The Astrophysical Journal, 887:212 (13pp), 2019 December 20 https://doi.org/10.3847/1538-4357/ab4f75

MAESTROeX: A Massively Parallel Low Mach Number Astrophysical Solver

Duoming Fan[1], Andrew Nonaka[1], Ann S. Almgren[1], Alice Harpole[2], and Michael Zingale[2]
[1] Lawrence Berkeley National Laboratory Center for Computational Sciences and Engineering One Cyclotron Road, MS 50A-3111 Berkeley, CA 94720, USA
DFan@lbl.gov
[2] Stony Brook University Department of Physics and Astronomy Stony Brook, NY 11794-3800, USA
Received 2019 August 5; revised 2019 October 2; accepted 2019 October 15; published 2019 December 19

Abstract

We present MAESTROeX, a massively parallel solver for low Mach number astrophysical flows. The underlying low Mach number equation set allows for efficient, long-time integration for highly subsonic flows compared to compressible approaches. MAESTROeX is suitable for modeling full spherical stars as well as well as planar simulations of dynamics within localized regions of a star, and can robustly handle several orders of magnitude of density and pressure stratification. Previously, we have described the development of the predecessor of MAESTROeX, called MAESTRO, in a series of papers. Here, we present a new, greatly simplified temporal integration scheme that retains the same order of accuracy as our previous approaches. We also explore the use of alternative spatial mapping of the one-dimensional base state onto the full Cartesian grid. The code leverages the new AMReX software framework for block-structured adaptive mesh refinement (AMR) applications, allowing for scalability to large fractions of leadership-class machines. Using our previous studies on the convective phase of single-degenerate progenitor models of SNe Ia as a guide, we characterize the performance of the code and validate the new algorithmic features. Like MAESTRO, MAESTROeX is fully open source.

Unified Astronomy Thesaurus concepts: Stellar convective zones (301); Hydrodynamics (1963); Computational methods (1965); Nuclear astrophysics (1129); Nucleosynthesis (1131); Nuclear abundances (1128)

Software reviewed by the Journal of Open Source Software

1. Introduction

Many astrophysical flows are highly subsonic. In this regime, sound waves carry sufficiently little energy that they do not significantly affect the convective dynamics of the system. In many of these flows, modeling long-time convective dynamics are of interest, and numerical approaches based on explicit compressible hydrodynamics are intractable, even on modern supercomputers. Our approach to this problem is to use a low Mach number model where sound waves are eliminated from the governing equations while retaining compressibility effects due to, e.g., nuclear energy release, stratification, compositional changes, and thermal diffusion. When the Mach number (the ratio of the characteristic fluid velocity over the characteristic sound speed; $\mathrm{Ma} = U/c$) is small, the resulting system can be numerically integrated with much larger time steps than a compressible model. Specifically, the time step increase is at least a factor of $\sim 1/\mathrm{Ma}$ larger. Each time step is more computationally expensive due to the presence of additional linear solves, but for many problems of interest the overall gains in efficiency can easily be an order of magnitude or more.

Low Mach number models that do not contain acoustic waves have been developed for a variety of contexts including combustion (Day & Bell 2000), terrestrial atmospheric modeling (Durran 1989; O'Neill & Klein 2014; Duarte et al. 2015), and elastic solids (Abbate et al. 2017). For astrophysical applications, a number of approaches to modeling low Mach number flows have been developed in recent years. One of the more similar approaches to ours is Lin et al. (2006); however, this approach is only first-order accurate and does not account for atmospheric expansion. There are several other approaches that retain the effects of acoustic wave propagation with various strategies to efficiently and robustly handle the low Mach number limit. In the reduced speed of sound technique and related methods, the speed of sound is artificially reduced by including a suitable scaling factor in the continuity equation, reducing the restriction on the size of the time step (Rempel 2005; Hotta et al. 2012; Wang et al. 2015; Takeyama et al. 2017; Iijima et al. 2019). There are semi-implicit all-Mach number solvers, where the Euler equations are split into an acoustic part and an advective part (Kwatra et al. 2009; Degond & Tang 2011; Cordier et al. 2012; Haack et al. 2012; Happenhofer et al. 2013; Chalons et al. 2016; Padioleau et al. 2019). The fast acoustic waves are then solved using implicit time integration, while the slow material waves are solved explicitly. There are also fully implicit time integration codes for the compressible Euler equations (Kifonidis & Müller 2012; Viallet et al. 2015; Goffrey et al. 2017). The MUSIC code uses fully implicit time integration for the compressible Euler equations, which therefore allows for arbitrarily large time steps. To resolve potential pressure scaling issues in very low Mach number flow in explicit or implicit schemes, another approach is to use preconditioned all-Mach number solvers (Miczek et al. 2015; Barsukow et al. 2016) to properly capture pressure fluctuations in the low Mach number regime (see Guillard & Nkonga 2017 for discussion). The numerical flux is multiplied by a preconditioning matrix that reduces the stiffness of the system at low Mach numbers, while retaining the correct scaling behavior in an acoustic limit of the Euler equations that retains $\mathcal{O}(\mathrm{Ma})$ pressure fluctuations.

Previously, we developed the low Mach number astrophysical solver, MAESTRO. MAESTRO is a block-structured, Cartesian grid finite-volume, adaptive mesh refinement (AMR) code that has been successfully used for many years for a number of applications, detailed below. Unlike several of the references above, MAESTRO is not an all-Mach solver, but is suitable for flows where the Mach number is small (~ 0.1 or smaller). Furthermore, the low Mach number model in MAESTRO is specifically designed for, but not limited to, astrophysical settings with significant atmospheric stratification. This includes full spherical stars, as well as planar simulations of dynamics within localized regions of a star. The numerical methodology relies on an explicit Godunov approach for advection, a stiff ordinary differential equation solver for reactions (VODE; Brown et al. 1989), and multigrid-based linear solvers for the pressure-projection steps. Thus, the time step is limited by an advective CFL constraint based on the fluid velocity, not the sound speed. Central to the algorithm are time-varying, one-dimensional stratified background (or base) state density and pressure fields that are held in hydrostatic equilibrium. The base state density couples to the full state solution through buoyancy terms in the momentum equation, and the base state pressure couples to the full state solution by constraining the evolution of the thermodynamic variables to match this pressure. The time-advancement strategy uses Strang splitting to integrate the thermodynamic variables, a second-order projection method to integrate the velocity subject to a divergence constraint, and a velocity-splitting scheme that uses a radially averaged velocity to hydrodynamically evolve the base state. The original MAESTRO code was developed in the pure-Fortran 90 FBoxLib software framework, whereas MAESTROeX is developed in the C++/F90 AMReX framework (AMReX Development Team et al. 2019; Zhang et al. 2019).

The key numerical developments of the original MAESTRO algorithm are presented in a series of papers which we refer to as Papers I–V:

1. In Paper I (Almgren et al. 2006a), we derive the low Mach number equation set for stratified environments from the fully compressible equations.
2. In Paper II (Almgren et al. 2006b), we incorporate the effects of atmospheric expansion through the use of a time-dependent background state.
3. In Paper III (Almgren et al. 2008), we incorporate reactions and the associated coupling to the hydrodynamics.
4. In Paper IV (Zingale et al. 2009), we describe our treatment of spherical stars in a three-dimensional Cartesian geometry.
5. In Paper V (Nonaka et al. 2010), we describe the use of block-structured AMR to focus spatial resolution in regions of interest.

Since then, there have been many scientific investigations using MAESTRO, which have included additional algorithmic enhancements. Topics include:

1. The convective phase preceding Chandrasekhar mass models for SNe Ia (Nonaka et al. 2011; Zingale et al. 2011; Malone et al. 2014a).
2. Convection in massive stars (Gilet et al. 2013; Gilkis & Soker 2016).
3. Sub-Chandrasekhar white dwarfs (Zingale et al. 2013; Jacobs et al. 2016).
4. Type I X-ray bursts (Malone et al. 2011, 2014b; Zingale et al. 2015).

In this paper, we present new algorithmic methodology that improves upon Paper V in a number of ways. First, the overall temporal algorithm has been greatly simplified without compromising second-order accuracy. The key design decisions were to eliminate the splitting of the velocity into average and perturbational components, and also to replace the hydrodynamic evolution of the base state with a predictor-corrector approach. Not only does this greatly simplify the dynamics of the base state, but this treatment is more amenable to higher-order multiphysics coupling strategies based on method-of-lines integration. In particular, schemes based on deferred corrections (Dutt et al. 2000) have been used to generate high-order temporal integrators for problems of reactive flow and low Mach number combustion (Pazner et al. 2016; Nonaka et al. 2018). Second, we explore the effects of alternative spatial mapping routines for coupling the base state and the Cartesian grid state for spherical problems. Finally, we examine the performance of our new MAESTROeX implementation in the new C++/F90 AMReX public software library (AMReX Development Team et al. 2019; Zhang et al. 2019). MAESTROeX uses MPI+OpenMP parallelism and scales well to over 10,000 MPI processes, with each MPI process supporting tens of threads. The resulting code is publicly available on GitHub (https://github.com/AMReX-Astro/MAESTROeX), uses the Starkiller-Astro microphysics libraries (The StarKiller Microphysics Development Team et al. 2019, https://github.com/starkiller-astro), as well as AMReX (https://github.com/AMReX-Codes/amrex).

The rest of this paper is organized as follows. In Section 2 we review our model for stratified low Mach number astrophysical flow. In Section 3 we present our numerical algorithm in detail, highlighting the new temporal integration scheme as well as spatial base state mapping options. In Section 4 we validate our new approach and examine the performance of our algorithm on full spherical star problems used in previous scientific investigations. We conclude in Section 5.

2. Governing Equations

Low Mach number models for reacting flow were originally derived using asymptotic analysis (Rehm & Baum 1978; Majda & Sethian 1985) and used in terrestrial combustion applications (Knio et al. 1999; Day & Bell 2000). These models have been extended to nuclear flames in astrophysical environments using adaptive algorithms in space and time (Bell et al. 2004). In Papers I–III, we extended this work and the atmospheric model by Durran (1989) by deriving a model and algorithm suitable for stratified astrophysical flow. We take the standard equations of reacting, compressible flow, and recast the equation of state (EOS) as a divergence constraint on the velocity field. The resulting model is a series of evolution equations for mass, momentum, and energy, subject to an additional constraint on velocity. The evolution equations are

$$\frac{\partial(\rho X_k)}{\partial t} = -\nabla\cdot(\rho X_k \boldsymbol{U}) + \rho\dot{\omega}_k, \tag{1}$$

$$\frac{\partial \boldsymbol{U}}{\partial t} = -\boldsymbol{U}\cdot\nabla\boldsymbol{U} - \frac{\beta_0}{\rho}\nabla\left(\frac{\pi}{\beta_0}\right) - \frac{\rho-\rho_0}{\rho}g\boldsymbol{e}_r, \tag{2}$$

$$\frac{\partial(\rho h)}{\partial t} = -\nabla\cdot(\rho h\boldsymbol{U}) + \frac{Dp_0}{Dt} + \rho H_{\rm nuc}. \tag{3}$$

Here ρ, $\boldsymbol{U}$, and h are the mass density, velocity, and specific enthalpy, respectively, and X_k are the mass fractions of species k with associated production rate $\dot{\omega}_k$ and energy release per time per unit mass $H_{\rm nuc}$. The species are constrained such that

$\sum_k X_k = 1$ giving $\rho = \sum_k (\rho X_k)$ and

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \boldsymbol{U}). \tag{4}$$

The total pressure is decomposed into a one-dimensional hydrostatic base state pressure, $p_0 = p_0(r, t)$, and a dynamic pressure, $\pi = \pi(\boldsymbol{x}, t)$, such that $p = p_0 + \pi$ and $|\pi|/p_0 = \mathcal{O}(\mathrm{Ma}^2)$ (we use $\boldsymbol{x}$ to represent the Cartesian coordinate directions of the full state and r to represent the radial coordinate direction for the base state). One way to mathematically think of the difference between p_0 and π is that π controls the velocity evolution in a way that forces the thermodynamic variables (ρ, h, X_k) to evolve in a manner that is consistent with the EOS and p_0.

By comparing the momentum Equation (2) to the momentum equation used in Equation (2) in Paper V, we note that we are using a formulation that enforces conservation of total energy in the low Mach number system in the absence of external heating or viscous terms (Klein & Pauluis 2012; Vasil et al. 2013). We have previously validated this approach in modeling sub-Chandrasekhar white dwarfs using MAESTRO (Jacobs et al. 2016). We also define a one-dimensional base state density, $\rho_0 = \rho_0(r, t)$, that represents the lateral average (see Section 3.1) of ρ and is in hydrostatic equilibrium with p_0, i.e.,

$$\nabla p_0 = -\rho_0 g \boldsymbol{e}_r, \tag{5}$$

where $g = g(r, t)$ is the magnitude of the gravitational acceleration and $\boldsymbol{e}_r$ is the unit vector in the outward radial direction. Here β_0 is a density-like variable that carries background stratification, defined as

$$\beta_0(r, t) = \rho_0(0, t) \exp\left(\int_0^r \frac{1}{\overline{\Gamma}_1 p_0} \frac{\partial p_0}{\partial r'} dr'\right), \tag{6}$$

where $\overline{\Gamma}_1$ is the lateral average of $\Gamma_1 = d(\log p)/d(\log \rho)|_s$ (evaluated with entropy, s, constant). We explored the effect of replacing Γ_1 with $\overline{\Gamma}_1$ as well as a correction term in Paper III. Thermal diffusion is not discussed in this paper, but we have previously described the modifications to the original algorithm required for implicit thermal diffusion in Malone et al. (2011); inclusion of these effects in the new algorithm presented here is straightforward.

Mathematically, Equations (2) and (3) must still be closed by the EOS. This is done by taking the Lagrangian derivative of the EOS for pressure as a function of the thermodynamic variables, substituting in the equations of motion for mass and energy, and requiring that the pressure is a prescribed function of altitude and time based on the hydrostatic equilibrium condition. See Papers I and II for details of this derivation. The resulting equation is a divergence constraint on the velocity field,

$$\nabla \cdot (\beta_0 \boldsymbol{U}) = \beta_0 \left(S - \frac{1}{\overline{\Gamma}_1 p_0} \frac{\partial p_0}{\partial t}\right). \tag{7}$$

The expansion term, S, incorporates local compressibility effects due to compositional changes and heat release from reactions,

$$S = -\sigma \sum_k \xi_k \dot{\omega}_k + \frac{1}{\rho p_\rho} \sum_k p_{X_k} \dot{\omega}_k + \sigma H_{\mathrm{nuc}}, \tag{8}$$

where $p_{X_k} \equiv \partial p / \partial X_k|_{\rho, T, X_{j, j \neq k}}$, $\xi_k \equiv \partial h / \partial X_k|_{p, T, X_{j, j \neq k}}$, $p_\rho \equiv \partial p / \partial \rho|_{T, X_k}$, and $\sigma \equiv p_T / (\rho c_p p_\rho)$, with $p_T \equiv \partial p / \partial T|_{\rho, X_k}$ and $c_p \equiv \partial h / \partial T|_{p, X_k}$ is the specific heat at constant pressure.

To summarize, we model evolution equations for momentum, mass, and energy, Equations (2) and (3), subject to a divergence constraint on the velocity, Equation (7), and the hydrostatic equilibrium condition, Equation (5).

3. Numerical Algorithm

3.1. Spatial Discretization

The spatial discretization and AMR methodology remains unchanged from Paper V. We now summarize some of the key points here before describing the new temporal integrator in the next section. We recommend the reader review Section 3 of Paper V for further details.

We shall refer to local atmospheric flows in two and three dimensions as problems in "planar" geometry, and full-star flows in three dimensions as problems in "spherical" geometry. The solution in both cases consists of the Cartesian grid solution and the one-dimensional base state solution. Figure 1 illustrates the relationship between the base state and the Cartesian grid state for both planar and spherical geometries in the presence of spatially adaptive grids. One of the key numerical modules is the "lateral average," which computes the average over a layer of constant radius of a Cartesian grid variable and stores the result in a one-dimensional base state array. In planar geometries, this is a straightforward arithmetic average of values in cells at a particular height since the base state cell centers are in alignment with the Cartesian grid cell centers. However for spherical problems, the procedure is much more complicated. In Section 4 of Paper V, we describe how there is a finite, easily computable set of radii that any three-dimensional Cartesian cell center can map to. Specifically, for every three-dimensional Cartesian cell, there exists an integer m such that the distance from the cell center to the center of the star is given by

$$\hat{r}_m = \Delta x \sqrt{0.75 + 2m}. \tag{9}$$

Figure 2 is a two-dimensional illustration (two dimensions is chosen in the figure for ease of exposition; this mapping is only used for three-dimensional spherical stars) of the relationship between the Cartesian grid state and the one-dimensional base state array. We compute the lateral average by first summing the values in all the cells associated with each radius, dividing by the number of contributing cells to obtain the arithmetic average, and then using quadratic interpolation to map this data onto a one-dimensional base state. Previously, for spherical problems MAESTRO only allowed for a base state with constant Δr (typically equal to $\Delta x/5$).

The companion "fill" module maps a base state array onto the full Cartesian grid state. For planar problems, direct injection can be used due to the perfect alignment of the base state and Cartesian grid state. For spherical problems, quadratic interpolation of the base state is used to assign values to each Cartesian cell center.

In this paper we explore a new option to retain an irregularly spaced base state to eliminate mapping errors from the "fill"

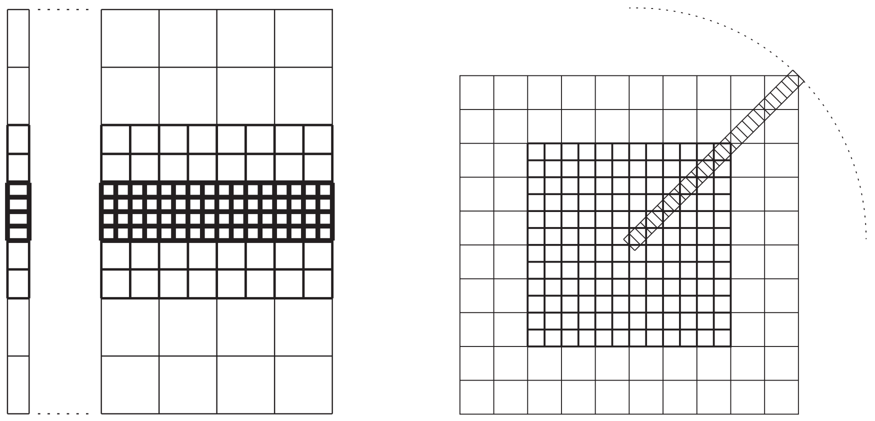


Figure 1. (Left) For multilevel problems in planar geometry, we force a direct alignment between the base state cell centers and the Cartesian grid cell centers by allowing the radial base state spacing to change with space and time. (Right) For multilevel problems in spherical geometry, since there is no direct alignment between the base state cell centers and the Cartesian grid cell centers, we choose to fix the radial base state spacing across levels. Reprinted from Paper V (Nonaka et al. 2010).

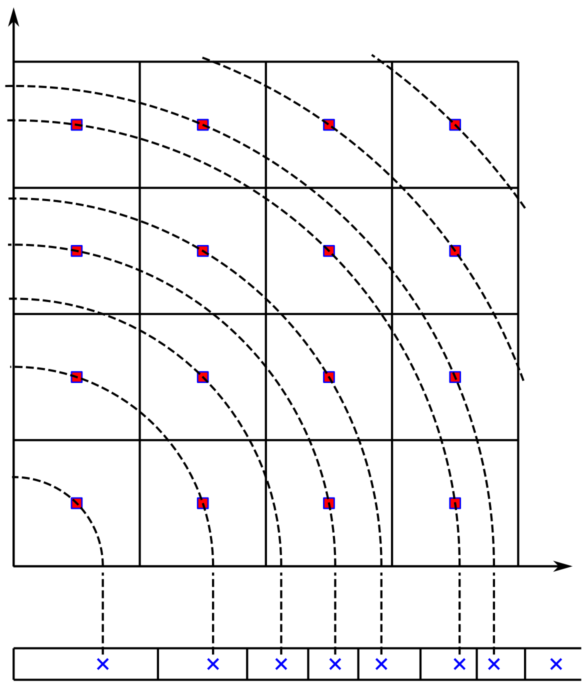


Figure 2. Direct mapping between the base state cell centers (red squares) and the Cartesian grid cell centers (blue crosses) is enforced by computing the average of the grid cell centers that share the same radial distance from the center of the star.

module. For the lateral average, as before we first sum the values in all the cells associated with each radius and divide by the number of contributing cells to obtain the arithmetic average. However we do not interpolate this onto a uniformly spaced base state and retain the use of this irregularly spaced base state. The advantage with this approach is that the fill module does not require any interpolation. A potential benefit to eliminating the mapping error is to consider a spherical star in hydrostatic equilibrium at rest. In the absence of reactions, the star should remain at rest. The buoyancy forcing term in the momentum equation contains ρ–ρ_0. With the original scheme, interpolation errors in computing ρ_0 by averaging would cause artificial acceleration in the velocity field due to the interpolation error from the Cartesian grid to and from the radial base state. By retaining the radial base state as an irregularly spaced array, the truncation error in the mapping scheme is completely eliminated. The only source of error is due to machine precision effects resulting from averaging a large number of values, which is not significant over the course of any simulation. We note that Δr decreases as the base state moves further from the center of the star, which results in far more total cells in the irregularly spaced array than the previous uniformly spaced array.

3.2. Temporal Integration Scheme

We now describe the new temporal integration scheme, noting that it can be used for the original base state mapping (with uniform base state grid spacing) as well as the new irregularly spaced base state mapping. Previously we adopted an approach where we split the velocity into a base state component, $w_0(r, t)$, and a local velocity $\widetilde{\boldsymbol{U}}(\boldsymbol{x}, t)$, so that

$$\boldsymbol{U} = \widetilde{\boldsymbol{U}}(\boldsymbol{x}, t) + w_0(r, t)\boldsymbol{e}_r, \tag{10}$$

where $\boldsymbol{e}_r$ is the normal vector in the outward radial direction. We used w_0 to provide an estimate of the base state density evolution over a time step. This resulted in some unnecessary complications to the temporal integration scheme including base state advection modules for density, enthalpy, and velocity, as well as more cumbersome split velocity dynamics evolution equations. Our new temporal integration scheme uses full velocities for scalar and velocity advection, and only uses the above splitting to satisfy the velocity divergence constraint due to boundary considerations at the edge of the star. This results in a much simpler numerical scheme than the one from Paper V since we use the velocity directly rather than more complex terms involving the perturbational velocity. Additionally, the new scheme uses a simpler predictor-corrector approach to the base state density and pressure that no longer requires evolution equations and numerical discretizations to update the base state, greatly simplifying the algorithm while retaining the same overall second-order accuracy in time.

At the beginning of each time step we have the cell-centered Cartesian grid state, $(\boldsymbol{U}, \rho X_k, \rho h)^n$, and nodal Cartesian grid state, $\pi^{n-1/2}$, and base state $(\rho_0, p_0)^n$. At any time, the associated density, composition, and enthalpy can be trivially computed using, e.g.,

$$\rho^n = \sum_k (\rho X_k)^n, \quad X_k^n = (\rho X_k)^n / \rho^n, \quad h^n = (\rho h)^n / \rho^n. \quad (11)$$

Temperature is computed using the EOS,[3] e.g., $T = T(\rho, p_0, X_k)$, where p_0 has been mapped to the Cartesian grid using the fill module, and $(\overline{\Gamma}_1, \beta_0)$ are similarly computed from (ρ, p_0, X_k); see Appendix A of Paper I and Appendix C of Paper III for details on how β_0 is computed.

The overall flow of the algorithm begins with a second-order Strang splitting approach to integrate the advection-reaction system for the thermodynamic variables $(\rho X_k, \rho h)$, followed by a second-order projection methodology to integrate the velocities subject to a divergence constraint. Within the thermodynamic variable update we use a predictor-corrector approach to achieve second-order accuracy in time. To summarize:

1. In Step 1 we react the thermodynamic variables over the first $\Delta t/2$ interval.
2. In Steps 2–4 we advect the thermodynamic variables over Δt. Specifically, we compute an estimate for the expansion term, S, compute face-centered, time-centered velocities that satisfy the divergence constraint, and then advect the thermodynamic variables.
3. In Step 5 we react the thermodynamic variables over the second $\Delta t/2$ interval.[4]
4. In Steps 6–8 we redo the advection in Steps 2–4 but are able to use the trapezoidal rule to time-center certain quantities such as S, ρ_0, etc.
5. In Step 9 we redo the reactions from Step 5 beginning with the improved results from Steps 6–8.
6. In Steps 10–11 we advect the velocity, and then project these velocities so they satisfy the divergence constraint while updating π.

There are a few key numerical modules we use in each time step.

1. Average $[\phi] \rightarrow [\overline{\phi}]$ computes the lateral average of a quantity over a layer at constant radius r, as described above in Section 3.1.
2. Enforce HSE $[\rho_0] \rightarrow [p_0]$ computes the base state pressure, p_0, from a base state density, ρ_0 by integrating the hydrostatic equilibrium condition in one dimension. This follows equation (A10) in Paper V, noting that for the irregularly spaced base state case, Δr is not constant, where $\Delta r_{j+1/2} = r_{j+1} - r_j$ for cell face with index $j+1/2$. The base state pressure remains equal to a constant value at the location of a prescribed cutoff density outward for the entire simulation.
3. React State $[(\rho X_k)^{\text{in}}, (\rho h)^{\text{in}}, p_0] \rightarrow [(\rho X_k)^{\text{out}}, (\rho h)^{\text{out}}, (\rho \dot{\omega}), (\rho H_{\text{nuc}})]$ uses the multistep variable-coefficient Adams–Moulton and Backward Differentiation Formula methods in the VODE (Brown et al. 1989) package to integrate the species and enthalpy due to reactions over $\Delta t/2$ by solving

$$\frac{dX_k}{dt} = \dot{\omega}_k(\rho, X_k, T);$$
$$\frac{dT}{dt} = \frac{1}{c_p}\left(-\sum_k \xi_k \dot{\omega}_k + H_{\text{nuc}}\right). \quad (12)$$

The inputs are the species, enthalpy, and base state pressure, and the outputs are the species, enthalpy, reaction rates, and nuclear energy generation rate. See Paper III for details.

Each time step is constrained by the standard advective CFL condition,

$$\Delta t = \sigma^{\text{CFL}} \min_i (\Delta x / U_i), \quad (13)$$

where for our simulations we typically use $\sigma^{\text{CFL}} \sim 0.7$ and the minimum is taken over all spatial directions over all cells. For simulations that are initialized with zero velocity, we use the buoyancy force in the momentum equation to give a proper timescale to start the simulation; see Section 3.4 in Paper III for details.

In stratified low Mach number models, due to the extreme variation in density, the velocity can become very large in low density regions at the edge of the star. These large velocities can severely affect the time step, so throughout Papers II–V, we have employed two techniques to help control these dynamics without significantly affecting the dynamics in the convective region. First, we use a cutoff density technique, where we hold the density constant outside a specified radius (typically near where the density is ∼4 orders of magnitude smaller than the largest densities in the simulation). Second, we employ a sponge technique where we artificially damp the velocities near and beyond the cutoff region. For more details, refer to Paper V and the previous references cited within.

Beginning with $(\boldsymbol{U}, \rho X_k, \rho h)^n$, $\pi^{n-1/2}$, and $(\rho_0, p_0)^n$, the temporal integration scheme contains the following steps:

Step 1: react the thermodynamic variables over the first Δt/2 interval.

Call React State $[(\rho X_k)^n, (\rho h)^n, p_0^n] \rightarrow [(\rho X_k)^{(1)}, (\rho h)^{(1)}, (\rho \dot{\omega}_k)^{(1)}, (\rho H_{\text{nuc}})^{(1)}]$.

Step 2: compute the time-centered expansion term, $S^{n+1/2,\star}$.

We compute an estimate for the time-centered expansion term in the velocity divergence constraint. Following Bell et al. (2004), we extrapolate to the half-time using S at the previous and current time steps,

$$S^{n+1/2,\star} = S^n + \frac{\Delta t^n}{2}\left(\frac{S^n - S^{n-1}}{\Delta t^{n-1}}\right). \quad (14)$$

Note that in the first time step we average S^0 and S^1 from the initialization step.

Step 3: construct a face-centered, time-centered advective velocity, $\boldsymbol{U}^{\text{ADV},\star}$.

The construction of face-centered time-centered states used to discretize the advection terms for velocity, species, and enthalpy, are performed using a standard multidimensional corner transport upwind approach (Colella 1990; Saltzman 1994) with the piecewise-parabolic method one-dimensional tracing (Colella & Woodward 1984). The full details of this Godunov advection approach for all steps in this algorithm are described in Appendix A of Zingale et al. (2015). Here we

[3] As described in Paper V, for planar problems we compute temperature using h instead of p_0, since we have successfully developed volume discrepancy schemes to effectively couple the enthalpy to the rest of the solution; see Malone et al. (2011). We are still exploring this option for spherical stars.

[4] After this step we could skip to the velocity advance in Steps 10–11, however, the overall scheme would be only first-order in time, so Steps 6–9 can be thought of as a trapezoidal corrector step.

use Equation (2) to compute face-centered, time-centered velocities, $\boldsymbol{U}^{\mathrm{ADV},\dagger,\star}$. The † superscript refers to the fact that the predicted velocity field does not satisfy the divergence constraint,

$$\nabla \cdot (\beta_0^n \boldsymbol{U}^{\mathrm{ADV},\star}) = \beta_0^n \left[S^{n+1/2**,\star} - \frac{1}{\overline{\Gamma}_1^n p_0^n} \left(\frac{\partial p_0}{\partial t} \right)^{n-1/2} \right]. \tag{15}$$

We project $\boldsymbol{U}^{\mathrm{ADV},\dagger,\star}$ onto the space of velocities that satisfy the constraint to obtain $\boldsymbol{U}^{\mathrm{ADV},\star}$. Each projection step in the algorithm involves the solution of a variable-coefficient Poisson solve using multigrid. Note that we still employ velocity-splitting as described by Equation (10) for this step in order to enforce the appropriate behavior of the system near the edge of the star as determined by the cutoff density. The details of this "MAC" projection are provided in Appendix.

Step 4: advect the thermodynamic variables over a time interval of Δt.

A. Update (ρX_k) using a discretized version of

$$\frac{\partial(\rho X_k)}{\partial t} = -\nabla \cdot (\rho X_k \boldsymbol{U}), \tag{16}$$

where the reaction terms have been omitted since they were already accounted for in React State. The update consists of two steps:

i. Compute the face-centered, time-centered species, $(\rho X_k)^{n+1/2,\mathrm{pred},\star}$, for the conservative update of $(\rho X_k)^{(1)}$ using a Godunov approach (Zingale et al. 2015). As described in Paper V, for robust numerical slope limiting we predict $\rho'^n = \rho^n - \rho_0^n$ and X_k^n to faces and here we spatially interpolate ρ_0^n to faces to assemble the fluxes.

ii. Evolve $(\rho X_k)^{(1)} \rightarrow (\rho X_k)^{(2),\star}$ using

$$(\rho X_k)^{(2),\star} = (\rho X_k)^{(1)} - \Delta t \{\nabla \cdot [\boldsymbol{U}^{\mathrm{ADV},\star} (\rho X_k)^{n+1/2,\mathrm{pred},\star}]\}. \tag{17}$$

B. Update ρ_0 by calling Average $[\rho^{(2),\star}] \rightarrow [\rho_0^{n+1,\star}]$.

C. Update p_0 by calling Enforce HSE $[\rho_0^{n+1,\star}] \rightarrow [p_0^{n+1,\star}]$.

D. Update the enthalpy using a discretized version of equation

$$\frac{\partial(\rho h)}{\partial t} = -\nabla \cdot (\rho h \boldsymbol{U}) + \frac{Dp_0}{Dt} + \rho H_{\mathrm{nuc}}, \tag{18}$$

again omitting the reaction terms since we already accounted for them in React State. This equation takes the form:

$$\frac{\partial(\rho h)}{\partial t} = -\nabla \cdot (\rho h \boldsymbol{U}) + \frac{\partial p_0}{\partial t} + (\boldsymbol{U} \cdot \boldsymbol{e}_r) \frac{\partial p_0}{\partial r}. \tag{19}$$

For spherical geometry, we solve the analytically equivalent form,

$$\frac{\partial(\rho h)}{\partial t} = -\nabla \cdot (\rho h \boldsymbol{U}) + \frac{\partial p_0}{\partial t} + \nabla \cdot (\boldsymbol{U} p_0) - p_0 \nabla \cdot \boldsymbol{U}. \tag{20}$$

The update consists of two steps:

i. Compute the face-centered, time-centered enthalpy, $(\rho h)^{n+1/2,\mathrm{pred},\star}$, for the conservative update of $(\rho h)^{(1)}$ using a Godunov approach (Zingale et al. 2015). As described in Paper V, for robust numerical slope limiting we predict $(\rho h)'^n = (\rho h)^n - (\rho h)_0^n$ to faces, where $(\rho h)_0^n$ is obtained by calling Average $[(\rho h)^n] \rightarrow [(\rho h)_0^n]$, and here we spatially interpolate $(\rho h)_0^n$ to faces to assemble the fluxes.

ii. Evolve $(\rho h)^{(1)} \rightarrow (\rho h)^{(2),\star}$ using

$$(\rho h)^{(2),\star} = (\rho h)^{(1)} - \Delta t \{\nabla \cdot [\boldsymbol{U}^{\mathrm{ADV},\star} (\rho h)^{n+1/2,\mathrm{pred},\star}]\} + \Delta t \frac{Dp_0}{Dt} \tag{21}$$

where here

$$\frac{Dp_0}{Dt} = \begin{cases} \dfrac{p_0^{n+1,*} - p_0^n}{\Delta t} + (\boldsymbol{U}^{\mathrm{ADV},\star} \cdot \boldsymbol{e}_r) \left(\dfrac{\partial p_0}{\partial r}\right)^n & \text{(planar)} \\ \dfrac{p_0^{n+1,*} - p_0^n}{\Delta t} + [\nabla \cdot (\boldsymbol{U}^{\mathrm{ADV},\star} p_0^n) - p_0^n \nabla \cdot \boldsymbol{U}^{\mathrm{ADV},\star}] & \text{(spherical)} \end{cases}, \tag{22}$$

and $p_0^{n+1/2} = (p_0^n + p_0^{n+1,*})/2$.

Step 5: react the thermodynamic variables over the second $\Delta t/2$ interval.

Call React State $[(\rho X_k)^{(2),\star}, (\rho h)^{(2),\star}, p_0^{n+1,\star}] \rightarrow [(\rho X_k)^{n+1,\star}, (\rho h)^{n+1,\star}, (\rho \dot{\omega}_k)^{n+1,\star}, (\rho H_{\mathrm{nuc}})^{n+1,\star}]$.

Step 6: compute the time-centered expansion term, $S^{n+1/2,\star}$.

First, compute $S^{n+1,\star}$ with

$$S^{n+1,\star} = \left(-\sigma \sum_k \xi_k \dot{\omega}_k + \frac{1}{\rho p_\rho} \sum_k p_{X_k} \dot{\omega}_k + \sigma H_{\mathrm{nuc}} \right)^{n+1,\star}. \tag{23}$$

Then, define

$$S^{n+1/2} = \frac{S^n + S^{n+1,\star}}{2}, \tag{24}$$

Step 7: construct a face-centered, time-centered advective velocity, $\boldsymbol{U}^{\mathrm{ADV}}$.

The procedure to construct $\boldsymbol{U}^{\mathrm{ADV},\dagger}$ is identical to the Godunov procedure for computing $\boldsymbol{U}^{\mathrm{ADV},\dagger,\star}$ in Step 3, but uses the updated value $S^{n+1/2}$ rather than $S^{n+1/2,\star}$. The † superscript refers to the fact that the predicted velocity field does not satisfy the divergence constraint,

$$\nabla \cdot (\beta_0^{n+1/2} \boldsymbol{U}^{\mathrm{ADV}}) = \beta_0^{n+1/2} \times \left[S^{n+1/2} - \frac{1}{\overline{\Gamma}_1^{n+1/2} p_0^{n+1/2}} \left(\frac{\partial p_0}{\partial t} \right)^{n+1/2} \right], \tag{25}$$

with

$$\beta_0^{n+1/2} = \frac{\beta_0^n + \beta_0^{n+1,\star}}{2},$$
$$\overline{\Gamma}_1^{n+1/2} = \frac{\overline{\Gamma}_1^n + \overline{\Gamma}_1^{n+1,\star}}{2}. \quad (26)$$

we project $\boldsymbol{U}^{\mathrm{ADV},\dagger}$ onto the space of velocities that satisfy the constraint to obtain $\boldsymbol{U}^{\mathrm{ADV}}$ using a MAC projection (see Appendix).

Step 8: advect the thermodynamic variables over a time interval of Δt.

A. Update (ρX_k). This step is identical to Step 4A except we use the updated values $\boldsymbol{U}^{\mathrm{ADV}}$ and $\rho_0^{n+1,\star}$ rather than $\boldsymbol{U}^{\mathrm{ADV},\star}$ and ρ_0^n. In particular:
   i. Compute the face-centered, time-centered species, $(\rho X_k)^{n+1/2,\mathrm{pred}}$, for the conservative update of $(\rho X_k)^{(1)}$ using a Godunov approach (Zingale et al. 2015). Again, we predict $\rho'^n = \rho^n - \rho_0^n$ and X_k^n to faces but here we spatially interpolate $(\rho_0^n + \rho_0^{n+1,*})/2$ to faces to assemble the fluxes.
   ii. Evolve $(\rho X_k)^{(1)} \to (\rho X_k)^{(2)}$ using

$$(\rho X_k)^{(2)} = (\rho X_k)^{(1)} - \Delta t\{\nabla\cdot[\boldsymbol{U}^{\mathrm{ADV}}(\rho X_k)^{n+1/2,\mathrm{pred}}]\}, \quad (27)$$

B. Update ρ_0 by calling Average $[\rho^{(2)}] \to [\rho_0^{n+1}]$.

C. Update p_0 by calling Enforce HSE $[\rho_0^{n+1}] \to [p_0^{n+1}]$.

D. Update the enthalpy. This step is identical to Step 4D except we use the updated values $\boldsymbol{U}^{\mathrm{ADV}}$, ρ_0^{n+1}, $(\rho h)_0^{n+1}$, and p_0^{n+1} rather than $\boldsymbol{U}^{\mathrm{ADV},\star}$, ρ_0^n, $(\rho h)_0^n$, and p_0^n. In particular:
   i. Compute the face-centered, time-centered enthalpy, $(\rho h)^{n+1/2,\mathrm{pred}}$, for the conservative update of $(\rho h)^{(1)}$ using a Godunov approach (Zingale et al. 2015). Again, we predict $(\rho h)'^n = (\rho h)^n - (\rho h)_0^n$ to faces but here we spatially interpolate $[(\rho h)_0^n) + (\rho h)_0^{n+1,*}]/2$ to faces to assemble the fluxes.
   ii. Evolve $(\rho h)^{(1)} \to (\rho h)^{(2)}$.

$$(\rho h)^{(2)} = (\rho h)^{(1)} - \Delta t\{\nabla\cdot[\boldsymbol{U}^{\mathrm{ADV}}(\rho h)^{n+1/2,\mathrm{pred}}]\} + \Delta t\frac{Dp_0}{Dt} \quad (28)$$

where here

$$\frac{Dp_0}{Dt} = \begin{cases} \frac{p_0^{n+1} - p_0^n}{\Delta t} + (\boldsymbol{U}^{\mathrm{ADV}}\cdot\boldsymbol{e}_r)\left(\frac{\partial p_0}{\partial r}\right)^{n+1/2} & \text{(planar)} \\ \frac{p_0^{n+1} - p_0^n}{\Delta t} + [\nabla\cdot(\boldsymbol{U}^{\mathrm{ADV}}p_0^{n+1/2}) - p_0^{n+1/2}\nabla\cdot\boldsymbol{U}^{\mathrm{ADV}}] & \text{(spherical)} \end{cases}, \quad (29)$$

and $p_0^{n+1/2} = (p_0^n + p_0^{n+1})/2$.

Step 9: react the thermodynamic variables over the second $\Delta t/2$ interval.

Call React State $[(\rho X_k)^{(2)}, (\rho h)^{(2)}, p_0^{n+1}] \to [(\rho X_k)^{n+1}, (\rho h)^{n+1}, (\rho\dot{\omega}_k)^{n+1}, (\rho H_{\mathrm{nuc}})^{n+1}]$.

Step 10: define the new-time expansion term, S^{n+1}.

A. Define

$$S^{n+1} = \left(-\sigma\sum_k \xi_k\dot{\omega}_k + \sigma H_{\mathrm{nuc}} + \frac{1}{\rho p_\rho}\sum_k p_{X_k}\dot{\omega}_k\right)^{n+1}. \quad (30)$$

Step 11: update the velocity.

First, we compute the face-centered, time-centered velocities, $\boldsymbol{U}^{n+1/2,\mathrm{pred}}$ using a Godunov approach (Zingale et al. 2015). Then, we update the velocity field $\boldsymbol{U}^n$ to $\boldsymbol{U}^{n+1,\dagger}$ by discretizing Equation (2) as

$$\boldsymbol{U}^{n+1,\dagger} = \boldsymbol{U}^n - \Delta t[\boldsymbol{U}^{\mathrm{ADV}}\cdot\nabla\boldsymbol{U}^{n+1/2,\mathrm{pred}}] - \Delta t\left[\frac{\beta_0^{n+1/2}}{\rho^{n+1/2}}\nabla\left(\frac{\pi^{n-1/2}}{\beta_0^{n-1/2}}\right) + \frac{(\rho^{n+1/2} - \rho_0^{n+1/2})}{\rho^{n+1/2}}g^{n+1/2}\boldsymbol{e}_r\right], \quad (31)$$

where

$$\rho^{n+1/2} = \frac{\rho^n + \rho^{n+1}}{2},$$
$$\rho_0^{n+1/2} = \frac{\rho_0^n + \rho_0^{n+1}}{2}. \quad (32)$$

Again, the † superscript refers to the fact that the updated velocity does not satisfy the divergence constraint,

$$\nabla\cdot(\beta_0^{n+1}\boldsymbol{U}^{n+1}) = \beta_0^{n+1}\left[S^{n+1} - \frac{1}{\overline{\Gamma}_1^{n+1}p_0^{n+1}}\left(\frac{\partial p_0}{\partial t}\right)^{n+1/2}\right]. \quad (33)$$

We use an approximate projection to project $\boldsymbol{U}^{n+1,\dagger}$ onto the space of velocities that satisfy the constraint to obtain $\boldsymbol{U}^{n+1}$ using a "nodal" projection. This projection necessarily differs from the MAC projection used in Step 3 and Step 7 because the velocities in those steps are defined on faces and $\boldsymbol{U}^{n+1}$ is defined at cell centers, requiring different divergence and gradient operators. Furthermore, as part of the nodal projection, we also define a nodal new-time perturbational pressure, $\pi^{n+1/2}$. Refer to Appendix for more details.

This completes one step of the algorithm.

To initialize the simulation we use the same procedure described in Paper III. At the beginning of each simulation, we define $(\boldsymbol{U}, \rho X_k, \rho h)$. We set initial values for $\boldsymbol{U}$, ρX_k, and ρh and perform a sequence of projections (to ensure the velocity field satisfies the divergence constraint) followed by a small number of steps of the temporal advancement scheme to iteratively find initial values for $\pi^{n-1/2}$ and S^0 and S^1 for use in the first time step.

Our approach to AMR is algorithmically the same as the treatment described in Section 5 of Paper V; we refer the reader

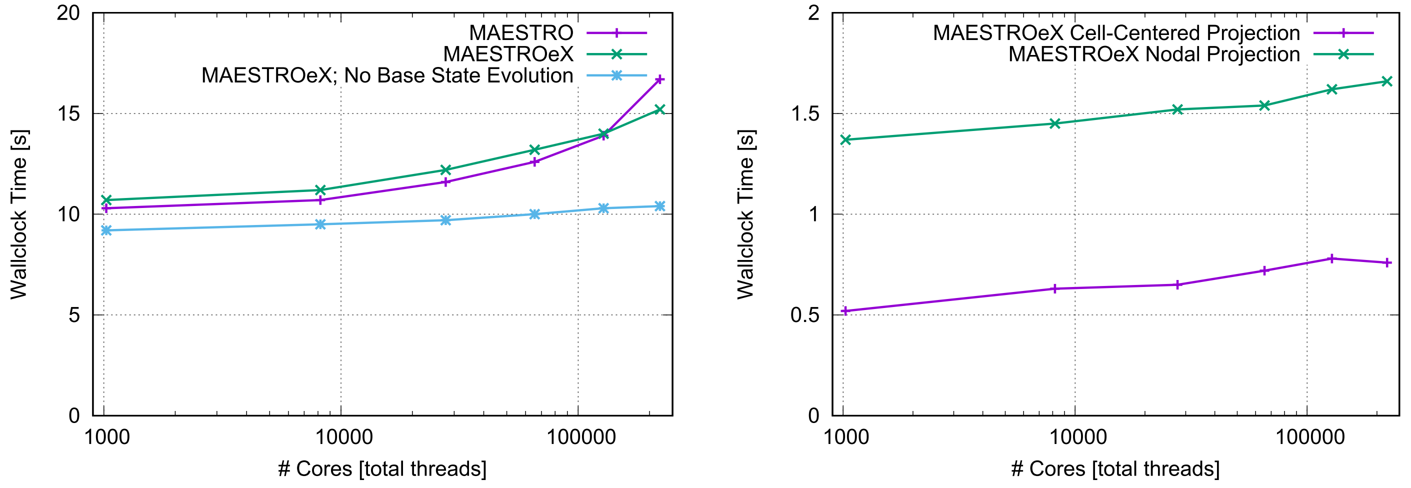


Figure 3. (Left) Weak scaling results for a spherical, full-star white dwarf calculation using the original MAESTRO code, MAESTROeX, and MAESTROeX with base state evolution disabled. Shown is the average wallclock time per time step. (Right) Weak scaling results showing the average wallclock time per time step spent in the cell-centered and nodal linear solvers within a full time step of the aforementioned simulations.

there for details. MAESTROeX supports refinement ratios of 2 between levels. We note that for spherical problems, AMR is only available for the case of a uniformly spaced base state.

4. Performance and Validation

4.1. Performance and Scaling

We perform weak scaling tests for simulations of convection preceding ignition in a spherical, full-star Chandrasekhar mass white dwarf. The simulation setup remains the same as reported in Section 3 of Nonaka et al. (2011) and originally used in Zingale et al. (2011), and thus we emphasize that these scaling tests are performed using meaningful, scientific calculations. Here, we perform simulations using 256^3, 512^3, 768^3, 1024^3, 1280^3, and 1536^3 grid cells on a spatially uniform grid (no AMR). We divide each simulation into 64^3 grids, so these simulations contain between 64 grids (256^3) and 13,824 grids (1536^3). These simulations were performed using the NERSC cori system on the Intel Xeon Phi (KNL) partition. Each node contains 68 cores, each capable of supporting up to four hardware threads (i.e., a maximum of 272 hardware threads per node). For these tests, we assign four MPI tasks to each node, and 16 OpenMP threads per MPI process. Each MPI task is assigned to a single grid, so our tests use between 64 and 13,824 MPI processes (i.e., between 1024 and 221,184 total OpenMP threads). For 64^3 grids we discovered that using more than 16 OpenMP threads did not decrease the wallclock time due to a lack of work available per grid; in principle, one could use larger grids, fewer MPI processes, and more threads per MPI process to obtain a flatter weak scaling curve; however, the overall wallclock time would increase except for extremely large numbers of MPI processes (beyond the range we tested here). Thus, the more accurate measure of weak scaling is to consider the number of MPI processes, since the scaling plot would look virtually identical for larger thread counts. Note that the largest simulation used roughly 36% of the entire computational system.

In the left panel of Figure 3 we compare the wallclock time per time step as a function of total core count (in this case, the total number of OpenMP threads) for the original FBoxLib-based MAESTRO implementation to the AMReX MAESTROeX implementation. These tests were performed using the original temporal integration strategy in Nonaka et al. (2010), noting that the new temporal integration with and without the irregular base state gives essentially the same results. We also include a plot of MAESTROeX without base state evolution. Comparing the original and new implementations, we see similar scaling results except for the largest simulation, where MAESTROeX performs better. We see that the increase in wallclock time from the smallest to largest simulations is roughly 42%. We also note that without base state evolution, the code runs 14% faster for small problems, and scales much better with wallclock time from the smallest to largest simulation increasing by only 13%. This is quite remarkable since there are three linear solves per time step (two cell-centered Poisson solves used in the MAC projection, and a nodal Poisson solve used to compute the updated cell-centered velocities). Contrary to our prior assumptions, the linear solves are not the primary scaling bottleneck in this code. In the right panel of Figure 3, we isolate the wallclock time required for these linear solves and see that (i) the linear solves only use 20%–23% of the total computational time, and (ii) the increase in the solver wallclock time from the smallest to largest simulations is only 28%. Further profiling reveals that the primary scaling bottleneck is the average operator. The averaging operator requires collecting the sum of Cartesian data onto one-dimensional arrays holding every possible mapping radius. This amounts to at least 24,384 double precision values (for the 256^3 simulation) up to 883,584 values (for the 1536^3 simulation). The averaging operator requires a global sum reduction over all processors, and the communication of this data is the primary scaling bottleneck. For the simulation with base state evolution, this averaging operator is only called once per time step (as opposed to 14 times per time step when base state evolution is included). The difference in total wallclock times with and without base state evolution is almost entirely due to the averaging. Note that as expected, advection, reactions, and calls to the EOS scale almost perfectly, since there is only a single parallel communication call to fill ghost cells.

4.2. White Dwarf Convection

To explore the accuracy of the new temporal algorithm, we now analyze in detail three-dimensional, full-star calculations

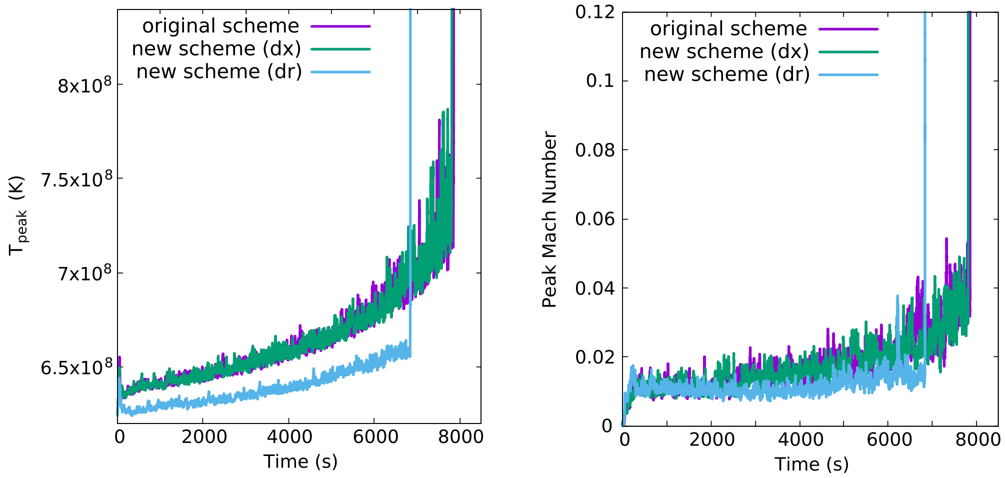


Figure 4. (Left) Peak temperature, T_{peak}, and (right) peak Mach number in a white dwarf until time of ignition at resolution of 256^3 for three different MAESTROeX algorithms.

of convection preceding ignition in a white dwarf. Again, we refer the reader to Nonaka et al. (2011) and Zingale et al. (2011) for setup details. We implement both uniformly and irregularly spaced base state with the new temporal algorithm, while only uniform base state spacing is used in the original algorithm. As in Section 3 of Nonaka et al. (2011), we choose the peak temperature and peak Mach number as the two diagnostics to compare the simulations. Figure 4 shows the evolution of both peak temperature and peak Mach number until time of ignition on a single-level grid with resolution of 256^3. The simulation using the new temporal scheme with uniformly spaced base state gives the same qualitative results as the original scheme, and predicts a similar time of ignition ($t = 7810$ s compared to $t = 7850$ s for original algorithm). The simulation using the new temporal scheme with irregularly spaced base state displays a slightly different peak temperature behavior during the initial transition period $t < 150$ s, which results in the difference between the curves post transition. We strongly suspect that this is a result of using a different initial model file (the resolution near the center of the star is much coarser with the irregular spacing than the uniform spacing). Fortunately, the simulation with irregular base state spacing still follows the same trend as with uniform spacing, and the star is shown to ignite at an earlier time $t = 6840$ s.

Figure 5 shows the peak temperature evolution over the first 1000 s on two grids of differing resolutions, 256^3 and 512^3. Limited allocations prevented us from running this simulation further. As previously suspected, the simulation using irregularly spaced base state agrees much closer with the results from using uniform spacing as the resolution increases. This is most likely due to the increased resolution of the initial model, which more closely matches the uniformly spaced counterpart. This is especially important when computing the base state pressure from base state density, which is particularly sensitive to coarse resolution near the center of the star.

In terms of efficiency, all three simulations on the 256^3 single-level grid were run on Cori haswell with 64 processors and 8 threads per core and their run times were compared. As a result of simplifying the algorithm by eliminating the evolution equations for the base state density and pressure, the simulation using the new temporal algorithm took only 6.75 s per time step with uniformly spaced base state, which is 13% faster than the 7.77 s per time step when using the original scheme. However,

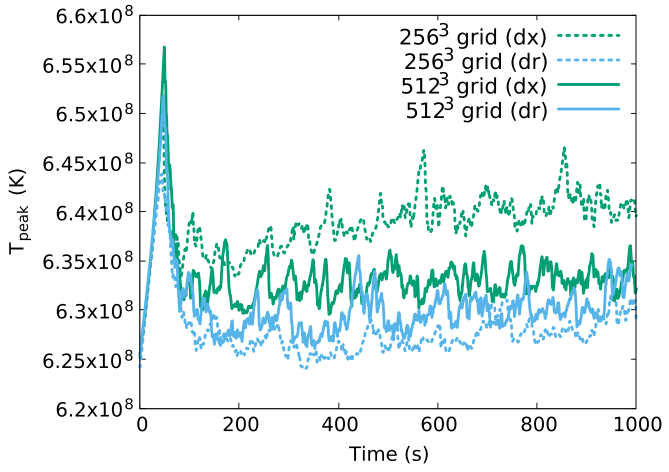


Figure 5. Peak temperature, T_{peak}, in a white dwarf at grid resolutions of 256^3 (dotted line) and 512^3 (solid line) until $t = 1000$ for uniform (dx) and irregular (dr) base state spacing. Note that the irregularly spaced solution agrees better with the uniformly spaced solution as the resolution increases.

we do observe a 25% increase in run time of 9.72 s when using irregularly spaced base state with the new algorithm. This can be explained by the irregularly spaced base state array being much larger in size than its uniformly spaced counterpart, and thus require additional communication and computation time. One possible strategy to significantly reduce the run time is to consider truncating the base state beyond the cutoff density radius.

4.3. AMR Performance

We now test the performance of MAESTROeX for adaptive, three-dimensional simulations to track localized regions of interest over time. Figure 6 illustrates the initial grid structures with two levels of refinement for both planar and spherical geometries where the grid is refined according to the temperature and density profiles, respectively. For each of the problems we tested, the single-level simulation was run using the original temporal scheme and the adaptive simulations using the original and new temporal algorithms. We want to show that the adaptive simulation can give similar results to the

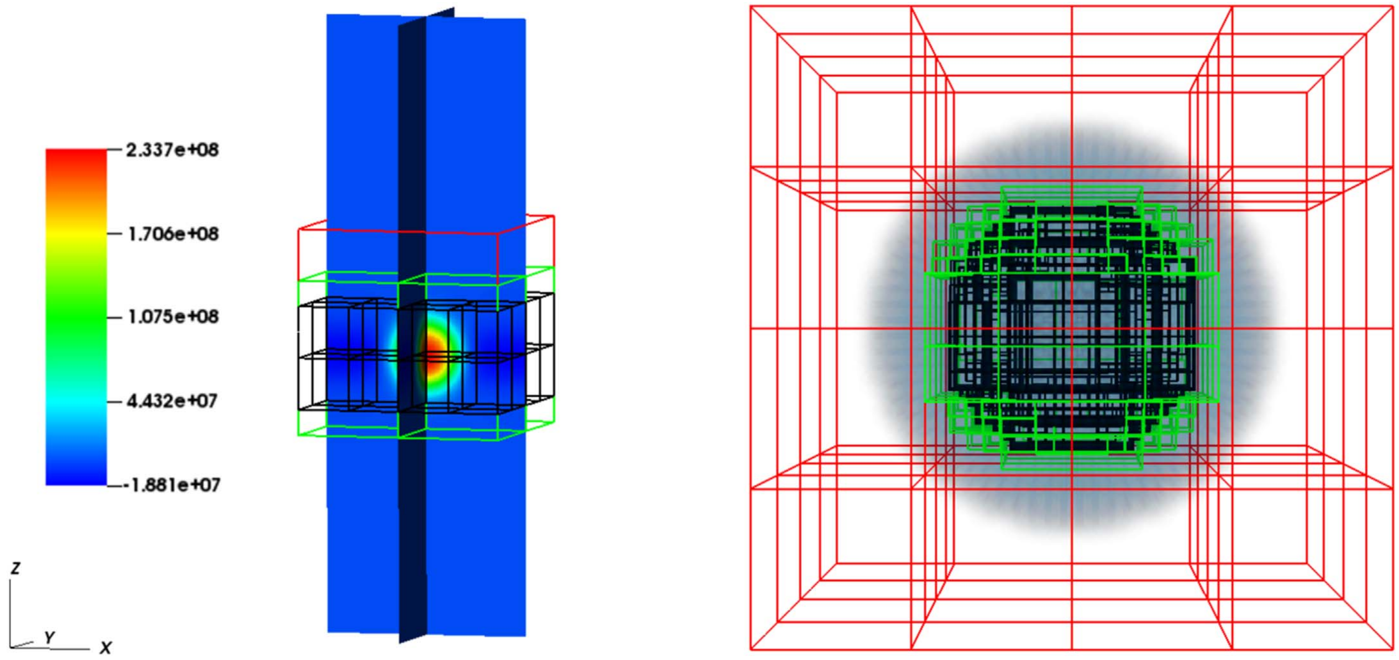


Figure 6. Initial grid structures with two levels of refinement. The red, green, and black lines represent grids of increasing refinement. (Left) Profile of $T - \bar{T}$ for a hot bubble in a white dwarf environment. (Right) Region of star where $\rho \geqslant 10^5$ g cm^{-3} in a full white dwarf star simulation.

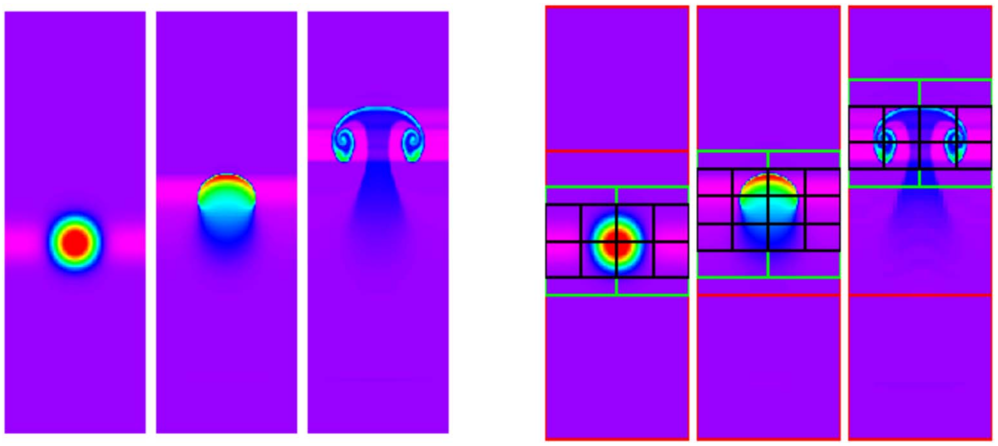


Figure 7. Time-lapse cross-section of a hot bubble in a white dwarf environment at $t = 0$, 1.25, and 2.5 s for (left) single-level simulation, and (right) adaptive simulation at the same effective resolution. The red, green, and black boxes indicate grids of increasing resolution.

single-level simulation and in a more computationally efficient manner.

In the planar case, we use the same problem setup for a hot bubble rising in a white dwarf environment as described in Section 6 of Paper V. Here we use a domain size of 3.6×10^7 cm by 3.6×10^7 cm by 2.88×10^8 cm, and allow the grid structure to change with time. The single-level simulation at a resolution of $128^2 \times 1024$ was run on Cori haswell with 48 processors and took approximately 33.5 s per time step (averaged over 10 time steps) using either the original or new temporal algorithm. The adaptive simulation has a resolution of $32^2 \times 256$ at the coarsest level, resulting in the same effective resolution at the finest level as the single-level simulation. We tag cells that satisfy $T - \bar{T} > 3 \times 10^7$ K as well as all cells at that height. The adaptive run took only 3.7 s per time step, and this 89% decrease in run time is mostly due to the fact that initially only 6.25% of the cells (1,048,576 out of $128^2 \times 1024$ cells) are refined at the finest level. Figure 7 shows a series of planar slices of the temperature profile at time intervals of 1.25 s, and verifies that the adaptive simulation is able to capture the same dynamics as the single-level simulation at much lower computational cost.

We continue to use the full-star white dwarf problem described in Section 4.2 to test adaptive simulations on spherical geometry. The adaptive grid is refined twice by tagging the density at $\rho > 10^5$ g cm^{-3} on the first level and $\rho > 10^8$ g cm^{-3} on the second level. These tagging values have been shown to work well previously in Paper V, but we have found that the code may encounter numerical difficulties when the tagging values are too close to each other in subsequent levels of refinement. The simulation on a single-level grid of 512^3 resolution took 12.7 s per time step (again averaged over 10 time steps). The adaptive grid has a resolution of 128^3 at the coarsest level and an effective resolution of 512^3 at the finest level. On this grid, 27.8% of the cells (4,666,880 out of 256^3 cells) are refined at the first level and 5.3% (7,176,192 out of

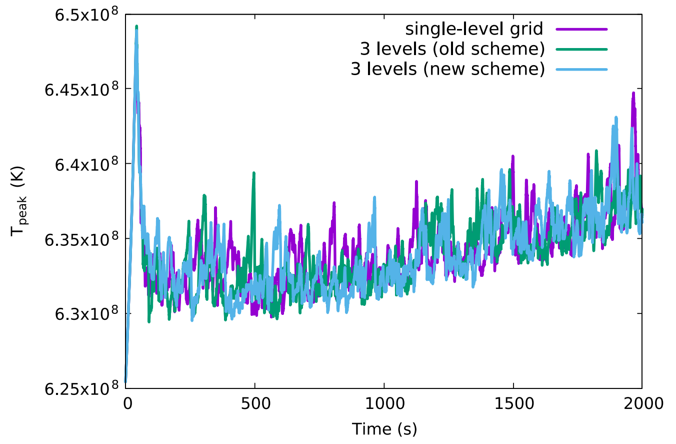


Figure 8. Peak temperature, $T_{\rm peak}$, in a white dwarf from $t = 0$ to 2000 s for grids with effective resolution of 512^3. We can see that the adaptive grids with two levels of refinement give very similar solution trends compared to the single-level grid.

512^3 cells) at the second. The adaptive simulation took 5.61 s, resulting in more than a factor of two in speedup. Both simulations are computed to $t = 2000$ s and we choose to use the peak temperature as the diagnostic to compare the results. Figure 8 shows the evolution of the peak temperature for all three runs and shows that the adaptive simulation gives the same qualitative result as the single-level simulation. We do not expect the curves to match up exactly because the governing equations are highly nonlinear, and slight differences in the solution caused by solver tolerance and discretization error can change the details of the results. Each simulation was run on Cori haswell with 512 processors and 4 threads per core.

5. Conclusions and Future Work

We have developed a new temporal integrator and spatial mapping options into our low Mach number solver, MAESTROeX. The new AMReX-enabled code scales well on large fractions of supercomputers with multicore architectures. Future software enhancements will include GPU implementation. In particular, the AMReX-based companion code, the compressible CASTRO code (Almgren et al. 2010), has recently ported hydrodynamics and reactions to GPUs (A. S. Almgren et al. 2019, in preparation). We plan to leverage the newly implemented mechanisms for offloading compute kernels to GPUs inside of the AMReX software library itself. Our future scientific investigations include convection in massive rotating stars (Heger et al. 2000), the convective Urca process in white dwarfs (Willcox et al. 2016), solar physics (Wood & Brummell 2018), and magnetohydrodynamics (Wood et al. 2011; Wood & Hollerbach 2015). Our future algorithmic enhancements include more accurate and higher-order multiphysics coupling strategies based on spectral deferred corrections (Dutt et al. 2000; Bourlioux et al. 2003). This framework has been successfully used in terrestrial combustion (Pazner et al. 2016; Nonaka et al. 2018)

The work at LBNL was supported by the U.S. Department of Energy's Scientific Discovery Through Advanced Computing (SciDAC) program under contract No. DE-AC02-05CH11231. The work at Stony Brook was supported by DOE/Office of Nuclear Physics grant DE-FG02-87ER40317 and through the SciDAC program DOE grant DE-SC0017955. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.

Facility: NERSC.

Software: AMReX (AMReX Development Team et al. 2019; Zhang et al. 2019), StarKiller Microphysics (The StarKiller Microphysics Development Team et al. 2019), MAESTROX (Fan et al. 2019).

Appendix
Projection Details

To enforce the divergence constraint in Steps 3, 7, and 11, we use a projection method analogous to the methods originally developed for incompressible flow (Bell et al. 1989; Almgren et al. 1998). The basic idea is to decompose the velocity field into a part that satisfies the divergence-satisfying component and a curl-free (gradient of a scalar field) component by solving a variable-coefficient Poisson equation for the scalar field. The details for the MAC projection in Step 3 and Step 7 are given in Appendix B of Paper III. The details of the nodal projection in Step 11 are given in Section 3.2 of Paper III. We note that in the nodal projection, the gradient of the scalar field is used to update the perturbational pressure, π.

Based on our past experience in the MAESTRO project, we have found it useful to split the velocity dynamics into a perturbational and base state velocity,

$$\boldsymbol{U} = \widetilde{\boldsymbol{U}}(\boldsymbol{x}, t) + w_0(r, t)\boldsymbol{e}_r, \tag{34}$$

solve for each term separately, and immediately combine them to find a full velocity that satisfies the constraint. We take that approach here, primarily because it allows us to enforce a boundary condition on w_0 at the edge of the star (i.e., the cutoff density location where we hold density constant). Namely, to enforce that $r^2 w_0$ remain constant is difficult to do when solving for the full velocity. This is demonstrated in Figure 9 (right) where the velocity magnitude is observed to incorrectly increase outside the cutoff density radius when we solve for the full velocity in the nodal projection. The resulting peak temperature also dips significantly as seen in Figure 9 (left), presumably because the dynamics of the overall expansion of the star are not being captured correctly.

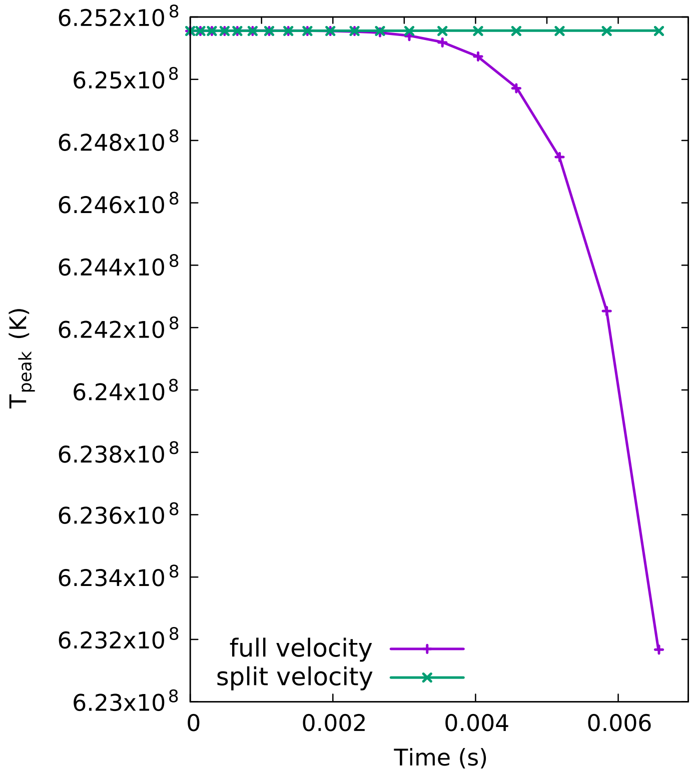
In practice, we solve the constraint over the lateral average,

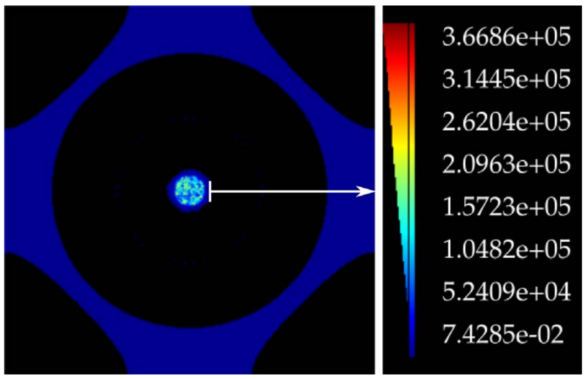
$$\nabla \cdot (\beta_0 \widetilde{\boldsymbol{U}}) = \beta_0 (S - \overline{S}) \tag{35}$$

and separately solve for w_0 using

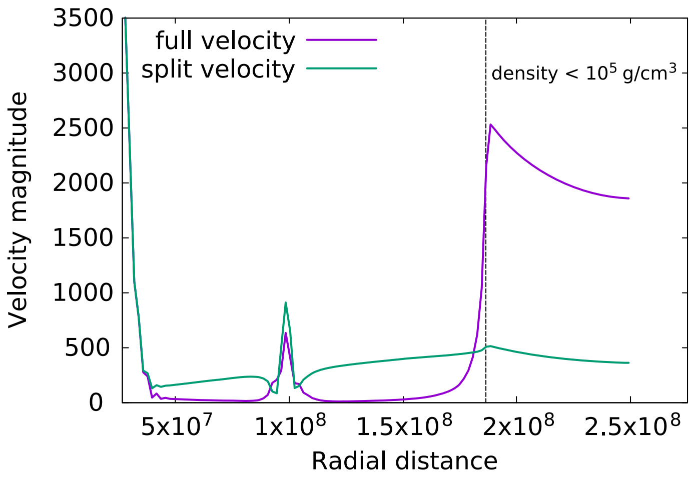
$$\nabla \cdot (\beta_0 w_0 \boldsymbol{e}_r) = \beta_0 \left(\overline{S} - \frac{1}{\overline{\Gamma}_1 p_0} \frac{\partial p_0}{\partial t} \right). \tag{36}$$

To solve for $\widetilde{\boldsymbol{U}}$ we use a projection method, which involves the solution of a variable-coefficient Poisson solver to extract the curl-free component of the unprojected velocity, leaving a velocity field that satisfies the divergence constraint. Note that MAESTRO contains alternate low Mach number formulations that conserve total energy in stratified systems, with minimal changes to the code (see Appendix A of Jacobs et al. 2016 for details). To find w_0 we integrate in one dimension using the procedure in Appendix B of Paper V, keeping in mind that the base state spacing (Δr) should be computed using the





(a) Velocity magnitude, solved using full **U**



(b) Comparison along white line

Figure 9. When solving for the full velocity $\boldsymbol{U}$ in the nodal projection step instead of the split velocity formulation, we illustrate (left) the peak temperature, T_{peak} in a white dwarf at resolution of 256^3 during the early evolution of the star and (right) the magnitude of velocity at early time. We observe much larger velocities outside the density cutoff region, $\rho < 10^5$ g cm^{-3}, that are not seen when we split the velocity dynamics.

appropriate cell-edge and cell-center locations when using irregularly spaced base state. Note that in this approach, we estimated the time-derivative of the pressure in part by examining how laterally averaged $\rho' = \rho - \rho_0$ changed over time (quantified by $\eta_\rho = \overline{\rho' \boldsymbol{U} \cdot \boldsymbol{e}_r}$). After evolving the species (Steps 4A/8A), we compute this term as reported in Paper V. For example, after Step 8A, we define a radial cell-centered $\eta_\rho^{n+1/2}$,

1. For planar geometry, $\eta_\rho = \overline{\rho'(\boldsymbol{U} \cdot \boldsymbol{e}_r)}$,

$$\eta_\rho^{n+1/2} = Average \sum_k [(\boldsymbol{U}^{\mathrm{ADV}} \cdot \boldsymbol{e}_r)(\rho X_k)^{n+1/2,\mathrm{pred}}]. \quad (37)$$

2. For spherical geometry, first construct $\eta_\rho^{\mathrm{cart},n+1/2} = [\rho'(\boldsymbol{U} \cdot \boldsymbol{e}_r)]^{n+1/2}$ on Cartesian cell centers using:

$$\eta_\rho^{\mathrm{cart},n+1/2} = \left[\left(\frac{\rho^{(1)} + \rho^{(2)}}{2}\right) - \left(\frac{\rho_0^n + \rho_0^{n+1}}{2}\right)\right] \cdot (\boldsymbol{U}^{\mathrm{ADV}} \cdot \boldsymbol{e}_r). \quad (38)$$

Then,

$$\eta_\rho^{n+1/2} = \mathbf{Average}(\eta_\rho^{\mathrm{cart},n+1/2}). \quad (39)$$

ORCID iDs

Duoming Fan https://orcid.org/0000-0002-3246-4315
Andrew Nonaka https://orcid.org/0000-0003-1791-0265
Ann S. Almgren https://orcid.org/0000-0003-2103-312X
Alice Harpole https://orcid.org/0000-0002-1530-781X
Michael Zingale https://orcid.org/0000-0001-8401-030X

References

Abbate, E., Iollo, A., & Puppo, G. 2017, JCoPh, 351, 1
Almgren, A., Beckner, V., Bell, J., et al. 2010, ApJ, 715, 1221
Almgren, A. S., Bell, J. B., Colella, P., Howell, L. H., & Welcome, M. L. 1998, JCoPh, 142, 1
Almgren, A. S., Bell, J. B., Nonaka, A., & Zingale, M. 2008, ApJ, 684, 449
Almgren, A. S., Bell, J. B., Rendleman, C. A., & Zingale, M. 2006a, ApJ, 637, 922
Almgren, A. S., Bell, J. B., Rendleman, C. A., & Zingale, M. 2006b, ApJ, 649, 927
AMReX Development Team T., Almgren, A., Beckner, V., et al. 2019, AMReX-Codes/amrex: AMReX 19.12, Zenodo, doi:10.5281/zenodo.2555438
Barsukow, W., Edelmann, P. V. F., Klingenberg, C., Miczek, F., & Roepke, F. K. 2016, arXiv:1612.03910
Bell, J., Day, M., Rendleman, C., Woosley, S., & Zingale, M. 2004, JCoPh, 195, 677
Bell, J. B., Colella, P., & Glaz, H. M. 1989, JCoPh, 85, 257
Bourlioux, A., Layton, A. T., & Minion, M. L. 2003, JCoPh, 189, 651

Brown, P. N., Byrne, G. D., & Hindmarsh, A. C. 1989, SIAM J. Sci. Stat. Comput., 10, 1038
Chalons, C., Girardin, M., & Kokh, S. 2016, CCoPh, 20, 188
Colella, P. 1990, JCoPh, 87, 171
Colella, P., & Woodward, P. R. 1984, JCoPh, 54, 174
Cordier, F., Degond, P., & Kumbaro, A. 2012, JCoPh, 231, 5685
Day, M. S., & Bell, J. B. 2000, CTM, 4, 535
Degond, P., & Tang, M. 2011, CCoPh, 10, 1
Duarte, M., Almgren, A. S., & Bell, J. B. 2015, JAtS, 72, 1605
Durran, D. R. 1989, 46, 1453
Dutt, A., Greengard, L., & Rokhlin, V. 2000, BIT Numerical Mathematics, 40, 241
Fan, D., Nonaka, A., Almgren, A., et al. JOSS, 4, 1757
Gilet, C., Almgren, A., Bell, J., et al. 2013, ApJ, 773, 137
Gilkis, A., & Soker, N. 2016, ApJ, 827, 40
Goffrey, T., Pratt, J., Viallet, M., et al. 2017, A&A, 600, A7
Guillard, H., & Nkonga, B. 2017, Handbook of Numerical Analysis, Vol. 18 (Amsterdam: Elsevier), 203
Haack, J., Jin, S., & Liu, J. 2012, CCoPh, 12, 955
Happenhofer, N., Grimm-Strele, H., Kupka, F., Löw-Baselli, B., & Muthsam, H. 2013, JCoPh, 236, 96
Heger, A., Langer, N., & Woosley, S. 2000, ApJ, 528, 368
Hotta, H., Rempel, M., Yokoyama, T., Iida, Y., & Fan, Y. 2012, A&A, 539, A30
Iijima, H., Hotta, H., & Imada, S. 2019, A&A, 622, A157
Jacobs, A. M., Zingale, M., Nonaka, A., Almgren, A. S., & Bell, J. B. 2016, ApJ, 827, 84
Kifonidis, K., & Müller, E. 2012, A&A, 544, A47
Klein, R., & Pauluis, O. 2012, JAtS, 69, 961
Knio, O. M., Najm, H. N., & Wyckoff, P. S. 1999, JCoPh, 154, 428
Kwatra, N., Su, J., Grétarsson, J. T., & Fedkiw, R. 2009, JCoPh, 228, 4146
Lin, D. J., Bayliss, A., & Taam, R. E. 2006, ApJ, 653, 545
Majda, A., & Sethian, J. 1985, CST, 42, 185
Malone, C., Nonaka, A., Almgren, A., Bell, J., & Zingale, M. 2011, ApJ, 728, 118
Malone, C., Nonaka, A., Woosley, S., et al. 2014a, ApJ, 782, 11
Malone, C., Zingale, M., Nonaka, A., Almgren, A., & Bell, J. 2014b, ApJ, 788, 115
Miczek, F., Röpke, F. K., & Edelmann, P. V. 2015, A&A, 576, A50
Nonaka, A., Almgren, A., Bell, J., et al. 2010, ApJS, 188, 358
Nonaka, A., Aspden, A., Zingale, M., et al. 2011, ApJ, 745, 73
Nonaka, A., Day, M. S., & Bell, J. B. 2018, CTM, 22, 156
O'Neill, W., & Klein, R. 2014, AtmRe, 142, 133
Padioleau, T., Tremblin, P., Audit, E., Kestener, P., & Kokh, S. 2019, ApJ, 875, 128
Pazner, W. E., Nonaka, A., Bell, J. B., Day, M. S., & Minion, M. L. 2016, CTM, 20, 521
Rehm, R. G., & Baum, H. R. 1978, JRNBS, 83, 2
Rempel, M. 2005, ApJ, 622, 1320
Saltzman, J. 1994, JCoPh, 115, 153
Takeyama, K., Saitoh, T. R., & Makino, J. 2017, NewA, 50, 82
The StarKiller Microphysics Development Team, Bishop, A., Fields, C. E., et al. 2019, Starkiller-astro/Microphysics: Microphysics 19.05, Zenodo, doi:10.5281/zenodo.2656476
Vasil, G. M., Lecoanet, D., Brown, B. P., Wood, T. S., & Zweibel, E. G. 2013, ApJ, 773, 169
Viallet, M., Goffrey, T., Baraffe, I., et al. 2015, A&A, 586, A153
Wang, J., Liang, C., & Miesch, M. S. 2015, JCoPh, 290, 90
Willcox, D. E., Townsley, D. M., Calder, A. C., Denissenkov, P. A., & Herwig, F. 2016, ApJ, 832, 13
Wood, T. S., & Brummell, N. H. 2018, ApJ, 853, 97
Wood, T. S., & Hollerbach, R. 2015, PhRvL, 114, 191101
Wood, T. S., McCaslin, J. O., & Garaud, P. 2011, ApJ, 738, 47
Zhang, W., Almgren, A., Beckner, V., et al. 2019, JOSS, 4, 1370
Zingale, M., Almgren, A. S., Bell, J. B., Nonaka, A., & Woosley, S. 2009, ApJ, 704, 196
Zingale, M., Malone, C. M., Nonaka, A., Almgren, A. S., & Bell, J. B. 2015, ApJ, 807, 60
Zingale, M., Nonaka, A., Almgren, A., et al. 2013, ApJ, 764, 97
Zingale, M., Nonaka, A., Almgren, A. S., et al. 2011, ApJ, 740, 8