

LA-UR-16-26272

Approved for public release; distribution is unlimited.

Title: xRage Equation of State

Author(s): Grove, John W.

Intended for: Report

Issued: 2016-08-16

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

xRage Equation of State

John W. Grove
CCS-2 Los Alamos National Laboratory

1 Introduction

The xRage code supports a variety of hydrodynamic equation of state (EOS) models. In practice these are generally accessed in the executing code via a pressure-temperature based table look up. This document will describe the various models supported by these codes and provide details on the algorithms used to evaluate the equation of state.

2 General Theory

The continuum model for compressible hydrodynamics assumes that the thermodynamics of a material component is described by an equilibrium equation of state that relates the pressure (P), temperature (T), specific internal energy (e), specific entropy (S), and specific volume (V) of the material. An excellent summary of equations of state for equilibrium components can be found in the article of Menikoff and Plohr [15]. Briefly, the thermodynamic equilibrium properties are described by a free energy for the component. These include:

The specific internal energy:

$$e = e(V, S), \quad de = TdS - PdV. \quad (1)$$

The specific enthalpy:

$$H = H(P, S), \quad dH = TdS + VdP. \quad (2)$$

The Gibb's free energy:

$$G = G(P, T), \quad dG = -SdT + VdP. \quad (3)$$

The Helmholtz free energy:

$$F = F(V, T), \quad dF = -SdT - PdV. \quad (4)$$

These free energies are related via Legendre transformations so that any one determines the others. In particular we have:

$$\begin{aligned} F(V, T) &= \inf_S [e(V, S) - TS], \quad e(V, S) = \sup_T [F(V, T) + TS] \\ H(P, S) &= \inf_V [e(V, S) + PV], \quad e(V, S) = \sup_P [e(V, S) - PV] \\ G(P, T) &= \inf_{V, S} [e(V, S) - TS + PV], \quad e(V, S) = \sup_{P, T} [G(V, T) + TS - PV]. \end{aligned} \quad (5)$$

The free energies are constrained by thermodynamic stability so that $e(V, S)$ is a convex function:

$$e((1 - \alpha)V_0 + \alpha V_1, (1 - \alpha)S_0 + \alpha S_1) \leq (1 - \alpha)e(V_0, S_0) + \alpha e(V_1, S_1), \quad 0 \leq \alpha \leq 1. \quad (6)$$

The invertability of the Legendre transform then implies that the Gibb's free energy is a concave function of pressure and temperature:

$$G((1 - \alpha)P_0 + \alpha P_1, (1 - \alpha)T_0 + \alpha T_1) \geq (1 - \alpha)G(P_0, T_0) + \alpha G(P_1, T_1), \quad 0 \leq \alpha \leq 1. \quad (7)$$

The free energies are piecewise smooth (C^2) away from phase transitions and the differential relations in equations (1.1)(1.4) hold in smooth regions of the domain. The convexity constraint for the specific internal energies implies the positivity of the specific heat at constant pressure (C_P), the specific heat at constant volume (C_V), and that the adiabatic sound speed (c) is real:

$$\begin{aligned} C_P &= \left. \frac{\partial H}{\partial T} \right|_P = T \left. \frac{\partial S}{\partial T} \right|_P \geq C_V = \left. \frac{\partial e}{\partial T} \right|_V T \left. \frac{\partial S}{\partial T} \right|_V \geq 0, \\ c^2 &= -V^2 \left. \frac{\partial P}{\partial V} \right|_S \geq 0. \end{aligned} \quad (8)$$

Incomplete equations of state: It is very common in practice to specify the equation of state of a material via a specification of the pressure as a function of the specific internal energy and specific volume. Such a formulation is called an incomplete equation of state. For inviscid hydrodynamics, such a formulation is in fact sufficient to solve to the compressible Euler equations. What is missing is the specification of the temperature or specific entropy of the material. If one wishes to derive a complete equation of state that is compatible with a given incomplete equation of state, the basic issue is to find a temperature and specific entropy that satisfies the first law of thermodynamics:

$$TdS = de + P(e, V) dV \Leftrightarrow \left. \frac{\partial 1/T}{\partial V} \right|_e = \left. \frac{\partial P/T}{\partial e} \right|_V \Leftrightarrow P \frac{\partial T}{\partial e} - \frac{\partial T}{\partial V} = T \frac{\partial P}{\partial e}. \quad (9)$$

Equation(1.9) specifies a partial differential equation for temperature as a function of specific internal energy and specific volume. This equation is a scalar hyperbolic equation whose characteristics are the isentropes (constant specific entropy) curves defined by the incomplete equation of state:

$$\frac{de}{dV} = -P(e, V). \quad (10)$$

General theory implies that a solution to (1.9) can be found by specifying the temperature along any non-characteristic curve. The most common case is that of constant C_V so that the specific internal energy is related to the temperature by a relation of the form:

$$e - e_r(V) = C_V (T - T_r(V)). \quad (11)$$

It can further be shown that the assumption of a constant C_V implies that the incomplete equation of state is of the Grüneisen form:

$$P - P_r(V) = \frac{\Gamma(V)}{V} (e - e_r(V)). \quad (12)$$

In this case the reference curves, $P_r(V)$ and $e_r(V)$ together with the Grüneisen exponent $\Gamma(V)$ are assumed to given as part of the specification of the incomplete equation of state. Substituting equations (1.11) and (1.12) into equation (1.9) we obtain an ordinary differential equation for the reference temperature:

$$\frac{dT_r}{dV} + \frac{\Gamma(V)}{V} T_r = \frac{P_r(V) + e'_r(V)}{C_V}. \quad (13)$$

The solution of (1.13) is given by:

$$T_r(V) = e^{-\int_{V_0}^V \frac{\Gamma(v)}{v} dv} \left[T_0 + \int_{V_0}^V \frac{P_r(v) + e'_r(v)}{C_V} e^{\int_{V_0}^v \frac{\Gamma(x)}{x} dx} dv \right]. \quad (14)$$

Here T_0 is the value for the reference temperature at the specific volume V_0

At this writing, there is a bug in many of the analytic equations of state implemented in xRage that assumes the relation $e = C_V T$, which implies $e_r(V) = C_V T_r(V)$, $P_r(V) V = C_V \Gamma(V) T_r(V)$, and hence $PV = \Gamma(V) e$. Since most of the non-perfect gas equations of state have non-trivial reference curves, this means that the temperature computed by these functions is not thermodynamically consistent with the given incomplete equation of state.

2.1 Mixtures

The xRage hydrodynamic solvers are implementations of what are commonly called mixed cell hydrodynamic algorithms. Basically these are models that assume the thermodynamics of a fluid mixture can be computed from the properties of the separate component equations of state together with information about the relative quantity and distribution of the components within a computational element, specifically the mass and volume fractions of the components in the mixture. Three basic models are currently implemented:

1. Pressure-temperature equilibrium which assumes that all of the components in the mixture are at the same pressure and temperature with respect to the separate equations of state.
2. Thermal Isolation which assumes that the components specific entropies advect with the volume while remaining in pressure equilibrium.
3. Uniform strain which assumes that the components volume fractions advect with the flow while remaining in pressure equilibrium.

All three of these models assume that the components in the mixture are in velocity equilibrium with each other.

2.1.1 Pressure-Temperature Equilibrium Mixtures

Physically, the pressure-temperature equilibrium model is based on the assumption that the thermodynamics of a moving fluid element is described by a Gibb's free energy that is the mass average of the component Gibb's free energy:

$$G(P, T, \mu_1, \dots, \mu_N) = \sum_{k=1}^N \mu_k G_k(P, T). \quad (15)$$

Here μ_k is the mass fraction of component k , and $\sum_{k=1}^N \mu_k = 1$. Since conservation of mass for a non-reacting flow imply that the mass fractions advect with the common velocity of a fluid element $\frac{D\mu_k}{Dt} = \frac{\partial\mu_k}{\partial t} + \mathbf{v} \bullet \nabla \mu_k = 0$ one can regard equation (1.15) as specifying an equation of state for the mixture as a mass average of the equation of state for the components. In particular, one can show that equation (1.15) implies that the mixture specific internal energy and specific volume are obtained as mass averages of the component values at their equilibrated pressure and temperature:

$$\begin{aligned} e &= \sum_{k=1}^N \mu_k e_k(P, T) \\ V &= \sum_{k=1}^N \mu_k V_k(P, T). \end{aligned} \tag{16}$$

In practice a pressure-temperature equilibrium equation of state is implemented by solving equation (1.16) for P and T given the mixture specific internal energy e , mixture specific volume V , and the component mass fractions μ_1, \dots, μ_N . The component volume fractions α_k are then determined by the relation:

$$\alpha_k \rho_k = \mu_k \rho \Leftrightarrow \alpha_k V = \mu_k V_k = \mu_k V_k(P, T). \tag{17}$$

Perfect Gas Mixtures In general it is not possible to solve the system (1.16) analytically for general equations of state; however this is possible for the important special case of mixtures all of whose components have perfect gas equations of state:

$$\begin{aligned} PV_k &= (\gamma_k - 1) e_k \\ e_k &= C_{V,k} T. \end{aligned} \tag{18}$$

Inserting the relations in (1.18) into system (1.16) we obtain:

$$\begin{aligned} e &= \left(\sum_{k=1}^N \mu_k C_{V,k} \right) T = C_V T \\ PV &= \sum_{k=1}^N \mu_k PV_k = \sum_{k=1}^N \mu_k (\gamma_k - 1) e_k = \left(\sum_{k=1}^N \mu_k (\gamma_k - 1) C_{V,k} \right) T = (\gamma - 1) e \\ \alpha_k &= \mu_k \frac{V_k}{V} = \mu_k \frac{(\gamma_k - 1) C_{V,k}}{(\gamma - 1) C_V}. \end{aligned} \tag{19}$$

The solution is that for constant mass fractions the mixture is effectively a perfect gas with a specific heat C_V that is the mass average of the component values, and a Grüneisen exponent $\Gamma = (\gamma - 1)$ that is a heat capacity average of the component values.

2.1.2 Non-Temperature Equilibrium Mixtures

Both the thermal isolation and uniform strain models include dynamical equations for the volume fractions and thus for the purposes of the equation of state evaluation both the mass fractions and volume fractions are assumed to be given in addition to the mixture specific internal energy and specific volume. In this case we only need the incomplete equation of state $P = P_k(e_k, V_k)$ for the components in order to solve for the mixture pressure. The system of equations to be solved becomes:

$$\begin{aligned} P &= P_k(e_k, V_k) = P_k\left(e_k, \frac{\mu_k}{\alpha_k} V\right), k = 1, \dots, N \\ e &= \sum_{k=1}^N \mu_k e_k. \end{aligned} \quad (20)$$

If the component equations of state are inverted to give $e_k = e_k(P, V_k)$ then (1.20) reduces to the single equation for pressure:

$$e = \sum_{k=1}^N \mu_k e_k\left(P, \frac{\mu_k}{\alpha_k} V\right). \quad (21)$$

Thermal Isolation/Uniform Strain mixtures of Grüneisen equations of state If all of the components in the mixture are of the Grüneisen form (1.12) then the linear relation between pressure and specific internal energy for fixed specific volume means that equation (1.21) can be solved explicitly:

$$P = \frac{e - \sum_{k=1}^N \mu_k e_{r,k}(V_k)}{\sum_{k=1}^N \mu_k \frac{V_k}{\Gamma_k(V_k)}} + \frac{\sum_{k=1}^N \mu_k \frac{V_k}{\Gamma_k(V_k)} P_{r,k}(V_k)}{\sum_{k=1}^N \mu_k \frac{V_k}{\Gamma_k(V_k)}}, V_k = \frac{\mu_k}{\alpha_k} V. \quad (22)$$

It is of some interest to note that the incomplete equation of state (1.22) is also of Grüneisen form for fixed mass and volume fractions with:

$$\frac{1}{\Gamma} = \sum_{k=1}^N \frac{\alpha_k}{\Gamma_k}, P_r = \Gamma \sum_{k=1}^N \frac{\alpha_k}{\Gamma_k} P_{r,k}, e_r = \sum_{k=1}^N \mu_k e_{r,k}. \quad (23)$$

3 Implemented Equation of State Models

The xRage code supports a variety of equation of state models. In the following we will summarize details of some of the more popular models that can be used for a hydrodynamic simulation.

3.1 Perfect Gas/Gamma Law Mixtures

A mixture of perfect gases is the most common analytic equation of state for theoretical applications. This model was described in the example on page 5. A user requests this EOS model via the input specifications:

3.1.1 Perfect Gas Mixtures Input File Specifications

The list below summarizes the input deck specifications needed to use a perfect gas mixture. Again the reader is reminded that xRage only supports one equation of state model per simulation specified by the value of the input variable “**keos**”. Simulations that require mixtures of different EOS models will need to use the tabular equation of state as described in the next section.

1. **keos**: Equation of state type, **keos**= 0
2. **nummat**: Number of material components, **nummat**= N a positive integer
3. **matdef**(16,m) $1 \leq m \leq N$: Grüneisen exponent, **matdef**(16, m) = Γ a real number > 1 , $\Gamma = \gamma - 1$ where γ is the usual ratio of specific heats.
4. **matdef**(30,m): Specific heat at constant volume, **matdef**(30, m) = CV a positive real number with units of ergs/(gram eV)

3.2 Gittings Tabular Equation of State

The Gittings tabular equation of state is the workhorse EOS model in xRage. This package was designed by M. Gittings of SAIC to allow the relatively rapid computation of the pressure-temperature thermodynamics of a mixture of material components. The structure of this EOS model is a set of pressure-temperature tables of densities and specific internal energies as functions of

pressure and temperature for each material component. The key feature of this table is that all of the material component data are given along the same isobars/isotherms. In other words the tables are of the form:

$$\begin{aligned} \rho_{i,j,m} &= \rho_m(P_i, T_j) \\ e_{i,j,m} &= e_m(P_i, T_j) \end{aligned}, \quad 1 \leq i \leq n_P, \quad 1 \leq j \leq n_T, \quad 1 \leq m \leq N. \quad (24)$$

Here N is the number of material components represented in the table. In the current code (as of this writing) evaluation of the table is based on bilinear interpolation of specific volume and specific internal energy as functions of pressure and temperature:

$$\begin{aligned} V_m(P, T) &= \left(\frac{P_{i+1} - P}{P_{i+1} - P_i} \right) \left(\left(\frac{T_{j+1} - T}{T_{j+1} - T_j} \right) V_m(P_i, T_j) + \left(\frac{T - T_j}{T_{j+1} - T_j} \right) V_m(P_i, T_{j+1}) \right) + \\ &\quad \left(\frac{P - P_i}{P_{i+1} - P_i} \right) \left(\left(\frac{T_{j+1} - T}{T_{j+1} - T_j} \right) V_m(P_{i+1}, T_j) + \left(\frac{T - T_j}{T_{j+1} - T_j} \right) V_m(P_{i+1}, T_{j+1}) \right) \\ e_m(P, T) &= \left(\frac{P_{i+1} - P}{P_{i+1} - P_i} \right) \left(\left(\frac{T_{j+1} - T}{T_{j+1} - T_j} \right) e_m(P_i, T_j) + \left(\frac{T - T_j}{T_{j+1} - T_j} \right) e_m(P_i, T_{j+1}) \right) + \\ &\quad \left(\frac{P - P_i}{P_{i+1} - P_i} \right) \left(\left(\frac{T_{j+1} - T}{T_{j+1} - T_j} \right) e_m(P_{i+1}, T_j) + \left(\frac{T - T_j}{T_{j+1} - T_j} \right) e_m(P_{i+1}, T_{j+1}) \right) \end{aligned}$$

$$\begin{aligned} P_i &\leq P < P_{i+1} \\ T_j &\leq T < T_{j+1} \\ 1 &\leq m \leq N \end{aligned} \quad (25)$$

Given masses m_k (or equivalently mass fractions $\mu_k = m_k/m$, $m = \sum_{j=1}^N m_j$) for the components in a cell together with the total cell specific internal energy e and specific volume $V = 1/\rho$ the Gittings TEOS method seeks to determine a pressure and temperature such that equation (1.16) holds. The algorithm is based on a bisection type search of the tabular data. Note that since the mass fractions are fixed in this calculation it is mathematically equivalent to an inversion of a Gibb's free energy given by equation(1.15) to compute pressure and temperature given specific internal energy and specific volume. Thus for this purpose the mass fractions can be regarded as "known" EOS parameters within the solution algorithm, and we will frequently suppress the explicit dependence of the specific volume and specific internal energy as

functions of the mass fractions:

$$\begin{aligned} V(P, T) &\equiv V(P, T, \mu_1, \dots, \mu_N) = \sum_{k=1}^N \mu_k V_k(P, T) \\ e(P, T) &\equiv e(P, T, \mu_1, \dots, \mu_N) = \sum_{k=1}^N \mu_k e_k(P, T). \end{aligned} \quad (26)$$

A similar suppression is used for the thermodynamic derivatives of the mixture.

3.2.1 Tabular Equation of State Input File Specifications

The input file specifications needed to use a tabular equation of state are summarized below:

1. **keos**: Equation of state type, **keos**=3
2. **nummat**: Number of material components, **nummat**= N a positive integer
3. **eosfile**: Name of the file containing the tabular equation of state data, **eosfile**=*'file'* where the TEOS is contained in the indicated file
4. **matdef**(1,m) $1 \leq m \leq N$: Material index identifier, an unique integer identifier for a material in the table file, **matdef**(1,m)= ID a positive integer corresponding to a material EOS contained in the TEOS file.
5. **matident**(m) $1 \leq m \leq N$: Material component name (e.g. 'air', 'Sn', ...), **matident**(m)=*'NAME'* where NAME is the label that will be used for this material in printed diagnostics. The names must be unique.
6. **matdef**(25,m) $1 \leq m \leq N$: Specific internal energy / Specific Volume / Specific entropy scale, **matdef**(25,m)= sr a dimensionless positive real number. Scales the table values for specific internal energy and specific volume by the indicated value. This is basically a ratio of molecular weights and is used to create an EOS for one isotope from the EOS of another isotope of the same material. It is also used as a method to attempt to match a desired mass for a component when the state of that material is specified by a pressure and temperature. The default value of this parameter is zero, but internally a value for matdef(25,m)

$= 0$ is translated into the equivalent value of 1 (identity scaling). It is possible to have the code attempt to set the value for `matdef(25,m)` automatically. This is done by not specifying a value for `matdef(25,m)` in your input deck and instead specifying either a desired mass or density for the material using the arrays `material_density` or `material_mass`, but not both. If `material_mass(m)` is given, then a value for `matdef(25,m)` is computed to yield that specified mass and the masses in the cells containing this component are modified to yield the desired mass. Similarly if `material_density(m)` is specified then the total mass of this component is modified so the the ratio of total mass of this material divided by its total volume is equal to the input density. Note: this process can produce a mesh dependency in the resulting EOS evaluations.

7. **matdef(26,m)** $1 \leq m \leq N$: Specific internal energy translation, **matdef(26,m)** $= \Delta e_m$ a real number with units ergs/gram. The specific energy of a component at pressure P and temperature T becomes $e_m(P, T) = e_{m,table}(P, T) - \Delta e_m$. Default = 0
8. **use_old_ss**: Revert to old (wrong) sound speed formula, A logical variable that allows the reversion to an old incorrect formula for the sound speed. Default = .false.

3.2.2 Generating TEOS Tables

The data files used and input for using the Gittings' tabular equation of state are generated by running xRage using a special input deck that contains the specification of the EOS models to be incorporated into the pressure-temperature based tables together with a file name for the outputted data. The key field is to specify a value **teos_file**=*filename* in the input deck; if such an entry is found the code branches to a non-returning equation of state setup routine. The only action of the run will be to produce the TEOS file using the desired material specifications. The following list outlines the main input deck options needed to generate a TEOS table.

1. **teos_file**: TEOS file name. The name of the file that will contain the computed TEOS data, a character string at most 64 characters long.
2. **numm**: Number of EOS models. The number of EOS models to be inserted in the table. **numm**= N a positive integer.

3. **support_tension**: A logical variable which when true will allow the tabular EOS to contain negative pressures. Default = false. Note that when a run uses an EOS file that was created with the support_tension option turned on, this value is turned on as the default value in the hydro run. This will affect various cutoff's used in the hydro solvers.
4. **matid**(m), $1 \leq m \leq N$: Material index. An array of material indices, one different array per material. These are the values to be entered as **matdef**(1,m) in simulation input files. A positive integer.
5. **eostype**(m), $1 \leq m \leq N$: Equation of state type. An array of N integer identifiers for the equation of state type to be used for each material. The legal values for **eostype**(m) include:
 - (a) Notation.
 - i. P is the pressure with units microbars.
 - ii. E is the specific internal energy with units ergs/gram.
 - iii. T is the thermodynamic temperature with units of temperature equivalent electron volts.
 - iv. ρ is the mass density with units grams/cc $V = \frac{1}{\rho}$
 - (b) **eostype**(m)=0: Sesame [2, 13] equation of state using the Gittings' sesame interface
 - i. Use **matid**(m) to select the desired SESAME material.
 - (c) **eostype**(m)=99: Sesame equation of state using the EOSPAC [19] interface
 - i. Use **matid**(m) to select the desired SESAME material.
 - (d) **eostype**(m)=1: JWL [12, 4] equation of state
 - i. $P = P_r(V) + \frac{w}{V}(E - E_r(V))$
 - ii. $E = E_r(V) + C_V T$
 - iii. $P_r(V) = A e^{-R_1 \frac{V}{V_0}} + B e^{-R_2 \frac{V}{V_0}}$
 - iv. $E_r(V) = \frac{A V_0}{R_1} e^{-R_1 \frac{V}{V_0}} + \frac{B V_0}{R_2} e^{-R_2 \frac{V}{V_0}}$
 - v. $A = \text{matdef}(3,m)$ units microbars
 - vi. $B = \text{matdef}(4,m)$ units microbars
 - vii. $R_1 = \text{matdef}(10,m)$ dimensionless

- viii. $R_2 = \text{matdef}(11, \text{m})$ dimensionless
- ix. $w = \text{matdef}(16, \text{m})$ dimensionless
- x. $\rho_0 = \text{matdef}(21, \text{m})$ dimensionless, $V_0 = \frac{1}{\rho_0}$
- xi. $\rho_0 C_V = \text{matdef}(30, \text{m})$ units ergs/cc/eV
- (e) **eostype(m)=2**: Polynomial equation of state $e = C_V(T - T_0 + P_0/(\Gamma \rho_0 C_V))$,
 - i. $P = (\kappa_a + (\kappa_b + \kappa_c \mu)\mu)\mu + \Gamma \rho E$ $\mu > 0$,
 - ii. $P = \kappa_a \mu + \Gamma \rho E$ $\mu < 0$
 - iii. $\mu = \frac{\rho}{\rho_0} - 1$
 - iv. $\kappa_a = \text{matdef}(3, \text{m})$ units microbars
 - v. $\kappa_b = \text{matdef}(10, \text{m})$ units microbars
 - vi. $\kappa_c = \text{matdef}(11, \text{m})$ units microbars
 - vii. $\Gamma = \text{matdef}(16, \text{m})$
 - viii. $\rho_0 = \text{matdef}(21, \text{m})$ units grams/cc
 - ix. $T_0 = \text{matdef}(26, \text{m})$ units eV
 - x. $P_0 = \text{matdef}(28, \text{m})$ units microbars
 - xi. $C_V = \text{matdef}(30, \text{m})$ units ergs/gram/eV
- (f) **eostype(m)=3**: The Steinberg [20, 21] variation of the Mie-Grüneisen equation of state based on principal Hugoniot as reference curve with

$$u_s = c_0 + [s_1 + s_2(u_p/u_s) + s_3(u_p/u_s)^2]u_p \quad (27)$$

i.

$$P = \begin{cases} \frac{\rho_0 c_0^2 \mu [1 + (1 - \frac{\Gamma_0}{2})\mu - \frac{b}{2}\mu^2]}{[1 - (s_1 - 1)\mu - s_2 \frac{\mu^2}{1 + \mu} - s_3 \frac{\mu^3}{(1 + \mu)^2}]^2} + (\Gamma_0 + b\mu) \rho E, & \text{for } \rho \geq \rho_0 \\ \rho_0 c_0^2 \mu + \Gamma_0 \rho E, & \text{for } \rho < \rho_0 \end{cases}$$

where $\mu = \frac{\rho}{\rho_0} - 1$.

- ii. $E = C_V(T - T_0)$
- iii. NOTE: There are two inconsistencies in the above formulas. The formula for pressure is based on a misinterpretation of formula (17) in [20, 21]. In the reference the quantity E is energy per reference volume not energy per volume. Consequently, the above formula for P does not give the correct

Hugoniot.

The second inconsistency is the specific internal energy temperature relation of which the above formula is only the linearization of the thermodynamically consistent equation near the reference values.

- iv. If $\text{matdef}(2,m)$ is non-zero, then the above mentioned inconsistencies in the specific heat and specific internal energy are corrected. In this case the value of $\text{matdef}(30,m)$ specifies the reference density times the specific heat at constant volume $\rho_0 C_V = \text{matdef}(30,m)$ and the incomplete equation of state becomes:

$$P = \begin{cases} \frac{\rho_0 c_0^2 \mu [1 + (1 - \frac{\Gamma_0}{2})\mu - \frac{b}{2}\mu^2]}{[1 - (s_1 - 1)\mu - s_2 \frac{\mu^2}{1+\mu} - s_3 \frac{\mu^3}{(1+\mu)^2}]^2} + (\Gamma_0 + b\mu) \rho_0 E, & \text{for } \rho \geq \rho_0 \\ \rho_0 c_0^2 \mu + \Gamma_0 \rho E, & \text{for } \rho < \rho_0 \end{cases}$$

where $\mu = \frac{\rho}{\rho_0} - 1$. See also **eostype**(m)=11.

- v. $c_0 = \text{matdef}(3,m)$ units cm/sec
- vi. $s_1 = \text{matdef}(4,m)$ units dimensionless
- vii. $s_2 = \text{matdef}(10,m)$ units dimensionless
- viii. $s_3 = \text{matdef}(11,m)$ units dimensionless
- ix. $\Gamma_0 = \text{matdef}(16,m)$
- x. $b = \text{matdef}(17,m)$
- xi. $\rho_0 = \text{matdef}(21,m)$ units grams/cc
- xii. $T_0 = \text{matdef}(26,m)$ units eV
- xiii. $P_0 = \text{matdef}(28,m)$ units microbars - this input is ignored
- xiv. $C_V = \text{matdef}(30,m)$ units ergs/gram/eV.

NOTE: when using the quantity c_p to approximate C_V from the tables in reference [21] the user should be aware that Steinberg's c_p is a specific heat per reference specific volume. As article specifically states, the specific heat c_p was only intended to be used to compute a melt energy, but if the user wishes to use his value to approximate C_V then the value $\frac{c_p}{\rho_0}$ is an appropriate value.

- (g) **eostype**(m)=4: The Nadyozhin [18] equation of state
- (h) **eostype**(m)=5: The Mader HOM solid equation of state [14]

- i. HOM solid is a Mie-Grüneisen form equation of state with constant Grüneisen and linear $U_s - U_p$ coefficient s . See the high explosive reference manual and related documents for more details. The temperature reference curve is approximated by a logarithmic fit.
- (i) **eostype(m)=6**: The Mader HOM gas equation of state [14]
 - i. HOM gas is a Grüneisen form equation of state with logarithmic fits for the reference curves which are assumed to be isentropic curves. See the high explosive reference manual and related documents for more details.

$$P = P_r(V) + \frac{\Gamma(V)}{V} [E - E_r(V)], \quad E = E_r(V) + C_V [T - T_r(V)]$$

$$\log [P_r(V)] = \sum_{k=0}^4 p_k [\log(V)]^k, \quad \log [E_r(V) + z] = \sum_{k=0}^4 e_k [\log(V)]^k,$$

$$\log [T_r(V)] = \sum_{k=0}^4 t_k [\log(V)]^k, \quad \Gamma(V) = -\frac{V}{T_r(V)} T'_r(V)$$

- ii. $p_0 = \text{matdef}(2, \text{m})$
- iii. $p_1 = \text{matdef}(3, \text{m})$
- iv. $p_2 = \text{matdef}(4, \text{m})$
- v. $p_3 = \text{matdef}(5, \text{m})$
- vi. $p_5 = \text{matdef}(6, \text{m})$
- vii. $e_0 = \text{matdef}(7, \text{m})$
- viii. $e_1 = \text{matdef}(8, \text{m})$
- ix. $e_2 = \text{matdef}(9, \text{m})$
- x. $e_3 = \text{matdef}(10, \text{m})$
- xi. $e_5 = \text{matdef}(11, \text{m})$
- xii. $t_0 = \text{matdef}(12, \text{m})$
- xiii. $t_1 = \text{matdef}(13, \text{m})$
- xiv. $t_2 = \text{matdef}(14, \text{m})$
- xv. $t_3 = \text{matdef}(15, \text{m})$
- xvi. $t_5 = \text{matdef}(16, \text{m})$
- xvii. $cv = \text{matdef}(17, \text{m})$

- xviii. $z = \text{matdef}(18, m)$
- (j) **eostype**(m)=7: National Bureau of Standards() [8] / S-CUBED [6] water-ice equation of state
- (k) **eostype**(m)=8: The LLNL QEOS [17] equation of state
- (l) **eostype**(m)=9: Stiffened polytropic (gamma law) equation of state [10]
 - i. $P + (\Gamma + 1)P_\infty = \frac{\Gamma}{V} (E + E_\infty)$
 - ii. $E + E_\infty = C_V T + P_\infty V$
 - iii. $P_\infty = \text{matdef}(3, m)$ units microbars
 - iv. $E_\infty = \text{matdef}(10, m)$ units ergs/gram
 - v. $\Gamma = \text{matdef}(16, m)$ dimensionless
 - vi. $C_V = \text{matdef}(30, m)$ units ergs/gram/eV
- (m) **eostype**(m)=10: JWL [12, 4] equation of state with temperature specified at a given density/energy. See section 2 for the discussion of how the temperature is computed for this model.
 - i. $P = P_r(V) + \frac{w}{V} (E - E_r(V))$
 - ii. $E - E_r(V) = C_V (T - T_r(V))$
 - iii. $P_r(V) = A e^{-R_1 \frac{V}{V_0}} + B e^{-R_2 \frac{V}{V_0}}$
 - iv. $E_r(V) = -\Delta E + \frac{A V_0}{R_1} e^{-R_1 \frac{V}{V_0}} + \frac{B V_0}{R_2} e^{-R_2 \frac{V}{V_0}}$
 - v. $T_r(V) = T_0 \left(\frac{V}{V_0} \right)^{-w}$
 - vi. $A = \text{matdef}(3, m)$ units microbars
 - vii. $B = \text{matdef}(4, m)$ units microbars
 - viii. $R_1 = \text{matdef}(10, m)$ dimensionless
 - ix. $R_2 = \text{matdef}(11, m)$ dimensionless
 - x. $w = \text{matdef}(16, m)$ dimensionless
 - xi. $\rho_0 = \text{matdef}(21, m)$ dimensionless, $V_0 = \frac{1}{\rho_0}$
 - xii. $\Delta E = \text{matdef}(29, m)$ units ergs/gram
 - xiii. $C_V = \text{matdef}(30, m)$ units ergs/gram/eV
 - xiv. $T_0 = \text{matdef}(32, m)$ units eV. $T_0 = 0$ corresponds to the standard JWL EOS, see **eostype**(m)=1. Larger value of T_0 can limit the domain at low temperatures.

- (n) **eostype(m)=11**: The Steinberg [20, 21] variation of the Mie-Grüneisen equation of state with temperature specified at a given density/energy; alternative to eostype(m)=3. See section 2 for the discussion of how the temperature is computed for this model.

i.

$$P = P_r(V) + \frac{\Gamma(V)}{V} [E - E_r(V)]$$

$$E - E_r(V) = C_V [T - T_r(V)]$$

$$C_V \left[\frac{d}{dV} T_r(V) + \frac{\Gamma(V)}{V} T_r(V) \right] = P_r(V) + E_r'(V), T_r(V_0) = T_0$$

$$\Gamma(V) = \begin{cases} \Gamma_0(1 - \eta) + b\eta & , 0 \leq \eta \leq 1 \\ \Gamma_0 & , \eta < 0 \end{cases}$$

$$P_r(V) = P_0 + \rho_0 c_0^2 \begin{cases} \frac{\eta}{[1 - \eta s(\eta)]^2} & , 0 \leq \eta \leq 1 \\ \frac{[(1 - \eta)^{-(\Gamma_0 + 1)} - 1]}{\Gamma_0 + 1} & , \eta < 0 \end{cases}, \eta = \frac{V_0 - V}{V_0}$$

$$s(\eta) = s_1 + s_2\eta + s_3\eta^2$$

$$E_r(V) = E_0 + \begin{cases} \frac{P_r(V) + P_0}{2} (V_0 - V) & , 0 \leq \eta \leq 1 \\ \frac{c_0^2}{\Gamma_0 + 1} \left[\frac{(1 - \eta)^{-\Gamma_0 - 1}}{\Gamma_0} + \frac{\eta}{\Gamma_0 + 1} \right] & , \eta < 0, P_r(V) = -E_r'(V) \end{cases}$$

- ii. $c_0 = \text{matdef}(3, m)$ units cm/sec
- iii. $s_1 = \text{matdef}(4, m)$ units dimensionless
- iv. $s_2 = \text{matdef}(10, m)$ units dimensionless
- v. $s_3 = \text{matdef}(11, m)$ units dimensionless
- vi. $\Gamma_0 = \text{matdef}(16, m)$
- vii. $b = \text{matdef}(17, m)$
- viii. $\rho_0 = \text{matdef}(21, m)$ units grams/cc
- ix. $T_0 = \text{matdef}(26, m)$ units eV
- x. $P_0 = \text{matdef}(28, m)$ units microbars

- xi. $E_0 = \text{matdef}(29, \text{m})$ units microbars
 - xii. $C_V = \text{matdef}(30, \text{m})$ units ergs/gram/eV, NOTE: when using the quantity c_p to approximate C_V from the tables in reference [21] the user should be aware that Steinberg's c_p is a specific heat per reference specific volume. As article specifically states, the specific heat c_p was only intended to be used to compute a melt energy, but if the user wishes to use his value to approximate C_V then the value $\frac{c_p}{\rho_0}$ is an appropriate value.
- (o) **eostype**(m) = 16: Ideal gas with a polynomial in T specific heat at constant pressure, C_p . Two sets of coefficients are used for both a high temperature range and low temperature range. The cutoff between the two ranges is denoted by T_s below.

i.

$$\begin{aligned}
 PV &= RT \\
 \frac{C_p}{R} &= \begin{cases} \sum_{i=1}^5 a_i T^{i-1}, & T > T_s \\ \sum_{i=1}^5 a_{i+7} T^{i-1}, & T < T_s \end{cases} \\
 \frac{h}{RT} &= \begin{cases} \frac{a_6}{T} + \sum_{i=1}^5 \frac{a_i}{i} T^{i-1}, & T > T_s \\ \frac{a_{13}}{T} + \sum_{i=1}^5 \frac{a_{i+7}}{i} T^{i-1}, & T < T_s \end{cases} \\
 \frac{S}{R} &= \begin{cases} a_1 \ln T + a_7 + \sum_{i=2}^5 \frac{a_i}{i-1} T^{i-1}, & T > T_s \\ a_8 \ln T + a_{14} + \sum_{i=2}^5 \frac{a_{i+7}}{i-1} T^{i-1}, & T < T_s \end{cases}
 \end{aligned}$$

- ii. $a_i = \text{matdef}(3+i, \text{m})$ $1 \leq i \leq 14$

- A. a_1, a_8 units dimensionless
- B. a_2, a_9 units (eV)⁻¹
- C. a_3, a_{10} units (eV)⁻²
- D. a_4, a_{11} units (eV)⁻³
- E. a_5, a_{12} units (eV)⁻⁴
- F. a_6, a_{13} units eV

G. a_7, a_{14} units dimensionless

NOTE: A table of a_i values for various common atoms and molecules can be found here: <http://www.me.berkeley.edu/gri-mech/version30/files30/thermo30.dat>. An explanation of the data in the tables is given here: http://www.me.berkeley.edu/gri-mech/data/nasa_plnm.html. These tables are accurate only within the temperature ranges given in the table (typically 200 K to a few 1000 K, hence the default limits for T_{\min} and T_{\max} below), and it is not recommended to use them above that range.

- iii. $T_{\min} = \text{matdef}(18, \text{m})$ units eV. Default = 200/EV_PER_KELVIN eV $\simeq 0.017$ eV (i.e., 200 K). For temperatures below this value, EOS quantities are extrapolated based on constant specific heats.
- iv. $T_{\max} = \text{matdef}(19, \text{m})$ units eV. Default = 3500/EV_PER_KELVIN eV $\simeq 0.302$ eV (i.e., 3500 K). For temperatures above this value, EOS quantities are extrapolated based on constant specific heats.
- v. $T_s = \text{matdef}(20, \text{m})$ units eV. Default = 1000/EV_PER_KELVIN eV $\simeq 0.086$ eV (i.e., 1000 K).
- vi. $R = \text{matdef}(30, \text{m})$ units ergs/grams/eV
- vii. $\rho_0 = \text{matdef}(21, \text{m})$ units grams/cc
- viii. $T_0 = \text{matdef}(26, \text{m})$ units eV
- ix. $P_0 = \text{matdef}(28, \text{m})$ units μbar
NOTE: In the current implementation of this EOS, ρ_0 , T_0 , and P_0 have no purpose, so it does not matter what you set them to.
- x. NOTE: There is no assumption that C_p , h , or S are continuous at T_s . The user can impose this condition by ensuring that

$$\sum_{i=1}^5 a_i T_s^{i-1} = \sum_{i=1}^5 a_{i+7} T_s^{i-1}$$

$$\frac{a_6}{T_s} + \sum_{i=1}^5 \frac{a_i}{i} T_s^{i-1} = \frac{a_{13}}{T_s} + \sum_{i=1}^5 \frac{a_{i+7}}{i} T_s^{i-1}$$

$$a_1 \ln T_s + a_7 + \sum_{i=2}^5 \frac{a_i}{i-1} T_s^{i-1} = a_1 \ln T_s + a_{14} + \sum_{i=2}^5 \frac{a_{i+7}}{i-1} T_s^{i-1}$$

6. Pressure/Temperature Mesh Control

(a) Standard pressure grid controls:

- i. **prsl0**: Minimum pressure. Minimum isobar pressure value, P_{lo} . A real number with units of microbars. Default = $10^4 \mu\text{bar}$.
- ii. **prshi**: Maximum pressure. Maximum isobar pressure value, P_{hi} . A real number greater than P_{lo} with units of microbars. Default = $10^{16} \mu\text{bar}$
- iii. **numprsdec**: Number of pressure decades. Number of pressure decades, N_{dec} , to insert in the table. A positive integer. Default = 20

When $P_{lo} > 0$ the table will contain isobars at pressures

$$P_i = P_{lo} 10^{\frac{i-1}{N_{dec}}}, 1 \leq i \leq N_p = 1 + N_{dec} \log_{10} \frac{P_{hi}}{P_{lo}}.$$

(b) Finer pressure grid controls.

- i. **pscale** Pressure scale P_{σ} . A positive real number of units microbars. This becomes the default value for `pscale_neg` and `pscale_pos` as used in the above formula for the pressure grid points. Default = $10^4 \mu\text{bar}$.
- ii. **pscale_pos** Positive pressure scale $P_{\sigma+}$. A positive real number of units microbars. Defines the pressure scale coefficient for positive pressures. Default = `pscale`.
- iii. **pscale_neg** Negative pressure scale $P_{\sigma-}$. A positive real number of units microbars. Defines the pressure scale coefficient for positive pressures. Default = `pscale`.
- iv. **numprsdec_pos** Number of positive pressure decades N_{Pdec+} . A positive integer. Controls the number of points for positive pressures. Default = `numprsdec`.
- v. **numprsdec_neg** Number of negative pressure decades N_{Pdec-} . A positive integer. Controls the number of points for negative pressures. Default = `numprsdec`.

These options were primarily introduced to provide better control of the pressure grid when tension support is requested. When $P_{lo} < 0$ the pressure grid is given by:

$$P_i = \begin{cases} P_{lo} 10^{\frac{1-i}{N_{Pdec-}}} + P_{\sigma-} \left(1 - 10^{\frac{1-i}{N_{Pdec-}}}\right), & 1 \leq i \leq N_{p-} \\ 0, & i = N_{p-} + 1 \\ P_{\sigma+} \left(10^{\frac{i-N_{p-}-1}{N_{Pdec+}}} - 1\right) & N_{p-} + 2 \leq i \leq N_p \end{cases}$$

$$N_{p-} = N_{Pdec-} \log_{10} \left(1 - \frac{P_{lo}}{P_{\sigma-}}\right), N_{p+} = N_{Pdec+} \log_{10} \left(1 + \frac{P_{hi}}{P_{\sigma+}}\right), N_p = N_{p-} + 1 + N_{p+}$$

The above options allow for separate exponential spacing for the positive and negative pressure axes and help ensure the EOS is adequately fine to capture the structure of the original EOS. A suggested value for **prsl** might be the minimum pressure supported by the equation of state. For SESAME table one can use the Amhctools utility "SesameRange" to quickly obtain these values. They can also be discovered using the T-1 EOS tool plot-sesamebin.x available on Los Alamos ICN machines. The choice for the **pscale_pos** and

- (c) **pscale_neg** should be based on how much resolution is required in the EOS table near zero pressure. Usually a value on the order of a bar or so is sufficient. Remember that the pressure units for the code are microbar, so to get a one bar pressure scale means using a value of a million in the input deck. The number of decades variables should be used to ensure a sufficiently fine pressure grid so that the bilinear interpolation used to evaluate the EOS are relatively accurate. There is no hard and fast number here, and the spacing should really depend on estimates of the second derivatives of the EOS data with respect to pressure and temperature. In practice using a value in the range of 20-50 seems to work. The higher values will greatly increase the size of the EOS table.
- (d) Standard temperature grid controls:
 - i. **tevlo**: Minimum temperature Minimum isotherm temperature value, T_{lo} A positive real number of units electron volts. Default = 10^{-2} eV

- ii. **tevhi**: Maximum temperature. Minimum isotherm temperature value, T_{lo} . A positive real number greater than T_{lo} of units electron volts. Default = 10^6 eV
- iii. **numtevedec**: Number of temperature decades. Number of temperature decades N_{Tdec} to insert in the table. A positive integer. Default = 20

If **tevlo** is positive, then by default the temperature grid contains isotherms at temperatures:

$$T_j = T_{lo} 10^{\frac{j-1}{N_{Tdec}}}, 1 \leq j \leq N_T = 1 + N_{Tdec} \log_{10} \frac{T_{hi}}{T_{lo}}$$

- (e) **Finer temperature grid Controls**: When tension is being supported or when the EOS needs more data in the colder regions of phase space, the default controls for the temperature grid result in very few low temperature isotherms. For applications with tension or even for those for which the lower temperature limit is desired to be zero, the temperature mesh can be modified by the following values.

- i. **tevlo** = 0. When there are no other modifications to the temperature grid controls other than setting the lower bound temperature to be zero, the isotherms in the EOS table are given by:

$$T_j = 10^{\frac{j-1}{N_{Tdec}}} - 1, 1 \leq j \leq N_T = 1 + N_{Tdec} \log_{10} (1 + T_{hi})$$

- ii. **tevhi_tension**. Specifies an upper bound, T_{ten} of temperatures whose isotherms should contain tension regions. A positive real number of units electron volts. Default = **tevlo**.
- iii. **num_tevtension**. The number of isotherms, N_{Tten} required in the table for temperatures less than or equal to **tevhi_tension**. Default = 0.

When $T_{ten} > T_{lo}$ and $N_{Tten} > 0$ then tabulated isotherms are given by:

$$T_j = \begin{cases} T_{lo} + (j-1) \frac{T_{ten}-T_{lo}}{N_{Tten}}, & 1 \leq j \leq N_{Tten} \\ T_{ten} 10^{\frac{j-(N_{Tten}+1)}{N_{Tdec}}}, & N_{Tten} + 1 \leq j \leq N_T = N_{Tten} + 1 + N_{Tdec} \log_{10} \frac{T_{hi}}{T_{ten}} \end{cases}$$

When using SESAME, a suggested value for **tevhi_tension** might be the maximum temperature for all of the tables being considered whose isotherm data contains a negative pressure, and the value for **num_tevtension** might be the maximum of the number of isotherms that contain negative pressures.

7. **sesfiles**: Sesame file list. An array of up to twenty file names for files containing SESAME equation of state data. Array of character strings Default = array of blanks (Appropriate if no sesame tables are being used.)
8. **matdef**(*m)), $1 \leq m \leq N$: EOS parameters. Specific values and array indices depend on the EOS model. Real numbers or integers as appropriate, the units are variable.
9. **force_code**: Force absolute zero isotherm. Sets the minimum temperature isotherm value to $10^{-6}eV$. Errors will occur if **tevlo** is less than this value. A logical variable. Default = .false.
10. **auto_teos**: Automatic EOS table setup. If true, create isotherms according to formula (28). Ignores the input values of **tevlo**, **tevhi**, and **numtevdec** A logical variable. Default = .false.
The isotherm formula for **auto_tev** is given by:

$$\begin{aligned}
 T_1 = \begin{cases} T_{cold} & \text{force_cold} = \text{.true.} \\ \Delta T_{fixed} eV & \text{force_cold} = \text{.false.} \end{cases}, T_{cold} = 10^{-6}eV, \Delta T_{fixed} = 10^{-3}eV \\
 T_j = \Delta T_{fixed} j, T_j \leq T_{fixed} = 0.05eV \\
 T_j = \sigma_j T_{j-1}, \sigma_j = 1 + \frac{\Delta T_{fixed}}{T_{fixed}}, T_{fixed} < T_j \leq T_{super} = 0.1eV \\
 T_j = \sigma_j T_{j-1}, T_{super} < T_j, \sigma_j = \min(10^1, 1.005 \times \sigma_{j-1}), T_j \leq T_{hi} = 10^6eV.
 \end{aligned}
 \tag{28}$$

11. **mxdome_itr**: Maximum Maxwell Construction iterations. Sets a maximum value for the number of internal iterations used to perform a Maxwell construction on a sesame table to eliminate van der Waals loops. The default value should be sufficient to process any reasonable sesame table. The user can increase this value as an attempt to force the code to succeed. But if the loop construction is failing, this is indicative of a faulty sesame table and it is unlikely that the code will be able to process this file. An integer variable. Default = 10,000

Debugging and Other Diagnostic Variables

1. Sesame Debugging Flags:
 - (a) **list_301**: List 301 tables. A logical flag, if true the SESAME 301 tables for the requested materials will be printed and no TEOS table will be created. A logical variable. Default = .false.
 - (b) **list_index**: List table indices. A logical flag, if true the SESAME material indices for the requested materials will be printed and no TEOS table will be created. A logical variable. Default = .false.
 - (c) **list_tid**: An unused integer input variable. Default = 0
2. **list_cross**: List crossed isotherms, Logical flag, if true then print any crossed isotherms detected during the table setup. A logical variable. Default = .true
3. Table build tests: If both **test_prs_sig** = σ_P and **test_tev_sig** = σ_T are positive the inverted P-T tables will be checked against the original sesame data at the points $(P_{lo}\sigma_P^i, T_{lo}\sigma_T^j), P_{lo} \leq P_{lo}\sigma_P^i \leq P_{hi}, T_{lo} \leq T_{lo}\sigma_P^j \leq T_{hi}$
 - (a) **test_prs_sig**, Pressure check points. A non-negative real number of units microbars. Default = 0 μbar
 - (b) **test_tev_sig**, Temperature check points. A non-negative real number of units electron volts. Default = 0 eV
4. **debug_eos**: Debug EOS. A logical flag to print out the table data if requested. The filenames are of the form “*matI.J.txt*” where *I* is the material id for a component and *J* corresponds to a step in the table setup algorithm (see below). A logical variable. Default = .false.

3.2.3 EOS Modifications During Table Creation

Several modifications of the equation of state data can occur during the creation of the TEOS tables. In particular SESAME data may be modified using a Maxwell construction [7, 9] to replace van der Waals loops in isotherms.

3.2.4 Implicit assumptions on the Equation of State implemented by TEOS

In the description of the solution algorithm for the TEOS equations of state, several implicit assumptions regarding the nature of the component equation of states turn up. The list below summarizes these assumptions.

1. The equation of state is consistent in the sense that the partial derivatives of specific internal energy and density satisfy the condition that there exists a Gibb's free energy for the flow:

$$\begin{aligned} dG &= -SdT + VdP \\ e &= G - PV + TS. \end{aligned} \quad (29)$$

1. The equation of state is stable, in particular the specific heats at constant volume are non-negative and the sound speed is real:

$$\begin{aligned} C_P &= T \left. \frac{\partial S}{\partial T} \right|_P \geq C_V = T \left. \frac{\partial S}{\partial T} \right|_V \geq 0 \\ c^2 &= -V^2 \left. \frac{\partial P}{\partial V} \right|_S \geq 0. \end{aligned} \quad (30)$$

Conditions 1 and 2 are actually generic restrictions for all thermodynamic equations of state and should be satisfied with the domain of validity of the EOS. The other conditions in this list need not be satisfied in general, but are assumed (or imposed) by the TEOS solution algorithm.

1. The specific internal energy is monotone non-decreasing as a function of temperature along an isobar:

$$\left. \frac{\partial e}{\partial T} \right|_P \geq 0. \quad (31)$$

1. The Grüneisen exponent is non-negative:

$$\Gamma = V \left. \frac{\partial P}{\partial e} \right|_V = - \left. \frac{\partial \log T}{\partial \log V} \right|_S \geq 0. \quad (32)$$

1. The pressure is a monotone non-increasing function of specific volume for constant energy:

$$\left. \frac{\partial P}{\partial V} \right|_e \leq 0 \quad (33)$$

1. For materials in a “*gas*” phase (see below), the specific internal energy does not depend on pressure along the lowest isobar in the table.

3.2.5 Table Data Setup

During initialization a user supplied data file containing the pressure/temperature based table of component specific internal energies and densities is read (see the subroutine *teos_load_select* () in module *teos.f90*). These files are usually called TEOS files and often are given a suffix of “.teos”. This file instantiates the arrays for energy and density defined in equation(2.1). This process includes a “culling” of the data provided in the TEOS file to limit the number of materials in the internal arrays to only those materials specifically requested by the users input deck.

Once the density and specific internal energy tables are assigned the code will create and store estimates for the specific volume and specific internal energy together with their pressure derivatives along the minimum isobar in the table:

$$\begin{aligned} V_{k, floor}(T_j) &= V_k(P_1, T_j), \quad \left. \frac{\partial V_{k, floor}}{\partial P} \right|_T(T_j) = \left. \frac{\partial V_k}{\partial P} \right|_T(P_1, T_j) \\ e_{k, floor}(T_j) &= e_k(P_1, T_j), \quad \left. \frac{\partial e_{k, floor}}{\partial P} \right|_T(T_j) = \left. \frac{\partial e_k}{\partial P} \right|_T(P_1, T_j). \end{aligned} \quad (34)$$

The specific internal energy derivatives are approximated using simple finite differences:

$$\left. \frac{\partial e_k}{\partial P} \right|_T(P_1, T_j) \approx \frac{e_k(P_2, T_j) - e_k(P_1, T_j)}{P_2 - P_1}. \quad (35)$$

The pressure derivatives of the component specific volumes along the pressure floor isotherm are approximated by a rather complicated method. First the component density derivatives are approximated along floor isotherm using finite differences:

$$\left. \frac{\partial \rho_k}{\partial P} \right|_T(P_1, T_j) \approx \frac{\Delta \rho_k}{\Delta P} \Big|_T(P_1, T_j) = \frac{\rho(P_2, T_j) - \rho(P_1, T_j)}{P_2 - P_1}. \quad (36)$$

Next the pressure is extrapolated to zero density along each isotherm:

$$\tilde{P}_{0,j,k} = P_1 - \frac{\rho_k(P_1, T_j)}{\left. \frac{\Delta \rho_k}{\Delta P} \right|_T(P_1, T_j)}. \quad (37)$$

Two cases are now considered, if $\tilde{P}_{0,j,k} > -P_1$ the component is said to be in a *gas* phase. In this case it is assumed that the density goes to zero as the pressure vanishes along the isotherm, so that the derivative of specific

volume is approximated by:

$$\left. \frac{\partial V_k}{\partial P} \right|_T (P_1, T_j) = -\frac{1}{\rho_k^2} \left. \frac{\partial \rho_k}{\partial P} \right|_T (P_1, T_j) \approx -\frac{1}{\rho_k^2} \frac{\rho_k(P_1, T_j) - \rho_k(0, T_j)}{P_1 - 0} = -\frac{V_k(P_1, T_j)}{P_1}. \quad (38)$$

Otherwise the derivative is approximated using the extrapolated pressure $\tilde{P}_{0,j,k}$ with the assumption that density vanishes at this pressure:

$$\left. \frac{\partial V_k}{\partial P} \right|_T (P_1, T_j) = -\frac{1}{\rho_k^2} \left. \frac{\partial \rho_k}{\partial P} \right|_T (P_1, T_j) \approx -\frac{1}{\rho_k^2} \frac{\rho_k(P_1, T_j) - \rho_k(\tilde{P}_{0,j,k}, T_j)}{P_1 - \tilde{P}_{0,j,k}} = -\frac{V_k(P_1, T_j)}{P_1 - \tilde{P}_{0,j,k}}. \quad (39)$$

The intent of this approximate of derivatives is to ensure that the density goes to zero as $P \rightarrow \tilde{P}_{0,j,k}$ for fixed $T = T_j$

3.2.6 Pressure Floor Estimate

The first step is to locate bounding isotherms for the specific internal energy evaluated at the floor isobar. Specifically find j_s so that:

$$\sum_{k=1}^N \mu_k e_k(P_1, T_{j_s}) \leq e \leq \sum_{k=1}^N \mu_k e_k(P_1, T_{j_s+1}). \quad (40)$$

Inequality (2.16) points out an implicit assumption about the equation of state models represented by the tabular data, namely that for each component, the specific internal energy is a non-decreasing function of temperature for fixed pressure:

$$\left. \frac{\partial e_k}{\partial T} \right|_P \geq 0. \quad (41)$$

Using the identities in the reference of Menikoff and Plohr [15] it can be shown that inequality (2.17) is equivalent to the inequality:

$$C_{P,k} \left[1 - \Gamma_k \frac{P}{\rho_k c_k^2} \right] = C_{P,k} \left[1 - \frac{\Gamma_k}{\gamma_k} \right] \geq 0. \quad (42)$$

Here $c^2 = \left. \frac{\partial P}{\partial \rho} \right|_S$ is the adiabatic sound speed squared (ρc^2 is called the bulk modulus), $\gamma = c^2/PV$ is the adiabatic exponent, and $\Gamma = -\left. \frac{\partial \log T}{\partial \log V} \right|_S = V \left. \frac{\partial P}{\partial e} \right|_\rho$ is the Grüneisen exponent. Inequalities (2.17) or (2.18) need not be true for

general equations of state, but are generally satisfied for equations of state that are used in practice.

Given the isotherms bounding the specific internal energy at the floor pressure a floor volume and temperature are computed using linear interpolation:

$$\begin{aligned} T_{floor} &= T_{j_s} + s (T_{j_{s+1}} - T_{j_s}), \quad V_{floor} = V_{j_s} + s (V_{j_{s+1}} - V_{j_s}) \\ s &= \max \left(\frac{e - e_{j_s}}{e_{j_{s+1}} - e_{j_s}}, 0 \right) \\ e_j &= \sum_{k=1}^N \mu_k e_k (P_j, T_j), \quad V_j = \sum_{k=1}^N \mu_k V_k (P_j, T_j). \end{aligned} \quad (43)$$

The specific heat at constant volume is also estimated at the point (P_1, T_{floor}) using the finite difference approximation:

$$C_V (P_1, T_{floor}) = \left. \frac{\partial e}{\partial T} \right|_V (P_1, T_{floor}) \approx \frac{e(P_1, T_{j_{s+1}}) - e(P_1, T_{j_s})}{T_{j_{s+1}} - T_{j_s}}. \quad (44)$$

Note that this is in fact possibly a poor approximation for C_V since it is actually a finite difference approximation for $\left. \frac{\partial e}{\partial T} \right|_P = T \left. \frac{\partial S}{\partial T} \right|_P - P \left. \frac{\partial V}{\partial T} \right|_P = C_P - P \left. \frac{\partial V}{\partial T} \right|_P$ in the sense that it is a temperature finite difference along the pressure floor isobar. In fact, for a general equation of state one can show that (see Menikoff & Plohr [15] for example) that:

$$\begin{aligned} \left. \frac{\partial e}{\partial T} \right|_P &= C_P - P \left. \frac{\partial V}{\partial T} \right|_P = C_P \left[1 - \frac{\Gamma P}{\rho c^2} \right] = C_V \left[\frac{1 - \frac{\Gamma P}{\rho c^2}}{1 - \frac{\Gamma^2 C_V T}{c^2}} \right] \\ C_V &= \frac{\left. \frac{\partial e}{\partial T} \right|_P}{1 + \frac{\Gamma^2 T}{c^2} \left. \frac{\partial e}{\partial T} \right|_P - \frac{\Gamma P}{\rho c^2}}. \end{aligned} \quad (45)$$

Pressure is then extrapolated along the PV isotherm through the point (P_1, V_{floor}) to zero specific volume:

$$P_{zero} = P_1 + \frac{V_{floor}}{\left. \frac{\partial V}{\partial P} \right|_T (P_1, T_{floor})}. \quad (46)$$

The slope in the denominator of (2.22) is approximated by linear interpolation:

$$\begin{aligned} \left. \frac{\partial V}{\partial P} \right|_T (P_1, T_{floor}) &= \sum_{k=1}^N \mu_k \left. \frac{\partial V_k}{\partial P} \right|_T (P_1, T_{floor}) \approx \\ &\sum_{k=1}^N \mu_k \left[\left. \frac{\partial V_k}{\partial P} \right|_T (P_1, T_{j_s}) + s \left(\left. \frac{\partial V_k}{\partial P} \right|_T (P_1, T_{j_{s+1}}) - \left. \frac{\partial V_k}{\partial P} \right|_T (P_1, T_{j_s}) \right) \right]. \end{aligned} \quad (47)$$

Similarly we can compute the derivative of specific internal energy with respect to pressure as:

$$\begin{aligned} \frac{\partial e}{\partial P}\bigg|_T(P_1, T_{floor}) &= \sum_{k=1}^N \mu_k \frac{\partial e_k}{\partial P}\bigg|_T(P_1, T_{floor}) \approx \\ &\sum_{k=1}^N \mu_k \left[\frac{\partial e_k}{\partial P}\bigg|_T(P_1, T_{j_s}) + s \left(\frac{\partial e_k}{\partial P}\bigg|_T(P_1, T_{j_s+1}) - \frac{\partial e_k}{\partial P}\bigg|_T(P_1, T_{j_s}) \right) \right]. \end{aligned} \quad (48)$$

We will also need the derivatives of specific volume and specific internal energy with respect to temperature at the point (P_1, T_{floor}) . These are approximated using simple finite differences:

$$\begin{aligned} \frac{\partial V}{\partial T}\bigg|_P(P_1, T_{floor}) &= \sum_{k=1}^N \mu_k \frac{\partial V_k}{\partial T}\bigg|_P(P_1, T_{floor}), \quad \frac{\partial V_k}{\partial T}\bigg|_P(P_1, T_{floor}) \approx \frac{V_k(P_1, T_{j_s+1}) - V_k(P_1, T_{j_s})}{T_{j_s+1} - T_{j_s}} \\ \frac{\partial e}{\partial T}\bigg|_P(P_1, T_{floor}) &= \sum_{k=1}^N \mu_k \frac{\partial e_k}{\partial T}\bigg|_P(P_1, T_{floor}), \quad \frac{\partial e_k}{\partial T}\bigg|_P(P_1, T_{floor}) \approx \frac{e_k(P_1, T_{j_s+1}) - e_k(P_1, T_{j_s})}{T_{j_s+1} - T_{j_s}}. \end{aligned} \quad (49)$$

Once P_{zero} is computed using equation (2.22) two cases are again considered, *gases* for which $P_{zero} > -P_1$ and *solids* for $P_{zero} \leq -P_1$

Pressure floor estimates for gases: For *gases* it is assumed that P_{zero} is effectively zero and the derivative $\frac{dP}{d\rho}\bigg|_T(P_{zero}, T_{floor})$ is approximated using an extrapolated finite difference between P_1 and $P_{zero} = 0$ with the assumption that $\rho(0, T_{floor}) = 0$:

$$\begin{aligned} \frac{dP}{d\rho}\bigg|_T(P_{zero}, T_{floor}) &\approx \frac{P_1 - P_{zero}}{\rho(P_1, T_{floor}) - \rho(P_{zero}, T_{floor})} = \\ &\frac{P_1}{\rho(P_1, T_{floor}) - \rho(0, T_{floor})} = \frac{P_1}{\rho(P_1, T_{floor})} = P_1 V_{floor}. \end{aligned} \quad (50)$$

The pressure first guess is then computed using a linear approximation for the isotherm through the point (P_{zero}, T_{floor}) , $P^0 = P_1 V_{floor} \rho = \frac{P_1 V_{floor}}{V}$ and the first guess for the solution temperature is taken as $T^0 = T_{floor}$. First guesses for the Grüneisen exponent (actually $\rho\Gamma$) and the bulk modulus ρc^2

using the formula:

$$\begin{aligned}
\rho \Gamma^0(P^0, T^0) &= \frac{\partial P}{\partial e} \Big|_V(P^0, T^0) = \\
&= \frac{\sum_{k=1}^N \mu_k \frac{\partial V_k}{\partial T} \Big|_P}{\left(\sum_{k=1}^N \mu_k \frac{\partial V_k}{\partial T} \Big|_P \right) \left(\sum_{k=1}^N \mu_k \frac{\partial e_k}{\partial P} \Big|_T \right) - \left(\sum_{k=1}^N \mu_k \frac{\partial V_k}{\partial P} \Big|_T \right) \left(\sum_{k=1}^N \mu_k \frac{\partial e_k}{\partial T} \Big|_P \right)} (P^0, T^0) = \\
&= \frac{\frac{\partial V}{\partial T} \Big|_P}{\left(\frac{\partial V}{\partial T} \Big|_P \right) \left(\frac{\partial e}{\partial P} \Big|_T \right) - \left(\frac{\partial V}{\partial P} \Big|_T \right) \left(\frac{\partial e}{\partial T} \Big|_P \right)} (P^0, T^0) \\
\rho c^2(P^0, T^0) &= P^0 \Gamma^0 + \rho \frac{\partial P}{\partial \rho} \Big|_e(P^0, T^0).
\end{aligned} \tag{51}$$

The TEOS code also makes the implicit assumption that the Grüneisen exponent is non-negative, so if the quantity computed in (2.27) is less than zero it is set to zero. The derivatives in (2.27) along the floor isobar are approximated by the values computed for the point (P_1, T_{floor}) as described in equations (2.23)(2.25).

The derivative of pressure with respect to density at constant energy can be written:

$$\frac{\partial P}{\partial \rho} \Big|_e = \frac{\partial P}{\partial \rho} \Big|_T \left[1 - \frac{\left(\frac{\partial V}{\partial T} \Big|_P \right) \left(\frac{\partial e}{\partial P} \Big|_T \right)}{\left(\frac{\partial V}{\partial P} \Big|_T \right) \left(\frac{\partial e}{\partial T} \Big|_P \right)} \right]^{-1}. \tag{52}$$

For “*gases*” this derivative is approximated using the implicit assumption that energy does not depend on pressure along the floor isobar, $\frac{\partial e}{\partial P} \Big|_T(P_1, T) \approx 0$ and thus $\frac{\partial P}{\partial \rho} \Big|_e \approx \frac{\partial P}{\partial \rho} \Big|_T$ as calculated in equation(2.26).

Pressure floor estimates for solids: In the case of a solid ($P_{zero} \leq -P_1$) the density derivative at the pressure floor isobar is estimated as:

$$\frac{\partial P}{\partial \rho} \Big|_T(P_1, T_{floor}) = -\frac{V_{floor}^2}{\frac{\partial V}{\partial P} \Big|_T(P_1, T_{floor})}. \tag{53}$$

Where the right hand side quantities are given in equations (2.19) and(2.23). Pressure is then extrapolated linearly assuming that the density vanishes along the isotherm at pressure P_{zero} :

$$\begin{aligned}
P^0 &= P_{zero} + \frac{\partial P}{\partial \rho} \Big|_T(P_{zero}, T_{floor}) (\rho - \rho(P_{zero}, T_{floor})) = P_{zero} + \frac{1}{V} \frac{\partial P}{\partial \rho} \Big|_T(P_{zero}, T_{floor}) \\
&\quad \frac{\partial P}{\partial \rho} \Big|_T(P_{zero}, T_{floor}) \approx \frac{\partial P}{\partial \rho} \Big|_T(P_1, T_{floor}).
\end{aligned} \tag{54}$$

The derivative of pressure with respect to density at fixed energy is approximated at the solution point:

$$\begin{aligned} \left. \frac{\partial P}{\partial \rho} \right|_e (P^0, T^0) &= -V^2 \left. \frac{\partial P}{\partial V} \right|_e (P^0, T^0) = - \frac{V^2}{\left(\frac{\frac{\partial V}{\partial P}|_T \frac{\partial e}{\partial T}|_P - \frac{\partial V}{\partial T}|_P \frac{\partial e}{\partial P}|_T \right)} (P^0, T^0) \approx \\ &= - \frac{V^2}{\left(\frac{\frac{\partial V}{\partial P}|_T \frac{\partial e}{\partial T}|_P - \frac{\partial V}{\partial T}|_P \frac{\partial e}{\partial P}|_T \right)} (P_1, T_{floor}). \end{aligned} \quad (55)$$

As before, the derivatives in the above relation are approximated using those in equations (2.23)(2.25). Finally the bulk modulus at (P^0, T^0) is approximated as in equation(2.27)

3.2.7 Iteration Scheme for Computing the P-T Equilibrium Solution

The calculation described in the previous section returns an initial guess for the P-T equilibrium solution together with thermodynamic derivatives at that location. Specifically it returns:

1. The isotherm indices j_s and $j_s + 1$ bounding the desired specific internal energy e at the floor pressure P_1 $e(P_1, T_{j_s}) \leq e \leq e(P_1, T_{j_s+1})$.
2. The initial estimates for the solution interpolated or extrapolated off the pressure floor isobar (P^0, T^0) .
3. Estimates for the specific heat at constant volume C_V , the bulk modulus ρc^2 , and the density times the Grüneisen exponent $\rho \Gamma = \left. \frac{\partial P}{\partial e} \right|_V$ at the point (P^0, T^0)

The first step in the solution iteration algorithm is to test the estimated pressure value P^0 against the pressure floor value P_1 . If $P^0 \leq P_1$ then it is assumed that the solution to the equilibrium equation lies below the range of the pressure table and the initial guesses described above are taken as the answer to the equilibrium calculation. This is perhaps a very risky assumption given the crudeness of the approximations made in computing this initial guess and could be a source of error for solutions that lie near the bottom of and pressure table. This bound check is in fact the only usage of the estimated pressure P^0 in the solution algorithm.

Once it has been determined that $P_1 < P^0$ the iteration is started at a pressure equal to the value of pressure provided as input data to the subroutine, P_{old} . The basic iteration can be summarized as:

1. Given $P^n (P^0 = P_{old})$ find the temperature T^n so that $e(P^n, T^n) = e$
2. Perform a Newton iteration along the specific internal energy level curve $e(P, T) = e$ through (P^n, T^n) to compute a new pressure at the desired density:

$$\begin{aligned} P^{n+1} &= P^n + \left. \frac{\partial P}{\partial \rho} \right|_e (P^n, T^n) (\rho - \rho^n) \\ \rho &= \frac{1}{V}, \quad \rho^n = \frac{1}{V(P^n, T^n)}. \end{aligned} \quad (56)$$

1. The scheme converges when either of the two conditions:

$$\begin{aligned} |\rho - \rho^n| &\leq \varepsilon_\rho |\rho + \rho^n| \\ |P^{n+1} - P^n| &\leq \varepsilon_P |P^{n+1} + P^n|. \end{aligned} \quad (57)$$

The actual tolerances used in the code are currently set to $\varepsilon_\rho = 10^{-12}$ and $\varepsilon_P = 10^{-7}$

3.3 Sesame Equations of State

The SESAME equation of state [1] is a family of equation of state models supported type the Theoretical Division's Equation of State Group [3] at the Los Alamos National Laboratory that are made available as tables of pressure and specific internal energy as functions of density and temperature. A detailed description of the structure of the SESAME tables can be found at the web site [2]. In addition to hydrodynamic equations of state, the SESAME format also supports other constitutive functions such as mean opacities and conductivities but our discussion here will only focus on the thermodynamic equation of state models represented in a SESAME table.

The basic structure of a SESAME EOS table is to provide matrices of specific internal energy and pressure as functions of density and temperature:

$$\begin{aligned} e_{i,j} &= e(\rho_i, T_j) \\ P_{i,j} &= P(\rho_i, T_j) \end{aligned}, \quad 1 \leq i \leq n_\rho, \quad 1 \leq j \leq n_T \quad (58)$$

Equation of state evaluations at arbitrary densities and temperatures are made using various interpolation methods. In particular the two most common methods are bilinear interpolation (fast but not very accurate), and a

more accurate rational interpolation scheme described in the report of Kerley [11]. Both methods are implemented in the library EOSPAC [19] and are suitable for many hydrodynamic solvers. The main deficiency of these interpolation schemes (especially the bilinear method) is that the derivatives of the interpolant are discontinuous across the density/temperature mesh boundaries. This can cause some difficulties for higher-order Godunov type schemes that make extensive use of the sound speed since this quantity is evaluated using such derivatives. The range of the density and temperature arrays is generally taken to be sufficiently large so as to encompass the range of state values that will be encountered in most simulations. However this is not guaranteed to be the case and in fact many equation of state errors are associated with evaluations of the tables for densities or temperatures outside the defined range.

The xRage code does not use SESAME tables directly; instead these tables are used to create a tabular equation of state (TEOS) table described in the next section. Users should be aware that this process can involve certain modifications of the raw EOS data contained in the SESAME file that can lead to differences in the EOS evaluations using the TEOS file as compared to such direct SESAME interpolations as implemented in EOSPAC.

The current version of the codes does not allow the direct running of simulations that initialize EOS's from SESAME. Instead one creates a tabular equation of state as described in the next section to create a TEOS equation of state table and initializes the hydro simulation to use that file to determine its equation of state data.

For porous materials such as foams, a dynamic equation of state modification can be incorporated into the tabular EOS solution that accounts for irreversible crushing of the material pores [5, 16].

Input specifications:

1. **ramp_num**: Number of EOS ramps, **ramp_num**= M a non-negative integer. The number of material component to which a pore crush ramp modification is to be applied. Default = 0
2. **ramp_reverse**(n) $1 \leq n \leq M$: Reversible pore compression flag, **ramp_reverse**(n) = L where L is either `.true.` or `.false.`. If `.true.` pore compression for the material component is a reversible process. Default = `.false.`
3. **ramp_mat**(n) $1 \leq n \leq M$: Ramp material, **ramp_mat**(n) = m an integer $1 \leq m \leq N$. Apply ramp corrections to material m . An error

occurs if the material m appears in more than one ramp.

4. **ramp_alpha**(n) $1 \leq n \leq M$: Uncrushed/crushed volume ratio, **ramp_alpha**(n) = α a real number > 1 . This quantity is the ratio of the uncrushed specific volume of the component to the fully crushed specific volume.
5. **ramp_pe**(n) $1 \leq n \leq M$: Minimum crush pressure **ramp_pe**(n) = P_e a positive real number of units microbars. This is the minimum pressure needed to compress the pores in the component, i.e. the start of the pressure ramp.
6. **ramp_pc**(n) $1 \leq n \leq M$: Maximum crush pressure, **ramp_pc**(n) = P_c a positive real number $> P_e$ of units microbars. This is the pressure above which the pores in the material are fully compressed.
7. **ramp_pe_de**(n) Energy modification to be applied for pressures $< \text{ramp_pe}$. This quantity is subtracted from the tabulated specific internal energy and is in addition to any specification by the user for a total energy offset (via `matdef(26,n)`). For pressures between `ramp_pe` and `ramp_pc` the energy offset changes linearly between the values `ramp_pe_de` and `ramp_pc_de` (see below).
8. **ramp_pc_de**(n) Energy modification to be applied for pressures $> \text{ramp_pc}$. This quantity is subtracted from the tabulated specific internal energy and is in addition to any specification by the user for a total energy offset (via `matdef(26,n)`).

The basic modification is to scale the specific volume of the material by a pressure dependent factor. In addition we allow a pressure dependent translation on the specific internal energy. NOTE: This correction is **in addition** to any translation provided via the specification in `matdef(26,mat)`. The basic modification of the tabulated specific volumes and specific internal energy for a fixed material is given by:

$$\begin{aligned} V_{\text{ramp}}(P, T) &= \alpha(P) V_{\text{table}}(P, T) \\ e_{\text{ramp}}(P, T) &= e_{\text{table}}(P, T) - \left\{ \left(\frac{\alpha(P)-1}{\alpha_{pe}-1} \right) \Delta_{pe} e + \left(\frac{\alpha_{pe}-\alpha(P)}{\alpha_{pe}-1} \right) \Delta_{pc} e \right\} \end{aligned} \quad (59)$$

The ramp modification does not in fact produce a thermodynamically consistent equation of state, but as is pointed out in the reference of Menikoff and

Kober [16] formula (2.34) can be regarded as an approximation of the specific internal energy as computed from a consistent EOS. The ramp function $\alpha(P)$ is computed as follows. The ramp function depends on the parameters described in on page 9 under the section on ramp EOS modifications together with the maximum pressure that has occurred in the cell, **pmax**. Specifically we denote:

1. **ramp_alpha** = α_0
2. **ramp_pe** = P_e
3. **ramp_pe_de** = $\Delta_{pe}e$
4. **ramp_pc** = P_c
5. **ramp_pc_de** = $\Delta_{pc}e$
6. **pmax** = P_{\max}

If **ramp_reverse** is false, the formula for $\alpha(P)$ is given by:

$$\alpha(P) = \begin{cases} \alpha_0 & \max(P, P_{\max}) \leq P_e \\ \alpha_0 + \frac{\max(P, P_{\max}) - P_e}{P_c - P_e} (1 - \alpha_0) & P_e \leq \max(P, P_{\max}) \leq P_c \\ 1 & P_c \leq \max(P, P_{\max}) \end{cases} \quad (60)$$

While for **ramp_reverse** equal to true:

$$\alpha(P) = \begin{cases} \alpha_0 & P \leq P_e \\ \alpha_0 + \frac{P - P_e}{P_c - P_e} (1 - \alpha_0) & P_e \leq P \leq P_c \\ 1 & P_c \leq P \end{cases} \quad (61)$$

The distinction here is that when **ramp_reverse** is true the compression is reversible, i.e. when the pressure relaxes the pores can expand again, while for **ramp_reverse** equal to false the crush is irreversible, once crushed the pores do not expand again.

4 Legacy Equation of State (EOS)

4.1 Multi-Material Equation of State (MMEOS)

The Multi-Material Equation of State (MMEOS) was the original EOS used for xRage, and this section is included to preserve the background of this work. The code has moved through many iterations and currently only supports Temperature Equation of State (TEOS), which gets set with `keos`. Using TEOS, the tables are built and the table reader is a derivation of the old MMEOS. SESAME files as always can be used.

4.2 Creating an EOS file /TEOS

The xRage code has the ability to either build a tabular EOS or use an EOS that has been produced already. Some simple analytic EOS can be used. The reason the user needs an EOS file even if the user is calling the SESAME files is due to the fact that they are used to build the EOS tables and are inverted. xRage also has the ability to build TEOS/EOS tables. To create an EOS file you must have an input file. An example of an input file is found in Table 1. To build an EOS file you run `codename.x file.eos.in`.

Table 1: Creating an EOS File and TEOS File

Variable	Description
<code>teos_file</code>	Name for the EOS output file.
<code>sesfiles(1)</code>	Full path to the first SESAME file.
<code>sesfiles(n)</code>	Additional SESAME file specifications.
<code>numm</code>	Total number of materials.
<code>matid(1)</code>	First material ID.
<code>matid(n)</code>	Additional material IDs.

4.3 Equation of State Types

The code xRage has the ability to either build a tabular Equation of State (EOS) or use an EOS that has been produced already. Some simple analytic EOS can be used (must be very simple). The user needs an EOS file, even if you are calling the SESAME files, because the EOS files are used to build the

EOS tables and are inverted. To build an EOS file one can run: `codename.x file.eos.in .`

4.3.1 keos

The type of equation of state xRage will use for a calculation is determined by the input variable `keos`. The default value of `keos` is 0, which specifies an iterative multi-material ideal gas EOS. Users will primarily use ideal gas and tabular equation of states. Table 2 below lists all of the `keos` options.

Table 2: Equation-of-State Options

keos	Type	Description
0	Multi-Material Ideal Gas	Default EOS.
1	Deprecated	
2	Deprecated	
3	TEOS	Built from SESAME EOS.
obsolete	Standard MMEOS Tabular	Multi-cell, multi-material, vectorized, tabular equation of state with crush curves. User is required to supply the proper tabular EOS file.
-1	Multi-Material Stiff Ideal Gas	Similar to Ideal Gas EOS but requires additional inputs to define stiffness properties.
-2	Multi-Material Analytic EOS with Crush Curves	Not yet supported.
-3	JWL Explosive	Not yet supported. Will allow only 1 explosive material with several other stiff ideal gas materials.
-4	Non-standard MMEOS Tabular	Not yet supported. Requires specially built tabular EOS files.

4.3.2 Analytic EOS

To use an ideal gas equation of state, the user first sets the input variable `keos=0`. This is the default value for `keos`. `keos>0` is used if the `eosfile` is

specified. The code will then look for an EOS table with filename `eosfile`. Else, if `eosver` is greater than 0, and `eosver` is less than 9999, then look for an EOS table with filename `estbXXXX`, where `XXXX` is a four-digit number equal to `eosver` and padded on the left with zeros. If `keos=-1`, this setting will utilize multi-material stiff ideal gas. `keos=-2` is multi-material analytic EOS with crush curves. `keos=3` uses TEOS built from SESAME EOS. `keos=1` and `keos=2` have been removed from the code.

4.3.3 Tabular

To use an xRage Tabular Equation of State, the user first sets the input variable `keos > 0`. The xRage tables can be built from SESAME EOSs. The code, xRage, will expect a default filename of `meostab` for the Tabular EOS file. The user can specify a unique file name with the input variable `eosfile`. Simply set this variable to the filename of the Tabular EOS file and place this file in the local file space with the xRage executable code. Table 3 displays input examples for ideal gas EOS.

Table 3: Ideal Gas EOS Input Example

Variable Setting	Description
<code>keos=0</code>	Ideal gas equation of state
<code>nummat=1</code>	1 material in problem
<code>matdef(16,1) = 0.66666667</code>	Specifying $(\gamma - 1)$; $\gamma = 5/3$, (air)
<code>matdef(30,1) = 1.0e-4</code>	Specifying C_v - specific heat [erg/gm/ev] (1 [J/(mol*K)] \approx 1e7*11604/(molar mass) [erg/(g*eV)])

4.3.4 SESAME EOS

SESAME is the typical tabular temperature-based EOS used. Since xRage is an energy-based code, the EOS tables need to be inverted to work.

References

- [1] An invitation to participate in the LASL equation of state library. Los Alamos Scientific Report LASL-79-62, Los Alamos National Laboratory, 1979.
- [2] SESAME: THE LOS ALAMOS NATIONAL LABORATORY EQUATION OF STATE DATABASE. http://t1web.lanl.gov/doc/SESAME_3Ddatabase_1992.html, 1992.
- [3] T-1 equation of state and mechanics of materials. <http://t1web.lanl.gov/>, 2008.
- [4] Llnl explosives reference guide, 2013. Registration required.
- [5] M. Baer and J. Nunziato. A two-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive granular materials. *Int. J. Multiphase Flows*, 12:861–889, 1986.
- [6] J. Baker, S. Peyton, F. Brown, and J. Champion. Computational equation of state AQUA water. Technical report, S-CUBED, 1988.
- [7] Radu Balescu. *Equilibrium and nonequilibrium statistical mechanics*. Wiley, New York, 1975.
- [8] R. D. Goodwin and L. A. Weber. Comparison of two melting-pressure equations constrained to the triple point using data for eleven gases and three metals. technical note (final). Technical Report COM-75-10052 NBS-TN-183, 1963. Previously announced as N-64-14291 United States System Entry Date: 06/03/2001 Source: NSA-33-000133; GRA (NTIS NTS) Language: English.
- [9] Jean Pierre Hansen and Ian R. McDonald. *Theory of simple liquids*. Elsevier, Boston, 2006.
- [10] F. Harlow and A. Amsden. Fluid dynamics. LANL Monograph LA-4700, National Technical Information Service, 1971.
- [11] Gerald I. Kerley. Rational function method of interpolation. Technical Report LA-6903-MS, 1977.

- [12] E. Lee, M. Finger, and W. Collins. JWL equation of state coefficients for high explosives. Technical report, Lawrence Livermore National Laboratory, 1973.
- [13] S. P. Lyon and J. D. Johnson. SESAME: The los alamos national laboratory equation of state database. Technical Report LA-UR-92-3407, 1992.
- [14] C. Mader. *Numerical Modeling of Detonations*. University of California Press, Berkeley, 1979.
- [15] R. Menikoff and B. Plohr. The riemann problem for fluid flow of real materials. *Rev. Mod. Phys.*, 61:75–130, 1989.
- [16] Ralph Menikoff and Edward Kober. Equation of state and hugoniot locus for porous materials: P-alpha model revisited. *AIP conference proceedings*, 505:129–132, 2000.
- [17] R. More, K. Warren, D. Young, and G. Zimmerman. QEOS: An equation of state for hot dense matter. *Phys. Fluids*, 31:3059–3078, 1988.
- [18] D. K. Nadyozhin and A. V. Yudin. Equation of state under nuclear statistical equilibrium conditions. *Astronomy Letters*, 30(9):634–646, 2004.
- [19] David A. Pimentel and Olga V. Chotinun. EOSPAC VERSION 6 USER’S MANUAL. Technical Report LA-UR-06-0213, 2006.
- [20] D. Steinberg, S. Cochran, and M. Guinan. A constitutive model for metals applicable at high strain-rate. *J. Appl. Phys.*, 51:1498–1504, 1980.
- [21] D. J. Steinberg. Equation of state and strength properties of selected materials. Technical Report UCRL-MA-106439-rev, Lawrence Livermore National Laboratory, 1996.