.1 General

Ideally, an adequate treatment of equilibrium melting, vaporization, solid-solid transitions, and chemical reactions will be included in the model used to construct the tabular EOS. In many cases, therefore, the only input variable required for STAT7 will be the material number for the Sesame EOS table (see Section 3.2). However, STAT7 offers a two-state model that is useful in a variety of problems, particularly those involving *nonequilibrium* phenomena. In such cases, an analytic EOS describes the initial state, the tabular EOS describes the final state, and an internal state variable is used to keep track of the state. In addition, STAT7 allows the user to scale the density and energy to account for small variations in chemical composition.

The analytical model used in STAT7 is essentially a simplified version of the Mie-Grüneisen EOS that is used in STAT1, with input parameters taken from the Hugoniot. However, we have developed a new form of the equations that is readily generalized for use in codes requiring the temperature variable. These equations will be discussed more fully in a separate report.

3.2 Definition of Input Variables

As with the other WONDY material models, the EOS constants are entered on records 15-19 and saved internally in the array CES (common block /CON/). In STAT7, records 18 and 19 are used for defining the shear modulus and yield strength, respectively, and are identical to those for STAT1. (See Ref. 1, p. 205.) The other constants are defined below. All can be defaulted by a zero or blank field except for the Sesame material number IDS2.

Variable	Symbol	Definition	Default
CES(1)	ρ_0	initial density	from 201 table
CES(2)	C_0	initial sound speed	from 302 table
CES(3)	IDS2	Sesame material number	none
CES(4)	S	slope of U_s - U_p curve	$S = 0$
CES(5)	γ_0	Grüneisen parameter	$\gamma_0 = 0$
CES(6)	ν	Poisson's ratio (for strength model)	$\nu = 0$
CES(7)	S_R	density and energy scaling factor	$S_R = 1$
CES(8)	P_0	initial pressure	$P_0 = 0$
CES(9)	β_0	bulk modulus (computed internally)	$\beta_0 = \rho_0 C_0^2$
CES10)	E_0	initial energy	$E_0 = 0$
CES(11)	P_T	crush pressure for initial state	$P_T = .01\rho_0 C_0^2$
CES(12)	β_T	isothermal modulus in crush region	$\beta_T = 0$
CES(13)	ρ_{min}	minimum density for initial state	$\rho_{min} = .75\rho_0$
CES(14)	E_{max}	maximum energy for initial state	$E_{max} = 0$
CES(15)	TYP	type of phase transition model	TYP=0
CES(16)	ρ_{max}	maximum density for initial state	$\rho_{max} = 2\rho_0$

Constants CES(17)-CES(21) are not currently used by STAT7.

The following tips should help users to decide which parameters to define and which to default. (A reminder - mks units are used for all variables.)

- The material number IDS2 is always required. Also specify Poisson's ratio ν when using the material strength model.
- Specify ρ_0 , P_0 , and E_0 when the initial conditions differ from the standard reference state of the material, given in the 201 table as described in Sec. 2.1 and in reference 11. (If $P_0 \neq 0$ or $E_0 \neq 0$, it is also necessary to enter the initial conditions using record 12, the same as for STAT1.)
- In addition, specify γ_0 , C_0 , S , P_T , β_T , ρ_{min} , and E_{max} when the initial state of the material is actually different from that described by the tabular EOS. If $\gamma_0 = 0$, the routines will choose $\beta_T = 0$ regardless of user input; therefore, it is best to enter a finite value for γ_0 when using the two-state option. Also note that a default value is set for P_T when $P_T \leq 0$ on input; the reason for this fact is to insure an upper density limit for the initial state.
- Set parameter TYP to select the transition mechanism for the two-state model, as discussed in Sec. 3.5, below. Also specify parameter ρ_{max} for TYP=2.

- Set parameter S_R to scale the density and energy, as discussed in Sec. 3.6, below.

3.3 Analytic EOS

The EOS used to describe the initial state of the material has the form

$$P(\rho, E) = P_R(\rho) + \gamma_0 \rho_0 [E - E_R(\rho)], \quad (7)$$

where P_R and E_R are the pressure and internal energy along an isentrope passing through the initial density and energy. The energy and pressure on the isentrope are given by

$$E_R(\rho) = E_0 + (\eta/\rho_0)[P_0 + a_1 \eta \exp(a_2 \eta + a_3 \eta^2 + a_4 \eta^3)], \quad (8)$$

$$P_R(\rho) = P_0 + (2 + a_2 \eta + 2a_3 \eta^2 + 3a_4 \eta^3)a_1 \eta \exp(a_2 \eta + a_3 \eta^2 + a_4 \eta^3), \quad (9)$$

where

$$\eta = 1 - \rho_0/\rho. \quad (10)$$

The constants $a_1 - a_4$ in Eqs. 8 and 9 are computed by fitting Eq. 7 to the Hugoniot. Assuming a linear relation between shock velocity U_s and particle velocity U_p ,

$$U_s = C_0 + S U_p, \quad (11)$$

it can be shown that

$$a_1 = \rho_0 C_0^2 / 2, \quad (12)$$

$$a_2 = 4S/3, \quad (13)$$

$$a_3 = S(1.5S - \gamma_0/6) - a_2^2/2, \quad (14)$$

and

$$a_4 = S(1.6S^2 - 0.3\gamma_0 S - \gamma_0^2/30) - a_2(a_3 + a_2^2/6). \quad (15)$$

This form of the EOS, which will be derived in a separate report, accurately represents the Hugoniot for most materials to pressures at least as high as 100 GPa. Unlike the EOS used in STAT1, which uses the Hugoniot as the reference curve, Eq. 7 does not give unphysical results at high densities, above the Hugoniot density asymptote. It also gives reasonable results in mild tension, for use with fracture models. Finally, it is easily generalized for use in wave codes that require the temperature variable.

3.4 Crush Region

In Eq. 9, the pressure is a monotonically increasing function of density. The STAT7 option also provides for a break in the EOS and a second region that describes the crushing or transformation to the final state. The transition begins above a density ρ_T , corresponding to a Hugoniot pressure P_T , and uses β_T as the isothermal bulk modulus in the transition region. The EOS is given by

$$P(\rho, E) = \rho_0[a_6 - \gamma_0(a_5 + a_6\eta)] + \gamma_0\rho_0E, \quad (16)$$

where

$$a_6 = P_R(\rho_T)/\rho_0 - \beta_T\rho_T/\gamma_0\rho_0^2, \quad (17)$$

and

$$a_5 = E_R(\rho_T) - a_6\eta_T - \beta_T\rho_T/\gamma_0^2\rho_0^2. \quad (18)$$

3.5 Types of Transitions

To complete the definition of the two-state model, it is necessary to prescribe how a material transforms between the initial and final states. First, a transition to the final state always occurs if $\rho < \rho_{min}$ or $E > E_{max}$. The density condition forces the material to vaporize upon expansion, while the energy condition is intended to account for melting or other thermally-induced transition mechanism. The parameter TYP is used to select additional mechanisms for the transition. The three standard options described below can be considered prototypes for designing other models.

The options that are specified by TYP=0 and TYP=1 are used when the initial state has a lower density than the final state and the transformation proceeds under pressure. This situation arises if the material undergoes a phase transition or if it is initially porous and undergoes void collapse; in defining the analytic EOS, parameters P_T and β_T are used to describe the partially transformed or crushed region, as discussed above. Let P_i and P_f describe the EOS for the initial and final states, respectively. During the first stages of the calculation, the pressure in each zone is computed from the initial state, $P(\rho, E) = P_i(\rho, E)$. As the calculation proceeds, a particular zone will transform to the final state when it becomes sufficiently compressed; this condition is

$$P(\rho, E) = P_f(\rho, E), \quad \text{if } 0 \leq P_i(\rho, E) \leq P_f(\rho, E). \quad (19)$$

(The lower limit in Eq. 19 is used so that the initial state can go into tension even when the tabular EOS does not have a tension region.)

These two options differ in only one respect. For TYP=0, the transition is reversible; the material can revert to the initial state after having transformed to the final state. For TYP=1, the transition is made irreversible through the use of a history variable, called VF in routine S2EOSI. Initially, VF=0 for all zones. For each zone, the code sets VF=1 when a transformation to the final state occurs, as in Eq. 19 or when $\rho \leq \rho_{min}$ or $E \geq E_{max}$. A zone with VF=1 is not allowed to revert to the initial state.

For TYP=2, the condition for transformation to the final state is

$$\rho \geq \rho_{max}. \quad (20)$$

As in TYP=1, the history variable VF is used to prevent the material from reverting to the initial state after transformation has occurred. This model is closely related to the so-called CJ volume burn that is often used for detonations¹³. For explosives, the initial state corresponds to a metastable condition (the unreacted explosive), with a *higher* density than the final state (the detonation products), so that the transition leads to expansion. In that case, the model will describe either an overdriven or underdriven detonation wave if ρ_{max} is set to the density at the CJ state corresponding to the initial density and energy of the explosive¹³. However, the model can also be used to describe a variety of other materials, including plastics, polymers, and organic liquids, which undergo chemical decomposition when shocked but do not detonate.

3.6 Scaling the Density and Energy

For metallic alloys, minerals, and other types of mixtures, variations in density can occur because of differences in chemical and isotopic composition. In such cases, the input parameter S_R can be used to scale the density and internal energy for the tabular EOS to those for the real material. This parameter can be defined in terms of either the average atomic weights A or the densities ρ , as follows.

$$S_R = \frac{A_{table}}{A_{real}} = \frac{\rho_{table}}{\rho_{real}}. \quad (21)$$

If $S_R \leq 0$ on input, the package resets $S_R = 1$. The scaling option should *not* be used if the density variation is due to porosity; in that case, the two-state option should be used to model the effects of void closure.

4 Examples

4.1 General Procedures

Sample problems for the new EOS option have been saved on IFS, under node /E00024305/HYDR/WONDY/TESTS. The file RUNTST, listed below, gives the CTSS-CCL commands to get files from IFS, update the code using the UPDATER procedure (from SLTLIB), run the code, and save the output files on IFS.

```
select savelog='temp'
! get update deck, EOS library, and input file from IFS
mass get wait-on dir=/e00024305/hydr/wondy/updates wonses
mass get wait-on dir=/e00024305/hydr/wondy/tests sesame
mass get wait-on dir=/e00024305/hydr/wondy/tests infile:inptpl
! run UPDATER to make executable file
sltlib
updater wondy7 wonses
switch uabs wondy7
! run WONDY problem
wondy7
-infile
! save files on IFS
log file=logxx
cconcat wonout logxx out
store wait-on dir=/e00024305/hydr/wondy/tests out:outtpl
store wait-on dir=/e00024305/hydr/wondy/tests pltfil:pltpl
```

Three sample problems are discussed below. Plots of the results from these calculations, shown in the figures, were obtained with the plotting module WNPLT7^{1,2}. Input files for obtaining the plots are not discussed here because they do not differ from those using other EOS options.

4.2 Electron Beam Deposition in Aluminum

Oswald, *et. al.*¹⁴ studied the dynamic response of 6061 aluminum exposed to a 185-keV pulsed electron beam. The e-beam energy is absorbed in a thin surface layer, creating high pressures and temperatures. This hot material expands, and a stress wave propagates through the material. The rear-surface velocity time history was measured for various incident fluences and plate thicknesses. Test problem #1 is a calculation of an experiment having an incident fluence of 36 cal/cm² and an aluminum thickness of 1.5 mm. The STAT7 option is used with an aluminum EOS developed by Kerley¹⁵

and also includes material strength and fracture. The input file, which is listed below, follows the standard WONDY format¹, and we will discuss only those features unique to this problem.

```

!
!           WONDY/SESAME TEST PROBLEM # 1
!           OSWALD AL E-BEAM EXPERIMENT
!           03/07/88
! 1.50 mm thick piece of aluminum, energy deposition using MORSTOR
! option. Al EOS of Kerley, with material strength and spall.
1  WONDY/SESAME TEST PROB 1 - AL E-BEAM EXPERIMENT
2  1 1 11 3 3 3000 1 1 20 1
3  1.0E-6 1.0E-9 -1.0E6 1.0E3
5  0.0E-6 .10E-6 1.00E-6
6  0.0E-6 0.5E-8 1.00E-6
10 1 7. 100. 1.5E-3 1.5E-5 1.5E-5
! Fracture criterion - maximum tensile stress failure
14 1 2.0 -1.4E9
! Aluminum EOS - tabular, with material strength
15 1 205 0.361
19 1 3.0 0.25e+09 12.0 7.0e+06
! MORSTOR cards - energy deposition in 106 ns. Multiply by
! 4.185E+04 to go from cal/cm**2 to Joule/m**2.
31 0.106E-06 1.507E+06
38 27 0.000e+00 2.880e+01 9.439e-06 2.799e+01 2.056e-05 2.694e+01
3.404e-05 2.549e+01 4.434e-05 2.428e+01 5.465e-05 2.307e+01 6.494e-05 2.170e+01
7.521e-05 2.016e+01 8.471e-05 1.888e+01 9.579e-05 1.742e+01 1.061e-04 1.573e+01
1.163e-04 1.420e+01 1.250e-04 1.283e+01 1.369e-04 1.106e+01 1.487e-04 9.131e+00
1.590e-04 7.601e+00 1.693e-04 6.071e+00 1.804e-04 4.940e+00 1.923e-04 3.968e+00
2.058e-04 2.914e+00 2.186e-04 2.181e+00 2.321e-04 1.528e+00 2.457e-04 9.543e-01
2.608e-04 6.191e-01 2.744e-04 2.856e-01 2.864e-04 3.360e-02 3.064e-04 0.000e+00

```

The only parameter specified for the Sesame EOS is the material number, given on record 15, the other input being defaulted. Poisson's ratio, needed for the material strength model, is also given on record 15; the value $\nu = .361$ was computed from sound speed data¹⁶. The material strength model, given on record 19, uses the following expression for yield strength Y ($\text{NOY}=3$)¹.

$$Y = Y_0(1 + y_1\eta)(1 - E/y_2). \quad (22)$$

An approximate fit to the data of Asay, *et. al.*¹⁷ was obtained using $Y_0 = 0.25 \times 10^9$ Pa, $y_1 = 12$, and $y_2 = 7.0 \times 10^6$ J/kg. The fracture model, specified on record 13, uses the maximum tensile stress criterion with a spall strength of 1.4×10^9 Pa; this value, which is slightly higher than that reported by Ek and Asay¹⁸, was chosen to match the late-time behavior of the velocity profile.

Deposition of the electron beam energy was modeled with the MORSTOR option¹. Input for this option includes the incident fluence and deposition time (record 31) and the dose-depth profile (record 38 and following cards), which is assumed to be

independent of time. A deposition time of 106 ns was obtained from Tokheim¹⁹, and the dose-depth curve was taken from Ref. 14.

The WONDY predictions are compared with experiment in Fig. 1. The agreement is fairly good. Both curves exhibit an elastic precursor, followed by a plastic wave that attenuates with increasing distance from the deposition surface. The calculated amplitude of the plastic wave is less than observed. Because this problem is presented mainly for illustrative purposes, we have not made a detailed sensitivity study of the model parameters. However, the energy deposition model is likely to be a major cause of the discrepancy. Tokheim has shown that the dose-depth profile is not time-independent¹⁹, and we have found that halving the deposition time doubles the amplitude of the second peak.

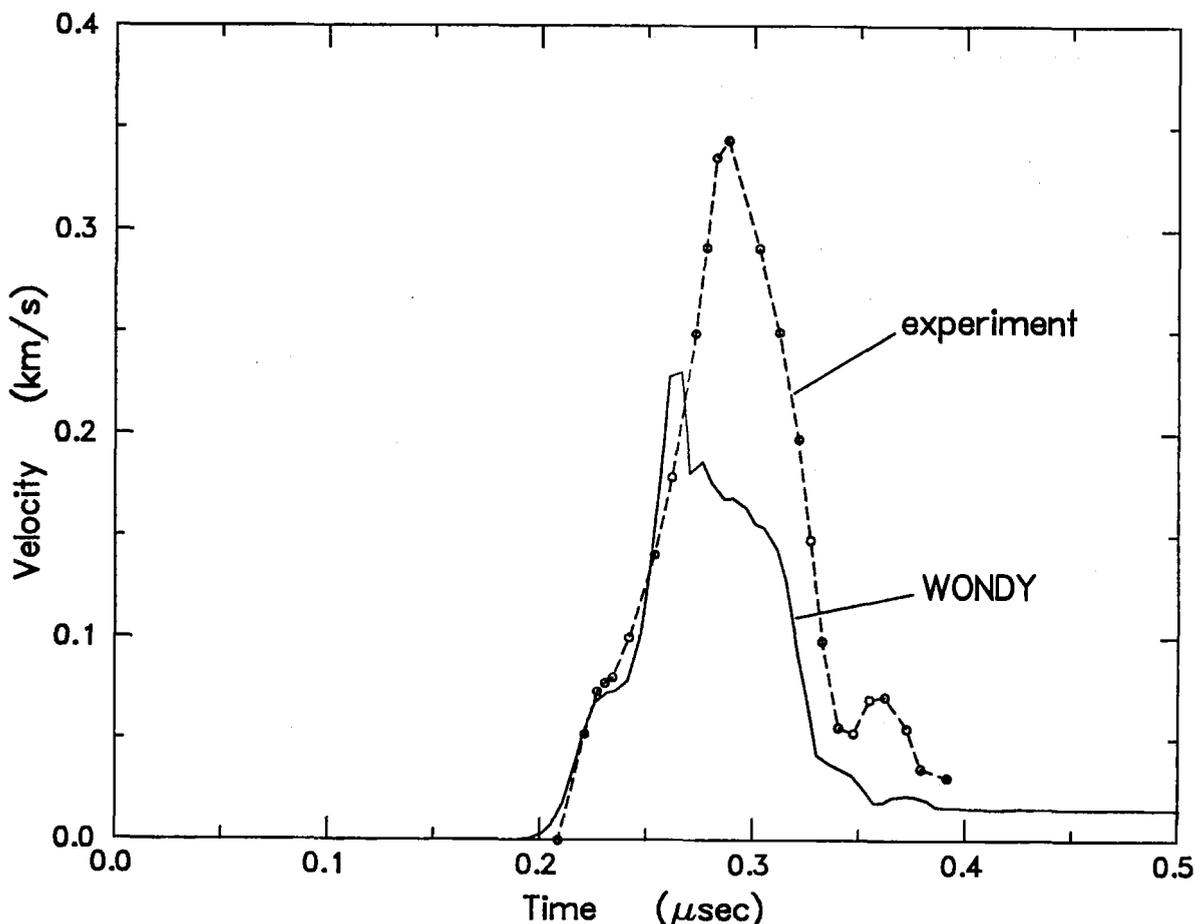


Figure 1: Rear-surface velocity for electron-beam experiment in aluminum, test problem # 1. Solid line is WONDY calculation and dashed line with circles is experimental¹⁴.

4.3 Explosive Charge Inside a Brass Shell

The second example consists of a spherical brass shell containing a charge of explosive that is detonated at its center. The input file is given below.

```

!                WONDY/SESAME TEST PROBLEM #2
!                PBX9404 IN BRASS SHELL
!                03/17/88
! 10 mm thick charge of PBX9404 inside a 10 mm thick brass shell.
! Uses tabular EOS with TYP=2 for explosive, TYP=0 for brass.
1  WONDY/SESAME TEST PROB 2 - PBX9404 IN BRASS SHELL
2      3      3      10      1      3 3000      1      1      15
3      1.0E-5      -1.0E6      1.0E6
5      0.0      1.0E-6      1.0
6      0.0      1.0E-8      1.0
10     1      7.      2.0      2.5E-4      1.25E-4      1.25E-4
10     2      7.      78.0      9.75E-3      1.25E-4      1.25E-4
10     3      7.      160.0      0.01      6.25E-5      6.25E-5
12     1      6.39E+6      8.0E+3      4.0E+3
12     2      6.39E+6
! PBX9404 tabular EOS - Sesame table 8200
15     1      1.84E+3      2.494E+3      8200      2.093      1.0      0      0
16     1      0      0      6.39E+6      40.E+9      0      0      8.0E+6
17     1      2.0      2.2E+3
! PBX9404 tabular EOS - Sesame table 8200
15     2      1.84E+3      2.494E+3      8200      2.093      1.0      0      0
16     2      0      0      6.39E+6      40.E+9      0      0      8.0E+6
17     2      2.0      2.2E+3
! Brass EOS - Sesame table 4100
15     3      4100      0.32
19     3      1.0      2.0E+8

```

This problem is similar to the one described in Appendix D of Ref. 1, except that it uses tabular EOS with TYP=2 for the explosive and TYP=0 for the brass. Sesame EOS 4100, from the Los Alamos data base¹⁰, is used for the brass. For the explosive, Sesame EOS 8200, for PBX-9502¹⁰, is used to describe the detonation products. (Although this table was not developed for the purpose of describing detonation wave propagation, it is satisfactory for the present purpose, to illustrate use of the TYP=2 option.) As discussed in Sec. 3, an analytic EOS is used to describe the unreacted explosive. The initial density and energy were chosen to be 1840 g/m³ and 6.39 × 10⁶ J/kg, respectively. Using these values, the following detonation parameters were obtained with the PANDA code¹¹:

$$\begin{aligned}
 D_{CJ} &= 8800 \text{ m/s} \\
 P_{CJ} &= 27.0 \times 10^9 \text{ Pa} \\
 \rho_{CJ} &= 2269 \text{ g/m}^3
 \end{aligned}$$

$$E_{CJ} = 7.78 \times 10^6 \text{ J/kg}$$

The density criterion for the transition to the final state was chosen as $\rho_{max} = 2200$, slightly below the CJ density. The other parameters for the model were taken from Ref 20.

The treatment of the detonation itself is significantly different from the STAT2 option that was used in the sample problem of Ref. 1. In that case, the explosive was treated by the "programmed burn" method; energy is deposited in the explosive cells as a function of time, forcing the detonation wave to propagate at the CJ velocity (specified by the user). In the present example, no such special treatment is needed; the explosive is not treated differently from an inert material. The detonation is initiated by assigning an initial velocity of 8000 km/s to the first two zones of the explosive. Thereafter, the usual momentum and energy transfer from the propagating shock front "burns" the explosive, resulting in a steady-state detonation. The detonation conditions are determined by the initial state of the unreacted explosive and by the detonation product EOS, not by user input. (However, the parameter ρ_{max} should be reasonably close to the CJ value.) Hence the model will automatically account for variations in the initial density or energy of the explosive. Moreover, this method will also work for overdriven detonations, where the programmed burn option fails.

Plots of the calculated stress vs. distance, shown in Fig. 2, are similar to those given in Ref 1. Fig. 2a, at 1 μs , shows the detonation wave in the explosive charge, with the peak pressure close to the CJ value computed with PANDA. Fig. 2b, at 2 μs , shows a shock wave propagating into the brass shell and a reflected shock traveling back into the explosive detonation products. (The difference in amplitudes for the shock in the brass between the present calculation and that in Ref. 1 arises from differences in the explosive model.) In Fig. 2c, at 3.5 μs , the reflected shock has reached the center of the sphere and a second reflection has occurred in the detonation products. In Fig. 2d, at 5.0 μs , a rarefaction wave from the free surface has propagated back into the brass shell.

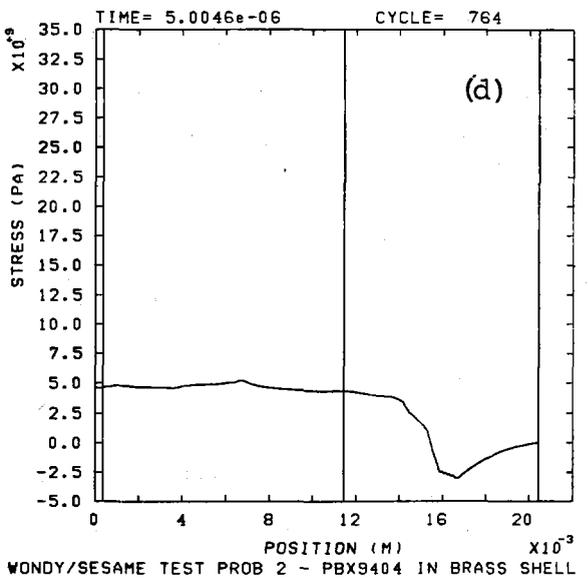
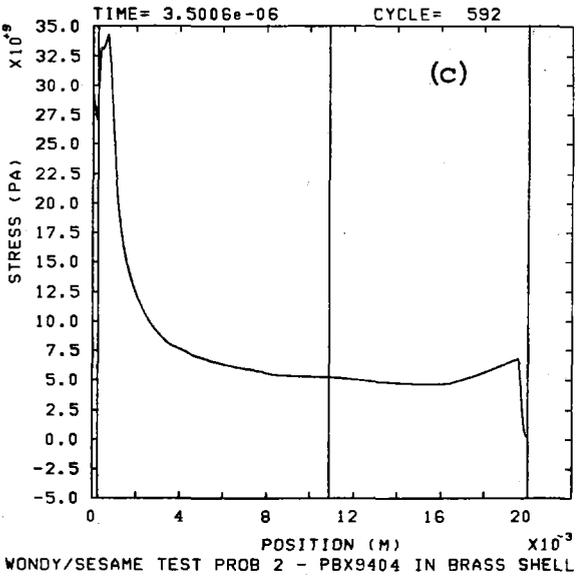
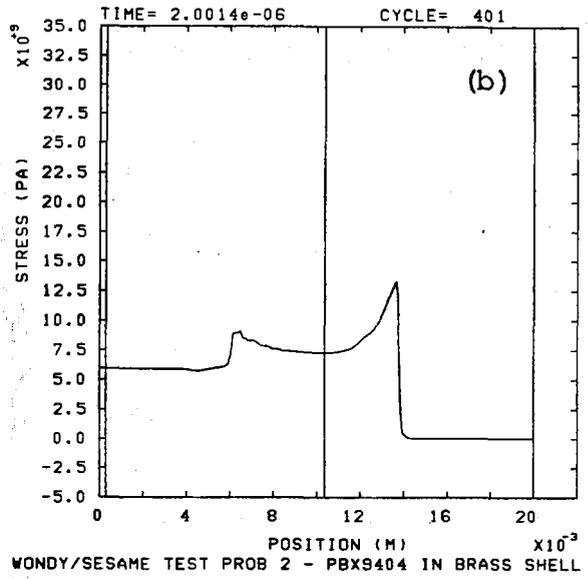
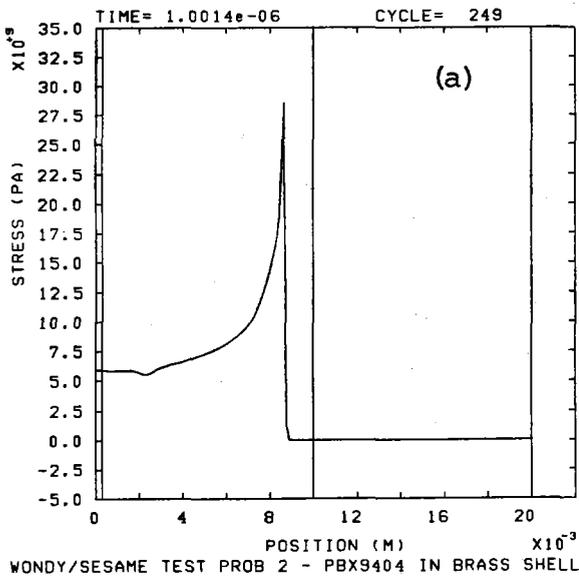


Figure 2: Calculated stress waves for detonation of an explosive charge inside a brass shell, test problem # 2.

4.4 Aluminum Shock Vaporization

The third example is a shock-induced vaporization experiment that was described in Ref. 21 (shot ALFOAM1). A 1.284 mm plate of porous aluminum, having a density of 1666 g/m³, was impacted with a Ta plate having a velocity of 6044 m/s. The shocked material was allowed to expand across a 9.902 mm gap and stagnate against a witness plate backed by a LiF window. The particle velocity history at the interface between the witness plate and the window was measured with a VISAR interferometer.

The problem setup for this calculation is slightly more complicated than for the previous problems because two updates are needed in addition to those for the tabular EOS option. It was found that better numerical results were obtained by using the artificial viscosity in expansion as well as in compression. Furthermore, it is necessary to drop the time step for a short interval, while the expanding aluminum vapors collide with the witness plate, to insure that the zone boundaries are tracked correctly. The file BATTP3, listed below, gives the CCL-CTSS commands to get the files, update and run the code, and save the output.

```
select savelog='temp'
! make updates for this run
copytext ups
-*ID HYDGIK
-*D WONDY7.1768,1769
-c This change turns on artificial viscosity in expansion
-   QSN(J)=.5*(RS(J)+RSN(J))*ABS(RHODOT(J))*QSN(J)
-*I WONDY7.1899
-c Fix to change time step for impacting surfaces
-   TBEFOR=1.05E-6
-   TAFTER=1.15E-6
-   IF(T.GT.TBEFOR.AND.T.LT.TAFTER) THEN
-     DO 6306 J=3,JACT
-6306 DELT3(J)=DELT3(J)*.1
-   ENDIF
! get files from ifs, run updater, and run problem
mass get wait=on /e00024305/hydr/wondy/updates/wonses
mass get wait=on /e00024305/hydr/wondy/tests/sesame
stlib
cconcat ups wonses upfile
updater wondy7 upfile
switch uabs wondy7
mass get wait=on /e00024305/hydr/wondy/tests/inptp3
wondy7
-inptp3
! save files on ifs
log file=logxx
cconcat wonout logxx out
store wait=on dir=/e00024305/hydr/wondy/tests out:outtp3
store wait=on dir=/e00024305/hydr/wondy/tests pltfil:plf1tp3
```


data and the theoretical implications is given in Ref. 21.

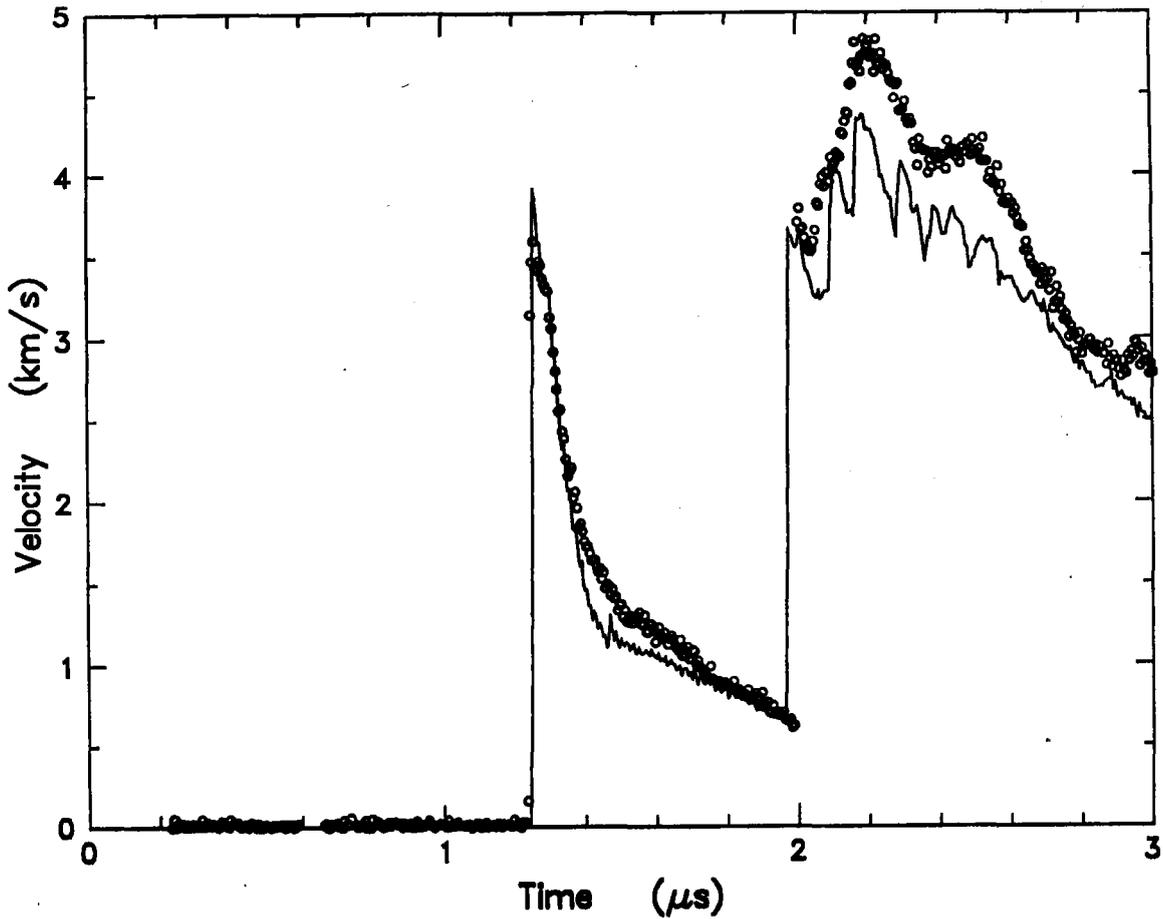


Figure 3: Wave profile for aluminum shock vaporization experiment. Solid line is WONDY calculation; circles are experimental²¹.

5 Acknowledgements

The author appreciates the assistance of M. E. Kipp, T. G. Trucano, and the late F. J. Zeigler with various WONDY calculations and in developing this package. He also thanks R. E. Tokheim for discussions about the electron beam experiments and for sharing the results of his own theoretical work. Finally, he is grateful to M. E. Kipp and T. G. Trucano for their critical evaluation and suggestions concerning this manuscript.

6 References

1. M. E. Kipp and R. J. Lawrence, "WONDY V - A One-Dimensional Finite-Difference Wave Propagation Code," Sandia National Laboratories report SAND81-0930, June 1982.
2. F. J. Zeigler, "WONDY VII - A Vectorized Version of the WONDY Wave Propagation Code," Sandia National Laboratories report SAND85-2338, February 1986.
3. J. R. Asay and G. I. Kerley, "The Response of Materials to Dynamic Loading," *Int. J. Impact Engng.* 5, 69-99 (1987).
4. S. L. Thompson and H. S. Lauson, "Improvements in the Chart-D Radiation-Hydrodynamic Code III: Revised Analytic Equations of State," Sandia Laboratories report SC-RR-710714, March 1972.
5. M. E. Kipp, private communication.
6. B. I. Bennett, J. D. Johnson, G. I. Kerley, and G. T. Rood, "Recent Developments in the Sesame Equation-of-State Library," Los Alamos Scientific Laboratory report LA-7130, 1978.
7. N. G. Cooper, "An Invitation to Participate in the LASL Equation of State Library," Los Alamos Scientific Laboratory report LASL-79-62, 1979.
8. J. Abdallah, Jr., G. I. Kerley, B. I. Bennett, J. D. Johnson, R. C. Albers, and W. F. Huebner, "HYDSES: A Subroutine Package for Using Sesame in Hydrodynamic Codes," Los Alamos Scientific Laboratory report LA-8209 1980.
9. C. W. Cranfill, "EOSPAC: A Subroutine Package for Accessing the Los Alamos Sesame EOS Data Library," Los Alamos National Laboratory report LA-9728-M, 1983.
10. K. S. Holian, "T-4 Handbook of Material Properties Data Bases," Los Alamos Scientific Laboratory report LA-10160-MS, 1984.
11. G. I. Kerley, "User's Manual for PANDA: A Computer Code for Calculating Equations of State," Los Alamos National Laboratory report LA-8833-M, 1981.

12. G. I. Kerley, "Rational Function Method of Interpolation," Los Alamos National Laboratory report LA-6903-MS, 1977.
13. C. L. Mader, Numerical Modeling of Detonations (University of California, Berkeley, 1979).
14. R. B. Oswald Jr., F. B. Mclean, D. R. Schallhorn, and T. R. Oldham, "Dynamic Response of Aluminum to Pulsed Energy Deposition in the Melt-Dominated Regime," J. Appl. Phys. 44, 3563-3574 (1973).
15. G. I. Kerley, "Theoretical Equation of State for Aluminum," Int. J. Impact Engng. 5, 441-449 (1987).
16. J. F. Thomas, "Third-Order Elastic Constants of Aluminum," Phys. Rev. 175, 955 (1968).
17. J. R. Asay, L. C. Chhabildas, G. I. Kerley, and T. G. Trucano, "High Pressure Strength of Shocked Aluminum," in Shock Waves in Condensed Matter, edited by Y. M. Gupta (Plenum Press, NY, 1986) pp 145-149.
18. D. R. Ek and J. R. Asay, "The Stress and Strain-Rate Dependence of Spall Strength in Two Aluminum Alloys," in Shock Waves in Condensed Matter, edited by Y. M. Gupta (Plenum Press, NY, 1986) pp 413-418.
19. R. E. Tokheim, SRI International, Menlo Park, CA, private communication.
20. T. R. Gibbs and A. Popolato, LASL Explosive Property Data, (University of California, Berkeley, 1980) pp. 84-97.
21. G. I. Kerley and J. L. Wise, "Shock-Induced Vaporization of Porous Aluminum," in Shock Waves in Condensed Matter 1987, edited by S. C. Schmidt and N. C. Holmes, (Elsevier Science Publishers, B.V., 1988) pp 155-158.

APPENDIX - EOS Updates for WONDY VII

```
*ID SESAME
*I WONDY7.215
c Test for comment card - ! in column 1
  IF(TESTR(1:1).EQ.'!') GO TO 1001
*I WONDY7.360
c Here call routine SESLIB to load EOS tables from Sesame library.
  CALL SESLIB
*D WONDY7.438,440
c Add branch to statement 1500 for State 7, Sesame tabular option.
  GO TO(1098,1099,1100,1110,1101,1102,1500,11021,11021,11021,11021,
    111021,11021,11021,11021,11021,11021,11021,11021,11021,11021,11021,
    211021,11021,11021), ISTATE
*I WONDY7.452
c Write out EOS data for Sesame option - State 7
1500 WRITE(KPT6,1501)
1501 FORMAT('0',3X,'SESAME EOS WITH MATERIAL STRENGTH')
  GO TO 1103
*D WONDY7.4217,4222
  SUBROUTINE STAT7
-----
c
c SUBROUTINE. STAT7
c
c PURPOSE. Tabular equation of state with analytic initial state
c and elastic plastic equations with work hardening
c
c ARGUMENTS. All arguments are in common, as indicated below
c
c The following variables are used in common /ST1/
c
c GCONST - defined by STIN7, used for constant shear modulus
c DELE - deviatoric work, computed in DEV1 for energy balance
c TX - stress deviator, computed in DEV1
c
c The following variables are used in common /CON/
c
c CES - EOS constants for plates
c FCONST - constant for fracture criterion for plate
c FCRT - fracture criterion for plate
c SIGMAF - constant for fracture criterion for plate
c TDEP - deposition time
c
c The following variables are used in common /VAR/
c
c DELRHO - density change divided by 2*rho**2
c DELT - various time differences
c DEP - fraction of energy added
c
c The following variables are used in common /IND/
c
c JOL - zone at left of plate
c JOR - zone at right of plate
c MORE - flag for morstor
c NONE - indicator for first cycle
c PLATE - plate index
c
```

```

c The following variables are used in common / /
c
c CS - sound speed
c ES, ESN - internal energies at old and new time step
c MS - mass
c PS - pressure
c RSN - density at new time step
c SS - stress
c DATB1 - internal state variable used for energy deposition
c
c The following variable is used in common /S2VAR/
c
c VEOS - internal state variable (for STAT7 only)
c
c The following variable is used in common /UNIT/
c
c KPT6 - output unit used in WRITE statement
c
c The following variables are local - PE, SMIN, KCONST, VHF, QDEP.
c
c REMARKS. This calculation is almost the same as state 1, except
c that the EOS is computed from Sesame tables. The EOS
c calculation is not vectorized.
c
c EXTERNALS. DEVI - material strength calculation
c S2EOSI - equation of state calculation
c
c PROGRAMMER. G. I. Kerley
c
c DATE. 16 March 1988

```

```

INTEGER PLATE
LOGICAL NONE
REAL KCONST, MS
COMMON /ST1/ GCONST(20),DELE(2000),TX(2000)
COMMON /CON/ ADDATA(100), ANGLE, B1, B2,
1CES(42,20), CHG(6,20), DE, DELTAX(20), DELTX1(20),
2DXMAX, DXMIN, DXMN(20), DXMX(20), EZERO(20),
3FCONST(20), FCNSTI(20), FCRTI(20), FCRTI(20), KM(3),
4KTL, NAME1(20), NAME2(20), NOMESH(20), PNUM(20),
5PZERO(20), RCCOMB, RCMB(20), RSC(20), RSCRIT,
6RSCO(20), RSC1(20), RZCO, RZC1, RZERO(20),
7RZTIME, RZTM(20), SIGACT, SIGMAF(20), SIGMIF(20),
8SIGMAO(20), SIGMOI(20), SIGMAX, SIGSEP, STATE(20),
9SZERO(20), TDEP, THKNS(20), UZERO(20),
AUZEROI(20), XGAP(20), RBCN, LBCN, XRATIO(20),
BXZERO, ZZERO(20)
COMMON /VAR/ C, CN, CAPE, CAPH,
1CAPK, DATB(100), DELRHO(2000), DELT(5), DELXJ(2000),
2DELXJN(2000), DELXJ2, DEP, E, EN,
3ETOT, HTOT, ITABLE(50), M, P,
4PA, PN, Q, QA, QN,
5R, RA, RN, RHODOT(2000), R1,
6R2, S, SA, SBN, SN,
7SUMH, SUMIE, SUMKE, SUMQE, T,
8TABLE(2,50), U, UA, UB, UBBN,
9UBN, UN, WL, WR, X,
AXA, XB, XBN, XN, Z,
BZA, ZN
COMMON /IND/ ALPHA2, EXIT, IALPHA, IDUMP,
1JACT, JDT, JDUMP, JMAX,
2JOL, JOR, JONE, JTAPE, K1Z,
3LHBT, LPHA, MCTR, MORE, N,
4NE, NEWPLA, NEXTEM, NONE, NOP,
5NSTART, NTWO, NVAR, PLATE, PRINTE,

```

```

6PRINTL,      PRINTN,      RHBT,      W4020,      IPL1,
7IPL2,        IPL3,        IPL4,        IPL5
COMMON /      / PFRAC(2000),QFRAC(2000),AS(2000),
1CS(2000),ES(2000),ESN(2000),MS(2000),PS(2000),QS(2000),QSN(2000),
2RS(2000),RSN(2000),SS(2000),US(2000),USN(2000),XS(2000),XSN(2000),
3ZS(2000),ZSN(2000),DATB1(2000),DATB2(2000),DATB3(2000),
4DATB4(2000),DATB5(2000),DATB6(2000),DATB7(2000),DATB8(2000),
5DATB9(2000),DATB10(2000)
COMMON/S2VAR/VEOS(2000)
COMMON /UNIT/ KFILIN,KPT6,KFILWR,KFILRR
DIMENSION KCONST(20)
DIMENSION PE(2000), SMIN(2000), VHF(2000), QDEP(2000), PX(3)

```

c

c Special cards for energy deposition

c

```

IF(MORE.EQ.0) THEN
DO 7101 J=JOL,JOR
QDEP(J)=0.
7101 CONTINUE
GO TO 7130
END IF
IF(PLATE.GT.1) GO TO 7110
IF(NONE) SUMDEP=0.
IF(TDEP.EQ.0.) SUMDEP=1.
IF(SUMDEP.LT.1.) GO TO 7103
DEP=0.
QDEP(2)=0.
GO TO 7110
7103 DEP=DELT(1)/TDEP
IF(SUMDEP+DEP-1.)7105,7104,7104
7104 DEP=1.-SUMDEP
7105 SUMDEP=SUMDEP+DEP
QDEP(2)=DEP*DATB1(2)
7110 CONTINUE
JOL1=JOL
IF(PLATE.EQ.1) JOL1=JOL+1
DO 7120 J=JOL1,JOR
QDEP(J)=DEP*DATB1(J)
7120 SUMQE=SUMQE+QDEP(J)*MS(J)
7130 CONTINUE

```

c

c Call routine DEV1 to do material strength calculation

c

```
CALL DEV1
```

c

c Iterative solution to the energy balance equation calling Sesame EOS.

c Accuracy criterion on energy may be important for some problems.

c FSTAT is average number of iterations/zone to solve energy equation.

c

```

FSTAT = 0
NSES = 0
DO 7510 J=JOL,JOR
NSES = NSES+1
NITER = 0
ECST = ES(J)+DELRHO(J)*(PS(J)+QS(J)+QSN(J))+DELE(J)+QDEP(J)
PX(1) = PS(J)+2.*(RSN(J)*CS(J)**2*DELRHO(J)
ESN(J) = DELRHO(J)*PX(1)+ECST
7505 NITER = NITER+1
IF(NITER.GT.100) THEN
WRITE(KPT6,7506) J,RSN(J),ESN(J),PX(1)
7506 FORMAT(/IX,' STATE 7 FAILED TO CONVERGE, J,RSN,ESN,PS=',I5,
1 3(1PE12.4))
GO TO 7507
ENDIF
FSTAT = FSTAT+1

```

c Old value of history variable is used until iteration is

```

c complete, unless there is a problem with convergence.
  IF(NITER.LT.50) VTMP=VEOS(J)
  CALL S2EOSI(PLATE,RSN(J),ESN(J),PX,VTMP)
  DESV = MIN(2.,MAX(.5,1.-DELRHO(J)*PX(3)))
  DESV = (DELRHO(J)*PX(1)+ECST-ESN(J))/DESV
  IF(ABS(DESV).LT.1.E-06*(ABS(ESN(J))+1.E+03)) GO TO 7507
  ESN(J) = ESN(J)+DESV
  GO TO 7505
7507 PS(J) = PX(1)
     VEOS(J) = VTMP
     CS(J) = PX(2)+PX(1)*PX(3)/RSN(J)**2
7510 CONTINUE
     FSTAT = FSTAT/NSES
     WRITE(KPT6,7511)FSTAT,PLATE
7511 FORMAT(/' STATE 7 -',F6.2,' ITERATIONS/ZONE FOR PLATE',I3)
c
c Array CS contains sound speed squared. Check for imaginary sound
c speed and also correct bulk value to longitudinal value.
c
  NOY=CES(29,PLATE)
  IF(NOY.NE.0) GO TO 8040
  DO 8005 J=JOL,JOR
8005 VHF(J)=0.
     DO 8010 J=JOL,JOR
     IF(CS(J).GT.0.) VHF(J)=1.
8010 CS(J)=VHF(J)*SQRT(ABS(CS(J)))+(1.-VHF(J))*CES(2,PLATE)
     GO TO 8100
8040 DO 8045 J=JOL,JOR
8045 VHF(J)=0.
     DO 8050 J=JOL,JOR
     IF(CS(J).GT.0.) VHF(J)=1.
8050 CS(J)=VHF(J)*SQRT(KCONST(PLATE)*ABS(CS(J)))
     1+(1.-VHF(J))*CES(2,PLATE)
8100 CONTINUE
c
c Compute total stress
c
  DO 7810 J=JOL,JOR
7810 SS(J)=PS(J)-TX(J)
c
c Corrections for minimum stress in tension. Option FCRT=1 is a
c primitive fracture option discussed on pp 72 and 73 in SAND81-0930.
c The coding here differs from STAT1 in that the sound speed is not
c corrected after the energy and pressure are corrected.
c
  IF(FCRT(PLATE).EQ.1) THEN
  DO 7870 J=JOL,JOR
7870 SMIN(J)=SIGMAF(PLATE)*AMAX1(0.,(1.-ESN(J)/FCONST(PLATE)))
     DO 7875 J=JOL,JOR
     VHF(J)=0.
7875 IF(SS(J).GE.SMIN(J)) VHF(J)=1.
     DO 7880 J=JOL,JOR
     SS(J)=VHF(J)*SS(J)+(1.-VHF(J))*SMIN(J)
     PE(J)=VHF(J)*PE(J)+(1.-VHF(J))*(SS(J)+TX(J))
     ESN(J)=ESN(J)+(1.-VHF(J))*(PE(J)-PS(J))*DELRHO(J)
7880 PS(J)=VHF(J)*PS(J)+(1.-VHF(J))*PE(J)
     ENDIF
     GO TO 8300
c
c Entry STIN7 is used to initialize equation of state constants
c
  ENTRY STIN7
  CES(9,PLATE)=CES(1,PLATE)*CES(2,PLATE)**2
  CES(23,PLATE)=1.5*CES(9,PLATE)*(1.0-2.0*CES(6,PLATE))/(1.0+CES(6,PLATE))
  LLATE)
c

```

```

c GCONST is the shear modulus, used in the constant g equation
c KCONST is used to correct from bulk to longitudinal sound speed
c
      GCONST(PLATE)=(1.0-2.0*CES(6,PLATE))/(2.0-2.0*CES(6,PLATE))
      KCONST(PLATE)=3.0*(1.0-CES(6,PLATE))/(1.0+CES(6,PLATE))
      JJ=CES(29,PLATE)
      IF (JJ.LE.0) GO TO 8156
c initialization of various yield parameters
      GO TO (8156,8156,8152,8153,8154), JJ
8152 IF (CES(32,PLATE).EQ.0.0) CES(32,PLATE)=1.0E+30
      GO TO 8156
8153 CES(33,PLATE)=2.0*CES(31,PLATE)*CES(32,PLATE)/(CES(31,PLATE)-CES(3
12,PLATE))
      GO TO 8156
8154 CES(33,PLATE)=3.0*CES(9,PLATE)*(1.0-2.0*CES(6,PLATE))
      YTEMP=CES(33,PLATE)*(CES(33,PLATE)/(CES(31,PLATE)*CES(32,PLATE)))
1*(1.0/(CES(31,PLATE)-1.0))
      IF (YTEMP.LE.CES(30,PLATE)) GO TO 8155
      CES(30,PLATE)=1.1*YTEMP
      WRITE(KPT6,8158) CES(30,PLATE),PLATE
8155 CES(34,PLATE)=CES(30,PLATE)/CES(33,PLATE)
8156 DEP=0.0
8158 FORMAT (31H0 Y(0) HAS BEEN INCREASED TO ,E12.5,12H FOR PLATE ,I
14)
8300 RETURN
      END
      SUBROUTINE DEVI

```

```

-----
c
c SUBROUTINE. DEVI
c
c PURPOSE. Material strength computation for STAT1 and STAT7
c
c ARGUMENTS.
c
c The following variables are used in common /ST1/
c
c GCONST - defined by STIN7, used for constant shear modulus
c DELE - deviatoric work, computed here for energy balance in STAT7
c TX - stress deviator, computed here for stress in STAT7
c
c The following variables are used in common /CON/
c
c CES - EOS constants for plates
c
c The following variables are used in common /VAR/
c
c DELT - various time differences
c DELXJ - zone thickness for old time step
c DELXJN - zone thickness for new time step
c
c The following variables are used in common /IND/
c
c IALPHA - geometry constant
c JOL - zone at left of plate
c JOR - zone at right of plate
c N - cycle number (used only in WRITE statement)
c PLATE - plate index
c
c The following variables are used in common / /
c
c QFRACT - fractured zone
c CS - sound speed
c ES, ESN - internal energies at old and new time step
c PS - pressure
c RS,RSN - density at old and new time step

```

c SS - stress
 c USN - velocity at new time step
 c ZS,ZSN - stress difference at old and new time step
 c DATB3 - internal state variable

c The following variables are used in common /FRAC/ and /UNIT/

c UF - velocity at right of fracture
 c KPT6 - output unit used in WRITE statement

c The following variables are local to this subroutine - DX, DZ, EPl,
 c ETA1, G, SUMG, TZ, Y, YEP, YIELDF, ETA, and ETAP.

c REMARKS. This routine does the elastic-plastic calculation as in
 c state 1. Coding was extracted from STAT1 in WONDY7.

c EXTERNALS. none

c PROGRAMMER. G. I. Kerley

c DATE. 15 September 1987

```

    INTEGER PLATE
    LOGICAL QFRACT
    COMMON /ST1/ GCONST(20),DELE(2000),TX(2000)
    COMMON /CON/ ADDATA(100), ANGLE, B1, B2,
    1CES(42,20), CHG(6,20), DE, DELTAX(20), DELTX1(20),
    2DXMAX, DXMIN, DXMN(20), DXMX(20), EZERO(20),
    3FCONST(20), FCNSTI(20), FCRTI(20), FCRTI1(20), KM(3),
    4KT1, NAME1(20), NAME2(20), NOMESH(20), PNUM(20),
    5PZERO(20), RCCOMB, RCMB(20), RSC(20), RSCRIT,
    6RSCO(20), RSC1(20), RZCO, RZC1, RZERO(20),
    7RZTIME, RZTM(20), SIGACT, SIGMAF(20), SIGMIF(20),
    8SIGMAO(20), SIGMOI(20), SIGMAX, SIGSEP, STATE(20),
    9SZERO(20), TDEP, THKNS(20), UZERO(20),
    AUZEROI(20), XGAP(20), RBCN, LBCN, XRATIO(20),
    BXZERO, ZZERO(20)
    COMMON /VAR/ C, CN, CAPE, CAPH,
    1CAPK, DATB(100), DELRHO(2000), DELT(5), DELXJ(2000),
    2DELXJN(2000), DELXJ2, DEP, E, EN,
    3ETOT, HTOT, ITABLE(50), M, P,
    4PA, PN, Q, QA, QN,
    5R, RA, RN, RHODOT(2000), R1,
    6R2, S, SA, SBN, SN,
    7SUMH, SUMIE, SUMKE, SUMQE, T,
    8TABLE(2,50), U, UA, UB, UBBN,
    9UBN, UN, WL, WR, X,
    AXA, XB, XBN, XN, Z,
    BZA, ZN
    COMMON /IND/ ALPHA2, EXIT, IALPHA, IDUMP,
    1JACT, JDT, JDUMP, JMAX,
    2JOL, JOR, JONE, JTape, K1Z,
    3LHBT, LPHA, MCTR, MORE, N,
    4NE, NEWPLA, NEXTEM, NONE, NOP,
    5NSTART, NTWO, NVAR, PLATE, PRINTE,
    6PRINTL, PRINTN, RHBT, W4020, IPL1,
    7IPL2, IPL3, IPL4, IPL5
    COMMON / / PFRACT(2000),QFRACT(2000),AS(2000),
    1CS(2000),ES(2000),ESN(2000),MS(2000),PS(2000),QS(2000),QSN(2000),
    2RS(2000),RSN(2000),SS(2000),US(2000),USN(2000),XS(2000),XSN(2000),
    3ZS(2000),ZSN(2000),DATB1(2000),DATB2(2000),DATB3(2000),
    4DATB4(2000),DATB5(2000),DATB6(2000),DATB7(2000),DATB8(2000),
    5DATB9(2000),DATB10(2000)
    COMMON /FRAC/ UF(2000), XF(2000), JFRACT(2000), NFRACT
    COMMON /UNIT/ KFILIN,KPT6,KFILWR,KFILRR
  
```

DIMENSION DX(2000), DZ(2000), EP1(2000), ETA(2000), ETAP(2000),
 1ETA1(2000), G(2000), SUMG(2000), TZ(2000), Y(2000), YEP(2000),
 2YIELDF(2000)

```

C
C   SET CONSTANTS FOR THIS PLATE
   NG=CES(22,PLATE)-.9
   NOY=CES(29,PLATE)
   IF(NOY.LE.0) GO TO 7210
   Y0=CES(30,PLATE)
   Y1=CES(31,PLATE)
   Y2=CES(32,PLATE)
   Y3=CES(33,PLATE)
   IF(NOY.LT.5) GO TO 7210
   Y4=CES(34,PLATE)
   Y5=Y4**Y1
   Y6=.5/Y3
   Y7=.5*Y0**2/Y3
   Y8=Y1*Y2/(Y1+1.)
   Y9=Y4**(Y1+1.)
7210 CONTINUE
      DO 7240 J=JOL,JOR
      ETA(J)=1.-CES(1,PLATE)/RSN(J)
7240 IF(ABS(ETA(J)).LT.5.E-10) ETA(J)=0.
      IF(NOY.GT.0) GO TO 7252
C   MATERIAL HAS NO STRENGTH
      DO 7250 J=JOL,JOR
      DELE(J)=0.
      TX(J)=0.
7250 ZSN(J)=0.
      GO TO 7500
C   MATERIAL HAS STRENGTH
C
C   COMPUTE G
7252 IF(CES(22,PLATE).NE.0.) GO TO 7262
      DO 7261 J=JOL,JOR
7261 G(J)=GCONST(PLATE)*RS(J)*CS(J)**2
      GO TO 7290
7262 DO 7265 J=JOL,JOR
      ETA1(J)=1.-2.*CES(1,PLATE)/(RSN(J)+RS(J))
7265 IF(ABS(ETA1(J)).LT.5.E-10) ETA1(J)=0.
C
C   THE PURPOSE OF DEFINING THE FACTORS FIE, AND THE REPETITION OF
C   CERTAIN LINES IS TO AID IN VECTORIZATION
7275 DO 7285 J=JOL,JOR
      ETAP(J)=0.
      SUMG(J)=1.
      FIE2=1.
      FIE3=1.
      FIE4=1.
      FIE5=1.
      IF(NG.LE.1) FIE2=0.
      IF(NG.LE.2) FIE3=0.
      IF(NG.LE.3) FIE4=0.
      IF(NG.LE.4) FIE5=0.
      ETAP(J)=ETAP(J)*ETA1(J)
      SUMG(J)=SUMG(J)+CES(24,PLATE)*ETAP(J)
      ETAP(J)=(1.-FIE2)*ETAP(J)+FIE2*ETAP(J)*ETA1(J)
      SUMG(J)=SUMG(J)+FIE2*CES(25,PLATE)*ETAP(J)
      ETAP(J)=(1.-FIE3)*ETAP(J)+FIE3*ETAP(J)*ETA1(J)
      SUMG(J)=SUMG(J)+FIE3*CES(26,PLATE)*ETAP(J)
      ETAP(J)=(1.-FIE4)*ETAP(J)+FIE4*ETAP(J)*ETA1(J)
      SUMG(J)=SUMG(J)+FIE4*CES(27,PLATE)*ETAP(J)
      ETAP(J)=(1.-FIE5)*ETAP(J)+FIE5*ETAP(J)*ETA1(J)
      SUMG(J)=SUMG(J)+FIE5*CES(28,PLATE)*ETAP(J)
7285 G(J)=SUMG(J)*CES(23,PLATE)
C   COMPUTE YIELD STRENGTH

```

```

7290 GO TO (7295,7295,7310,7320,7330) NOY
7295 DO 7300 J=JOL,JOR
7300 Y(J)=Y0
      GO TO 7350
7310 DO 7315 J=JOL,JOR
7315 Y(J)=AMAX1(Y0*(1.+Y1*ETA(J))*(1.-ES(J)/Y2),0.)
      GO TO 7350
7320 DO 7325 J=JOL,JOR
7325 Y(J)=SQRT(Y3*DATAB3(J)+Y0**2)
      GO TO 7350
7330 DO 7340 J=JOL,JOR
      EP1(J)=DATAB3(J)
      IF(EP1(J).LE.0.) GO TO 7340
C     ITERATION FOR NOY=5
      NITER=0
      Y(J)=Y0+((Y1+1.)*EP1(J)*Y2**((1./Y1))**((Y1/(Y1+1.)))
7332 D6=(Y(J)-Y0)/Y2+Y5
      NITER=NITER+1
      EPTEMP=Y(J)*D6**((1./Y1)-Y6*Y(J)**2-Y7+Y8*Y9-Y8*D6**((Y1+1.)/
1Y1)-EP1(J))
      EPRIME=Y(J)/(Y1*Y2)*D6**((1.-Y1)/Y1)-Y(J)/Y3
      Y(J)=Y(J)-EPTMP/EPRIME
      IF(ABS(EPTMP/EPRIME).LT.1.E-7*Y0) GO TO 7340
      IF(Y(J).LT.Y0) GO TO 7334
      IF(NITER.GE.50) GO TO 7334
      GO TO 7332
7334 WRITE(KPT6,8159) N,J,Y0,Y(J),EP1(J)
7340 CONTINUE
7350 DO 7355 J=JOL,JOR
7355 YEP(J)=Y(J)**2/1.5
C
C     compute deviators
C
      IF(IALPHA) 7370,7390,7370
7370 IF(NOY.NE.2) GO TO 7373
      DO 7372 J=JOL,JOR
      DX(J)=2.*(USN(J)-USN(J-1))
1/(DELXJN(J)+DELXJ(J))+RHODOT(J)/3.
      IF(QFRACT(J-1)) DX(J)=2.*(USN(J)-UF(J-1))
1/(DELXJN(J)+DELXJ(J))+RHODOT(J)/3.
      TX(J)=PS(J)-SS(J)+2.*DELT(1)*G(J)*DX(J)
      DELE(J)=1.5*DELT(1)*(TX(J)+PS(J)-SS(J))*DX(J)/(RSN(J)+RS(J))
7372 ZSN(J)=1.5*TX(J)
      GO TO 7500
7373 IF(NOY.GE.4) GO TO 7377
      DO 7374 J=JOL,JOR
      DX(J)=2.*(USN(J)-USN(J-1))
1/(DELXJN(J)+DELXJ(J))+RHODOT(J)/3.
      IF(QFRACT(J-1)) DX(J)=2.*(USN(J)-UF(J-1))
1/(DELXJN(J)+DELXJ(J))+RHODOT(J)/3.
      TX(J)=PS(J)-SS(J)+2.*DELT(1)*G(J)*DX(J)
7374 YIELDF(J)=1.5*TX(J)**2
      DO 7376 J=JOL,JOR
      IF(YIELDF(J).GT.YEP(J)) TX(J)=Y(J)/1.5*SIGN(1.,TX(J))
      DELE(J)=1.5*DELT(1)*(TX(J)+PS(J)-SS(J))*DX(J)/(RSN(J)+RS(J))
7376 ZSN(J)=1.5*TX(J)
      GO TO 7500
7377 DO 7378 J=JOL,JOR
      DX(J)=2.*(USN(J)-USN(J-1))
1/(DELXJN(J)+DELXJ(J))+RHODOT(J)/3.
      IF(QFRACT(J-1)) DX(J)=2.*(USN(J)-UF(J-1))
1/(DELXJN(J)+DELXJ(J))+RHODOT(J)/3.
      TX(J)=PS(J)-SS(J)+2.*DELT(1)*G(J)*DX(J)
7378 YIELDF(J)=1.5*TX(J)**2
      DO 7380 J=JOL,JOR
      IF(YIELDF(J).GT.YEP(J)) TX(J)=Y(J)/1.5*SIGN(1.,TX(J))

```

```

      DATB3(J)=DATB3(J)+YIELDF(J)/(2.*G(J))
      1*(SQRT(YEP(J)/YIELDF(J))-YEP(J)/YIELDF(J))
      DELE(J)=1.5*DELT(1)*(TX(J)+PS(J)-SS(J))*DX(J)/(RSN(J)+RS(J))
7380  ZSN(J)=1.5*TX(J)
      GO TO 7500
7390  DO 7400 J=JOL,JOR
      DX(J)=2.*(USN(J)-USN(J-1))
      1/(DELXJN(J)+DELXJ(J))+RHODOT(J)/3.
      IF(QFRACT(J-1)) DX(J)=2.*(USN(J)-UF(J-1))
      1/(DELXJN(J)+DELXJ(J))+RHODOT(J)/3.
      DZ(J)=RHODOT(J)/3.
      TX(J)=PS(J)-SS(J)+2.*DELT(1)*G(J)*DX(J)
      TZ(J)=ZS(J)-2.*(PS(J)-SS(J)-DELT(1)*G(J)*DZ(J))
      IF(NOE.EQ.2) GO TO 7398
      YIELDF(J)=2.*(TX(J)**2+TX(J)*TZ(J)+TZ(J)**2)
      IF(YIELDF(J).LE.YEP(J)) GO TO 7398
      TX(J)=TX(J)*Y(J)/SQRT(1.5*YIELDF(J))
      TZ(J)=TZ(J)*Y(J)/SQRT(1.5*YIELDF(J))
      IF(NOE.LT.4) GO TO 7398
      DATB3(J)=DATB3(J)+YIELDF(J)/(2.*G(J))
      1*(SQRT(YEP(J)/YIELDF(J))-YEP(J)/YIELDF(J))
7398  DELE(J)=DELT(1)/(RSN(J)+RS(J))*((TX(J)+PS(J)-SS(J))
      1*(2.*DX(J)+DZ(J))+(TZ(J)+ZS(J)-2.*(PS(J)-SS(J)))
      2*(2.*DZ(J)+DX(J)))
7400  ZSN(J)=2.*TX(J)+TZ(J)
7500  CONTINUE
8159  FORMAT (48H0      NOY=5 ITERATION DID NOT CONVERGE FOR CYCLE ,I6,7H
      1 MESH ,I5/9H      Y(0)=,E12.5,5H      Y=,E12.5,8H      E(P)=,E12.5)
      RETURN
      END
      SUBROUTINE SESLIB

```

```

-----
C
C SUBROUTINE.  SESLIB
C
C PURPOSE.     Setup routine for Sesame EOS option - state 7
C
C ARGUMENTS.   Blocks /CON/, /IND/, and /UNIT/ are from WONDY code.
C               Blocks /S2DIR/ and /S2TBS/ are local to Sesame routines.
C
C /CON/      CES (in/out) - EOS constants for each plate. Constants
C             CES(1)-CES(14) are entered by the user. All can be defaulted to
C             zero except material number.
C             CES(1) - R0, initial density (if 0, takes from 201 table)
C             CES(2) - C0, initial sound speed (if 0, takes from 302 table)
C             CES(3) - Sesame material number (required)
C             CES(4) - S, slope of US-UP curve (default is 0)
C             CES(5) - G0, Gruneisen parameter (default is 0)
C             CES(6) - Poissons ratio (used for strength calculation)
C             CES(7) - SR, density scaling factor (if 0, sets SR=1),
C             CES(8) - P0, initial pressure (default is 0)
C             CES(9) - B0=R0*C0**2, computed internally, not used for input
C             CES(10) - E0, initial energy (default is 0)
C             CES(11) - PT, crush pressure (if 0, sets PT=.01*B0)
C             CES(12) - BT, isothermal modulus for crush curve (default is 0)
C             CES(13) - RMIN, minimum ramp density (default is .75*R0)
C             CES(14) - EMAX, maximum ramp energy (default is 0)
C             CES(15) - TYP, type of transition for 2-state model
C                   TYP=0, reversible phase transition or porous material
C                   TYP=1, irreversible phase transition or porous material
C                   TYP=2, density driven transition (CJ volume burn)
C             CES(16) - RMAX, maximum ramp density for TYP=2 (default is 2*R0)
C             CES(17)-CES(21), not presently used in STAT7
C             CES(22)-CES(35) are for strength model, not used in this routine.
C /CON/      STATE (input) - EOS type for each plate
C /IND/      NOP (input) - number of plates or material regions

```

```

c /UNIT/ KPT6 (input) - output unit used in WRITE statement
c /S2TBS/ TBL5 (output) - Array for storage of tables
c /S2DIR/ LCMX (output) - length of array TBL5
c /S2DIR/ NRS (output) - total number of plates
c /S2DIR/ LCFW (output) - pointers to data in array TBL5
c /S2VAR/ VEOS (output) - internal state variable array

```

REMARKS.

```

c Calls routine S2GETI to load EOS data for each plate designated
c state 7. Data are read from a file named 'sesame', assigned to
c unit 50. Calls TB302 to compute sound speed, if not input. Sets
c defaults in array CES. Units of table from Sesame values (Mg/m**3,
c MJ/Kg, GPa) to SI values (Kg/m**3, J/Kg, Pa). The temperature
c calculations are commented out in this version of S2GETI.

```

EXTERNALS. S2GETI, TB302

PROGRAMMER. G. I. Kerley

DATE. 16 March 1988

```

PARAMETER (MXTB=25000)
COMMON/S2DIR/LCMX,NRS,LCFW(20,2)
COMMON/S2TBS/TBL5(MXTB)
COMMON/S2VAR/VEOS(2000)
COMMON /CON/ ADDATA(100), ANGLE, B1, B2,
1CES(42,20), CHG(6,20), DE, DELTAX(20), DELTX1(20),
2DXMAX, DXMIN, DXMN(20), DXMX(20), EZERO(20),
3FCNST(20), FCNSTI(20), FCRT(20), FCRTI(20), KM(3),
4KT1, NAME1(20), NAME2(20), NOMESH(20), PNUM(20),
5PZERO(20), RCCOMB, RCMB(20), RSC(20), RSCRIT,
6RSCO(20), RSC1(20), RZCO, RZC1, RZERO(20),
7RZTIME, RZTM(20), SIGACT, SIGMAF(20), SIGMIF(20),
8SIGMAO(20), SIGMOI(20), SIGMAX, SIGSEP, STATE(20),
9SZERO(20), TDEP, THKNS(20), UZERO(20),
AUZEROI(20), XGAP(20), RBCN, LBCN, XRATIO(20),
BXZERO, ZZERO(20)
COMMON /IND/ ALPHA2, EXIT, IALPHA, IDUMP,
1JACT, JDT, JDUMP, JMAX,
2JOL, JOR, JONE, JTAPE, KLZ,
3LHBT, LPHA, MCTR, MORE, N,
4NE, NEWPLA, NEXTEM, NONE, NOP,
5NSTART, NTWO, NVAR, PLATE, PRINTE,
6PRINTL, PRINTN, RHBT, W4020, IPL1,
7IPL2, IPL3, IPL4, IPL5
COMMON /UNIT/ KFILIN,KPT6,KFILWR,KFILRR
DIMENSION P(3)

```

c Initialize variables in /S2DIR/ and /S2VAR/

```

LCMX = MXTB
NRS = 0
DO 1 I=1,2000
1 VEOS(I) = 0.
IOPN = 0
LCNT = 1
DO 5 IR=1,NOP

```

c First load EOS tables from library into TBL5 and set up pointers.

c Note that, if STATE(IR)<0, EOS number is obtained from another plate.

```

NRS = NRS+1
LCFW(IR,1) = 0
JR = IR
IF(STATE(IR).LT.0.) JR=-STATE(IR)
IF(STATE(JR).NE.7) GO TO 5
IF(IOPN.EQ.0) OPEN (UNIT=50,FILE='sesame',STATUS='old',
1 FORM='unformatted',ERR=6)

```

```

IOPN = 1
IDS2 = CES(3,JR)
CALL S2GETI(IR,IDS2,TBLS,LCNT,50,IFL)
IF(IFL.EQ.0) THEN
  WRITE(KPT6,2)IDS2
2  FORMAT(/1X,'SESAME EOS',I6,' NOT ON SESAME FILE')
  STOP
  ELSE IF(IFL.LT.0) THEN
    IFL=-IFL
    WRITE(KPT6,3)IFL
3  FORMAT(/1X,'INSUFFICIENT STORAGE FOR SESAME EOS, IFL=',I5)
  STOP
ENDIF
c Next check parameters in CES and set up two-state model.
LOC = LCFW(IR,1)
IF(CES(7,JR).LE.0.) CES(7,JR)=1.0
SR = CES(7,JR)
IF(CES(1,JR).LE.0.) CES(1,JR)=TBLS(LOC+1)/SR
IF(CES(2,JR).LE.0.) THEN
  CALL TB302(IR,TBLS(LOC+1),CES(10,JR)/SR,1,P)
  CSSQ = (P(2)+P(1)*P(3)/TBLS(LOC+1)**2)*SR
  CES(2,JR) = SQRT(MAX(1.E+05,CSSQ))
ENDIF
CES(9,JR) = CES(1,JR)*CES(2,JR)**2
IF(CES(11,JR).LE.0.) CES(11,JR)=.01*CES(9,JR)
IF(CES(13,JR).LE.0.) CES(13,JR)=.75*CES(1,JR)
IF(CES(16,JR).LE.0.) CES(16,JR)=2.0*CES(1,JR)
A1 = .5*CES(9,JR)
A2 = 4.*CES(4,JR)/3.
A3 = CES(4,JR)*(1.5*CES(4,JR)-CES(5,JR)/6.)-.5*A2**2
A4 = CES(4,JR)*(1.6*CES(4,JR)**2-.3*CES(5,JR)*CES(4,JR)
1  -CES(5,JR)**2/30.)-A2*(A3+A2**2/6.)
IF(CES(4,JR).EQ.0.) THEN
  UT = (CES(11,JR)-CES(8,JR))/CES(9,JR)
ELSE
  UT = 1.+5*CES(9,JR)/(CES(4,JR)*(CES(11,JR)-CES(8,JR)))
  UT = (UT-SQRT(UT*UT-1.))/CES(4,JR)
ENDIF
RT = CES(1,JR)/(1.-UT)
X1 = A2*UT
X2 = X1+A3*UT*UT
X3 = X2+A4*UT**3
X1 = 2.-X1-X2+3.*X3
X3 = A1*EXP(X3)
IF(CES(5,JR).NE.0.) THEN
  A7 = CES(12,JR)*RT/(CES(5,JR)*CES(1,JR))**2
ELSE
  A7 = 0.
ENDIF
A6 = (CES(8,JR)+X1*X3*UT)/CES(1,JR)-A7*CES(5,JR)
A5 = CES(10,JR)+(CES(8,JR)+X3*UT)*UT/CES(1,JR)-A6*UT-A7
c Load parameters for two-state model into TBLS
NMAX = LCMX-LCNT-15
IF(NMAX.LT.0) THEN
  WRITE(KPT6,2)NWDS
  STOP
ENDIF
LCFW(IR,2) = LCNT
TBLS(LCNT) = CES(7,JR)
TBLS(LCNT+1) = CES(1,JR)
TBLS(LCNT+2) = CES(8,JR)
TBLS(LCNT+3) = CES(10,JR)
TBLS(LCNT+4) = CES(5,JR)
TBLS(LCNT+5) = CES(13,JR)
TBLS(LCNT+6) = CES(14,JR)
TBLS(LCNT+7) = A1

```

```

      TBLS(LCNT+8) = A2
      TBLS(LCNT+9) = A3
      TBLS(LCNT+10) = A4
      TBLS(LCNT+11) = RT
      TBLS(LCNT+12) = A5
      TBLS(LCNT+13) = A6
      TBLS(LCNT+14) = CES(15,JR)
      TBLS(LCNT+15) = CES(16,JR)
      LCNT = LCNT+16
      WRITE(KPT6,4)IR
4     FORMAT(/1X,'SESAME TABLE SUCCESSFULLY LOADED FOR PLATE',I3)
5     CONTINUE
      IF(IOPN.EQ.1) CLOSE(UNIT=50)
      RETURN
c Error trying to open file
6     WRITE(KPT6,7)
7     FORMAT(/1X,'ERROR OPENING SESAME FILE')
      STOP
      END
      SUBROUTINE S2GETI(IR,IDS2,TBLS,LCNT,LU,IFL)
-----
c
c
c SUBROUTINE. S2GETI(IR,IDS2,TBLS,LCNT,LU,IFL)
c
c PURPOSE. Load inverted EOS tables from library
c
c ARGUMENTS. IR (input) - material region or plate number
c             IDS2 (input) - Sesame 2 id number
c             TBLS (in/out) - array for storage of tables
c             LCNT (in/out) - pointer to TBLS, updated by routine
c             LU (input) - logical unit number for library
c             IFL (output) - error flag
c             IFL = 1, if data is loaded with no problems
c             IFL = 0, if data cannot be located
c             IFL = -no. of words needed, if storage is insufficient
c
c User also supplies a common block having the form
c COMMON/S2DIR/LCMX,NRS,LCFW(20,2)
c LCMX = length of array TBLS
c NRS = no. of regions, including non-Sesame regions
c LCFW = array used for directory to TBLS
c This common block also used by routine S2EOSI.
c
c REMARKS. Loads data into array TBLS, beginning with TBLS(LCNT),
c           inverts 301 table, and computes new value of LCNT, the
c           value chosen here so as to overwrite temperature array.
c           Converts units of density, pressure, and energy.
c
c EXTERNALS. TGTNDX, TGTMAT, INV301
c
c PROGRAMMER. G. I. Kerley
c
c DATE. 16 March 1988
c
-----
      DIMENSION TBLS(1),INDX(200)
      COMMON/S2DIR/LCMX,NRS,LCFW(20,2)
      IFL = 0
      IF(NRS.LT.1) RETURN
      ID = IABS(IDS2)
c First check to see if EOS table has already been loaded
      DO 1 I=1,NRS
          LOC = LCFW(I,1)
          IF(LOC.EQ.0) GO TO 1
          ITEST = TBLS(LOC)
          IF(ID.EQ.ITEST) THEN

```

```

        LCFW(IR,1) = LOC
        IFL = 1
        RETURN
    ENDIF
1    CONTINUE
c    Load and invert EOS table.
    CALL TGTNDX(ID,INDX,LU,200,IFL)
    IF(IFL.LT.1) RETURN
    LOC = LCNT+2
    NMAX = LCMX-LOC+1
    CALL TGTMAT(201.,TBLS(LOC),INDX,LU,NMAX,IFL)
    IF(IFL.LT.1) RETURN
    TBLS(LCNT+1) = TBLS(LOC+2)
    CALL TGTMAT(301.,TBLS(LOC),INDX,LU,NMAX,IFL)
    IF(IFL.LT.1) RETURN
    LOC = LCNT
    NR = TBLS(LOC+2)
    NT = TBLS(LOC+3)
    NWDS = IFL+NR+3*NT
    IFL = NMAX-NWDS
    IF(IFL.LT.0) RETURN
    R0 = TBLS(LOC+1)
    CALL INV301(TBLS(LOC+2),R0,LDS)
    TBLS(LCNT) = ID
    LCNT = LCNT+2+LDS
c    The following line writes over the temperature array
    LCNT = LCNT-NR*NT
    LCFW(IR,1) = LOC
    IFL = 1
c    Change units using factors RFAC, EFAC, and PFAC, defined below.
    RFAC = 1.E+03
    EFAC = 1.E+06
    PFAC = 1.E+09
    TBLS(LOC+1) = RFAC*TBLS(LOC+1)
    LOCX = LOC+3
    DO 2 I=1,NR
        TBLS(LOCX+I) = RFAC*TBLS(LOCX+I)
2    TBLS(LOCX+NR+NT+I) = EFAC*TBLS(LOCX+NR+NT+I)
    LOCX = LOCX+NR
    DO 3 J=1,NT
3    TBLS(LOCX+J) = EFAC*TBLS(LOCX+J)
    LOCX = LOCX+NR+NT
    DO 4 I=1,NR
    DO 4 J=1,NT
        LOCX = LOCX+1
4    TBLS(LOCX) = PFAC*TBLS(LOCX)
    RETURN
    END
    SUBROUTINE S2EOSI(IR,R,E,P,VF)
-----
c
c    SUBROUTINE.  S2EOSI(IR,R,E,P,VF)
c
c    PURPOSE.    Compute P(V,E) from tabular EOS with two-state model
c
c    ARGUMENTS.  IR (input) - material region or plate number
c                R (input) - density
c                E (input) - internal energy
c                P (output) - pressure vector
c                P(1) - pressure
c                P(2) - density derivative
c                P(3) - energy derivative
c                VF (in/out) - fraction of final state
c                /S2TBS/ - EOS tables (see routine SESLIB)
c                /S2DIR/ - pointers to EOS tables (see routine SESLIB)
c

```

```

c REMARKS.      Calls S2INIT for EOS initial state and TB302 for EOS of
c               final state (EOS table).
c
c EXTERNALS.    S2INIT, TB302
c
c PROGRAMMER.   G. I. Kerley
c
c DATE.         3 March 1988
c
c

```

```

-----
c PARAMETER (MXTB=25000)
c COMMON/S2DIR/LCMX,NRS,LCFW(20,2)
c COMMON/S2TBS/TBLS(MXTB)
c DIMENSION P(3),PF(3)
c Parameter SR is used to scale density and internal energy
c Parameter TYP defines type of transition in two-state model
  LOC = LCFW(IR,2)
  SR = TBLS(LOC)
  TYP = TBLS(LOC+14)
  IF(R.LE.TBLS(LOC+5).OR.E.GE.TBLS(LOC+6)) THEN
    CONTINUE
  ELSE IF(TYP.EQ.2.0) THEN
    IF(VF.NE.1.0.AND.R.LT.TBLS(LOC+15)) THEN
      CALL S2INIT(IR,R,E,P)
      RETURN
    ENDIF
  ELSE IF(TYP.EQ.1.0.AND.VF.EQ.1.0) THEN
    CONTINUE
  ELSE
    CALL S2INIT(IR,R,E,P)
    CALL TB302(IR,SR*R,E/SR,1,PF)
    IF(P(1).LT.PF(1).AND.P(1).GT.0.) THEN
      P(1) = PF(1)
      P(2) = SR*PF(2)
      P(3) = PF(3)/SR
      VF = 1.0
    ENDIF
    RETURN
  ENDIF
  CALL TB302(IR,SR*R,E/SR,1,P)
  P(2) = SR*P(2)
  P(3) = P(3)/SR
  VF = 1.0
  RETURN
END
SUBROUTINE S2INIT(IR,R,E,PI)

```

```

-----
c
c SUBROUTINE.    S2INIT(IR,R,E,PI)
c
c PURPOSE.      Compute P(V,E) for initial state in two-state model
c
c ARGUMENTS.    IR (input) - material region or plate number
c               R (input) - density
c               E (input) - internal energy
c               PI (output) - pressure vector
c               PI(1) = pressure
c               PI(2) = density derivative
c               PI(3) = energy derivative
c               /S2TBS/ - EOS tables (see routine SESLIB)
c               /S2DIR/ - pointers to EOS tables (see routine SESLIB)
c
c REMARKS.      The initial state parameters are in TBLS with LCFW(IR,2)
c               as pointers to the data.
c
c EXTERNALS.    none

```

```

C
C PROGRAMMER. G. I. Kerley
C
C DATE. 3 March 1988
C
C

```

```

-----
PARAMETER (MXTB=25000)
COMMON/S2DIR/LCMX,NRS,LCFW(20,2)
COMMON/S2TBS/TBLS(MXTB)
DIMENSION PI(3)
LOC = LCFW(IR,2)
R0 = TBLS(LOC+1)
G0 = TBLS(LOC+4)
PI(3) = G0*R0
U = 1.-R0/R
RT = TBLS(LOC+11)
IF(R.LT.RT) THEN
  X1 = TBLS(LOC+8)*U
  X2 = X1+TBLS(LOC+9)*U*U
  X3 = X2+TBLS(LOC+10)*U**3
  X1 = 2.-X1-X2+3.*X3
  X3 = TBLS(LOC+7)*EXP(X3)
  ER = TBLS(LOC+3)+(TBLS(LOC+2)+X3*U)*U/R0
  PR = TBLS(LOC+2)+X1*X3*U
  PI(1) = PR+PI(3)*(E-ER)
  PI(2) = ((X1*(2.+X1)-6.-2.*X2)*X3-G0*PR)*R0/R**2
ELSE
  PI(1) = R0*TBLS(LOC+13)+PI(3)*(E-TBLS(LOC+12)-U*TBLS(LOC+13))
  PI(2) = -G0*TBLS(LOC+13)*(R0/R)**2
ENDIF
RETURN
END
SUBROUTINE INBUF(LU,Z,N,IM)

```

```

-----
C
C SUBROUTINE. INBUF(LU,Z,N,IM)
C
C PURPOSE. Unformatted read into local array
C
C ARGUMENTS. LU (input) - logical unit number for read
C             Z (in/out) - name of local array
C             N (input) - number of words to be read
C             IM (output) - end file flag
C             IM=0 if end file mark encountered
C             IM=1 if no end file mark
C
C REMARKS. This routine is just a binary read and test for end of
C           file. It assumes the data records were written with the
C           corresponding routine, OUTBUF, and must be compatible
C           with that routine.
C
C EXTERNALS. none
C
C PROGRAMMER. G. I. Kerley
C
C DATE. 27 March 1984
C
C

```

```

-----
DIMENSION Z(1)
READ(LU,END=1)(Z(I),I=1,N)
IM = 1
RETURN
1 IM = 0
RETURN
END
SUBROUTINE INV301(DSTR,R0,LDS)

```

```

C-----
C
C SUBROUTINE.  INV301(DSTR,R0,LDS)
C
C PURPOSE.    Invert data string of type 301 to type 302.
C
C ARGUMENTS.  DSTR (input) - data string to be inverted
C              R0  (input) - approximate density of solid
C              LDS (output) - length of new data string
C
C REMARKS.    The routine overwrites locations following the data
C              string.  It expands the string by NR words, where NR is
C              the number of densities.  It also uses 3*NT words as
C              temporary storage, where NT is number of temperatures.
C
C EXTERNALS.  QSRCHK, RATFN1.
C
C PROGRAMMER. G. I. Kerley
C
C DATE.       24 February 1988
C-----

```

```

C
C DIMENSION Z(2),DSTR(1)
C COMMON/RTBLK1/LOCX,NR,LOCY,KY,JX,NT
C NR = DSTR(1)
C NT = DSTR(2)
C LOCT = 3+NR
C LCEC = LOCT+NT
C LOCP = LCEC+NR
C LOCE = LOCP+NR*NT
C LOCN = LOCE+NR*NT
C IMAX = 2*NR*NT
C DO 1 I=1,IMAX
1  DSTR(LOCN-I) = DSTR(LOCN-I-NR)
C DO 2 I=1,NR
C JJ = LOCE+I-1
C Q = 1.E-08*ABS(DSTR(JJ))
C DSTR(LCEC+I-1) = DSTR(JJ)
C DSTR(JJ) = 0.
C DO 2 J=2,NT
C JJ = JJ+NR
C DSTR(JJ) = DSTR(JJ)-DSTR(LCEC+I-1)
C IF(DSTR(JJ)-DSTR(JJ-NR).LT.Q) DSTR(JJ)=DSTR(JJ-NR)+Q
2  CONTINUE
C CALL QSRCHK(R0,DSTR(4),NR-2,1,I)
C I = I+1
C DO 3 J=1,NT
C DSTR(LOCN+J-1) = DSTR(LOCT+J-1)
3  DSTR(LOCT+J-1) = DSTR(LOCE+I-1+NR*(J-1))
C DO 5 I=1,NR
C LOCX = LOCE+I-1
C DO 4 J=1,NT
C ET = DSTR(LOCT+J-1)
C CALL QSRCHK(ET,DSTR(LOCX+NR),NT-2,NR,JX)
C JX = JX+1
C LOCY = LOCP+I-1
C KY = NR
C CALL RATFN1(ET,DSTR,Z)
C DSTR(LOCN+NT+J-1) = Z(1)
C LOCY = LOCN
C KY = 1
C CALL RATFN1(ET,DSTR,Z)
4  DSTR(LOCN+NT+NT+J-1) = Z(1)
C DO 5 J=1,NT
C DSTR(LOCP+I-1+NR*(J-1)) = DSTR(LOCN+NT+J-1)
5  DSTR(LOCX+NR*(J-1)) = DSTR(LOCN+NT+NT+J-1)

```

```

LDS = LOCN-1
RETURN
END
SUBROUTINE QSRCH(X,T,N,I)

```

```

-----
C
C SUBROUTINE.  QSRCH(X,T,N,I)
C
C PURPOSE.     Search for index of X in monotonic increasing table T
C
C ARGUMENTS.   X (input) - value to be located
C               T (input) - table to be searched
C               N (input) - number of values in table T
C               I (in/out) - index of X in array T
C
C REMARKS.     The index I is defined so that  $T(I) \leq X < T(I+1)$ , or  $I=N-1$ 
C               if  $X=T(N)$ . Returns  $I=0$  or  $I=N$  if X is outside range of
C               table. If  $I=0$  on input, does binary search. Otherwise,
C               does sequential search.
C
C EXTERNALS.   None
C
C PROGRAMMER.  G. I. Kerley
C
C DATE.        8 November 1987
C
-----

```

```

      DIMENSION T(1)
      IF(X.EQ.T(N)) THEN
        I = N-1
        RETURN
C Binary search
      ELSE IF(I.LE.0.OR.I.GE.N) THEN
        I = 0
        J = N+1
1       IF(J-I.EQ.1) RETURN
        JP = .5*(J+I)
        IF(X.GE.T(JP)) THEN
          I = JP
        ELSE
          J = JP
        ENDIF
        GO TO 1
C Sequential search
      ELSE
        IF(X.LT.T(I)) THEN
2         I = I-1
          IF(I.EQ.0) RETURN
          IF(X.GE.T(I)) RETURN
          GO TO 2
        ELSE
3         IF(X.LT.T(I+1)) RETURN
          I = I+1
          IF(I.EQ.N) RETURN
          GO TO 3
        ENDIF
      ENDIF
      END
      SUBROUTINE QSRCHK(X,TBLS,N,K,I)

```

```

-----
C
C SUBROUTINE.  QSRCHK(X,TBLS,N,K,I)
C
C PURPOSE.     Binary search for index of X in an array TBLS. Table
C               values need not be contiguous and can be in either
C               ascending or descending order.
C

```

```

C
C ARGUMENTS.  X   (input) - value to be located
C             TBLs (input) - table to be searched
C             N   (input) - number of values to be searched
C             K   (input) - spacing between values in table
C             I   (output) - index of X in array TBLs
C
C REMARKS.    The value of the function is the index I, where
C             TBLs(1+K*(I-1)).LE.X.LT.TBLs(1+K*I), or
C             TBLs(1+K*(I-1)).GE.X.GT.TBLs(1+K*I), else
C             I=0 or I=N if X is outside range of table.
C
C EXTERNALS.  None
C
C PROGRAMMER. G. I. Kerley
C
C DATE.       3 November 1987
C

```

```

-----
C
C   DIMENSION TBLs(1)
C   I = 0
C   J = N+1
C   KI = 1-K
C   S = TBLs(KI+K*N)-TBLs(1)
1  IF(J-I.EQ.1) RETURN
C   JP = .5*(J+I)
C   IF(S*(X-TBLs(KI+K*JP)).GE.0.) THEN
C     I = JP
C   ELSE
C     J = JP
C   ENDIF
C   GO TO 1
C   END
C   SUBROUTINE RATFN1(X,TBLs,Y)

```

```

-----
C
C SUBROUTINE.  RATFN1(X,TBLs,Y)
C
C PURPOSE.    Interpolate for a function Y(X) and its derivative
C             from tables located in array TBLs.
C
C ARGUMENTS.  X   (input) - independent variable
C             TBLs (input) - array containing tables
C             Y   (output) - vector of length 2, where
C             Y(1) = value of function
C             Y(2) = derivative of function
C             The routine also requires a common block,
C             COMMON/RTBLK1/LOCX,KX,LOCY,KY,I,N
C             LOCX = location of X vector
C             KX   = spacing of X vector
C             LOCY = location of Y vector
C             KY   = spacing of Y vector
C             I    = index of X and Y vectors
C             N    = length of X and Y vectors
C
C REMARKS.    Uses rational function method with quadratic estimate
C             of derivatives at the mesh points.
C
C EXTERNALS.  None, but first call search routine to compute index I.
C
C PROGRAMMER. G. I. Kerley
C
C DATE.       3 November 1987
C

```

```

-----
C
C   DIMENSION TBLs(1),Y(2)

```

```

COMMON/RTBLK1/LOCX,KX,LOCY,KY,I,N
IX = LOCX+KX*(I-1)
IY = LOCY+KY*(I-1)
Q = X-TBLS(IX)
D = TBLS(IX+KX)-TBLS(IX)
R = D-Q
S = (TBLS(IY+KY)-TBLS(IY))/D
IF(I.EQ.1) THEN
  SP = (TBLS(IY+KY+KY)-TBLS(IY+KY))/(TBLS(IX+KX+KX)-TBLS(IX+KX))
  C2 = (SP-S)/(TBLS(IX+KX+KX)-TBLS(IX))
  IF(S*(S-D*C2).LE.0.) C2=S/D
  Y(1) = TBLS(IY)+Q*(S-R*C2)
  Y(2) = S+(Q-R)*C2
ELSE IF (I.EQ.N-1) THEN
  DM = TBLS(IX)-TBLS(IX-KX)
  SM = (TBLS(IY)-TBLS(IY-KY))/DM
  C1 = (S-SM)/(D+DM)
  Y(1) = TBLS(IY)+Q*(S-R*C1)
  Y(2) = S+(Q-R)*C1
ELSE
  DM = TBLS(IX)-TBLS(IX-KX)
  SM = (TBLS(IY)-TBLS(IY-KY))/DM
  C1 = (S-SM)/(D+DM)
  IF(I.EQ.2.AND.SM*(SM-DM*C1).LE.0.) C1=(S-SM-SM)/D
  SP = (TBLS(IY+KY+KY)-TBLS(IY+KY))/(TBLS(IX+KX+KX)-TBLS(IX+KX))
  C2 = (SP-S)/(TBLS(IX+KX+KX)-TBLS(IX))
  C3 = ABS(C2*R)
  C4 = C3+ABS(C1*Q)
  IF(C4.GT.0.) C3=C3/C4
  C4 = C2+C3*(C1-C2)
  Y(1) = TBLS(IY)+Q*(S-R*C4)
  Y(2) = S+(Q-R)*C4+D*(C4-C2)*(1.-C3)
ENDIF
RETURN
END
SUBROUTINE RATFN2(X,Y,TBLS,Z)

```

```

c
c -----
c
c SUBROUTINE.  RATFN2(X,Y,TBLS,Z)
c
c PURPOSE.    Interpolate for a function Z(X,Y) and its derivatives
c              from tables located in array TBLS.
c
c ARGUMENTS.  X,Y (input) - independent variables
c              TBLS (input) - array containing tables
c              Z (output) - vector of length 3, where
c                  Z(1) = value of function
c                  Z(2) = X derivative of function
c                  Z(3) = Y derivative of function
c
c The routine also requires a common block,
c COMMON/RTBLK2/LOCX,IX,NX,LOCY,IY,NY,LOCZ,NZ,IFN
c LOCX = location of X vector
c IX = index of X vector
c NX = length of X vector
c LOCY = location of Y vector
c IY = index of Y vector
c NY = length of Y vector
c LOCZ = location of Z(X,Y) array
c NZ = spacing of Z array
c IFN = interpolation flag - set IFN=1 for bilinear
c
c REMARKS.    Unless bilinear form is specified, routine uses rational
c              function method with quadratic estimate of derivatives
c              at the mesh points.
c
c EXTERNALS.  None, but call search routine to compute IX and IY.

```

C
 C PROGRAMMER. G. I. Kerley
 C
 C DATE. 4 November 1987
 C
 C

```

DIMENSION ZZ(4), ZD(4), TBLS(1), Z(3)
COMMON/RTBLK2/LOCX, IX, NX, LOCY, IY, NY, LOCZ, NZ, IFN
I = LOCX+IX-1
IZ = LOCZ+NZ*(IX-1+NX*(IY-1))
KZ = NZ
IBR = IX
NBR = NX-IX
Q = X-TBLS(I)
D = TBLS(I+1)-TBLS(I)
R = D-Q
DO 1 K=1,4
  S = (TBLS(IZ+KZ)-TBLS(IZ))/D
  ZZ(K) = .5*(TBLS(IZ)+Q*S)
  ZD(K) = .5*S
  IF(IFN.NE.1) THEN
    IF(IBR.LE.1) THEN
      SP = (TBLS(IZ+KZ+KZ)-TBLS(IZ+KZ))/(TBLS(I+2)-TBLS(I+1))
      C2 = (SP-S)/(TBLS(I+2)-TBLS(I))
      IF(S*(S-D*C2).LE.0.) C2=S/D
      ZZ(K) = ZZ(K)-Q*R*C2
      ZD(K) = ZD(K)+(Q-R)*C2
    ELSE IF(NBR.LE.1) THEN
      DM = TBLS(I)-TBLS(I-1)
      SM = (TBLS(IZ)-TBLS(IZ-KZ))/DM
      C1 = (S-SM)/(D+DM)
      ZZ(K) = ZZ(K)-Q*R*C1
      ZD(K) = ZD(K)+(Q-R)*C1
    ELSE
      DM = TBLS(I)-TBLS(I-1)
      SM = (TBLS(IZ)-TBLS(IZ-KZ))/DM
      C1 = (S-SM)/(D+DM)
      IF(IBR.EQ.2.AND.SM*(SM-DM*C1).LE.0.) C1=(S-SM-SM)/D
      SP = (TBLS(IZ+KZ+KZ)-TBLS(IZ+KZ))/(TBLS(I+2)-TBLS(I+1))
      C2 = (SP-S)/(TBLS(I+2)-TBLS(I))
      C3 = ABS(C2*R)
      C4 = C3+ABS(C1*Q)
      IF(C4.GT.0.) C3=C3/C4
      C4 = C2+C3*(C1-C2)
      ZZ(K) = ZZ(K)-Q*R*C4
      ZD(K) = ZD(K)+(Q-R)*C4+D*(C4-C2)*(1.-C3)
    ENDIF
  ENDIF
  IF(K.EQ.1) THEN
    IZ = IZ+NX*NZ
  ELSE IF(K.EQ.2) THEN
    DX = D
    QX = Q
    I = LOCY+IY-1
    KZ = KZ+NX
    IZ = IZ-KZ
    IBR = IY
    NBR = NY-IY
    Q = Y-TBLS(I)
    D = TBLS(I+1)-TBLS(I)
    R = D-Q
  ELSE IF(K.EQ.3) THEN
    IZ = IZ+NZ
  ENDIF
  CONTINUE
  ZZ(2) = (ZZ(2)-ZZ(1))/D

```

```

ZZ(4) = (ZZ(4)-ZZ(3))/DX
Z(1) = ZZ(1)+ZZ(2)*Q+ZZ(3)+ZZ(4)*QX
Z(2) = ZZ(4)+ZD(1)+(ZD(2)-ZD(1))*Q/D
Z(3) = ZZ(2)+ZD(3)+(ZD(4)-ZD(3))*QX/DX
RETURN
END
SUBROUTINE TB302(IP,R,E,NV,F)

```

```

-----
C
C
C SUBROUTINE.   TB302(IP,R,E,NV,F)
C
C PURPOSE.     Search/interpolate for pressure or temperature as
C              functions of density and internal energy.
C
C ARGUMENTS.   IP (input) - material region or plate number
C              R  (input) - density
C              E  (input) - energy
C              NV (input) - NV=1 for pressure, NV=2 for temperature
C              F  (output) - function and derivatives
C              F(1) = value of the function
C              F(2) = density derivative, dF/dR at constant E
C              F(3) = energy derivative, dF/dE at constant R
C
C REMARKS.     Uses rational function interpolation. To use bilinear,
C              set IFN=1 in first executable statement.
C
C EXTERNALS.   QSRCH, RATFN1, RATFN2.
C
C PROGRAMMER.  G. I. Kerley
C
C DATE.        3 March 1988
C
-----

```

```

PARAMETER (MXTB=25000)
DIMENSION F(3)
COMMON/S2DIR/LCMX,NRS,LCFW(20,2)
COMMON/S2TBS/TBLS(MXTB)
COMMON/RTBLK1/LOCX,KX,LOCY,KY,IX,N
COMMON/RTBLK2/LOCR,IR,NR,LOCE,IE,NE,LOCZ,NZ,IFN
SAVE IPSV,RSV,ESV,ET,DECDR
DATA IPSV,RSV,ESV/0,0.,0./
IFN = 0
c If a new region, zero saved values and reset /RTBLK1/ and /RTBLK2/.
IF(IP.NE.IPSV) THEN
  LOC = LCFW(IP,1)+2
  NR = TBLS(LOC)
  NE = TBLS(LOC+1)
  LOCX = LOC+2
  KX = 1
  LOCY = LOCX+NR+NE
  KY = 1
  N = NR
  LOCR = LOCX
  LOCE = LOCX+NR
  NZ = 1
  IR = 0
  IE = 0
  IPSV = IP
c If density and energy unchanged, skip search, cold curve calculation.
ELSE IF(R.EQ.RSV.AND.E.EQ.ESV) THEN
  GO TO 1
ENDIF
c Otherwise, search for new indices and compute cold curve.
CALL QSRCH(R,TBLS(LOCX),NR,IR)
IR = MAX(1,MIN(NR-1,IR))
IX = IR

```

```

CALL RATFN1(R,TBLS,F)
ET = MAX(0.,E-F(1))
DECDR = F(2)
CALL QSRCH(ET,TBLS(LOCE),NE,IE)
IE = MAX(1,MIN(NE-1,IE))
RSV = R
ESV = E
1 LOCZ = LOCY+NR+(NV-1)*NR*NE
CALL RATFN2(R,ET,TBLS,F)
F(2) = F(2)-DECDR*F(3)
RETURN
END
SUBROUTINE TGTMAT(TYP,TBLS,RNDX,LU,NMAX,NWDS)

```

```

C-----
C
C SUBROUTINE. TGTMAT(TYP,TBLS,RNDX,LU,NMAX,NWDS)
C
C PURPOSE. Get a data table from SESAME 2 library.
C
C ARGUMENTS. TYP (input) - catalogue number for type of data table
C TBLS (in/out) - array where table is to be stored
C RNDX (input) - material index (obtained with TGTNDX)
C LU (input) - logical unit number for library
C NMAX (input) - maximum length of array TBLS
C NWDS (output) - number of words loaded into TBLS
C NWDS=0, if the table cannot be found
C NWDS=-number of words needed, if NMAX is too small
C
C REMARKS. Assumes sequential binary file structure, with pointer
C on unit LU set to first word of material file (index).
C After locating data, it backspaces to return pointer to
C the original position.
C
C EXTERNALS. INBUF, IFIX
C
C PROGRAMMER. G. I. Kerley
C
C DATE. 3 November 1987
C-----

```

```

C
C DIMENSION TBLS(1),RNDX(1)
C NWDS = 0
C J = 5
C NREC = IFIX(RNDX(J))
C DO 1 I=1,NREC
C IF(TYP.EQ.RNDX(J+I)) GO TO 2
1 CONTINUE
C RETURN
2 NRD = IFIX(RNDX(J+I+NREC))
C IF(NRD.LE.NMAX) THEN
C DO 3 K=1,I
C CALL INBUF(LU,TBLS,1,IEOF)
C IF(IEOF.EQ.0) RETURN
3 CONTINUE
C CALL INBUF(LU,TBLS,NRD,IEOF)
C IF(IEOF.EQ.0) RETURN
C I = I+1
C DO 4 K=1,I
4 BACKSPACE LU
C NWDS = NRD
C ELSE
C NWDS = NMAX-NRD
C ENDIF
C RETURN
C END
C SUBROUTINE TGTNDX(MATID,RNDX,LU,NMAX,NWDS)

```

```

-----
C
C
C SUBROUTINE.  TGTNDX(MATID,RNDX,LU,NMAX,NWDS)
C
C PURPOSE.    Get index for a material from SESAME 2 library.
C
C ARGUMENTS.  MATID (input) - material ID number
C              RNDX (in/out) - array where index is to be stored
C              LU   (input) - logical unit number for library
C              NMAX (input) - maximum length of array RNDX
C              NWDS (output) - number of words loaded into RNDX
C              NWDS=0, if the index cannot be found
C              NWDS--number of words needed, if NMAX is too small
C
C REMARKS.    Assumes sequential binary file structure.  First rewinds
C              file LU and reads in directory, then sequentially reads
C              file until it obtains specified index.  After index has
C              been read in, it backspaces LU one record, to locate
C              pointer at the first word of the material file.
C
C EXTERNALS.  INBUF, IFIX
C
C PROGRAMMER. G. I. Kerley
C
C DATE.       3 November 1987
C
-----

```

```

DIMENSION RNDX(1)
REWIND LU
NWDS = 0
CALL INBUF(LU,RNDX,1,IEOF)
IF(IEOF.EQ.0) RETURN
NMAT = IFIX(RNDX(1))
NUM = 2*NMAT
IF(NUM.GT.NMAX) GO TO 5
CALL INBUF(LU,RNDX,NUM,IEOF)
IF(IEOF.EQ.0) RETURN
DO 1 I=1,NMAT
  IMAT = IFIX(RNDX(I))
  IF(IMAT.EQ.MATID) GO TO 2
1  CONTINUE
RETURN
2  NUM = IFIX(RNDX(NMAT+I))
  IF(NUM.GT.NMAX) GO TO 5
  DO 4 K=1,I
3    CALL INBUF(LU,RNDX,1,IEOF)
    IF(IEOF.NE.0) GO TO 3
4    CONTINUE
  CALL INBUF(LU,RNDX,NUM,IEOF)
  IF(IEOF.EQ.0) RETURN
  IMAT = IFIX(RNDX(1))
  IF(IMAT.NE.MATID) RETURN
  NWDS = NUM
  BACKSPACE LU
  RETURN
5  NWDS = NMAX-NUM
  RETURN
END

```

EXTERNAL DISTRIBUTION:

C. E. Anderson
Southwest Research Institute
P.O. Drawer 28510
San Antonio, TX 78284

M. Lax
Department of Physics
The City University of New York
New York, NY 10031

R. M. Schmidt
Boeing Aerospace Co.
P.O. Box 3999
Seattle, WA 98124

Ktech Corporation (2)
911E Pennsylvania NE
Albuquerque, NM 87112
Attn: L. Lee
E. Young

J. A. Zukas
Computational Mechanics Assoc.
P.O. Box 11314
Baltimore, MD 21239

Ballistics Research Laboratory
Aberdeen Proving Ground
Aberdeen, MD 21005
Attn: K. Kimsey

Pacifica Technology (2)
P.O. Box 148
Del Mar, CA 92014
Attn: R. Bjork
R. T. Allen

S. Peyton
S-Cubed
P.O. Box 1620
La Jolla, CA 92038

S. Schuster
California Research & Technology
20943 Devonshire St.
Chatsworth, CA 91311

R. Boade
GB105
Phillips Petroleum Co.
Bartlesville, OK 74004

Naval Surface Weapons Center
White Oak Laboratories
Silver Springs, MD 20910
Attn: J. O. Erkman, R-12

M. E. Graham
Effects Technology, Inc.
5383 Hollister Avenue
P.O. Box 30400
Santa Barbara, CA 93105

D. L. Hicks
Department of Mathematics
Michigan Technological University
Houghton, MI 49931

Kaman Sciences Corporation (3)
1500 Garden of the Gods Road
P.O. Box 7463
Attn: J. L. May
P. B. Snow
D. C. Williams

Orlando Technology, Inc. (2)
P.O. Box 855
Shalimar, Florida 32579
Attn: D. A. Matuska
J. Osborn

Lockheed Missiles & Space Co.
D/81-11 Building 154
Aerothermodynamics & Vulnerability
Sunnyvale, CA 94088
Attn: S. Chang

Lawrence Livermore National
Laboratory (9)
P.O. Box 808
Livermore, CA 94550
Attn: J. O. Hallquist, L122

D. B. Larson, L201
E. L. Lee, L368
C. E. Rosenkilde, L84
M. Ross, L299
S. Sack, L35
D. J. Steinberg, L35
M. L. Wilkins, L321
D. A. Young, L299

Los Alamos National Laboratory (9)
P.O. Box 1663
Los Alamos, NM 87545

Attn: R. G. Dupree, MS-F665
J. N. Johnson, MS-B267
R. A. Krajcik, MS-F664
J. W. Hopson, MS-B216
E. J. Linnebur, MS-B257
D. A. Marshall, MS-B257
D. A. Poling, MS-F669
J. W. Shaner, MS-J970
M. S. Shaw, MS-B214

SRI International (2)
333 Ravenswood Avenue
Menlo Park, CA 94025
Attn: L. Seaman
R. E. Tokheim

Gordon R. Johnson
Mail Station MN 11-2440
Honeywell, Inc.
Defense Systems Division
600 Second Street NE
Hopkins, MN 55343

D. P. Dandekar
Army Materials & Mechanics
Research Center
Watertown Arsenal
Watertown, MA 02171

Sandia Internal

1131 B. Morosin
1131 R. A. Graham
1131 R. E. Setchell

1131 P. A. Taylor
1221 M. M. Widner
1261 M. J. Clauser
1265 J. P. Quintenz
1265 M. A. Sweeney
1273 M. K. Matzen
1510 J. W. Nunziato
1511 D. K. Gartling
1512 M. R. Baer
1513 D. W. Larson
1520 L. W. Davison
1521 R. D. Krieg
1522 R. C. Reuter
1522 K. W. Schuler
1523 J. H. Biffle
1523 E.-P. Chen
1524 A. K. Miller
1530 W. Herrmann (Acting)
1531 S. L. Thompson
1531 T. K. Bergstresser
1531 M. Elrick
1531 L. N. Kmetyk
1531 J. M. McGlaun
1531 S. L. Passman
1531 S. Rottler
1531 J. W. Swegle
1533 P. Yarrington
1533 W. T. Brown
1533 P. J. Chen
1533 W. R. Davey
1533 A. V. Farnsworth
1533 G. I. Kerley (30)
1533 M. E. Kipp
1533 S. T. Montgomery
1533 F. R. Norwood
1533 J. Prentice
1533 A. C. Robinson
1534 J. R. Asay
1534 L. M. Barker
1534 L. C. Chhabildas
1534 J. E. Dunn
1534 M. D. Furnish
1534 D. E. Grady
1534 M. Shahinpoor

1534 T. G. Trucano
1534 J. L. Wise
1550 C. W. Peterson
1556 W. L. Oberkamp
2510 D. H. Anderson
2513 D. E. Mitchell
2514 P. L. Stanton
2515 P. D. Wilcox
3141 S. A. Landenberger (5)
3151 W. A. Garner (3)
3154-3 C. Dalin (8)
 (for DOE/OSTI)
5220 A. A. Lieber
6230 W. C. Luth
6232 W. W. Wawersik
6321 R. P. Sandoval
6321 B. D. Zak
6332 B. M. Butcher
7111 R. C. Bass
7112 A. J. Chabai
7116 R. P. Reed
7251 J. L. Russell
7533 R. A. Benham
8241 G. A. Benedetti
8241 M. L. Chiesa
8241 L. E. Voelker
8441 A. T. Jones
8524 P. W. Dean
9014 R. J. Lawrence
9122 R. H. Braasch