CEE convection notebook

Yisheng Tu

May 5, 2020

Contents

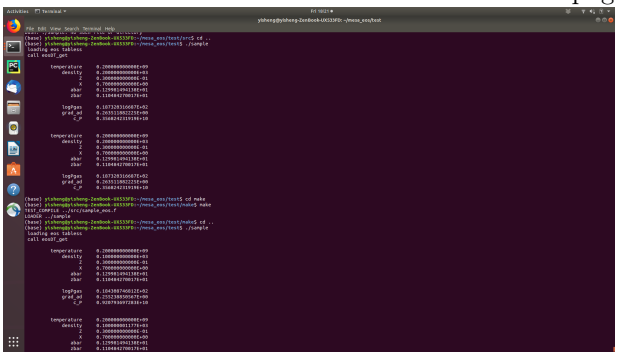
1 10-18-2019

New work

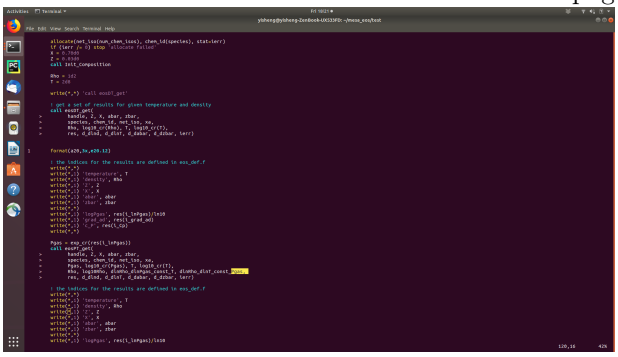
- The EOS data files are in **/mesa-r11701/data/eosDT_data**
- To call the EOS function, go to **/mesa-r11701/eos/test/**. In **/mesa-r11701/eos/test/src/**, see **sample_eos.f**. This is the script that use the equation of state. Then go to **/mesa-r11701/eos/test/make/**, type "make". This creates a executable in **/mesa-r11701/eos/test/** called "**sample**". Type **./sample** to run it.

- Within **sample_eos.f**, there are functions use the EOS. There seems to be a precision problem. if density $= 1d2$, then the first output is different from the second; if density $= 2d2$, then the output is the same

from 2019-10-18 18-21-07.png



from 2019-10-18 18-21-44.png



Next step

see how the **sample_eos.f** actually works. Then invert all tables it uses.

2 10-24-2019/10-25-2019

New work

- Met with Luke on 10-25-2019
- Looked into MESA about how MESA used the EOS tables
- MESA store most of the reading functions in **/eos/private** or **/eos/public**. The ones in **/eos/test** are functions that uses those in these two directories

Next step

Make a clear structure of **/mesa/eos/***.

3 11-01-2019

New work

- Found how mesa eos works, especially worked out eosDE_get: density and internal energy as independent variables.

MESA structure:

1. On Bluehive, to study how MESA EOS works, I created a directory called MESA_eos, and put the essentials into it. They are **MESA/eos/**; **MESA/util/**; **MESA/data/**; **MESA/include/**; **MESA/libs/**.
2. The MESA EOS has three built-in ways to calculate the equation of state (3 different sets of independant variables) - DT: density and temperature; PT: pressure and temperature and DE: density and internal energy.
3. **eos/public/**: The most useful and instructive files are in here. By naming conventions these are the files that MESA want us to use.
4. **eos/private/**: have all the evaluation files.
5. **eos/test/**: examples of how to use MESA EOS

eos/public/

- **eos_def.f**: This file names all variables used in the EOS. such as

```
eosDT_result_names(i_lnPgas) = 'lnPgas'
eosDT_result_names(i_lnE) = 'lnE'
eosDT_result_names(i_lnS) = 'lnS'
eosDT_result_names(i_grad_ad) = 'grad_ad'
eosDT_result_names(i_chiRho) = 'chiRho'
eosDT_result_names(i_chiT) = 'chiT'
eosDT_result_names(i_Cp) = 'Cp'
eosDT_result_names(i_Cv) = 'Cv'
eosDT_result_names(i_dE_dRho) = 'dE_dRho'
eosDT_result_names(i_dS_dT) = 'dS_dT'
eosDT_result_names(i_dS_dRho) = 'dS_dRho'
eosDT_result_names(i_mu) = 'mu'
eosDT_result_names(i_lnfree_e) = 'lnfree_e'
eosDT_result_names(i_gamma1) = 'gamma1'
eosDT_result_names(i_gamma3) = 'gamma3'
eosDT_result_names(i_eta) = 'eta'
```

  defined all outputs by eos module. Although it says "DT", these variable names also works for DE and PT.

- **eos_lib.f90**: This file has all the "get" functions, which gives eos given input independent variables. The following three function are especially important for our concern.

```
subroutine eosDT_get(...)
subroutine eosPT_get(...)
subroutine eosDE_get(...)    ! density + internal energy
```

eos/test/

- **./mk**: makes a file called "sample". if one do **./sample**, it will output eos data as specified in **test/src/sample_eos.f** (compile it).
- **test/src/sample_eos.f**. In here is where the eos gets used. I modified here to use **eosDE_get**. Following code comes from **/eos/public/eos_lib.f90**

```
subroutine eosDE_get( &
        ! input
       handle, Z, X, abar, zbar, &
       species, chem_id, net_iso, xa, &
       energy, log10E, rho, log10Rho, log10T_guess, &
        ! output
       T, log10T, res, d_dlnRho_const_T, d_dlnT_const_Rho, &
       d_dabar_const_TRho, d_dzbar_const_TRho, &
       dlnT_dlnE_c_Rho, dlnT_dlnd_c_E, &
       dlnPgas_dlnE_c_Rho, dlnPgas_dlnd_c_E, &
       ierr)
```

  Some of the variable types are different in DE then in DT and PT (see **sample_eos.f**). The effect of **log10T_guess** on the output can be ignored (as long as it is not way out of range, otherwise ierr != 0, which means the eos failed)

Backup of today's work is in /home/ytu7/CEE_convection/backup_01-11-2019/eos.gz. This is the eos folder in **/scratch/ytu7/CEE_convection/MESA_eos/**

4 11-06-2019

met with Jonathan about integrating tabulated EOS into AstroBear

MESA EOS is hard to operate from outside, so use the existing test routine and create our own tables. Try with a fixed x and z values, create a table of values in format

$\rho_{\min}$	$\rho_{\max}$	$d\rho$		
$T_{\min}$	$T_{\max}$	dT		
p_{11}	p_{11}	p_{11}	...	p_{1n}
...				
p_{m1}	p_{m2}	p_{m3}	...	p_{mn}
E_{11}	E_{11}	E_{11}	...	E_{1n}
...				
E_{m1}	E_{m2}	E_{m3}	...	E_{mn}

Table 1: Caption

We need 6 sets of these tables, and 3 others:

- $P(\rho, T)$
- $P(\rho, E)$
- $T(\rho, P)$
- $T(\rho, E)$
- $E(\rho, P)$
- $E(\rho, T)$
- $C_V(\rho, E)$
- $c_s(\rho, E)$
- $c_s(\rho, P)$

5 11-07-2019

For terms such as **d_dlnT**, look into **eos_lib.f90**. They are defined where the script define variables. Speed of sound can be found in **eos_def.f**

```
names(i_gamma1) = 'gamma1'   ! dlnP_dlnRho at constant S
names(i_gamma3) = 'gamma3'   ! gamma3 - 1) = '' dlnT_dlnRho at constant S
names(i_eta) = 'eta'
```

$$c_s = \left(\frac{dp}{d\rho}\Big|_s\right)^{1/2} = \left(\frac{dp}{d\ln p}\frac{d\ln p}{d\ln \rho}\frac{d\ln \rho}{d\rho}\Big|_s\right)^{1/2} = \left(\frac{p}{\rho}\cdot\frac{d\ln p}{d\ln \rho}\Big|_s\right)^{1/2}$$

Other updates

Added a do-loop in **sample_eos.f**.
Added the comments to write the result pressure to a file for **eosDT_get**, the output is in **/test/output/***.
Judging by **/data/eosDT_data/mesa-eosELM**, the step width of $\log \rho$ is 0.025. So we need at least this precision to keep up with the accuracy.

Next step

Do the same for **eosPT** and **eosDE**. Then do the other 3 tables. Also need to run for the speed of sound. Send email to Jonathan for this update, and double check self-consistency.

6 11-10-2019

Added the get_PT function into **sample_eos.f**. The code is tested to be compilable. A copy of today's code is in D:\Research_CEE_convection\MESA_eos\

Next step

- Find the range of value of each variable, ρ, P, T, E, etc. And use that range to calculate EOS.
- Add get_DE function
- Write inverting function, and use to get the other three tables.

7 11-13-2019

Added get_DE into **sample_eos.f**. Tested for self-consistency.
Again, need the range at which the values are valid.
Now a backup is also available in **D:\Research_CEE_convection\MESA_eos**

Next step

- Find the range of value of each variable, ρ, P, T, E, etc. And use that range to calculate EOS.
- Write inverting function, and use to get the other three tables.
- test the codes first before using the actual range to reduce the computation time.

8 11-18-2019

- Developed new algorithm for inverting the EOS table. Now takes an initial guess value, then use the Nelder-Mead algorithm to find the closest value possible.
- Ran production tables in MESA. Stored and backuped in **MESA_EOS/eos/test/output/**.

Next Step

- Need to write the initial guess function. Probably use the ideal gas law.
- the one more complicated is $c_s(\rho, P)$. This probably need yet another inverting routine, by first find $P(\rho, E)$, then $c_s(\rho, P(\rho, E))$. **Can we do this in AstroBEAR?**

9 11-19-2019

Meeting with the group
Also get Opacity table from MESA.

10 01-09-2020

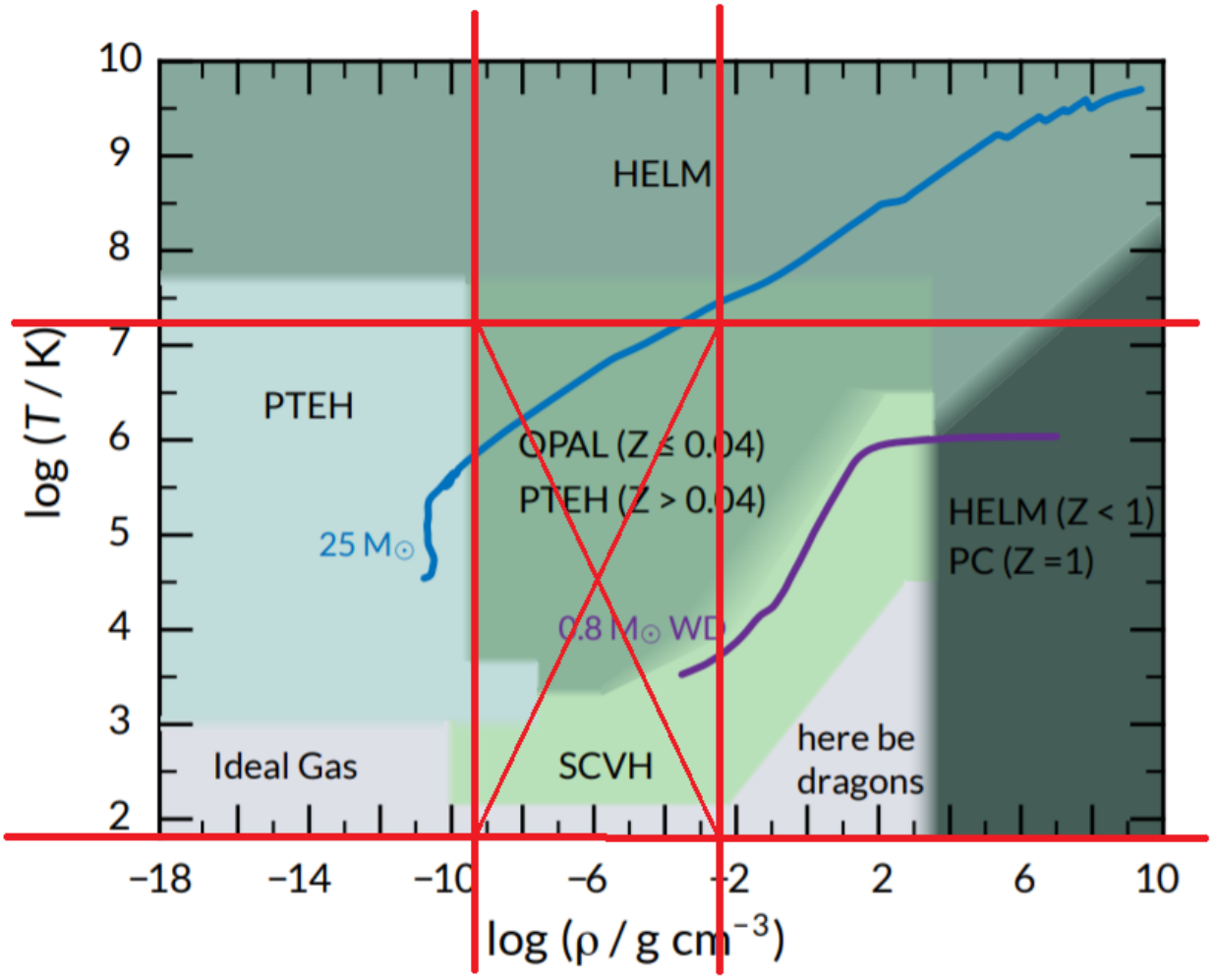


Figure 1: Range of temperature and density in Run 143

Physics

Ideal Gas EOS

1. All gas particles are in constant motion and collisions between the gas molecules and the walls of the container cause the pressure of the gas.
2. The particles are so small that their volume is negligible compared with the volume occupied by the gas.
3. The particles don't interact. There are no attractive or repulsive forces between them.
4. The average kinetic energy of the gas particles is proportional to temperature.

When there are multiple atoms in a molecule, some of the internal energy goes into molecular vibrations rather than in translational motion

(Ref: http://butane.chem.uiuc.edu/pshapley/GenChem1/L14/2.html)

OPAL EOS (http://articles.adsabs.harvard.edu/pdf/1996ApJ...456..902R)

1. used "the activity expansion method"
2. treats gas as individual particles
3. expends pressure to two-body, three-body clusters.
4. considers: non-relativistic Fermi-Dirac electrons, classical ions, all stages of ionization and excitation, molecular hydrogen, degenerate Coulomb corrections, quantum electron diffraction electron exchange, pressure ionization and terms arising from the so-called ladder diagrams of full quantum theory.
5. The OPAL EOS only considers elements up to neon. The abundance of heavier elements (by number fraction) are added to neon.
6. Ignoring hydrogen and helium, this is the relative abundance of heavy elements used in the EOS tables

Element	Relative mass Fraction	Relative number fraction
C	0.1906614	0.2471362
N	0.0558489	0.0620778
O	0.5429784	0.5283680
Ne	0.2105114	0.1624178

Table 2: Caption

7. Independent variable temperature & density. Directly calculated quantities are pressure, internal energy, entropy and $\chi_T = \left(\frac{\partial \ln P}{\partial \ln T}\right)_\rho, \chi_P = \left(\frac{\partial \ln P}{\partial \ln \rho}\right)_T, \Gamma_1 = \left(\frac{\partial \ln P}{\partial \ln \rho}\right)_S, \frac{\Gamma_2}{\Gamma_2 - 1} = \left(\frac{\partial \ln P}{\partial \ln T}\right)_S, C_V = \left(\frac{\partial E}{\partial T}\right)_V = \left(\frac{\partial E}{\partial \rho}\right)_T$. Additional quantities can be calculated analytically from the tabulated quantities.
8. reasons for boundaries of the table:
   above $T \sim 10^9 K$: pair production becomes an issue;
   Below $T \sim 5000 K$: numerical difficulty
   Above $\rho \sim 10^5$ g cm^{-3}: electron are relativistic.

SCVH EOS (http://articles.adsabs.harvard.edu/pdf/1995ApJS...99..713S)

1. free-energy minimization (FMIN) technique which is well described by Graboske et al. (1969), FGVH, and Hummer & Mihalas (1988).

2. Computed in detail a pure hydrogen EOS, and a pure Helium EOS. The values in between are obtained via interpolation.

3. Uses the "chemical picture". *On the other hand, the chemicalpicture assumes that bound configurations, such as atoms and molecules, retain a definite identity and interact through pair potentials. This amounts to solving the quantum problem first (with a heuristic approach) and then applying statistical mechanics to populate available states. This approximation has a serious drawback. At densities corresponding to pressure ionization, the electrons in bound configurations become delocalized, pair potentials become meaningless and consequently, bound species lose their identity. Nevertheless, the chemical picture emerges as a useful alternative in view of the practical limitations of the physical picture.*
   I assume this has something to do with the particle identity getting lost?

4. Uses "Free-energy minimization technique" (FMIN). Basically everything was derived from Helmholtz free energy $F(V, T, \{N_i\})$ of the system as a function of the total volume V, temperature T and particle numbers N_i for each species i.

5. **Helmholtz free energy** (ref: https://en.wikipedia.org/wiki/Helmholtz_free_energy):
   the useful work obtainable from a closed thermodynamic system at a constant temperature and volume (isothermal, isochoric).
   Derivation:
   $$dU = dQ + dW \tag{1}$$
   By assuming reversible process $dW = -pdV$, $dQ = TdS$
   $$dU = Tds - pdV = d(TS) - SdT - pdV \tag{2}$$
   Rearranging
   $$dF \equiv d(U - TS) = -SdT - pdV \tag{3}$$
   where F is the Helmholtz free energy. Because F is a function of state (a function defined for a system relating several state variables or state quantities that depends only on the current equilibrium state of the system, path independent for changing state), the function is also valid for a irreversible process (without electrical work or composition change).

6. Hydrogen:

   - "plasma phase transition" (PPT): Pressure ionization of hydrogen may not be a gradual process at all temperatures but may occur discontinuously through a first-order phase transition. $\log T_c = 4.185$
   - considers H_2, H, H^+ and e. Other species are too low abundances and have a negligible effect on the EOS
   - assume fractorizability of the partition function
   - account for weak diffraction effects in the interactions between heavy particles through Wigner-Kirkwood $\hbar^2$ correction
   - see section 3.2 for more details of interaction considerations

7. Helium:

- Considers He, He^+, He^{2+} and e.
- includes Helium "PPT"
- see section 4.2 for more details

8. In between: interpolation and approximations (see section 6)

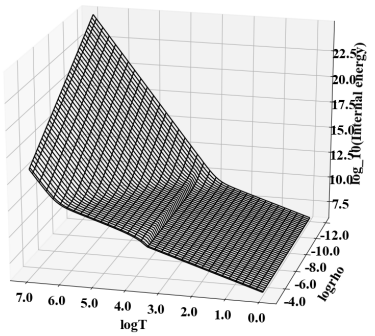
PTEH EOS

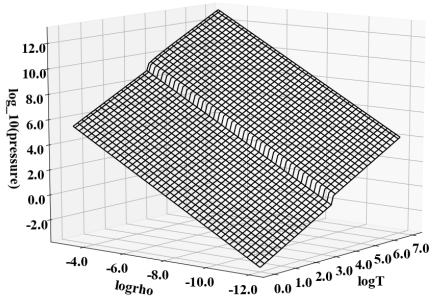
1. Based on Helmholtz free energy minimization.
2. Elements included are H^+, H, H_2, He^{2+}, He^+, He; as well as 7 heavier elements C, N, O, Ne, Mg, Si and Fe, which are assumed to be fully ionized, at all temperatures and densities. Each of these elements contributes one free electron foe every two baryons, except for Fe contributes fractionally more.
   The effect of heavier elements on EOS is insignificant, but on opacity it is.

HELM

1. Assumes fully ionized gas (at very high temperature or density)
2. Deals with electron-positron physics
3. Based on Helmholtz free energy

Upon putting into astrobear ...





so $E(\rho, T)$, $P(\rho, T)$ are both monotone, so others ($E(\rho, P)$, etc.) can be obtained through directly inverting these two tables.
$CV(\rho, E) = CV(\rho, E(\rho, T)) = CV(\rho, T)$

11 01-26-2019

- X and Z interpolation: viable
- putting into AstroBEAR

12 02-02-2019

Opacity:
Opacity in MESA: see /mesa-r11701/kap/notes for more details.
Opacity tables are much more complicated,

- "for each Z_init = 0.000, 0.001, 0.004, 0.010, 0.020, 0.030, 0.050, 0.100
  for each X = 0.00, 0.03, 0.10, 0.35, 0.70
  There are about 60 tables for various (XC, XO) pairs (carbon and oxygen abudance).
- conductive opacity?

02-03-2019

Putting EOS into AstroBEAR
Now that it works.. Added EOS_TABULAR case into all subroutines, tested that ρ, T, P, E all works out. Couple of things to notice.

1. Changing the internal energy unit would result in lost of metalicity. So the underlining table stays in the unit of erg/g, while converting to erg/cm3 in the run.
2. TempScale and rScale are not a thing in implimenting the tables.

Next Step:

1. The sound speed tables are wrong, need to get the correct tables.
2. Enthalpy tables.
3. Roe tables.

02-16-2020

1. So MESA EOS $d\log(\rho) = 0.0125$; $d\log T = 0.02$.

02-27-2020

Total energy in a simulation:
Let 1 be Neutral gas, 2 be ionized. Let E_{ext} be the internal binding energy (such as ionization energy)

$$E_1 = E_{\text{kin }1} + E_{\text{pot }1} + E_{\text{int }1} + E_{\text{ext }1}$$

$$E_2 = E_{\text{kin }2} + E_{\text{pot }2} + E_{\text{int }2} + E_{\text{ext }2}$$

If a gas evolves from state 1 to 2, then energy is conserved

$$E_1 = E_2$$

Since E_{ext} is not measurable in simulation, define

$$\tilde{E}_1 = E_1 - E_{\text{ext }1}$$

$$\tilde{E}_2 = E_2 - E_{\text{ext }2}$$

If $E_{\text{ext }1} = 0$, neutral gas has ionization energy 0, then $E_{\text{ext }2} < 0$. Thus

$$\tilde{E}_2 > E_2 = E_1 = \tilde{E}_1 \tag{4}$$

Thus Ionization increases "measured energy"

13 05-04-2020

```
subroutine get_result_names(names)
   character (len=8) :: names(nv)
   names(i_lnPgas) = 'lnPgas'
   names(i_lnE) = 'lnE' ! internal energy per gram
   names(i_lnS) = 'lnS' ! entropy per gram
   names(i_grad_ad) = 'grad_ad' ! dlnT_dlnP at constant S
   names(i_chiRho) = 'chiRho' ! dlnP_dlnRho at constant T
   names(i_chiT) = 'chiT' ! dlnP_dlnT at constant Rho
   names(i_Cp) = 'Cp' ! dE_dT at constant P, specific heat at constant pressure
   names(i_Cv) = 'Cv' ! dE_dT at constant Rho, specific heat at constant volume
   names(i_dE_dRho) = 'dE_dRho' ! at constant T
   names(i_dS_dT) = 'dS_dT' ! at constant Rho
   names(i_dS_dRho) = 'dS_dRho'  ! at constant T
   names(i_mu) = 'mu'
   names(i_lnfree_e) = 'lnfree_e'
   names(i_gamma1) = 'gamma1'  ! dlnP_dlnRho at constant S
   names(i_gamma3) = 'gamma3'  ! gamma3 - 1) = '' dlnT_dlnRho at constant S
   names(i_eta) = 'eta'
end subroutine get_result_names
```

Figure 2: screenshot of EOS definitions from eos/public/eos_def.f

Table 3. `eos` output quantities and units

Output	Definition	Units
$P_{\rm gas}$	gas pressure	ergs cm^{-3}
E	internal energy	ergs g^{-1}
S	entropy per gram	ergs g^{-1} K^{-1}
$dE/d\rho\|_T$		ergs cm^3 g^{-2}
$C_{\rm V}$	specific heat at constant $V \equiv 1/\rho$	ergs g^{-1} K^{-1}
$dS/d\rho\|_T$		ergs cm^3 g^{-2} K^{-1}
$dS/dT\|_\rho$		ergs g^{-1} K^{-2}
χ_ρ	$\equiv d\ln P/d\ln\rho\|_T$	none
χ_T	$\equiv d\ln P/d\ln T\|_\rho$	none
$C_{\rm P}$	specific heat at constant pressure	ergs g^{-1} K^{-1}
$\nabla_{\rm ad}$	adiabatic T gradient with pressure	none
Γ_1	$\equiv d\ln P/d\ln\rho\|_S$	none
Γ_3	$\equiv d\ln T/d\ln\rho\|_S + 1$	none
η	ratio of electron chemical potential to $k_B T$	none
μ	mean molecular weight per gas particle	none
$1/\mu_e$	mean number of free electrons per nucleon	none

Figure 3: Variables with unit in MESA EOS, adapted from MESA 1: https://arxiv.org/pdf/1009.1622.pdf

Convective stability requires

$$\left|\frac{dT}{dz}\right| < \left|\frac{dT}{dz}\right|_{ad}$$

Ideal gas EOS, with a polytrophic $p - \rho$ relation
(see http://www.pas.rochester.edu/ blackman/ast462/ast462lecture9.pdf)

$$\left|\frac{dT}{dz}\right| < \left(1 - \frac{1}{\gamma}\right)\frac{T}{P}\left|\frac{dP}{dz}\right|$$

MESA EOS outputs $\frac{d\ln T}{d\ln P}|_{ad}$, so that

$$\frac{dT}{dP} = \frac{dT}{d\ln T}\frac{d\ln T}{d\ln P}\frac{d\ln P}{dP} = \frac{T}{P}\frac{d\ln T}{d\ln P}$$

multiply the stability condition by $\frac{dz}{dP}$ on both sides

$$\left|\frac{dT}{dP}\right| < \left|\frac{dT}{dP}\right|_{ad}$$