# EOS_Omni

Christian D. Ott <cott@tapir.caltech.edu>,
Erik Schnetter <eschnetter@perimeterinstitute.ca>

February 24, 2013

**Abstract**

This thorn provides a unified EOS (Equation Of State) interface and implements multiple analytic EOS, and also provides table reader and interpolation routines for finite-temperature microphysical EOS available from `http://www.stellarcollapse.org`. In addition cold and barotropic tabulated EOS are provided for use in initial data thorns. Currently, the implemented analytic EOS are the polytropic EOS, the gamma-law EOS, and a hybrid EOS consisting of a 2-piece piecewise-polytrope with a thermal, gamma-law component.

## 1 Introduction

Equations of State (EOS) are crucial for hydrodynamics and hydro codes (as well as other codes needing/providing microphysics) are closely coupled to EOS and call EOS routines many times for each grid point during the calculation of a time update.

`EOS_Omni` is presently coded for cold and hot EOS, including those based on microphysical models. It does currently assume nuclear statistical equilibrium (NSE) with rest-mass density $\rho$, specific internal energy $\epsilon$ (or temperature $T$), and electron fraction $Y_e$ being the independent variables. `EOS_Omni` can be called on arrays or on single grid points.

## 2 Units

This thorn uses *solar* units where $c = G = M_\odot = 1$. Temperatures are measured in MeV.

## 3 Using This Thorn

### 3.1 Basic Usage

`EOS_Omni` works via the aliased-function interface, and EOS functions to be used must be declared in `interface.ccl`. Here is an example `interface.ccl` entry:

```
void FUNCTION EOS_Omni_press(CCTK_INT IN eoskey,         \
                             CCTK_INT IN havetemp,       \
                             CCTK_REAL IN rf_precision,  \
                             CCTK_INT IN npoints,        \
                             CCTK_REAL IN ARRAY rho,     \
                             CCTK_REAL INOUT ARRAY eps,  \
                             CCTK_REAL INOUT ARRAY temp, \
                             CCTK_REAL IN ARRAY ye,      \
                             CCTK_REAL OUT ARRAY press   \
                             CCTK_INT OUT ARRAY keyerr,  \
                             CCTK_INT OUT anyerr)
```

Here,

- `eoskey` is the type of EOS to be used in this call.
  - `eoskey = 1`: Polytropic EOS
  - `eoskey = 2`: Gamma-Law EOS
  - `eoskey = 3`: Hybrid EOS (2 Polys, 1 Gamma-Law), used for stellar core collapse simulations.
  - `eoskey = 4`: Finite-temperature microphysical EOS
  - `eoskey = 5`: Cold tabulated EOS with Gamma-Law
  - `eoskey = 6`: Tabulated barotropic EOS
- `havetemp` determines whether the EOS is to be called as a function of $(\rho, \epsilon, Y_e)$ (`havetemp = 0`), or as a function of $(\rho, T, Y_e)$ (`havetemp = 1`). `havetemp = 0` is the method of choice for analytic EOS during evolution, but at the initial data stage one may need to set `havetemp = 1` (with $T = 0$) to obtain initial values for $\epsilon$. In the case of a finite-temperature EOS (that usually is a function of $(\rho, T, Y_e)$), a call with `havetemp = 0` will first lead to the solution of $T(\rho, \epsilon, Y_e)$ via a Newton-Raphson-type iteration (using the supplied value of $T$ as the starting point) and will then calculate the requested dependent variable as a function of $X = X(\rho, T, Y_e)$. Both $X$ and the updated $T$ are returned.
- `npoints` tells the EOS how many data points are passed in,
- `rho,eps,temp,ye,press` are obvious,
- `rf_precision` is a real number telling the root finding algorithm (for finding $T(\rho, \epsilon, Y_e)$) at what relative error to terminate the iteration. $10^{-10}$ is a good value.
- `keyerr` is an array (with $n$ entries for $n$ data points) containing error codes (relevant only for tabulated microphysical EOS),
- `anyerr` is an integer $> 0$ in case any error occured.

## 3.2 Parameter Settings

Many hydro codes require a fallback EOS in case something goes wrong. This is also true for the Einstein Toolkit GR hydro code `EinsteinEvolve/GRHydro`. If you want to use `EOS_Omni` with `GRHydro`, you must set parameters for the EOS of your choice and, *in addition*, the following parameters must be set to sensible values:

```
eos_omni::poly_gamma
eos_omni::poly_gamma_initial
eos_omni::poly_k
```

The only non-obvious parameter here is `poly_gamma_initial`. In most simulations it should not be set. In simulations that are run with a different adiabatic index than what was used to set up the initial data, `poly_gamma` should be the evolution value, and `poly_gamma_initial` should be the initial data value. `EOS_Omni` then rescales `poly_k` such that the *cgs* value of `poly_k` is the same for initial data and evolution. Since the units of `poly_k` ($[K] = \frac{\mathrm{N}}{\mathrm{m}^2}\left(\frac{\mathrm{m}^3}{\mathrm{kg}}\right)^\Gamma$) depend on `poly_gamma` this is not fully trivial.

Check `param.ccl` for parameters for the other EOS.

# 4 Equations of State Details

## 4.1 Polytropic

The `poly` EOS is a *polytropic* equation of state, which does not allow for changes in entropy:

$$p = K\rho^\gamma \tag{1}$$

where $p$ is the pressure, $\rho$ the density, $K$ the polytropic constant set via `poly_k`, and $\gamma$ is the adiabatic index set via `poly_gamma`.

If the internal energy $\epsilon$ is to be "calculated from the temperature" (`havetemp = 1`), then this is done using the relation

$$\epsilon = \frac{K}{\gamma - 1}\rho^{\gamma-1} \tag{2}$$

(which actually ignores the temperature).

Note: This polytropic EOS is also used as fall-back when other EOS fail.

## 4.2 Gamma-Law

The `gl` EOS is a *gamma-law* equation of state, corresponding to an ideal gas:

$$p = (\gamma - 1)\rho\epsilon \tag{3}$$

where $p$ is the pressure, $\rho$ the density, $\epsilon$ the internal energy, and $\gamma$ is the adiabatic index set via `gl_gamma`.

At the initial data stage, it may be necessary to set up initial values for $\epsilon$. For this, the `gl` EOS implements equation (**??**) just like the `poly` EOS and the parameters `poly_gamma_ini` and `gl_k` must be set for this.

## 4.3 Hybrid

The hybrid EOS was introduced by [**?**] for use in simplified simulations of stellar collapse to mimic (1) the stiffening of the nuclear EOS at nuclear density and (2) to include thermal pressure in the postbounce phase. It consists of two polytropes characterized by ($K_1$, $\gamma_1$) and ($K_2$, $\gamma_2$) and a thermal $\gamma-$law component described by $\gamma_{\rm th}$. Polytrope 1 is soft and describes a gas of relativistic degenerate electrons with $\gamma_1 \approx 4/3$. It is used below nuclear density ($\rho_{\rm nuc} \approx 2 \times 10^{14}\,{\rm g\,cm^{-3}}$), and is smoothly matched to polytrope 2 which applies above $\rho_{\rm nuc}$, is stiff, and models the repulsive core of the strong force above nuclear density ($\gamma_2 \gtrsim 2.5$). $K_2$ is completely determined by $P_1(\rho_{\rm nuc}) = P_2(\rho_{\rm nuc})$ and $K_1, \gamma_1$, and $\gamma_2$. The full functional form of the EOS $P = P(\rho, \epsilon)$ with the thermal component (which takes into account shock heating) is given by

$$P = \frac{\gamma - \gamma_{\rm th}}{\gamma - 1} K \rho_{\rm nuc}^{\gamma_1 - \gamma} \rho^{\gamma} - \frac{(\gamma_{\rm th} - 1)(\gamma - \gamma_1)}{(\gamma_1 - 1)(\gamma_2 - 1)} K \rho_{\rm nuc}^{\gamma_1 - 1} \rho + (\gamma_{\rm th} - 1)\rho\epsilon\,. \tag{4}$$

The `EOS_Omni` parameters for the hybrid EOS are the following:

| | |
|---|---|
| `hybrid_gamma1` | $\gamma_1$, $\gamma_1 = 1.325$ is an appropriate choice. |
| `hybrid_gamma2` | $\gamma_2$, $\gamma_2 = 2.5$ is an appropriate choice. |
| `hybrid_gamma_th` | $\gamma_{\rm th}$, perhaps 1.5. |
| `hybrid_k1` | $K_1$, 0.4640517 in solar units for relativistic degenerate $e^-$. |
| `hybrid_rho_nuc` | nuclear density, standard is $3.238607 \times 10^{-4}$ in solar units. |

## 4.4 Finite-Temperature Nuclear EOS

Complex microphysical finite-temperature equations of state come usually in tabulated form. `EOS_Omni` comes with routines provided as part of the `nuc_eos` package described in [**?**] and available at `http://www.stellarcollapse.org`. A variety of EOS tables for application in high-density astrophysical situations (i.e. in stellar collapse or in compact star mergers) are also available from there in HDF5 format.

The parameters controlling the finite-temperature nuclear EOS are the following:

| | | |
|---|---|---|
| `nuceos_read_table` | BOOLEAN | Set to `yes` to read table. |
| `do_energy_shift` | BOOLEAN | Set to `yes` to subtract the energy shift stored in the table to get correctly normalized $\epsilon$. |
| `nuceos_table_name` | STRING | Path/Name of the table file. |

## 4.5 Cold Tabulated Nuclear EOS with Gamma Law

Many equations of state for neutron stars are generated under the assumption of zero temperature. This is perfectly appropriate for cold old neutron stars. In simulations of binary mergers, however, shocks will drive the temperature up, adding a thermal pressure component, which can be accounted for approximately with a Gamma-Law: $P_{\rm th} = (\Gamma_{\rm th} - 1)\rho\epsilon_{\rm th}$ .

`EOS_Omni` implements such an equation of state. It reads in an ASCII EOS table (see subdirector `tables` for an example table for the SLy EOS [**?**, **?**], which was generated according to the prescription in [**?**, **?**]). All EOS parameters are read from the ASCII file, which has the following format:

```
EoSType = Tabulated
Nrho = 600    NYe = 1     NT = 1
RhoMin = 1e-09  RhoMax = 0.01
HeatCapacityE = 1
GammaTh = 2
```

```
Kappa = 1
RhoSpacing = Log
    1.57940636422747e-03    1.38773826035349e+00    2.62139412738900e-02
[...]
    2.81804006881059e+00    6.89967555695907e-01    8.30537612378975e-01
```

The header completely determines the range in baryon rest mass density (in $c = G = M_\odot = 1$ units), gives the number of zones (currently only `NYe = 1`, `NT = 1`, `HeatCapacityE = 1`, and `RhoSpacing = Log` are supported). `GammaTh` is the $\Gamma_{\mathrm{th}}$ of the thermal gamma law. The tabulated columns are $\epsilon$, $\Gamma$, $c_s$ (the speed of sound of the cold component of the EOS). The pressure is obtained via $P = \kappa\rho^\Gamma$, where $\kappa$ is the `Kappa` scaling parameter.

Generally, $P = P(\rho, \epsilon)$ in this EOS, but note that $\epsilon_{\mathrm{th}} = \epsilon - \epsilon_{\mathrm{cold}}$. `EOS_Omni` uses linear interpolation to first find $P(\rho)$ and $\epsilon_{\mathrm{cold}}(\rho)$ and then computes the thermal component analytically.

| | | |
|---|---|---|
| `coldeos_read_table` | BOOLEAN | Set to `yes` to read table. |
| `coldeos_use_thermal_gamma_law` | BOOLEAN | Set to `yes` to use the thermal gamma law (default). |
| `coldeos_table_name` | STRING | Path/Name of the table file. |

# 5 Converting Old Parameter Files

If you have a parameter file that uses the previous EOS interface in Cactus, you will have to convert it so that it runs with `EOS_Omni`. The following describes a set of simple rules for this conversion.

1. Add `EOS_Omni` to the thorn list. You can then remove all other `EOS_*` thorns from the thorn list (or you can leave them in; they are unused).
2. Activate `EOS_Omni` in the parameter file: Add `EOS_Omni` to your active thorns, and do not activate any other `EOS_*` thorns.
3. Translate all EOS parameters according to table **??**.
4. All thorns using this EOS interface will have a parameter that determines which EOS to use, typically via a string/keyword parameter specifying an EOS name. Convert these names using table **??**.

| Old Parameter | Old Value | New Parameter | New Value |
|---|---|---|---|
| EOS_Polytrope::eos_gamma | | EOS_Omni::poly_gamma | |
| EOS_Polytrope::eos_k | | EOS_Omni::poly_k | |
| EOS_Polytrope::use_cgs | yes | — | |
| EOS_Polytrope::use_cgs | no | unsupported | |
| EOS_Polytrope::gamma_ini | | EOS_Omni::poly_gamma_ini | |
| EOS_Ideal_Fluid::eos_ideal_fluid_gamma | | EOS_Omni::gl_gamma | |
| **TODO: complete this table** | | | |

Table 1: Parameter conversion table

# References


[1] Janka, H.-T., Zwerger, T., & Moenchmeyer, R. 1993, Astron. Astrophys., 268, 360

| EOS | Description | Old Name | New Name |
|---|---|---|---|
| poly | polytropic | ??? | 2D_Polytrope |
| gl | gamma-law | ??? | Ideal_Fluid |
| hybrid | hybrid | ??? | Hybrid |
| nuc | finite-temperature nuclear | ??? | nuc_eos |

Table 2: EOS name conversion table

[2] O'Connor, E., & Ott, C. D. 2010, Class. Quantum Grav., 27, 114103

[3] Douchin, F., & Haensel, P. 2001, Astron. Astrophys., 380, 151

[4] Haensel, P., & Potekhin, A. Y. 2004, Astron. Astrophys., 428, 191

[5] Shibata, M., Taniguchi, K., & Uryū, K. 2005, Phys. Rev. D, 71, 084021

[6] Corvino, G., Rezzolla, L., Bernuzzi, S., De Pietri, R., & Giacomazzo, B. 2010, Class. Quantum Grav., 27, 114104