AMReX-Astrophysics Microphysics: A set of microphysics routines for astrophysical simulation codes based on the AMReX library

AMReX-Astro Microphysics Team[†], **Khanak Bhargava**[1], **Abigail Bishop**[2], **Zhi Chen**[1], **Doreen Fan**[3], **Carl Fields**[4], **Adam M. Jacobs**[3], **Eric T. Johnson**[1], **Max P. Katz**[1], **Mark Krumholz**[5], **Chris Malone**[6], **Andy Nonaka**[7], **Piyush Sharda**[8], **Alexander Smith Clark**[1], **Frank Timmes**[9], **Ben Wibking**[10], **Don E. Willcox**[3], **and Michael Zingale**[1]

† https://github.com/amrex-astro/Microphysics **1** Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY, USA **2** Department of Physics, University of Wisconsin, Madison, Madison, WI, USA **3** affiliation not disclosed **4** Department of Astronomy, University of Arizona, Tucson, AZ, USA **5** Research School of Astronomy and Astrophysics, The Australian National University, Australia **6** Los Alamos National Laboratory, Los Alamos, NM, USA **7** Lawrence Berkeley National Laboratory, Berkeley, CA, USA **8** Leiden Observatory, Leiden, The Netherlands **9** Arizona State University, Tempe, AZ, USA **10** Department of Physics and Astronomy, Michigan State University, E. Lansing, MI, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 20 July 2025
Published: unpublished

Summary

The AMReX-Astrophysics Microphysics library provides a common set of microphysics routines (reaction networks and associated physics, equations of state, and various transport coefficients) as well as solvers (stiff ODE integrators, nonlinear system solvers) for astrophysical simulation codes built around the AMReX adaptive mesh refinement library (W. Zhang et al., 2019). Several multi-dimensional simulation codes, including the compressible hydrodynamics code Castro (Almgren et al., 2010), the low-Mach number hydrodynamics code MAESTROeX (Fan et al., 2019), and the radiation-hydrodynamics code Quokka (Wibking & Krumholz, 2022) use Microphysics to provide the physics and solvers needed to close the hydrodynamics systems that they evolve. The library is implemented in C++ with GPU-offloading a key design feature.

Statement of need

Astrophysical simulation codes need many different smallscale (microphysics) physics inputs to close the system of equations. There are many astrophysics simulation codes built around the AMReX library, with each specializing in different astrophysics phenomena. Each of these codes share some common needs. The Microphysics library was created to minimize developer effort across these codes and coordinate the approach to exascale compute architectures, in particular, GPU support for astrophysical simulation codes.

Design

The Microphysics project started in 2013 as a way to centralize the reaction networks and equations of state used by Castro and MAESTRO (Nonaka et al., 2010), the predecessor to MAESTROeX. Originally, Microphysics used Fortran and for a brief period, it was referred to as Starkiller Microphysics, which was an attempt to co-develop microphysics routines for the Castro and the Flash (Fryxell et al., 2000) simulation codes. As interest in GPUs grew (with early support added to Microphysics in 2015), Castro moved from a mix of C++ and Fortran to pure C++ to take advantage of GPU-offloading afforded by the AMReX library, and C++ ports of all physics routines and solvers were added to Microphysics. At this point, the project was formally named the AMReX-Astrophysics Microphysics library. Today, the library is completely written in C++ and relies heavily on the AMReX data structures to take advantage of GPUs. The GPU-enabled reaction network integrators led to the Quokka code adopting Microphysics for their simulations.

Microphysics provides several different types of physics: equations of state, reaction networks and screening methods, nuclear statistical equilibrium solvers and tabulations, thermal conductivities, and opacities, as well as the tools needed to work with them, most notably the suite of stiff ODE integrators for the networks. Several classic Fortran libraries have been converted to header-only C++ implementations, including the VODE integrator (Brown et al., 1989), the hybrid Powell method of MINPACK (Powell, 1970), and the Runge-Kutta Chebyshev (RKC) integration method (Sommeijer et al., 1998). The code was modernized where possible, with many `go to` statements removed and additional logic added to support our applications (see for example the discussion on VODE in Zingale et al. (2022)). We also make use of the C++ autodiff library (Leal, 2018) to compute thermodynamic derivatives required in the Jacobians of our reaction networks.

Microphysics uses header-only implementations of all functionality as much as possible to allow for easier compiler inlining, which is especially important in GPU kernels. We also leverage C++17 `if constexpr` templating to compile out unnecessary computations for performance. Generally, the physics routines and solvers are written to work on a single zone from a simulation code, and in AMReX, a C++ lambda-capturing approach is used to loop over zones (and offload to GPUs if desired). When used with an application code, this design permits the simulation state data to be allocated directly in GPU memory and left there for the entire simulation, with all physics run directly on the GPU. Since each zone in a simulation usually will have a different thermodynamic state, the integration of reaction networks can lead to thread divergence issues. To help mitigate this issue, we can cap the number of integration steps and either retry an integration on a zone-by-zone basis with different tolerances or Jacobian approximations or pass the failure back to the application code to deal with. This strategy has been successful for many large scale simulations (Zingale et al., 2025).

Another key design feature is the separation of the reaction network from the integrator. This allows us to easily experiment with different integration methods (such as the RKC integrator) and also support different modes of coupling reactions to a simulation code, including operator splitting and spectral deferred corrections (SDC) (see, e.g., Zingale et al. (2022)). The latter is especially important for explosive astrophysical flows. Tight integration with pynucastro (Smith et al., 2023; Willcox & Zingale, 2018), allows for the generation of custom reaction networks for a science problem.

There are two ways to use Microphysics: in a standalone fashion (via the unit tests) for simple investigations or as part of an (AMReX-based) application code. In both cases, the core (compile-time) requirement is to select a network—this defines the composition that is then used by most of the other physics routines. This compile-time requirement also allows Microphysics to provide the number of species as a `constexpr` value (which many application codes need), and greatly reduces the compilation time (due to the templating used throughout the library).

Research Impact Statement

Microphysics has been used for simulations of convective Urca (Boyd et al., 2025) and X-ray bursts (Guichandut et al., 2024) with MAESTROeX; and for simulations of nova (Smith Clark & Zingale, 2025), X-ray bursts (Harpole et al., 2021), thermonuclear supernovae (Zingale, Chen, Rasmussen, et al., 2024), and convection in massive stars (Zingale, Chen, Johnson, et al., 2024) with Castro. This Microphysics library has also enabled recent work in astrophysical machine learning to train deep neural networks modeling nuclear reactions (Fan et al., 2022; X. Zhang et al., 2025).

AI Usage Disclosure

No generative AI/LLM was used for producing code or documentation in the git repository or for this paper. We have experimented with using AI/LLM tools for code review and for suggesting places to focus our optimization efforts on, but the resulting coding, benchmarking, and testing is then done by humans.

Acknowledgements

The AMReX-Astro Microphysics library developers are an open scientific team with members contributing to various aspects of the library. We have thus chosen to display the members of our development team in the author list in alphabetical order. All developers who have contributed new features, substantial design input, and/or at least 3 commits were invited to be coauthors. The work at Stony Brook was supported by the US Department of Energy, Office of Nuclear Physics grant DE-FG02-87ER40317.

References

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

Boyd, B., Calder, A., Townsley, D., & Zingale, M. (2025). 3D convective urca process in a simmering white dwarf. *The Astrophysical Journal*, *979*(2), 216. https://doi.org/10.3847/1538-4357/ad9bb0

Brown, P. N., Byrne, G. D., & Hindmarsh, A. C. (1989). VODE: A variable coefficient ODE solver. *SIAM J. Sci. Stat. Comput.*, *10*, 1038–1051.

Fan, D., Nonaka, A., Almgren, A. S., Harpole, A., & Zingale, M. (2019). MAESTROeX: A Massively Parallel Low Mach Number Astrophysical Solver. *Astrophysical Journal*, *887*(2), 212. https://doi.org/10.3847/1538-4357/ab4f75

Fan, D., Willcox, D. E., DeGrendele, C., Zingale, M., & Nonaka, A. (2022). Neural networks for nuclear reactions in MAESTROeX. *The Astrophysical Journal*, *940*(2), 134. https://doi.org/10.3847/1538-4357/ac9a4b

Fryxell, B., Olson, K., Ricker, P., Timmes, F. X., Zingale, M., Lamb, D. Q., MacNeice, P., Rosner, R., Truran, J. W., & Tufo, H. (2000). FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes. *ApJS*, *131*, 273–334. https://doi.org/10.1086/317361

Guichandut, S., Zingale, M., & Cumming, A. (2024). Hydrodynamical simulations of proton ingestion flashes in type i x-ray bursts. *The Astrophysical Journal*, *975*(2), 250. https://doi.org/10.3847/1538-4357/ad81f7

Harpole, A., Ford, N. M., Eiden, K., Zingale, M., Willcox, D. E., Cavecchi, Y., & Katz, M. P. (2021). Dynamics of Laterally Propagating Flames in X-Ray Bursts. II. Realistic Burning and Rotation. *Astrophysical Journal*, *912*(1), 36. https://doi.org/10.3847/1538-4357/abee87

Leal, A. M. M. (2018). *Autodiff, a modern, fast and expressive C++ library for automatic differentiation*. `https://autodiff.github.io`. https://autodiff.github.io

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

Powell, M. J. D. (1970). A hybrid method for nonlinear equations. In P. Rabinowitz (Ed.), *Numerical methods for nonlinear algebraic equations* (pp. 87–114). Gordon; Breach Science Publishers, New York.

Smith, A. I., Johnson, E. T., Chen, Z., Eiden, K., Willcox, D. E., Boyd, B., Cao, L., DeGrendele, C. J., & Zingale, M. (2023). Pynucastro: A python library for nuclear astrophysics. *The Astrophysical Journal*, *947*(2), 65. https://doi.org/10.3847/1538-4357/acbaff

Smith Clark, A., & Zingale, M. (2025). Multidimensional Nova Simulations with an Extended Buffer and Lower Initial Mixing Temperatures. *The Open Journal of Astrophysics*, *8*. https://doi.org/10.33232/001c.136890

Sommeijer, B. P., Shampine, L. F., & Verwer, J. G. (1998). RKC: An explicit solver for parabolic PDEs. *Journal of Computational and Applied Mathematics*, *88*(2), 315–326. https://doi.org/10.1016/S0377-0427(97)00219-7

Wibking, B. D., & Krumholz, M. R. (2022). QUOKKA: a code for two-moment AMR radiation hydrodynamics on GPUs. *Monthly Notices of the Royal Astronomical Society*, *512*(1), 1430–1449. https://doi.org/10.1093/mnras/stac439

Willcox, D. E., & Zingale, M. (2018). pynucastro: an interface to nuclear reaction rates and code generator for reaction net work equations. *Journal of Open Source Software*, *3*(23), 588. https://doi.org/10.21105/joss.00588

Zhang, W., Almgren, A., Beckner, V., Bell, J., Blaschke, J., Chan, C., Day, M., Friesen, B., Gott, K., Graves, D., Katz, M. P., Myers, A., Nguyen, T., Nonaka, A., Rosso, M., Williams, S., & Zingale, M. (2019). AMReX: a framework for block-structured adaptive mesh refinement. *Journal of Open Source Software*, *4*(37), 1370. https://doi.org/10.21105/joss.01370

Zhang, X., Yi, Y., Wang, L., Xu, Z.-Q. J., Zhang, T., & Zhou, Y. (2025). Deep neural networks for modeling astrophysical nuclear reacting flows. *The Astrophysical Journal*, *990*(2), 105. https://doi.org/10.3847/1538-4357/adf331

Zingale, M., Bhargava, K., Brady, R., Chen, Z., Guichandut, S., Johnson, E. T., Katz, M., & Smith Clark, A. (2025). The challenges of modeling astrophysical reacting flows. *Journal of Physics: Conference Series*, *2997*(1), 012007. https://doi.org/10.1088/1742-6596/2997/1/012007

Zingale, M., Chen, Z., Johnson, E. T., Katz, M. P., & Smith Clark, A. (2024). Strong coupling of hydrodynamics and reactions in nuclear statistical equilibrium for modeling convection in massive stars. *The Astrophysical Journal*, *977*(1), 30. https://doi.org/10.3847/1538-4357/ad8a66

Zingale, M., Chen, Z., Rasmussen, M., Polin, A., Katz, M., Smith Clark, A., & Johnson, E. T. (2024). Sensitivity of simulations of double-detonation type ia supernovae to integration methodology. *Astrophysical Journal*, *966*(2), 150. https://doi.org/10.3847/1538-4357/ad3441

Zingale, M., Katz, M. P., Nonaka, A., & Rasmussen, M. (2022). An improved method for coupling hydrodynamics with astrophysical reaction networks. *The Astrophysical Journal*, *936*(1), 6. https://doi.org/10.3847/1538-4357/ac8478