GR1D v2.0

Nov 20th, 2014

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

Evan O'Connor, North Carolina State University & Christian D. Ott, California Institute of Technology

e-mail: 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! This is particularly probable if you are using GR1D in a way we have not thought of yet. 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.

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*, Class. Quantum Grav., 2009 27 114103

When using the neutrino transport model of GR1D in published work, in addition to the above reference, please make reference to the paper describing it:

> E. O'Connor, *An Open-Source Neutrino Radiation Hydrodynamics Code for Core-Collapse Supernovae*, *eubmitted to The Astrophysical Journal: Supplement Series*, 2014 arXiv:1411.????

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. The compiler options must be set in the *make.inc* file. The authors suggest using the MESA SDK available at http://www.astro.wisc.edu/~townsend/static.php?ref=mesasdk.

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
```

For the neutrino transport module you must have

```
HAVE_LAPACK=1
```

in addition to setting the LAPACK library above.

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 and opacity 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. This library is included with the MESA SDK linked above.

GR1D also requires LAPACK for the matrix inversion and the path must be set in the *make.inc* file. Here is the LAPACK website, http://www.netlib.org/lapack/. The LAPACK libraries are also available in the MESA SDK

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 (g/cm^3), (6) radial velocity (cm/s), (7) Y_e, and (8) Ω (rad/s). We include in this version of GR1D an $n = 3$ polytrope with a central density of 5×10^{10} 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 but note different groups define variables differently, zone center or zone interface values for example, *src/map_profile.F90* may need to be adjusted accordingly). For example, there is a parameter setting for the common KEPLER models that assumes the radial coordinate is the outer edge of the zone rather than the center. This can make a large difference depending on the intial profile, be careful.

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 progentior model, (5) Microphysical EOS core collapse with neutrino leakage/heating of a progenitor model. There are also sample parameter files for the neutrino radiation transport that reproduce the results in [2].

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:

unigrid	Devides the domain into zones of equal size.
log	Logarthmic 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 `grid_custom_dx1`
custom	Sets up a region of constant zone width near the origin and a logorthmic progression outside. The value for the inner region zone widths is set through `grid_custom_dx1` and the extent of this inner region is set through `grid_custom_rad1`.
custom2	Follows 'custom' closely, but the `grid_custom_number` innermost zones have zone widths logorithically increasing to a value of `grid_custom_inner` 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. Older restart files may not work with updated version of GR1D.

7 $Y_e(\rho)$

`GR1D` implements a parametrized $Y_e(\rho)$ for the prebounce evolution or stellar cores. `GR1D` can interpolate a numerical profile or use the fitting formula of [3]. 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,

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

Ω_c and A are set in the parameter file. In addition to this, Ω can be read in from the initial stellar profile.

9 Neutrino Transport

As of version 2, `GR1D` now has a neutrino transport module. The associated paper is [2] and is the main reference for all of the details. Below we describe the possible parameter file settings.

10 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','M1test'
  - `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 (g/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 Γ_{th}
  - `hybridgamma1`: real, Hybrid EOS low density index Γ_1
  - `hybridgamma2`: real, Hybrid EOS high density index Γ_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

- **M1 Parameters**
  - `do_M1`: integer, 0 disables transport, 1 enables it
  - `v_order`: integer, -1 full velocity dependence, 0 for none
  - `extraction_radii`: real, radii to evolve neutrinos out to and where data is extracted
  - `number_species`: integer, number of neutrino species, currently only 3 works without minimal adjustment
  - `number_groups`: integer, number of neutrino energy groups, must match table
  - `opacity_table`: string, path to nulib opacity table
  - `number_eas`: integer, number of variables in the eas dataset of the nulib table, usually 3
  - `M1closure`: string, denotes closure type, only a couple implemented
  - `testcases`: integer, denotes the number of testcase to perform, initial_data must be 'M1test'
  - `include_epannihil_kernels`: integer, 1 for including nux pair produciton with the kernels (triggers read from table)
  - `include_nes_kernels`: integer, 1 for including neutrino electron scattering with kernels (triggers read from table)
  - `nes_evolution_type`: integer, 0 for no inelastic scattering, 1 for including neutrino electron scattering explicitly, 2 for implicitly
  - `energy_coupling_type`: integer, 0 for no energy coupling, 1 for including energy coupling explicitly, 2 for implicitly
  - `M1_control`: integer, 0 for no control, parameters stay constant. 1 for primative control, parameters change as below
    - `M1_phase1phase2_density`: real, central density that triggers phase 2
    - `M1_phase2phase3_pbtime`: real, post bounce time that triggers phase 3
    - `M1_phase1_reconstruction`: string, reconstruction method for both matter and neutrinos (phase 1, value must match the value given above)
    - `M1_phase2_reconstruction`: string, reconstruction method for both matter and neutrinos (phase 2)
    - `M1_phase3_reconstruction`: string, reconstruction method for both matter and neutrinos (phase 3)
    - `M1_phase1_cffac`: real, Courant factor, (phase 1)
    - `M1_phase2_cffac`: real, Courant factor, (phase 2, typically needs to be reduced near bounce)
    - `M1_phase3_cffac`: real, Courant factor, (phase 3)
    - `M1_phase1_ns`: integer, number of species to evolve (phase 1; 1 is only nue, 3 is all three)
    - `M1_phase2_ns`: integer, number of species to evolve (phase 2; 1 is only nue, 3 is all three)
    - `M1_phase3_ns`: integer, number of species to evolve (phase 3; 1 is only nue, 3 is all three)
    - `M1_phase1_ies_way`: integer, 0 for no inelastic scattering, 1 for including neutrino electron scattering explicitly, 2 for implicitly (phase 1, value must match the value given above)
    - `M1_phase2_ies_way`: integer, 0 for no inelastic scattering, 1 for including neutrino electron scattering explicitly, 2 for implicitly (phase 2)

    * `M1_phase3_ies_way`: integer, 0 for no inelastic scattering, 1 for including neutrino electron scattering explicitly, 2 for implicitly (phase 3)
    * `M1_phase1_encpl_way`: integer, 0 for no energy coupling, 1 for including energy coupling explicitly, 2 for implicitly (phase 1, value must match the value given above)
    * `M1_phase2_encpl_way`: integer, 0 for no energy coupling, 1 for including energy coupling explicitly, 2 for implicitly (phase 2)
    * `M1_phase3_encpl_way`: integer, 0 for no energy coupling, 1 for including energy coupling explicitly, 2 for implicitly (phase 3)

- **Leakage 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_yeofrhofit`: integer, `0`: use profile `1`: use fit values, see section 7
    * `yeofrho_rho1`: real, lower density limit of $Y_e(\rho)$ fit, (g/cm^3)
    * `yeofrho_rho2`: real, upper density limit of $Y_e(\rho)$ fit, (g/cm^3)
    * `yeofrho_ye1`: real, Y_e at lower density limit
    * `yeofrho_ye2`: real, Y_e at upper density limit
    * `yeofrho_yec`: 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
  - `do_NNBrem`: integer, `0`: do not include Nucleon-Nucleon Bremsstrahlung, `1`: include it
- **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 then `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*

11 Version 1.01, 1.02, 1.03, & v2.0 Updates

11.1 Physics Updates

- We correct the source terms for the neutrino leakage (*src/leakage_rosswog/leak_rosswog.F90*) and neutrino pressure contributions (*src/neutrino_pressure.F90*) to the evolution equations.
- v1.01 now includes Newtonian neutrino leakage

- v1.0 had issues when leakage was preformed near the outer edge of the grid (for example with neutron stars). We have fixed all the loop bounds and initialized previously uninitialized variables.
- We correct the Newtonian source terms for neutrino pressure and leakage (v1.0 missing $\sqrt{\gamma} = r^2$)
- For `GR`=1, v1.01 now fully includes the effect of the moving fluid on the redshifted luminosity of the neutrinos (*src/leakage_rosswog/leak_rosswog.F90*).
- *custom2* grid setup had a bug in v1.0 that would put the switch between constant grid and log grid at larger radii then requested.
- v1.01 leakage scheme now uses the neutron rest mass as the conversion between ρ and baryon density instead of the atomic mass unit. This is to be more consistant with our EOS tables.
- $T/|W_{grav}|$ and J were defined wrong in v1.0, factors of $2/3 = \int \sin(\theta)^3 d\theta / \int \sin(\theta) d\theta$ to account for averages over the sphere.
- $T/|W_{grav}|$ now defined for Newtonian simulations.
- $T/|W_{grav}|$ is now calculated as a function of r and output both as an *.xg* file and a time series file (*ToverW_edge.dat*, with the value at the grid's outer edge).
- v1.01 will convert the mass variable in the initial data to solar masses if not already.
- v1.01 fixes a reconstruction bug where the temperature was being set to the value at the interface, v1.02 fixed a bug that wasn't apparent because of this reconstruction bug, keytemp=0 is now used exclusively to maintain accuracy in the eps variable.
- From tests we have been performing, we changed how we evolve our angular momentum. Our conserved variable is now $S_\phi = \rho h W^2 r v_\varphi$ this, along with the new source terms gives perfect angular momentum conservation for Newtonian simulations (as the source term is 0) and much better (factor of 40) angular momentum conservation for GR runs, as there are still source terms required but the largest is now gone. For Newtonian runs we reconstruct Ω at the interfaces and convert back to v_φ, for our GR runs this leads to problems as we are effectively evolving Ωr^2 and near the origin one can evolve to negative velocities if the wrong variable is reconstructed. Hence we reconstruct v_φ at the interfaces, note the side effect of this is a slight divergence of Ω near the orgin ($r < 1$km), while visuallly unappealing, this does not contribute to the angular momentum as for a constant density sphere the total angular momentum is $\propto r^5$. Given this, we recommend using the `grid_type` = 'custom' option for rotating runs which increases resolution near the origin.
- v1.03 now includes the option of include Nucleon-Nucleon Bremsstrahlung in the leakage scheme, the appropiate parameter file variable must be changed to `1` from the default value of `0`.
- v2.0 now includes a neutrino transport scheme.

11.2 Useability Updates

- In v1.0, some grids would be rejected by the accrection analysis, in v1.01 this is corrected and `GR1D` now gives a warning and disables the accretion analysis for simulations with $r_{max} < 300$ km this can be editted manually in *src/analysis.F90*.
- Also in the accrection analysis, bus errors could be generated for particular grid set ups, this was do to improper initialization and has been corrected.
- In v1.0 the shock radius was determined by locating the r corresponding to the max absolute velocity. This method fails for exploding stars and now is found locating the r corresponding to the minimum velocity which will be the material directly upstream of the shock. As of v1.03, the shock radius and zone index of the shock is now included in the restart file. **Old restart files will NOT be valid as the shock radius values will not be stored.**

- In v1.0 the statement 'Using Ye(rho) fit formula' was printed for a given set of neutrino parameters but not actually used.
- The conversion of units for Ω was done incorrectly for the output file *omega.xg*.
- We note that many pre-supernova input files define their quantities at different locations (cell interface vs. cell center), this can make a difference, for example, in the iron core mass of the initial profile if interpolated incorrectly. `GR1D` assumes the radial coordinate and density are both cell centered, adjustments should be made if this is not the case (e.g., the Woosley and Weaver, 1995 and Woosley et al., 2002 presupernova models).

References

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

[2] E. O'Connor, *submitted to ApJS*, 2014 arXiv:1411.????.

[3] M. Liebendörfer. *Astrophys. J.*, **633**, 1042, 2005.