

# GR1D v1.0

An Open-Source Code for Stellar Collapse and Black Hole Formation

Evan O'Connor & Christian D. Ott, California Institute of Technology

e-mail: [GR1D@stellarcollapse.org](mailto:GR1D@stellarcollapse.org)

## 1 Disclaimer and Software License

GR1D is open source and may be used at your own risk. The code comes with absolutely no warranty and we are unable to guarantee that we will be able to provide support or help if you run into problems running it. If you decide to use GR1D in published work, it is YOUR responsibility to test the code and ensure its physical correctness and consistency.

**If you find bugs or inconsistencies, please let us know via e-mail to [GR1D@stellarcollapse.org](mailto:GR1D@stellarcollapse.org)!** We are also open for collaboration on extending GR1D to include more physics. However, please avoid contacting us with questions like: “How do I get GR1D to read in my stellar profile?” Just remember that GR1D is plain Fortran 90 and that basic knowledge of that language and the ability to use a text editor will solve this problem for you.

While GR1D is open source, its copyright is held by Evan O'Connor and Christian D. Ott. In the absence of suitable open scientific software licenses, we release this version of GR1D to the community under the **Creative Commons** *attribution-noncommercial-share* alike license whose details can be found at <http://creativecommons.org/licenses/by-nc-sa/3.0/us>. Essentially, you may use GR1D, but must make reference to our work, must not use GR1D for commercial purposes, and any code including or using our routines or part of them may be made publically available, and if so, only under the same license.

When using GR1D in published work, please make reference to the paper describing it:

E. O'Connor and C. D. Ott, *A New Spherically-Symmetric General Relativistic Hydrodynamics Code for Stellar Collapse to Neutron Stars and Black Holes*, submitted to *Class. Quant. Grav.*, arXiv:0912.2393, 2009.

## 2 Compiling:

### 2.1 make.inc

All settings necessary for successful compilation of GR1D must go into the file *make.inc* which is read by the make system. In particular, the compiler, compiler flages (e.g., optimization, debugging options), and the location of external library include and library files must be provided. GR1D is written in Fortran 90. It has been tested to work with gcc (*gfortran*) 4.3.3 & 4.4.2 and ifort 11. The compiler options must be set in the *make.inc* file.

To activate support for finite-temperature EOS and neutrino leakage, the following preprocessor definitions must be set in *make.inc*:

```
HAVE_NUC_EOS=1
HAVE_LEAK_ROS=1
```

### 2.2 External libraries

GR1D requires a set of HDF5 libraries and include files compiled for use with your Fortran 90 compiler. We use HDF5 libraries to write restart files and also to read in our EOS tables. The source files for HDF5 are available from the HDF5 Group at <http://www.hdfgroup.org/HDF5/> along with straightforward installation instructions. The path to the libraries and include files must be set in the *make.inc* file. Our EOS table are made with HDF5 v1.8.2, therefore this version or later is recommended.

## 2.3 Compiling & Running

Once all the settings have been set, running `>make` will compile the code. If settings in `make.inc` have changed, first run `>make clean` then `>make`. To run GR1D, simply enter the command `>GR1D`.

## 3 .short initial data format

GR1D uses the '.short' format for input stellar profiles. On the first line of the input file is the number of lines in the input file. The rest of the file consists of lines with 8 columns each: (1) zone index, (2) enclosed mass (g), (3) radial coordinate (cm), (4) temperature (K), (5) density ( $\text{g/cm}^3$ ), (6) radial velocity (cm/s), (7)  $Y_e$ , and (8)  $\Omega$  (rad/s). We include in this version of GR1D an  $n = 3$  polytrope with a central density of  $5 \times 10^{10} \text{ g/cm}^3$  and total mass of  $1.435 M_\odot$ . The polytrope may be used for collapse calculations with the hybrid EOS. For runs with finite-temperature EOS and neutrino leakage, a presupernova stellar profile is necessary. Various presupernova models are available for download from stellar evolution groups, (c.f. [http://www.stellarcollapse.org/other\\_resources](http://www.stellarcollapse.org/other_resources)).

## 4 Sample parameter files

We include sample parameter files for test cases and some collapse simulations. These follow closely the simulations presented in our GR1D methods paper, [1], and include: (1) Shocktube problems #1 & #2, (2) Newtonian Sedov blast wave, (3) Oppenheimer-Snyder Collapse of a pressureless dust ball, (4) Hybrid-EOS core collapse of a progenitor model, (5) Microphysical EOS core collapse with neutrino leakage/heating of a progenitor model.

The sample parameter files live in the subdirectory `sample_parameter_files` and must be copied into the file `parameters` in the code's main directory.

## 5 Grid Types

GR1D has several pre-programmed grid arrangements we find work well for different situations. GR1D fully supports non-equally spaced grids and defines all variables at cell centers. Available grids through the `gridtype` parameter are:

<i>unigrid</i>	Divides the domain into zones of equal size.
<i>log</i>	Logarithmic progression of the zone size starting from the center out to the domain edge. The central grid spacing in 'log' is set to the parameter <code>grid_custom_dx1</code>
<i>custom</i>	Sets up a region of constant zone width near the origin and a logarithmic progression outside. The value for the inner region zone widths is set through <code>grid_custom_dx1</code> and the extent of this inner region is set through <code>grid_custom_rad1</code> .
<i>custom2</i>	Follows 'custom' closely, but the <code>grid_custom_number</code> innermost zones have zone widths logarithmically increasing to a value of <code>grid_custom_inner</code> in the innermost zone.

## 6 Restart Files

GR1D can create restart files that can be used to restart a simulation at a later time. These files are in HDF5 format. The files are checked for consistency of key parameters between original run and restart. Hence, it is important that the parameter file of the restarted run is the same as the original. If you happen to need to change these parameters between restarts, please comment out the parameter checks.

## 7 $Y_e(\rho)$

GR1D implements a parametrized  $Y_e(\rho)$  for the prebounce evolution of stellar cores. GR1D can interpolate a numerical profile or use the fitting formula of [2]. The fitting constants are set in the parameter file.

## 8 Rotation

If rotation is included, an option of analytically setting the angular velocity profile is available through the following formula,

$$\Omega = \Omega_c / \left[ 1 + \left( \frac{r}{A} \right)^2 \right] \quad (1)$$

$\Omega_c$  and  $A$  are set in the parameter file. In addition to this,  $\Omega$  can be read in from the initial stellar profile.

## 9 Parameter File

Here we list all the options in the parameter file and give a brief description.

### • Job Parameters

- `jobname`: text, Description of job.
- `GR`: integer, 1: for GR, 0: for Newtonian
- `outdir`: text, location of stored files, must exist, *Data/* suggested
- `initial_data`: text, type of simulation: 'Collapse', 'OSC', 'Sedov', 'Shocktube'
- `profile_name`: text, location of input profile used in 'Collapse' simulations
- `profile_type`: integer, 1: for .short format
- `gravity_active`: integer, 0: for no gravity, 1: for gravity
- `ntmax`: integer, maximum hydro steps
- `tend`: real, maximum simulation time (seconds)

### • Grid Parameters

- `geometry`: integer, 1: for linear grid, 2: for spherical
- `gridtype`: text, type of grid arrangement: 'unigrid', 'log', 'custom', 'custom2'. See 5 for description.
  - \* `grid_custom_dx1`: real, (cm)
  - \* `grid_custom_rad1`: real, (cm)
  - \* `grid_custom_number`: integer
  - \* `grid_custom_inner`: real, (cm)
- `rmax_from_profile`: integer, 0: specify `rmax` in `grid_rmax`, 1: set `rmax` from density in `rho_cut`.
- `rho_cut`: real, density to stop grid at ( $\text{g}/\text{cm}^3$ ).
- `grid_rmax`: real, radius to stop grid at (cm)
- `radial_zones`: integer, number of radial zones
- `ghosts1`: integer, number of ghost zones

### • Hydro Parameters

- `do_hydro`: integer, 0: no hydrodynamic step 1: do hydrodynamic step

- `cffac`: real, Courant factor
- `iorder_hydro`: integer, order of Runge-Kutta
- `reconstruction_method`: text, reconstruction method: 'ppm', 'tvd', 'pc'
- `ppm_origin_TVD`: integer, how many zones of TVD at the origin
- `tvd_limiter`: text, TVD limiter: 'MC', 'minmod'
- `flux_type`: text, Riemann solver, only 'HLLE'

- **EOS Parameters**

- `eoskey`: integer, type of EOS: 1: Hybrid EOS, 2: Polytrope EOS, 3: Nuclear EOS, 4: Gamma-Law EOS
- `eos_table_name`: text, location of `.h5` Nuclear EOS table
- `hybridgamma.th`: real, Hybrid EOS thermal index  $\Gamma_{th}$
- `hybridgamma1`: real, Hybrid EOS low density index  $\Gamma_1$
- `hybridgamma2`: real, Hybrid EOS high density index  $\Gamma_2$

- **Output Parameters**

- `ntinfo`: integer, prints info to stdout after this many hydro steps
- `dynamic_output_control`: integer, 0: output frequency stay constant, 1: increases output near bounce and black hole formation.
- `vs_mass`: integer, 0: for `.xg` files versus radial coordinate, 1: for `.xg` files versus enclosed baryonic mass
- `small_output`: integer, 0: output all `.xg` files, 1: output smaller subset of `.xg` files
- `dtout`: real, time between output of `.xg` files (seconds)
- `dtout_scalar`: real, time between output of `.dat` files (seconds)
- `ntout`: integer, hydro steps between output of `.xg` files, (-1 to ignore)
- `ntout_scalar`: integer, hydro steps between output of `.dat` files (-1 to ignore)

- **Restart Parameters**

- `ntout_restart`: integer, hydro steps between output of `.h5` restart files (-1 to ignore)
- `dtout_restart`: real, time between output of `.h5` restart files (seconds)
- `do_restart`: integer, 0: no restart, 1: restart
- `restart_file_name`: text, location of `.h5` file to restart from

- **Neutrino Parameters**

- `fake_neutrinos`: integer, 0: for no neutrino physics, 1: for neutrino physics
- `ye_of_rho`: integer, 0: no  $Y_e(\rho)$ , 1: use  $Y_e(\rho)$  prescription
  - \* `ye_profile_name`: text, profile name of numerical  $Y_e(\rho)$  table to interpolate
  - \* `do_yeofrho_fit`: integer, 0: use profile 1: use fit values, see section 7
  - \* `yeofrho_rho1`: real, lower density limit of  $Y_e(\rho)$  fit, ( $\text{g}/\text{cm}^3$ )
  - \* `yeofrho_rho2`: real, upper density limit of  $Y_e(\rho)$  fit, ( $\text{g}/\text{cm}^3$ )
  - \* `yeofrho_ye1`: real,  $Y_e$  at lower density limit
  - \* `yeofrho_ye2`: real,  $Y_e$  at upper density limit
  - \* `yeofrho_yc`: real,  $Y_e$  correction factor
- `neutrino_pressure`: integer, 0: no neutrino pressure, 1: include neutrino pressure
- `do_leak_ros`: integer, 0: no neutrino leakage/heating, 1: include neutrino leakage/heating
- `do_heating`: integer, 0: only neutrino leakage, 1: include heating
- `heat_fac`: real, ad-hoc factor for scaling heating

- **Atmosphere Parameters**

- `atmo_rho_rel_min`: real, sets the minimum relative value of the density for the atmosphere (fraction).
- `atmo_rho_abs_min`: real, sets the absolute value of the atmosphere density, recommended value,  $\sim 10\%$  less than `rho_cut`.
- `atmo_fac`: real, fraction of atmosphere density to above densities.

- **Rotation**

- `do_rotation`: integer, 0: no rotation, 1: rotation included
- `set_omega`: integer, 0: use rotation from `.short` file, 1: analytically set omega through Eq. 1
- `omega_c`: real, central value of omega in Eq. 1
- `omega_A`: real, radial factor in Eq. 1

- **Test problems**

- `shocktube_problem`: integer, various initial shocktube settings, see `src/shocktube.F90`

## References

- [1] E. O'Connor & C. D. Ott, *submitted to Class. Quant. Grav.*, **arXiv:0912.2393**, 2009.
- [2] M. Liebendörfer. *Astrophys. J.*, **633**, 1042, 2005.