

pyro: a framework for hydrodynamics explorations and prototyping

Alice Harpole¹, Michael Zingale¹, Ian Hawke², and Taher Chegini³

¹ Department of Physics and Astronomy, Stony Brook University ² University of Southampton ³ University of Houston

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Open Journals](#)

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))

Summary

pyro is a Python-based simulation framework designed for ease of implementation and exploration of hydrodynamics methods. It is built in an object-oriented fashion, allowing for the reuse of the core components and fast prototyping of new methods.

The original goal of pyro was to learn hydrodynamics methods through example, and it still serves this goal. At Stony Brook, pyro is used with new undergraduate researchers in our group to introduce them to the ideas of computational hydrodynamics. But the current framework has evolved to the point where pyro is used for prototyping hydrodynamics solvers before implementing them into science codes. An example of this is the 4th-order compressible solver built on the ideas of spectral deferred corrections (the `compressible_sdc` solver). This implementation was used as the model for the development of higher-order schemes in the Castro hydrodynamics code (Almgren et al., 2010). The low Mach number atmospheric solver (`lm_atm`) is based on the Maestro code (Nonaka et al., 2010) and the pyro implementation will be used to prototype new low Mach number algorithms before porting them to science codes.

In the time since the first pyro paper (Zingale, 2014), the code has undergone considerable development, gained a large number of solvers, adopted unit testing through `pytest` and documentation through `sphinx`, and a number of new contributors. pyro's functionality can now be accessed directly through a `Pyro()` class, in addition to the original commandline script interface. This new interface in particular allows for easy use within Jupyter notebooks. We also now use HDF5 for output instead of Python's `pickle()` function. Previously, we used Fortran to speed up some performance-critical portions of the code. These routines could be called by the main Python code by first compiling them using `f2py`. In the new version, we have replaced these Fortran routines by Python functions that are compiled at runtime by `numba`. Consequently, pyro is now written entirely in Python.

The current pyro solvers are:

- linear advection (including a second-order unsplit CTU scheme, a method-of-lines piecewise linear solver*, a 4th-order finite-volume scheme*, a WENO method*, and advection with a non-uniform velocity field*)
- compressible hydrodynamics (including a second-order unsplit CTU scheme, a method-of-lines piecewise linear solver*, and two 4th-order finite-volume schemes, one with Runge-Kutta integration and the other using a spectral deferred corrections method*)
- diffusion using a second-order implicit discretization
- incompressible hydrodynamics using a second-order approximate projection method.
- low Mach number atmospheric solver*, using an approximate projection method.

41 ▪ shallow water equations solver*

42 (solvers since the first pyro paper are marked with a *). Also, new is support for Lagrangian
43 tracer particles, which can be added to any solver that has a velocity field.

44 Acknowledgements

45 The work at Stony Brook was supported by DOE/Office of Nuclear Physics grant DE-FG02-
46 87ER40317 and DOE grant DE-SC0017955.

47 References

48 Almgren, A. S., Beckner, V. E., Bell, J. B., Day, M. S., Howell, L. H., Joggerst, C. C., Lijewski,
49 M. J., Nonaka, A., Singer, M., & Zingale, M. (2010). CASTRO: A New Compressible
50 Astrophysical Solver. I. Hydrodynamics and Self-gravity. *Astrophysical Journal*, 715,
51 1221–1238. <https://doi.org/10.1088/0004-637X/715/2/1221>

52 Nonaka, A., Almgren, A. S., Bell, J. B., Lijewski, M. J., Malone, C. M., & Zingale, M. (2010).
53 MAESTRO: An Adaptive Low Mach Number Hydrodynamics Algorithm for Stellar Flows.
54 *Astrophysical Journal Supplement*, 188, 358–383. [https://doi.org/10.1088/0067-0049/](https://doi.org/10.1088/0067-0049/188/2/358)
55 188/2/358

56 Zingale, M. (2014). pyro: A teaching code for computational astrophysical hydrodynamics.
57 *Astronomy and Computing*, 6, 52–62. <https://doi.org/10.1016/j.ascom.2014.07.003>