Some notes on SESAME

Answers to frequently asked questions
Santiago R. Simanca
Department of Applied Mathematics & Statistics
The University at Stony Brook

Since the equations of fluid dynamics do not form a closed system, an equation of state is added to make them so. This equation is intended to describe the thermodynamics properties of the fluid, and introduces a functional relation among the density, pressure, energy, temperature and entropy, for example, by expressing some of these quantities as functions of the others.

When the equation of state is given by an analytic formula, it usually involves some modeling assumptions on the fluid, rendering it somewhat restricted in its domain of validity. Simple analytical formulas fail to describe regions where phase transitions occur.

An equation of state given in tabular form provides values that give an accurate description over a wide range of parameters consistent with experimental data. The data is computed with the best physical models and, given realistic time constraints, it agrees with available experimental data. It has the potential to be more accurate than a purely theoretical model.

SESAME is a tabular equation of state developed at Los Alamos National Laboratory. This data is used in the front tracking gas code to close the equations of fluid dynamics, and solve the corresponding Riemann problems. Realistically, the accuracy of the result is limited by the interpolation order and grid spacing in thermodynamics space, in addition to the accuracy itself of the data used to construct the table. In the hierarchy of the front tracking code, the source related to the implementation of this capability resides in
/libs/gas/geos.

In these notes we succinctly explain how this works. The table of contents that follows will help you locate particular subjects of interest.

Contents

1 Brief introduction to gas dynamics

2
2 Tabular equation of state: sesame tables
2.1 201 tables ................................................................. 5
2.2 301 tables ................................................................. 5
2.3 401 tables ................................................................. 5
2.4 Grids ................................................................. 5

3 Description of the algorithm ........................................ 5

4 Library functions .......................................................... 8
4.1 Reading/inverting tables: sesame,F, sesadd,F  and sesinv.c .............. 9
4.2 generic-eos.c ............................................................... 9

1 Brief introduction to gas dynamics

The conservation of mass, momentum and energy give
\[
\partial_t \rho + \nabla \cdot (\rho u) = 0 \quad (1)
\]
\[
\partial_t (\rho u) + \nabla \cdot (\rho u^2) + \nabla p = -\rho \nabla \varphi \quad (2)
\]
\[
\partial_t \left[ \rho \left( \frac{1}{2} u^2 + E \right) \right] + \nabla \cdot \left[ \rho \left( \frac{1}{2} u^2 + E \right) u + pu \right] = -\rho u \cdot \nabla \varphi \quad (3)
\]

In this expression, \( u \) is the particle velocity, \( \rho \) is the mass density, \( E \) is the specific internal energy, \( p \) is the pressure, and \( \varphi \) is the gravitational potential energy. These are equations for the unknowns \( u, \rho, E \) and \( p \).

If \( n \) is the dimension, there are \( n + 2 \) equations in \( n + 3 \) unknowns — the three scalar quantities and the velocity. To obtain a closed system, the fluid equations are supplemented by a constitutive relation specifying the material properties, or its thermodynamics: the equation of state.

The first law of Thermodynamics says that the path integral of the difference of the differential amount of heat \( dQ \) absorbed by a system and the differential amount of work \( dW \) done by it is independent of the path. Hence, there exists a function \( E \) — the internal energy — such that
\[
dE = dQ - dW.
\]

This equation says that the non-conservative part of the heat absorption cancels the non-conservative part of the work done by the system, producing and exact differential. The expressions \( dQ \) and \( dW \) are to be interpreted as 1-forms and not as the differential of functions \( Q \) and \( W \). Sometimes these functions are defined, but often they are not.

A system state is thus determined by 3 parameters \( \rho, V, T \), where \( V = 1 / \rho \) is the specific volume. The equation of state gives a scalar relation among them, defining one in terms of the others.
In a infinitesimal reversible transformation, a system changes according to

\[ dW = p \, dV. \]

Therefore,

\[ dQ = dE + dW = \partial_p E \, dp + (\partial_V E + p) \, dV. \]

If we now express \( Q \) as a function of \((T, V)\) we obtain that

\[ dQ = \frac{\partial E}{\partial T} \, dT + \left( \frac{\partial E}{\partial V} + p \right) \, dV. \]

The second law of Thermodynamics now implies that the path integral of \( dQ/T \) only depends upon the initial and end points of the path. Hence, there must be a function –the entropy \( S\)– such that

\[ dS = \frac{dQ}{T}. \]

Therefore,

\[ T \, dS = \frac{\partial E}{\partial T} \, dT + \left( \frac{\partial E}{\partial V} + p \right) \, dV, \]

or what is the same,

\[ T \, dS = dE + pdV. \quad (4) \]

If \( u = u(t, x) \) is the speed of the non-linear wave equation, the characteristic curves are defined by

\[ \Gamma = \Gamma_x := \left\{ \begin{aligned}
\frac{dx}{dt} &= u(t, x) \\
x \big|_{t=0} &= x_0.
\end{aligned} \right. \]

Then, along \( \Gamma \) we have, for example,

\[ \frac{d}{dt} \rho(t, x(t)) = \rho_t + u \cdot \nabla \rho = -\rho \nabla \cdot u, \]

which follows by (1). Proceeding similarly by differentiation along characteristics, the fluid equations above can be recasted along \( \Gamma \) as

\[ \frac{d}{dt} \begin{pmatrix} \rho(t, x(t)) \\ u(t, x(t)) \\ E(t, x(t)) \end{pmatrix} = - \begin{pmatrix} \rho \nabla \cdot u \\ (1/\rho) \nabla p \\ (\rho/p) \nabla \cdot u \end{pmatrix}, \]

version known as \textit{Euler equations}. Notice that the third and first of these implies that

\[ \frac{d}{dt} E = -p \frac{d}{dt} \left( \frac{1}{\rho} \right) = -p \frac{d}{dt} V; \quad (5) \]
and so, by \( (4) \), we have
\[
TdS = 0,
\]
which says that, for smooth flows, the entropy is constant along particle trajectories (of course, \( T > 0 \)). When going through a shock, the particle path will see an increase in entropy.

Equation \( (4) \) is fundamental. It gives
\[
p(V, S) = -\frac{\partial}{\partial V} E(V, S), \quad T(V, S) = \frac{\partial}{\partial S} E(V, S).
\]
These equations produce several quantities important in the study of the equation of state:

- adiabatic exponent \( \gamma = -\frac{\partial \log p}{\partial \log V} \),
- Grüneisen coefficient \( \Gamma = -\frac{\partial \log T}{\partial \log V} \).

Obviously, \(-\gamma\) is the slope of the isentrope in the log \( p \)-log \( V \) plane, while \(-\Gamma\) is the slope in the log \( T \)-log \( V \) plane.

The quantities
\[
g = -\frac{pV}{T^2} \frac{\partial T}{\partial S}, \quad G = -\frac{V}{2} \frac{\partial^3 E}{\partial V^3} \frac{\partial^2 E}{\partial V^2},
\]
measure the spacing of the isentropes in the log \( T \)-log \( V \) plane, and the convexity of the isentropes in the \( p \)-\( V \) plane. Thermodynamics stability requires that \( E(V, S) \) be convex. So we must have
\[
\frac{\partial^2 E}{\partial S^2} \geq 0 \quad \Leftrightarrow \quad g \geq 0,
\]
\[
\frac{\partial^2 E}{\partial V^2} \geq 0 \quad \Leftrightarrow \quad \gamma \geq 0,
\]
\[
\frac{\partial^2 E \partial^2 E}{\partial S^2 \partial V^2} \geq \left( \frac{\partial^2 E}{\partial S \partial V} \right)^2 \quad \Leftrightarrow \quad \gamma g \geq \Gamma^2.
\]
The sign of \( G \) determines several properties of the equation of state.

## 2 Tabular equation of state: sesame tables

The tabular equation of state is compiled and maintained at Los Alamos National Laboratory. As of today, it contains 179 materials, encompassing gases, metals, minerals, polymers and mixtures. We point out that to the effect of numerical experiments, the equation of state is regarded as input data, and the quality of the input determines the quality of the solution.

Usually (but not always) there are 3 tables per material:
2.1 201 tables
The 201 table contains 5 words (floats) which are the atomic number, atomic weight, normal
density at room temperature and 1 atmosphere, pressure and energy (these last two at room
temperature and normal density). The temperature is measured in degrees Kelvin.

2.2 301 tables
The 301 table is a table of pressure, internal energy and, in some cases, free energy. The
independent variables are temperature and density. For some materials the pressure and
energy is separated into electronic and nuclear components, in fact creating several 300’s
tables: 301 total pressure and energy tables, 303 ionic EOS (including cold curve and zero-
point contributions), 304 thermal electronic tables, 305 thermal nuclear tables (including
zero-point contributions), 306 cold curve. The 301 tables are the sum of the 304, 305 and
306 tables.

2.3 401 tables
The 401 tables concern data involving phase transition. Many of the materials do not have
data concerning phase transition, and so, have empty 401 tables.

2.4 Grids
The independent variables $\rho$ and $T$ vary on an underlying rectangular grid $R$ in the $\rho$-
$T$ plane. The sesame subroutine library (which resides in /lib/gas/geo) contains the
software needed to access and use the tabulated data, reading $E(\rho, T)$ and $p(\rho, T)$ for a
given desired material, specified by an index or id.

Since the Riemann solution algorithm will often need the data as function of independent
variables other than $\rho$ and $T$ or $\rho$ and $E$, software was written to invert the original table
and to store the data at the outset.

When inverting the original table, we loose the rectangularity of the data array and data
boundary required by the SESAME format. This problem is overcome by using the front
tracking method for storing and interpolating data based on non-rectangular grid arrays.

3 Description of the algorithm
The algorithm solves the Riemann problem for (1), (2), (3) in dimension 1, with no gravita-
tional effect:

\[
\begin{align*}
\partial_t \rho + \partial_x (\rho u) &= 0 \\
\partial_t (\rho u) + \partial_x (\rho u^2 + p) &= 0 \\
\partial_t \left[ \rho \left( \frac{1}{2} u^2 + E \right) \right] + \partial_x \left[ \rho \left( \frac{1}{2} u^2 + E \right) u + pu \right] &= 0
\end{align*}
\]
such that

\[ S(t, x) \big|_{t=0} = (\rho, u, p) \big|_{t=0} = \begin{cases} (\rho_r, u_r, p_r) & \text{if } x \geq 0, \\ (\rho_l, u_l, p_l) & \text{if } x < 0. \end{cases} \quad (7) \]

The state data \( S(0, x) \) for \( x \geq 0 \) is called the right state \( S_r \). The state data \( S(0, x) \) for \( x < 0 \) is called the left state \( S_l \).

The exact solution consists of the initial states \( S_l \); \( S_r \) and a middle state \( S_m \), the latter separated from the first two by either a rarefaction or shock wave. The values of \( p_m \) and \( u_m \) are constant, but the density assumes a left value \( \rho_{ml} \) and right value \( \rho_{mr} \) with a discontinuity at the line of slope \( u_m \):

\[ \frac{dx}{dt} = u_m. \quad (8) \]

The algorithm below [1, 2] is for the case where there are no phase transitions and \( G > 0 \).

First of all, the pressure \( p_m \) and velocity \( u_m \) for \( S_m \) is evaluated. If

\[ m_r = \frac{p_r - p_m}{u_r - u_m}; \]
\[ m_l = -\frac{p_l - p_m}{u_l - u_m}; \]

then

\[ p_m = \frac{u_l - u_r + p_l + p_r}{\frac{1}{m_l} + \frac{1}{m_r}}; \quad (9) \]

and

\[ m_r = \varphi(p_m, p_r, \rho_r); \]
\[ m_l = \varphi(p_m, p_l, \rho_l); \quad (10) \]

where the function \( \varphi \) is defined by

\[ \varphi(p_m, p, \rho) = \begin{cases} \sqrt{\rho p_m - p} & \text{for } p_m > p, \\ (p - p_m) \left( \int_{p_m}^{p} \frac{dp}{\rho c p} \right)^{-1} & \text{for } p_m < p, \\ \frac{\rho}{p} & \text{for } p_m = p. \end{cases} \]

In this expression, \( \rho_m \) comes from the Hugoniot relation for a shock joining the given state and a state with pressure \( p_m \); the integral in the case when \( p_m = p \) is evaluated at constant entropy \( S \), and \( c \) is the sound speed defined by the formula

\[ c^2 = \frac{\partial P(\rho, S)}{\partial \rho}. \]
Notice that once \( m_r \) and \( m_l \) are known, we have
\[
 u_m = \frac{p_l - p_r + m_r u_r + m_l u_l}{m_r + m_l}
\]  
(11)

The algorithm now proceeds as follows:

1. Pick an initial value of \( p_m^0 \) (or \( m_r^0, m_l^0 \)) and using (9) obtain a value \( \tilde{p}_m^n \). Set
\[
 p_m^{n+1} = \max \{ \varepsilon, \tilde{p}_m^n \},
\]
where \( \varepsilon \) is a prescribed small positive real number. This ensures that \( p \) remains positive.

2. Using (10), update the values of \( m_r^{n+1} \) and \( m_l^{n+1} \):
\[
 m_r^{n+1} = \varphi(p_m^{n+1}, p_r, \rho_r),
 m_l^{n+1} = \varphi(p_m^{n+1}, p_l, \rho_l).
\]

3. Iterate this procedure until \( \max \{ |m_r^{n+1} - m_r^n|, |m_l^{n+1} - m_l^n| \} \leq \varepsilon \). When this is is achieved, set
\[
 m_r = m_r^{n+1},
 m_l = m_l^{n+1},
 p_m = p_m^{n+1},
\]
and calculate the value of \( u_m \) using (11).

**Remark:** the iteration may fail to converge in the presence of a strong rarefaction. In order to ensure its convergence in a finite number of steps, a minor modification of the original Godunov’s procedure must be introduced. This essentially involves a modification in the expression to obtain \( p_m^{n+1} \) from \( \tilde{p}_m^n \). The reader should see [2] for details.

As mentioned above, the fluid initially at \( x \leq 0 \) is separated from the fluid initially at \( x > 0 \) by the split line (8). In order to find the state values \((\bar{\rho}, \bar{u}, \bar{p})\) at grid points \( P = (x, t) = (\theta h, k/2) \), we must consider four cases:

a) The sample point \( P \) is to the left of the slip line, and the right wave is a shock, that is, \( p_m > p_r \). In this case,
\[
 m_r = -\rho_r(u_r - U_r) = -\rho_{mr}(u_m - U_r),
\]
where \( \rho_{mr} \) is the density in the portion of \( S_m \) adjoining the right shock, and \( U_r \) is the velocity of the right shock. This equation is used to find \( U_r \), and

- If \( P \) is to the right of the shock \( x = U_r \), then \( \bar{\rho} = \rho_r \), \( \bar{u} = u_r \) and \( \bar{p} = p_r \).
• If \( P \) is to the left of the shock \( \dot{x} = U_r \), then \( \bar{u} = u_{m_r}, \bar{p} = p_r \), and \( \bar{\rho} = \rho_{m_r} \) can be calculated from the second equation in (12).

b) The sample point \( P \) is to the right of the slip line, and the right wave is a rarefaction, that is, \( p_m \leq p_r \). In this case, we use the Riemann invariant

\[
\Gamma_r = u_m + \int_{\rho}^{\rho_{m_r}} \frac{c}{\rho} \, d\rho = u_r + \int_{\rho}^{\rho_{m_r}} \frac{c}{\rho} \, d\rho ,
\]

to calculate \( \rho_{m_r} \). The rarefaction is bounded by \( \dot{x} = u_r + c_r \) on the right and \( \dot{x} = u_m + c_m \) on the left, where \( c_r, c_m \) stand for the appropriate sound speeds. If \( P \) lies to the right of the rarefaction, \( \bar{\rho} = \rho_r, \bar{u} = u_r \) and \( \bar{p} = p_r \). If \( P \) lies to the left of the rarefaction, \( \bar{\rho} = \rho_{m_l}, \bar{u} = u_m \) and \( \bar{p} = p_m \). If \( P \) lies inside the rarefaction, the slope of the line through the origin and \( P \) is equated with \( \dot{x} = u + c \), yielding

\[
\bar{u} + \bar{c} = 2\theta h/k .
\]

Using the invariance of \( \Gamma_r \) together with the definition of \( c \) yields \( \bar{\rho}, \bar{u}, \bar{p} \).

c) \( P \) lies to the left of the slip line and the left wave is a shock, that is, \( p_m > p_l \). This case is a mirror image of (a) with \( m_r \) replaced by \( m_l \).

d) \( P \) lies to the left of the slip line and the left wave is a rarefaction, that is, \( p_m \leq p_l \). This case is a mirror image of (b) with \( \Gamma_r \) replaced by \( \Gamma_l \).

4 Library functions

The sphere library resides in /libs/gas/geos and contains several important files with functions that will be described below. The procedure to use the tabulated equations of state is twofold: first, the data is loaded and preprocessed. Second, the preprocessed data is used in the front tracking simulations.

In order to compile the Front Tracking code enabling this capability to read tabular equation of states, you might need to include in the /libs/gas/geos makefile the following source files:

- eosplot.c
generic-eos.c
giniteos.c
gphriem.c
gsehyp.c
gsesintrp.c
gsesphase.c
gsespline.c
gsesprint.c
gsesinout.c
mpoly-eos.c
poly-eos.c
sesame-eos.c
spoly-eos.c
gpergsub.F
gstoc.F
esadd.F
sesame.F
sesinv.c
esesplin.c
sesstate.c
jwl-eos.c
mg-eos.c

and include also the following header files

- geosdeca.h
geosprotos.h
gsesprotos.h
mpoly.h
poly.h
sesame.h
spoly.h
jwl.h
mg.h
4.1 Reading/inverting tables: sesame.F, sesadd.F and sesinv.c

The binary file with the tabulated data for all materials is in /lib/gas/sesmat/sesmat. The library contains FORTRAN routines that reads this data file. It also contains routines that modify the and updates the data file from a data source file.

In sesame.F you find the routines to read the data from the binary file, as well as generalized input/output routines, routines used to update the data.

1. s2get(ir,ids2,tbls,lcnt,lu,ifl): loads eos table from the binary library.
2. s2eos(ir,tbls,r,t,p,e): compute eos from sesame 2 tables.

The function s4get(ir,ids2,tbls,lcnt,lu,ifl) used to read the 401 table when doing numerical simulations involving phase transition is contained in sesadd.F.

In sesinv.c one finds the routines that invert the original data coming from the 301 table, which permits to calculate the adiabats, Riemann invariants and sound speeds as functions of density and temperature, and computes them as functions of density and specific internal energy, density and specific entropy, and pressure and specific entropy, respectively.

1. void setrt(float rho, float T, Locstate state, SESAME_EOS *seos):
2. void integrate_entropy():
3. void coldpe():

4.2 generic-eos.c

This is a generic file for the implementation of an equation of state mode. It contains the following functions:

1. GENERIC_pressure(): returns the thermodynamic pressure of a state

\[ p = - \frac{dE}{dV} |_{s} \] .

2. GENERIC_sound_speed_squared(): returns the square of the local sound speed of the state

\[ c^2 = \frac{dp}{d\rho} |_{s} \] .

3. GENERIC_specific_internal_energy(): returns the specific internal energy = internal energy per unit mass of the state.

4. GENERIC_gruneisen_gamma(): returns the dimensionless Gruneisen exponent

\[ \Gamma = - \frac{d\log T}{d\log V} |_{s} \] .
5. \texttt{GENERIC\_temperature()}: returns the thermodynamic temperature of a state

\[ T = \frac{dE}{dS} |_V . \]

6. \texttt{GENERIC\_entropy()}: returns the specific entropy of a state.

7. \texttt{GENERIC\_C\_V()}: returns the specific heat at constant volume

\[ C_V = T \frac{dS}{dT} |_V . \]

8. \texttt{GENERIC\_acoustic\_impedance\_squared()}: returns the square of the local acoustic impedance of the state

\[ i^2 = - \frac{dp}{dV} |_s . \]

9. \texttt{GENERIC\_specific\_enthalpy()}: computes the specific enthalpy of the given state

\[ H = E + pV . \]

10. \texttt{GENERIC\_adiabatic\_gamma()}: returns the dimensionless sound speed

\[ \gamma = - \frac{d(\log P)}{d(\log V)} |_s . \]

11. \texttt{GENERIC\_fundamental\_derivative()}: returns the fundamental derivative of gas dynamics for the state, which is defined by

\[ G = - \frac{1}{2} \frac{d^2 p}{dV^2} |_s . \]

12. \texttt{GENERIC\_C\_P()}: returns the specific heat at constant pressure

\[ C_P = T \frac{dS}{dT} |_P . \]

13. \texttt{GENERIC\_K\_S()}: returns the isentropic compressibility

\[ K_S = - \frac{1}{V} \frac{dV}{dp} |_s . \]
14. **GENERIC_K_T()**: returns the isothermal compressibility

\[ K_T = -\lim_{V \to 0} \frac{dV}{dp} |_{T}. \]

15. **GENERIC_dens_Hugoniot()**: given the state \( S_0 \) on one side of an oblique shock, and the pressure \( p_1 \) on the other side, this function returns the density \( \rho_1 \) of the state with pressure \( p_1 \). This density is found by solving the Hugoniot relation

\[ (p_1 + p_0)(\frac{1}{\rho_0} - \frac{1}{\rho_1}) = 2(\epsilon_1 - \epsilon_0), \]

where \( \epsilon_0 \) and \( \epsilon_1 \) are the specific internal energies of the two respective states. For a given equation of state, the specific internal energy can be expressed as a function of the pressure and density. Thus the above equation can be solved to give \( \rho_1 \) as a function of \( S_0 \) and \( p_1 \).

16. **GENERIC_pr_normal_vel_wave_curve()**: this function computes the pressure on the forward Riemann wave curve given the velocity difference across the wave. If \( du > 0 \), it returns the solution to the system:

\[
\begin{align*}
du^2 &= (p - p_0)(V_0 - V), \\
d\epsilon &= \frac{1}{2}(p + p_0)(V_0 - V).
\end{align*}
\]

If \( du < 0 \), it returns

\[ du = \int_{p0}^{p} \frac{dp}{\epsilon p} \mid_{s}. \]

17. **GENERIC_mass_flux()**: returns the mass flux across a wave

\[ m = \frac{|p - p_0|}{u - u_0}, \]

where \( p_0 \) and \( u_0 \) are the pressure and velocity ahead of the shock, while \( p \) and \( u \) are those behind it.

18. **GENERIC_oned_fan_state()**: this is a utility function provided for the evaluation of states in a simple wave. Given \( S_a \), it solves for \( S_m \) using the equation:

\[ w = c_m - c_a + \int_{p_a}^{p_m} \frac{dp}{\epsilon p} \mid_{s = S_a} = \int_{c_a}^{c_m} \frac{d\epsilon}{\mu^2} \mid_{s = S_a}. \]

Here \( c \) is the sound speed, \( \rho \) the density, \( S \) the specific entropy, \( p \) the pressure, and \( \mu^2 = (G - 1)/G \), where \( G \) is the fundamental derivative of gas dynamics. The returned \( S_m \) contains only the thermodynamics of the state in the rarefaction fan. In particular
$S_1$ can be used to evaluate the pressure, density, and sound speed of the state inside the rarefaction fan.

The input data are the value of $w$ as defined above ($w$), the state ahead of fan ($S_a$), and the state behind fan ($S_b$). The output data are the state inside fan ($S_m$) and vacuum ($= 1$ if $S_m$ is a vacuum, 0 otherwise). The function returns the sound speed of the answer state $S_m$.

19. **GENERIC_riemann_wave_curve()**: this evaluates the forward wave family wave curve defined by

$$
\begin{cases} 
\sqrt{(p_s - p_0)(V_0 - V)} & \text{if } p_s > p_0, \\
\int_{p_c}^{p_s} \frac{dp}{\rho} |_S & \text{if } p_s < p_0.
\end{cases}
$$

**References**
