In-class assignment # 15

Brian O'Shea,
PHY-905-002, Computational Astrophysics and Astrostatistics
Spring 2021

Instructions: In today's class we are going to work with the `pyro2` code (repository on GitHub; documentation), which you have already downloaded and installed. This is a Python code written by Michael Zingale (and documented in Zingale (2014)) which is intended to teach computational hydrodynamics, including the Euler equations. We are going to be experimenting specifically with the compressible solvers. Please consult the "Running" documentation for instructions on how to run the code, and the "Analysis Routines" documentation on plotting. We're going to do the following:

1. Examine the code in the `compressible/` subdirectory, which implements the algorithms we've looked at in class. Start in the file `compressible/simulation.py` (which is called from `pyro.py` in the top-level directory for `pyro2`), and make sure you understand what the code is doing in terms of setting up initial conditions, calculating the fluxes at cell faces, doing Riemann solves, and updating the solution.

2. Run some test problems with the default compressible solver, which is a 2^{nd} order finite volume method. In particular, try the Sod shock tube problem and the Sedov blast wave problem.

   Make plots of these and look at the solutions compared to the analytic solutions using the comparison scripts (follow the directions in the linked documentation).

3. For the Sod shock tube problem, try re-running it with varied parameters. See the compressible solver documentation;

   the general format to modify a parameter at the command line is to add an argument that looks like "`section.option=value`" – i.e., "`compressible.riemann=CGF`" to change the Riemann solver from the default (HLL. For the `compressible` solver, turn on and off the flattening, modify the artificial viscosity, change the limiter and Riemann solver, and change the EOS. Use the analysis tools to make plots compared to the analytic solution (which will end up in a file named `sod_compare.png`, probably in the `analysis/` directory). Rename those plots to something useful and submit them, along with an analysis of the variation. What trends do you see?

4. Time permitting, try doing the same thing with either the Sedov blast problem **or** with one of the other compressible solvers (`compressible_rk`, `compressible_fv4`, or `compressible_sdc`) and their modifiable parameters. How does this behave differently than the standard 2^{nd} order finite volume method?

As per usual, submit your code, plots, etc. via GitHub!