MAESTRO: AN ADAPTIVE LOW MACH NUMBER HYDRODYNAMICS ALGORITHM FOR STELLAR FLOWS

A. Nonaka[1], A. S. Almgren[1], J. B. Bell[1], M. J. Lijewski[1], C. M. Malone[2], AND M. Zingale[2]
[1] Center for Computational Science and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[2] Department of Physics & Astronomy, Stony Brook University, Stony Brook, NY 11794-3800, USA

ABSTRACT

Many astrophysical phenomena are highly subsonic, requiring specialized numerical methods suitable for long-time integration. In a series of earlier papers we described the development of MAESTRO, a low Mach number stellar hydrodynamics code that can be used to simulate long-time, low-speed flows that would be prohibitively expensive to model using traditional compressible codes. MAESTRO is based on an equation set derived using low Mach number asymptotics; this equation set does not explicitly track acoustic waves and thus allows a significant increase in the time step. MAESTRO is suitable for two- and three-dimensional local atmospheric flows as well as three-dimensional full-star flows. Here, we continue the development of MAESTRO by incorporating adaptive mesh refinement (AMR). The primary difference between MAESTRO and other structured grid AMR approaches for incompressible and low Mach number flows is the presence of the time-dependent base state, whose evolution is coupled to the evolution of the full solution. We also describe how to incorporate the expansion of the base state for full-star flows, which involves a novel mapping technique between the one-dimensional base state and the Cartesian grid, as well as a number of overall improvements to the algorithm. We examine the efficiency and accuracy of our adaptive code, and demonstrate that it is suitable for further study of our initial scientific application, the convective phase of Type Ia supernovae.

Key words: convection – hydrodynamics – methods: numerical – nuclear reactions, nucleosynthesis, abundances – supernovae: general – white dwarfs

1. INTRODUCTION

Many astrophysical phenomena of interest occur in the low Mach number regime, where the characteristic fluid velocity is small compared to the speed of sound. Some well-known examples are the convective phase of Type Ia supernovae (SNe Ia; Höflich & Stein 2002; Kuhlen et al. 2006; Zingale et al. 2009), classical novae (Glasner et al. 2007), convection in stars (Meakin & Arnett 2007), and Type I X-ray bursts (Lin et al. 2006). Such problems require a numerical approach capable of resolving phenomena over timescales much longer than the characteristic time required for an acoustic wave to propagate across the computational domain. In a series of papers (see Almgren et al. 2006a, henceforth Paper I; Almgren et al. 2006b, henceforth Paper II; Almgren et al. 2008, henceforth Paper III; and Zingale et al. 2009, henceforth Paper IV), we have described the initial development of MAESTRO, a low Mach number hydrodynamics code for computing stellar flows using a time step constraint based on the fluid velocity rather than the sound speed. MAESTRO is suitable for two- and three-dimensional local atmospheric flows as well as three-dimensional full-star flows. All simulations are performed in a Cartesian grid framework, but rely on the presence of a one-dimensional radial base state that describes the average state of the star or atmosphere. Starting with the development of the low Mach number equation set (see Paper I), we demonstrated how to capture the expansion of the base state in a local atmospheric simulation in response to large-scale heating (Paper II) and incorporate reaction networks (Paper III). In Paper IV, we presented the initial application of MAESTRO, following the last two hours of convection inside a white dwarf leading up to the ignition of an SN Ia using a three-dimensional, full-star simulation with a base state that is constant in time.

In general, astrophysical flows are highly turbulent. In the case of the convective period preceding an SN Ia explosion, the Reynolds number is $\mathcal{O}(10^{14})$ (Woosley et al. 2004), far larger than can be modeled on today's supercomputers. Nevertheless, to understand the role of turbulence in these events, we must use increasingly more accurate simulations. In this paper we describe how to incorporate adaptive mesh refinement (AMR), in which we locally refine the Cartesian grid in regions of interest, to allow us to efficiently push to higher spatial resolutions and better capture the turbulent flow in critical regions of the simulation. The primary difference between MAESTRO and other structured grid AMR approaches for incompressible and low Mach number flows is the presence of the time-dependent base state, whose evolution is coupled to the evolution of the full solution. We also describe how to incorporate a time-dependent base state for full-star problems, which involves a novel mapping technique between the one-dimensional base state and the Cartesian grid. This allows us to properly capture the effects of an expanding base state in full-star simulations. We have also made a number of overall improvements to the algorithm, and all together, these enhancements will allow us to compute more efficient and accurate solutions for our target applications, including the convective phase of SNe Ia and Type I X-ray bursts.

This paper is divided into several sections, along with three detailed appendices. In Section 2, we present the governing equations. In Section 3, we give an overview of the methodology, referring the reader to the appendices for full details. In Section 4, we describe the new mapping procedure between one-dimensional and three-dimensional data structures in full-star problems. In Section 5, we discuss the extension of the algorithm to include AMR. In Section 6, we describe the results of our test problems. We conclude with Section 7, which includes future plans for scientific investigation.

2. GOVERNING EQUATIONS

Stellar flows are well characterized by the compressible Euler equations (i.e., viscosity effects are negligible). These equations model all compressibility effects in a fluid, and allow for the

formation and propagation of shocks. For low speed convective flows in a hydrostatically stratified star or atmosphere, we do not need to explicitly follow the propagation of sound waves. However, we do need to include large-scale compressibility effects such as the expansion/contraction of a fluid element as it changes altitude in the stratified background, and the local changes to the density of the fluid element through heating and compositional changes. By reformulating the equations of hydrodynamics to filter out sound waves but preserve the correct large-scale fluid motions and hydrostatic balance, we can retain the compressibility effects we desire while allowing for much larger time steps than a corresponding compressible code. The full derivation of the low Mach number hydrodynamics equations is given in Papers I–III. The resulting equations are

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

$$\frac{\partial \mathbf{U}}{\partial t} = -\mathbf{U} \cdot \nabla \mathbf{U} - \frac{1}{\rho} \nabla \pi - \frac{\rho - \rho_0}{\rho} g \mathbf{e}_r, \tag{2}$$

$$\frac{\partial(\rho h)}{\partial t} = -\nabla \cdot (\rho h \mathbf{U}) + \frac{Dp_0}{Dt} + \rho H_{\text{nuc}} + \rho H_{\text{ext}}, \tag{3}$$

where ρ, $\mathbf{U}$, 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 $\dot{\omega}_k$. The species are constrained such that $\sum_k X_k = 1$ giving $\rho = \sum_k (\rho X_k)$ and

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

The source terms H_{ext} and H_{nuc} are the external heating rate and nuclear energy generation rate per unit mass. The pressure is decomposed into a hydrostatic base state pressure, $p_0 = p_0(r, t)$, and a dynamic pressure, $\pi = \pi(\mathbf{x}, t)$, such that $|\pi|/p_0 = \mathcal{O}(M^2)$ (we use $\mathbf{x}$ to represent the Cartesian coordinate directions of the full state and r to represent the radial coordinate direction for the base state). We also define a base state density, $\rho_0 = \rho_0(r, t)$, which is in hydrostatic equilibrium with p_0, i.e., $\nabla p_0 = -\rho_0 g \mathbf{e}_r$, where $g = g(r, t)$ is the magnitude of the gravitational acceleration and $\mathbf{e}_r$ is the unit vector in the outward radial direction.

Mathematically, this system must still be closed by the equation of state which we express as a divergence constraint on the velocity field (see Paper III),

$$\nabla \cdot (\beta_0 \mathbf{U}) = \beta_0 \left(S - \frac{1}{\overline{\Gamma_1} p_0} \frac{\partial p_0}{\partial t} \right), \tag{5}$$

where β_0 is a density-like variable that carries background stratification, defined as

$$\beta_0(r, t) = \rho_0(0, t) \exp \left(\int_0^r \frac{1}{\overline{\Gamma_1} p_0} \frac{\partial p_0}{\partial r'} dr' \right), \tag{6}$$

and $\overline{\Gamma_1}$ is the lateral average (see Section 4.1) of $\Gamma_1 = d(\log p)/d(\log \rho)$ at constant entropy. The expansion term, S, incorporates local compressibility effects due to heat release from reactions, compositional changes, and external sources,

$$S = -\sigma \sum_k \xi_k \dot{\omega}_k + \frac{1}{\rho p_\rho} \sum_k p_{X_k} \dot{\omega}_k + \sigma H_{\text{nuc}} + \sigma H_{\text{ext}}, \tag{7}$$

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.

It is important to note that if the Mach number of the fluid in a numerical simulation becomes $O(1)$, through large acceleration due to buoyancy or nuclear energy generation, for example, the solution of these equations would no longer be physically meaningful. The low Mach number equations do not enforce that the Mach number remain small; rather, if the dynamics of the flow are such that the Mach number does remain small, then these equations are valid approximations for the evolution of the flow.

As in Papers II and III, we decompose the full velocity field into a base state velocity, w_0, that governs the base state dynamics, and a local velocity, $\widetilde{\mathbf{U}}$, that governs the local dynamics, i.e.,

$$\mathbf{U} = w_0(r, t)\mathbf{e}_r + \widetilde{\mathbf{U}}(\mathbf{x}, t). \tag{8}$$

with $\overline{(\widetilde{\mathbf{U}} \cdot \mathbf{e}_r)} = 0$ and $w_0 = \overline{(\mathbf{U} \cdot \mathbf{e}_r)}$. The velocity evolution equations are then

$$\frac{\partial w_0}{\partial t} = -w_0 \frac{\partial w_0}{\partial r} - \frac{1}{\rho_0} \frac{\partial \pi_0}{\partial r}, \tag{9}$$

$$\begin{aligned}\frac{\partial \widetilde{\mathbf{U}}}{\partial t} = & -\mathbf{U} \cdot \nabla \widetilde{\mathbf{U}} - (\widetilde{\mathbf{U}} \cdot \mathbf{e}_r) \frac{\partial w_0}{\partial r} \mathbf{e}_r - \frac{1}{\rho} \nabla \pi \\ & + \frac{1}{\rho_0} \frac{\partial \pi_0}{\partial r} \mathbf{e}_r - \frac{\rho - \rho_0}{\rho} g \mathbf{e}_r, \end{aligned} \tag{10}$$

where π_0 is the base state component of the perturbational pressure. By laterally averaging to Equation (5), we obtain a divergence constraint for w_0:

$$\nabla \cdot (\beta_0 w_0 \mathbf{e}_r) = \beta_0 \left(\overline{S} - \frac{1}{\overline{\Gamma_1} p_0} \frac{\partial p_0}{\partial t} \right). \tag{11}$$

The divergence constraint for $\widetilde{\mathbf{U}}$ can be found by subtracting (11) into (5), resulting in

$$\nabla \cdot (\beta_0 \widetilde{\mathbf{U}}) = \beta_0 (S - \overline{S}). \tag{12}$$

In the present paper, we revert back to the method introduced in Paper II and define a base state enthalpy, $(\rho h)_0$. We use ρ_0 and $(\rho h)_0$ to define the perturbational quantities $\rho' = \rho - \rho_0$ and $(\rho h)' = (\rho h) - (\rho h)_0$, which are predicted to the Cartesian edges to compute fluxes for the conservative updates of ρ_0 and $(\rho h)_0$. Experience has shown that by advancing perturbational quantities, the slope limiters are more effective at reducing numerical oscillations since they are being applied to a normalized signal, rather than a signal that spans many orders of magnitude over a small number of cells. This is a departure from Paper III where we predicted temperature to the Cartesian edges. Evolution equations for ρ_0 and $(\rho h)_0$ are designed so that ρ_0 and $(\rho h)_0$ will remain the average over a layer of constant radius of ρ and (ρh). The fluxes for (ρX_k) are computed by first predicting ρ_0, ρ', and X_k to time-centered Cartesian edges. The flux for (ρh) is computed by first predicting $(\rho h)_0$ and $(\rho h)'$ to time-centered Cartesian edges.

We now derive the equations used to predict the time-centered Cartesian edge values in the actual algorithm. The species evolution equation is found by combining Equations (1) and (4):

$$\frac{\partial X_k}{\partial t} = -\mathbf{U} \cdot \nabla X_k + \dot{\omega}_k. \tag{13}$$
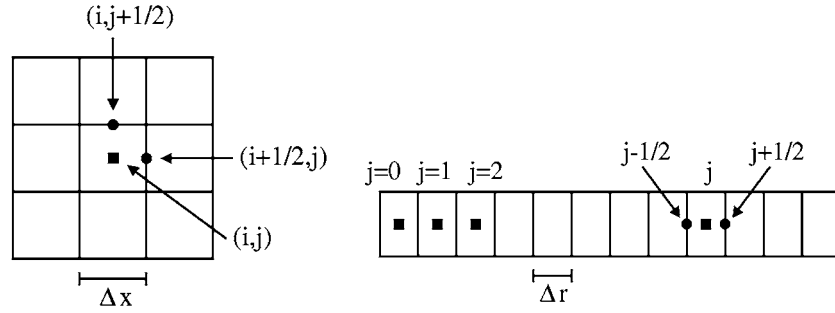
Figure 1. Left: for data on the Cartesian grid (shown here in two dimensions), we use a cell-centered convention with indices i, j, k (in three dimensions). Edges are denoted with a half-integer. Right: the base state lives on a radial array and uses a cell-centered convention with index j. Edges are denoted with a half-integer.

The base state evolution equations for density and enthalpy can be found by averaging (4) and (3) respectively over a layer of constant radius, resulting in

$$\frac{\partial \rho_0}{\partial t} = -\nabla \cdot (\rho_0 w_0 \mathbf{e}_r), \quad (14)$$

$$\frac{\partial (\rho h)_0}{\partial t} = -\nabla \cdot [(\rho h)_0 w_0 \mathbf{e}_r] + \psi + \overline{\rho H_{\mathrm{nuc}}} + \overline{\rho H_{\mathrm{ext}}}, \quad (15)$$

where ψ is the Lagrangian change in the base state pressure defined as $\psi \equiv D_0 p_0 / Dt \equiv \partial p_0 / \partial t + w_0 \partial p_0 / \partial r$ and is related to the total pressure by

$$\frac{Dp_0}{Dt} = \psi + \widetilde{\mathbf{U}} \cdot \nabla p_0. \quad (16)$$

Subtracting the base state evolution equations from the corresponding full state equations yields

$$\frac{\partial \rho'}{\partial t} = -\mathbf{U} \cdot \nabla \rho' - \rho' \nabla \cdot \mathbf{U} - \nabla \cdot (\rho_0 \widetilde{\mathbf{U}}), \quad (17)$$

$$\frac{\partial (\rho h)'}{\partial t} = -\mathbf{U} \cdot \nabla (\rho h)' - (\rho h)' \nabla \cdot \mathbf{U} - \nabla \cdot [(\rho h)_0 \widetilde{\mathbf{U}}]$$
$$+ \widetilde{\mathbf{U}} \cdot \nabla p_0 + (\rho H_{\mathrm{nuc}} - \overline{\rho H_{\mathrm{nuc}}}) + (\rho H_{\mathrm{ext}} - \overline{\rho H_{\mathrm{ext}}}). \quad (18)$$

In our treatment of enthalpy, we split the reactions and external heating from the hydrodynamics, i.e., during the hydrodynamics step, we neglect the $\dot{\omega}_k$, H_{nuc}, and H_{ext} terms. Also, in our treatment of species, we similarly split the reactions from the hydrodynamics.

While Equation (14) properly captures the change in ρ_0 due to atmospheric expansion caused by heating, it neglects changes that can occur due to significant convective overturning. We impose the constraint that $\overline{\rho'} = 0$ for all time. In Paper III, we quantified the drift in $\overline{\rho'}$ by introducing η_ρ in the equation

$$\frac{\partial \overline{\rho'}}{\partial t} = -\nabla \cdot (\eta_\rho \mathbf{e}_r). \quad (19)$$

However, we incorrectly derived η_ρ by assuming $\overline{\nabla \cdot (\rho' w_0 \mathbf{e}_r)} = 0$, when in general this is not true since ρ', when predicted to time-centered edges, does not in general satisfy $\overline{\rho'} = 0$. Therefore, the correct expression is $\eta_\rho = \overline{(\rho' \mathbf{U} \cdot \mathbf{e}_r)}$. In practice, we correct the drift by simply setting $\rho_0 = \overline{\rho}$ after the advective update of ρ. However we still need to explicitly compute η_ρ since it appears in other equations.
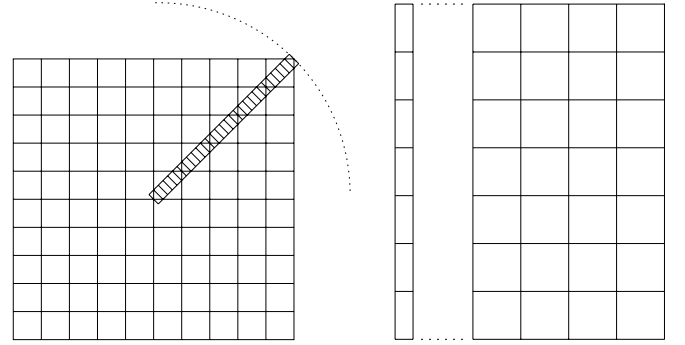
Figure 2. Left: for problems in spherical geometry, there is no direct alignment between the radial array cell centers and the Cartesian grid cell centers. Right: for problems in planar geometry, there is a direct alignment between the radial array cell centers and the Cartesian grid cell centers.

3. OVERVIEW OF NUMERICAL METHODOLOGY

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 ($\widetilde{\mathbf{U}}, \rho, h, X_k, T$) and the one-dimensional base state solution ($w_0, \rho_0, (\rho h)_0, p_0$), all of which are cell-centered except for w_0, which is edge-centered. Edge-centered data are denoted by a half-integer subscript. See Figure 1 for a representation of each grid structure. The time step index is denoted as a superscript.

For planar problems, $\mathbf{e}_r$ is in alignment with the Cartesian grid unit vector in the outward radial direction, $\mathbf{e}_y$ (in two dimensions) or $\mathbf{e}_z$ (in three dimensions). We choose $\Delta r = \Delta x$ so that there will be a simple, direct mapping between the radial array and the Cartesian grid. For spherical problems, $\mathbf{e}_r$ is not in alignment with any Cartesian coordinate direction. Our choice of Δr can be independent of Δx; as in Paper IV, we use $5\Delta r = \Delta x$. Note that for spherical problems, we place the center of the star at the center of the computational domain, and therefore the center of the star lies at a corner where eight Cartesian cells meet. See Figure 2 for an illustration of the relationships between the radial array and the Cartesian grid for spherical and planar geometries.

The time-advancement algorithm uses a predictor–corrector formalism. In the predictor step, we compute an estimate of the expansion of the base state, and then compute a preliminary estimate of the state at the new time level. In the corrector step, we use this preliminary state to compute a new estimate of the expansion of the base state, and then compute the final state at the new time level. We incorporate reactions and external heating using Strang-splitting. As in previous papers, our algorithm is second order in space and time.

The full details of the algorithm are presented in Appendix A. The main algorithm description in Appendix A.4 is similar to the description in Paper III, but has been significantly updated to show how we incorporate the time-dependent spherical base state. There are numerous other improvements we have made to the algorithm since Papers III and IV, which are described in Appendix A.1. Note that these changes have also been incorporated into the main algorithm description. Overall,

1. Appendix A.1 is a summary of algorithmic changes since Papers III and IV.
2. Appendix A.2 describes how we compute and discretize gravity.
3. Appendix A.3 is a description of shorthand notation we use in describing the algorithm.
4. Appendix A.4 steps through the algorithm in detail.
5. Appendix A.5 describes special treatment given to low density regions in the simulation.

4. MAPPING

At many points in the algorithm, we need to map the full state on the Cartesian grid onto a one-dimensional radial array and vice versa. Since Paper IV, we have greatly increased the accuracy of the numerical mapping to and from these data structures for spherical problems, most notably the lateral average routine described below. We refer to the procedure for mapping a cell-centered Cartesian field to a cell-centered radial array as a "lateral average," and we refer to the procedures for mapping an edge- or cell-centered radial array to an edge- or cell-centered Cartesian grid as a "fill."

4.1. Lateral Average

For any Cartesian cell-centered field, ϕ, we define $\overline{\phi} = \mathbf{Avg}(\phi)$ as the lateral average over a layer at constant radius r, Ω_H, as

$$\overline{\phi(r)} = \frac{1}{A(\Omega_H)} \int_{\Omega_H} \phi(r, \mathbf{x}) d\Omega; \qquad A(\Omega_H) = \int_{\Omega_H} d\Omega. \tag{20}$$

Planar. This is a straightforward arithmetic average of cells at a particular height since the radial cell centers are in alignment with the Cartesian grid cell centers.

Spherical. It can be shown that any Cartesian cell center is a radius $\hat{r}_m = \Delta x \sqrt{3/4 + 2m}$ from the center of the star, where $m \geqslant 0$ is an integer. For example, the Cartesian cell with coordinates $(i, j, k) = (1, 1, 1)$ relative to the center of the star lies at a distance of $\Delta x \sqrt{3/4 + 6}$ from the center of the star, corresponding to $m = 3$. The Cartesian cells with coordinates $(i, j, k) = (2, 0, 0)$, $(0, 2, 0)$, or $(0, 0, 2)$ relative to the center of the star also lie at that same distance. For the 384^3 resolution examples in this paper, we have verified that a non-zero set of Cartesian cell centers map into each radius $\hat{r}_m$ until m is large enough to correspond to a radius larger than half the width of the computational domain (i.e., the edge of the domain, not the corner of the domain). Figure 3 shows the number of Cartesian cells that map into each radius $\hat{r}_m$, which we refer to as the "hit count," for a 384^3 domain. We use this mapping to help construct the lateral average, using the following steps.

1. Create an itemized list, $\hat{\phi}_m$, where each element is associated with a radius $\hat{r}_m = \Delta x \sqrt{3/4 + 2m}$ from the center of the star.
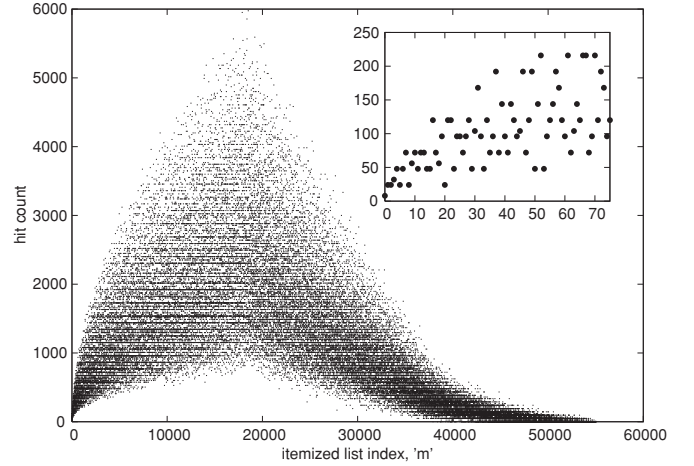
Figure 3. Number of Cartesian cells whose centers lie at a radius $\hat{r}_m$ (i.e., the "hit count") for a 384^3 domain vs. the itemized list index, m. Indices $m < 18{,}432$ correspond to locations within half the width of the computational domain. A non-zero set of Cartesian cell centers maps into the radius associated with every $m \leqslant 37{,}912$, which corresponds to approximately 0.72 times the width of the computational domain. The inset plot is a zoom-in of the innermost 75 values of m.

2. For each $\hat{\phi}_m$, compute the arithmetic average value of the Cartesian cells whose centers lie at the associated radius. As an additional element in the itemized list, include the center of the star (corresponding to a radius of $r = 0$). Compute this additional value of $\hat{\phi}$ at this location using quadratic interpolation with $\hat{\phi}_0$, $\hat{\phi}_1$, and a homogeneous Neumann condition at $r = 0$ as the stencil points. Note that for very large values of m, it is possible that no Cartesian cell centers exist at a radius $\hat{r}_m$ (i.e., the hit count is zero). If so, we say that $\hat{\phi}_m$ has an undefined/invalid value, and we ignore such values for the rest of this procedure.

3. To compute the lateral average, use quadratic interpolation using the value in the itemized list with the closest associated radius, $\hat{\phi}_k$, and the nearest values above and below, $\hat{\phi}_{k_+}$ and $\hat{\phi}_{k_-}$, using divided differences:

$$\overline{\phi(r)} = \hat{\phi}_{k_-} + \frac{\hat{\phi}_k - \hat{\phi}_{k_-}}{\hat{r}_k - \hat{r}_{k_-}} (r - \hat{r}_{k_-})$$
$$+ \frac{\frac{\hat{\phi}_{k_+} - \hat{\phi}_k}{\hat{r}_{k_+} - \hat{r}_k} - \frac{\hat{\phi}_k - \hat{\phi}_{k_-}}{\hat{r}_k - \hat{r}_{k_-}}}{\hat{r}_{k_+} - \hat{r}_{k_-}} (r - \hat{r}_{k_-})(r - \hat{r}_k),$$

where $\hat{r}_{k_-}$, $\hat{r}_k$, and $\hat{r}_{k_+}$ are the three radii associated with $\hat{\phi}_{k_-}$, $\hat{\phi}_k$, and $\hat{\phi}_{k_+}$, respectively. Finally, constrain $\overline{\phi(r)}$ to lie within the range of $\hat{\phi}_{k_-}$, $\hat{\phi}_k$, and $\hat{\phi}_{k_+}$ so as to not introduce any new maxima or minima.

In Section 6.1, we show the improvement of this averaging procedure over the Paper IV procedure.

4.2. Fill

There are four different mappings from a one-dimensional radial array to the three-dimensional Cartesian grid; below we describe the procedures for planar and spherical geometries separately.

4.2.1. Planar

1. To map a cell-centered radial array onto Cartesian cell centers, we use direct injection since the radial cell centers
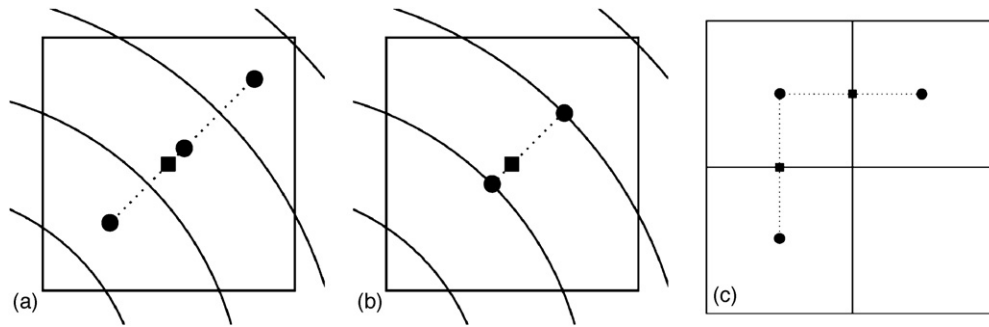
Figure 4. Illustrations of the fill operation for spherical geometry. (a) To fill the Cartesian cell center (fill type 1), represented by the square, from radial cell-centered data, represented by the circles, we use quadratic interpolation from the nearest three points. (b) To fill the Cartesian cell center (fill type 2), represented by the square, from radial edge-centered data, represented by the circles, we use linear interpolation from the nearest two points. (c) To fill a Cartesian edge (fill types 3 and 4), represented by the squares, first fill the Cartesian cell centers, represented by the circles, then average the two neighboring cell centers.

are in alignment with the Cartesian cell centers.

2. To map an edge-centered radial array onto Cartesian cell centers, we average the two nearest radial edge-centered values.

3. To map a cell-centered radial array onto Cartesian edges with normal in the radial direction, we use fourth-order spatial interpolation. For example, in two dimensions,

$$\phi_{i,j+1/2} = \tfrac{7}{12}(\phi_j + \phi_{j+1}) - \tfrac{1}{12}(\phi_{j-1} + \phi_{j+2}). \qquad (21)$$

We constrain $\phi_{i,j+1/2}$ to lie between the interpolated values, and lower the order of interpolation near domain boundaries. For the Cartesian edges transverse to the base state direction, we use direct injection since the radial cell centers are in alignment with these Cartesian edges.

4. To map an edge-centered radial array onto Cartesian edges, we use direct injection on Cartesian edges normal to the base state direction since the radial edges are in alignment with these Cartesian edges. For the remaining Cartesian edges, we average the two nearest radial edge-centered values.

4.2.2. Spherical

1. To map a cell-centered radial array onto Cartesian cell centers, we use quadratic interpolation from the nearest three radial cell centers (see Figure 4(a)). This is a departure from Paper IV, in which we used piecewise constant interpolation.

2. To map an edge-centered radial array onto Cartesian cell centers, we use linear interpolation from the nearest two points (see Figure 4(b)).

3. To map a cell-centered radial array onto Cartesian edges, we first map the radial array onto Cartesian cell centers (see (1)), then average the two neighboring centers to obtain the Cartesian edge values (see Figure 4(c)).

4. To map an edge-centered radial array onto Cartesian edges, we first map the radial array onto Cartesian cell centers (see (2)), then average the two neighboring centers to obtain the Cartesian edge values (see Figure 4(c)).

5. ADAPTIVE MESH REFINEMENT

Our approach to AMR uses a nested hierarchy of logically rectangular grids with successively finer grids at higher levels. This is based on the strategy introduced for gas dynamics by Berger & Colella (1989), extended to the incompressible Navier–Stokes equations by Almgren et al. (1998), and extended to low Mach number reacting flows by Pember et al. (1998) and Day & Bell (2000). We refer the reader to these works for more details. The key difference between our method and these earlier methods stems from the presence of a one-dimensional base state whose time evolution is coupled to that of the full solution. To the best of our knowledge, there are no existing AMR algorithms for astrophysics or any other field, for flows with a time-dependent base state coupled to the full solution. For simplicity, we present a version of the algorithm with no subcycling in time, i.e., the solution at all levels is advanced with the same time step.

We first summarize our AMR approach without the base state, and then discuss how the base state is incorporated in both the planar and spherical cases.

5.1. Creating and Managing the Grid Hierarchy

At each time step the state data are defined on a nested hierarchy of grids, ranging from the base level ($\ell = 1$), which covers the entire computational domain, to the finest level ($\ell = \ell_{\max}$). At each level there is a union of non-intersecting rectangular grids with the same spatial resolution. For simplicity, we require that the cells composing the grids be square ($\Delta x = \Delta y = \Delta z$), and that the refinement ratio between levels be 2. The grids in the interior of the computational domain are required to be properly nested, i.e., the union of grids at level $\ell + 1$ is completely contained in the union of grids at level ℓ. Additionally, in the interior, we require that each grid at level $\ell + 1$ be a distance of at least two level ℓ cells from the boundary between level ℓ and level $\ell - 1$ grids; this allows us to always fill "ghost cells" at level $\ell + 1$ from the level ℓ data (or the physical boundary conditions, if appropriate). We initialize the grid hierarchy and regrid following the procedure outlined in Bell et al. (1994). A user-specified error estimation routine is used to tag cells where more resolution is desired. The tagged cells are grouped into rectangular patches following Berger & Rigoutsos (1991), and subsequently refined to create new grids at next level. Refinement continues until the maximum level is reached.

During **Step 0**,[3] grids at all levels are filled directly from the initial data. As the simulation progresses, we periodically check our refinement criteria and regrid as necessary. This regridding takes places during **Step 12**, before computing the next time step. Newly created grids are filled by using data from previous grids

---

[3] The "**Step**" notation is used in describing the full details of the algorithm in Appendix A.4.

at the same refinement level (if available) or by interpolating from underlying coarser grids.

5.2. Communication Between Levels

Since we use the same time step to advance the solution at all levels, much of the complication associated with synchronization of data between levels in a subcycling algorithm (see Almgren et al. 1998) is eliminated. The MAC projections in **Steps 3** and **7** enforce that $\widetilde{U}^{ADV,\star}$ and $\widetilde{U}^{ADV}$, respectively, on any coarse edge underlying fine edges are the average of the values on the fine edges. Similarly, the nodal projection in **Step 12** enforces that at any coarse node underlying a fine node, the value of ϕ on the coarse node is identical to the value on the fine node above it. The additional communication of data between levels occurs as follows.

1. Before any explicit operation at level $\ell > 1$, data in ghost cells at that level are filled by interpolating from level $\ell - 1$, or imposing physical boundary conditions, as appropriate.
2. Edge-based fluxes at level $\ell < \ell_{max}$ that underlie edges at level $\ell + 1$ are defined to be the average of the fluxes on level $\ell + 1$ at that edge. This enforces conservation.
3. After any update to the solution, data at finer levels are conservatively averaged onto the underlying coarse grid cells, starting at the finest level.

5.3. AMR with a Time-dependent Base State

Our specific treatment of AMR is guided by our initial scientific applications, including Type I X-ray bursts and the convective phase of SNe Ia, as well as numerical concerns, most notably the presence of the one-dimensional base state. Our treatment of the base state in an AMR framework differs for planar versus spherical problems.

For planar problems, our approach is to define a radial base state array with variable mesh spacing. A general localized fine Cartesian grid would require either a base state that exists at multiple resolutions at a particular height, or an interpolation algorithm to obtain the base state value at a particular height if $\Delta r \neq \Delta x$. Both of these methods pose problems, as they generate oscillations in perturbational quantities (such as ρ', $(\rho h)'$ and the $S - \overline{S}$ term on the right-hand side of the divergence constraint) since the lateral average routine is only defined when the base state is aligned with the Cartesian grid across the width of the domain. Any attempt at interpolation will cause oscillations in the perturbational quantities directly related to the interpolation error. We have found that such oscillations can be detrimental to the results. With these issues in mind, we choose to only allow fine grids to exist that span the width of the domain. This way, the base state exists as a single seamless entity with multiple resolutions depending on height (see Figure 5). We will take advantage of this type of grid structure in our studies of Type I X-ray bursts.

Next, we define ghost cell values for the finer base state levels, and fill these values by interpolating coarser data. This makes the algorithm directly compatible with the one-dimensional time-centered edge state calculation used in **Advect Base Density**[4] and **Advect Base Enthalpy**. In particular, the slopes can be used with a consistent stencil at each level, that is not dependent on the data from any other level once the ghost cells are set.

---

[4] The boldface notation refers to numerical modules we have described in Appendix A.3.
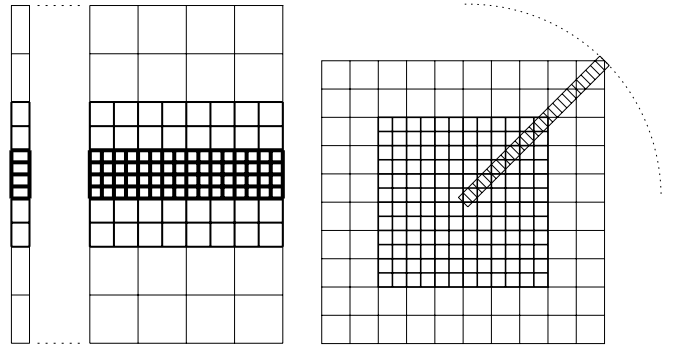
Figure 5. Left: for multi-level problems in planar geometry, we force a direct alignment between the radial array cell centers and the Cartesian grid cell centers by allowing the radial base state spacing to change with space and time. Right: for multi-level problems in spherical geometry, since there is no direct alignment between the radial array cell centers and the Cartesian grid cell centers, we choose to fix the radial base state spacing across levels.

Finally, whenever we regrid the Cartesian grid data, we regrid the base state to match the grid structure of the Cartesian grid. Then, we set $\rho_0 = \overline{\rho}$ and compute p_0 using **Enforce HSE**. To compute ψ and w_0 on the new base state array, we use piecewise linear interpolation of the coarser data to fill any new fine radial cells/edges.

For spherical geometry, we first note that even in the single-level case the radial base state is not aligned with the Cartesian grid. Therefore, we use a base state with a fixed Δr for all levels (see Figure 5). As in the single-level algorithm, we choose $\Delta r = \Delta x/5$, but here, Δx corresponds to resolution of the Cartesian grid at the finest level.

Our next consideration is defining the radial average. First, we first create an itemized list associated with each level of refinement using only Cartesian cells that are not covered by cells at a finer level. At this point one option would be to merge the lists and proceed as in the single-level algorithm; this was tested and found to be problematic. Instead, we choose the list from a chosen particular level and define the average using quadratic interpolation with only this list, as in the single-level case. To decide which list to use, we first examine the three points that would be used by quadratic interpolation at each level. The guiding principle is to avoid using interpolation points with low hit counts. Thus, at each level we find the minimum hit count of the three points; the level which has the largest minimum hit count is the level whose list we use for interpolation. We note that this multi-level average works particularly well when the center of the star is fully refined. This is the case since near the center of the star, there are relatively few Cartesian cells that contribute to each radial bin, so by fully refining the center of the star, we ensure that the multi-level averaging procedure retains the accuracy of a single-level spherical average near the center. For our studies of SNe Ia, we will take advantage of this fact by always refining the center of the star, which is our region of interest (see Figure 12 in Section 6.6, as an example). In Section 6.1, we present numerical tests of the new multi-level averaging procedure for spherical geometry.

For both planar and spherical problems, after regridding the Cartesian grid, we make the state thermodynamically consistent by computing $T = T(\rho, h, X_k)$ (for planar problems) or $T = T(\rho, p_0, X_k)$ (for spherical problems). Then, we recompute $\overline{\Gamma_1}$ and β_0 as described in **Steps 10** and **11**.

5.4. Parallel Implementation

We parallelize the algorithm by distributing the grids on each level across processors. Each grid carries a perimeter of ghost cells that are filled from neighboring grids at the same level or interpolated from coarser grids as needed. This allows the data on each grid to be updated independently of the other grids. A typical grid is large enough (e.g., 32^3 cells) that for explicit operations the cost of computation within each grid greatly exceeds the cost of communication between grids. The linear solves necessary for the MAC projection and approximation nodal projection have higher communication costs, but we still obtain good parallel efficiency for the overall algorithm. A scaling study for MAESTRO can be found in Almgren et al. (2007).

Since the one-dimensional base state arrays are so much smaller than the three-dimensional arrays holding the full solution, each processor owns a copy of the entire one-dimensional base state arrays. Operations such as averaging to define base state quantities require a collection operation among grids, followed by a distribution of the average state to each processor.

6. TEST PROBLEMS

We have developed a suite of test problems in order to test various aspects of our code.

1. In Section 6.1, we show that our new mapping procedure from Section 4 is much more accurate than the mapping procedure from Paper IV.
2. In Section 6.2, we show that we are able to properly capture the expansion of the base state in a three-dimensional full-star simulation due to heating at the center of the star.
3. In Section 6.3, we show that our multi-level algorithm is second-order accurate in space and time by tracking a hot bubble rising in a white-dwarf environment.
4. In Section 6.4, we show that an adaptive algorithm in three-dimensional planar geometry can properly track a hot bubble rising in a white-dwarf environment.
5. In Section 6.5, we demonstrate that a multi-level, two-dimensional planar simulation will properly capture the expansion of the base state due to a heating layer, and also that a multi-level simulation is able to capture the same fine-scale structure as a single-level simulation at the same effective resolution over a short time.
6. In Section 6.6, we demonstrate that a full-star simulation with AMR can be used to study the dynamics of convection in white dwarfs.

For test problems Sections 6.2–6.6, we compute flows in which the density spans at least four orders of magnitude. The large drop in density in the upper atmosphere results in high velocities due to conservation of momentum. This region should not affect the dynamics below the surface in the convecting regions of the star. However, because the time step in the low Mach number code is limited by the highest velocity in the computational domain, the efficiency gains of the low Mach number algorithm are reduced if those velocities persist. We employ a sponging technique to damp such velocities. Damping techniques are commonly used in modeling atmospheric convection (see, for example, Durran 1990). In Paper IV, for full-star convection, we explored the effects of sponging the velocity beginning at two different heights to demonstrate that the dynamics in the upper atmosphere do not affect the convecting regions of the star.

Full details for the sponge implementation can be found in Papers III and IV, but in summary, we add a forcing term to the velocity update before the final projection. We use the parameters r_{sp}, r_{md}, and κ to describe the sponge. The sponge forcing turns on at radius r_{sp} and reaches half of its peak strength at radius r_{md}. We can control the strength of the forcing with the parameter, κ. For full-star problems, we also use an outer sponge which prevents the velocities near the domain boundaries from becoming too large.

For all of these tests, we use a publicly available, general stellar equation of state (Fryxell et al. 2000; Timmes & Swesty 2000), with contributions from ions, radiation, and degenerate/relativistic electrons.

6.1. Mapping

To test the new spherical fill and lateral average routines from Section 4, we first create a unit cube with 384^3 resolution and no refinement. We create a radial array with $\Delta r = \Delta x/5$, and initialize the radial array cell centers with the Gaussian profile $\phi_{\text{exact}}(r) = e^{-10r^2}$. We map ϕ_{exact} to Cartesian cell centers with the fill operation, then compute the lateral average of this Cartesian field, $\mathbf{Avg}(\phi)$. We repeat this process by choosing the grid with two levels of refinement used in the full-star simulation test in Section 6.6, shown in Figure 12.

Figure 6 (left) shows the relative error between ϕ_{exact} and $\mathbf{Avg}(\phi)$ for the new mapping procedures and the mapping procedures from Paper IV for the single-level test. The new mapping procedure greatly decreases the relative error. Figure 6 (right) is a zoom-in of the relative error for the new mapping. For the single-level grid, the relative error is $\mathcal{O}(10^{-8})$ for $r \in [0, 0.036]$ and is $\mathcal{O}(10^{-13})$ for $r \in [0.036, 0.7]$. For the test with two levels of refinement, the relative error is $\mathcal{O}(10^{-8})$ for $r \in [0, 0.7]$, which is still a vast improvement when compared to the Paper IV mapping applied to a single-level simulation.

6.2. Spherical Base State

To test the base state expansion for spherical geometry, we perform a series of tests similar to those in Paper II which tested the base state expansion in planar problems. We run the same test using three codes—a one-dimensional version of the compressible code, CASTRO (Almgren et al. 2010), in spherical coordinates; a one-dimensional version of MAESTRO in spherical coordinates, and a full three-dimensional spherical star in MAESTRO.

Our initial model is generated by specifying a core density $(2.6 \times 10^9 \text{ g cm}^{-3})$, temperature $(6 \times 10^8 \text{ K})$, and a uniform composition $(X(^{12}\text{C}) = 0.3, X(^{16}\text{O}) = 0.7)$ and integrating the equation of hydrostatic equilibrium outward while constraining the specific entropy, s, to be constant. In discrete form, we solve

$$p_{0,j+1} = p_{0,j} + \tfrac{1}{2}\Delta r(\rho_{0,j} + \rho_{0,j+1})g_{j+1/2}. \tag{22}$$

$$s_{0,j+1} = s_{0,j}. \tag{23}$$

We begin with a guess of $\rho_{0,j+1}$ and $T_{0,j+1}$ and use the equation of state and Newton–Raphson iterations to find the values that satisfy our system. Since this is a spherical, self-gravitating star, the gravitation acceleration, $g_{j+1/2}$, is updated each iteration based on the current value of the density. Once the temperature falls below 10^7 K, we keep the temperature constant, and continue determining the density via hydrostatic equilibrium. This uniquely determines the initial model.
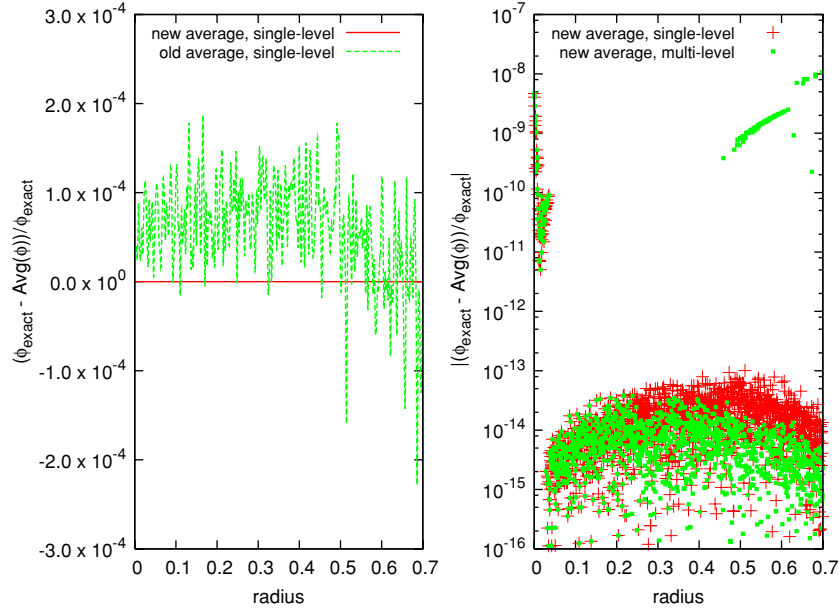
Figure 6. Left: relative error between ϕ_{exact} and $\mathbf{Avg}(\phi)$ from averaging a mapped Gaussian profile centered on a unit cube with 384^3 resolution and no refinement for the current mapping (solid line) and the mapping from Paper IV (dashed line). The new mapping procedure greatly decreases the relative error. Right: a zoom-in of the relative error for the new mapping. The red markers correspond to the test with no refinement. The relative error is $\mathcal{O}(10^{-8})$ for $r \in [0, 0.036]$ and is $\mathcal{O}(10^{-13})$ for $r \in [0.036, 0.7]$. The green dots show the relative error for a test with two levels of refinement using the grid structure shown in Figure 12. The relative error is $\mathcal{O}(10^{-8})$ for $r \in [0, 0.7]$, which is still a vast improvement when compared to the Paper IV mapping applied to a single-level simulation.

For the one-dimensional simulations, we map the inner 5×10^8 cm of the model onto a one-dimensional array with 1280 elements with $\Delta r = 3.90625 \times 10^5$ cm. For the full-star three-dimensional simulation, we map the model onto a 5×10^8 cm^3 domain with 256^3 Cartesian grid cells with $\Delta x = 5\Delta r = 19.53125 \times 10^5$ cm. For the one- and three-dimensional MAESTRO calculations, we use cutoff densities (see Appendix A.5) of $\rho_{\text{cutoff}} = 10^5$ g cm^{-3} and $\rho_{\text{anelastic}} = 10^6$ g cm^{-3}, corresponding to radii of approximately 1.8×10^8 cm and 1.9×10^8 cm, so the star easily fits within the computational domain for each problem. For the full-star three-dimensional simulation, we use an inner sponge with r_{sp} equal to the radius where $\rho_0 = 10^7$ g cm^{-3}, r_{md} equal to the radius where $\rho_0 = 3 \times 10^6$ g cm^{-3}, and $\kappa = 10$ s^{-1}. We use the same outer sponge as in Paper IV. All boundary conditions are outflow, except for the center of the one-dimensional simulations, which uses a symmetry boundary condition. We run each simulation using a CFL number of 0.5.

We heat the center of the star for 0.5 s and look at the solution at 2.0 s (after the compressible solution has had time to re-equilibrate). We use $H_{\text{ext}} = H_0 e^{-(r/10^7 \text{ cm})^2}$, with $H_0 = 10^{16}$ erg g^{-1} s^{-1} chosen to be much larger than the nuclear energy generation rate during the convective phase of SN Ia, in order to see a measurable effect over a few seconds. A comparison of ρ_0, p_0, and $\overline{T}$ at $t = 0$ and $t = 2$ s for each code is shown in Figure 7. There is excellent agreement between each of the simulations.

6.3. Convergence Test

In Paper III, we demonstrated that our single-level algorithm is second order in space and time by tracking a hot bubble rising in a white-dwarf environment using two-dimensional planar geometry. Here, we perform the same test to show that our algorithm, with a level of refinement, is second order in space and time.

We choose a domain size of 7.2×10^7 cm by 2.88×10^8 cm and generate a high resolution initial model with $\Delta r = 7.03125 \times 10^4$

cm (which is equal to Δx for our highest resolution, "exact" solution) using the method described in Appendix C with $r_{\text{base}} = 0$. For each of the remaining, lower resolution simulations for this test, we generate an initial model using Δr equal to the effective Δx of each simulation by linearly interpolating values from the high resolution model. Next, we add a temperature perturbation of the form

$$T_{ij} = T_{0,j} + 0.3 \left[1 + \tanh\left(2 - \frac{d_{ij}}{\sigma} \right) \right], \quad (24)$$

where $\sigma = 2.5 \times 10^6$ cm and d_{ij} is the physical distance between the cell center corresponding to cell (i, j) and the location $(3.6 \times 10^7$ cm, 3.2×10^7 cm). Then, we call the equation of state to compute a consistent ρ, $h = \rho$, $h(p_0, T, X_k)$ everywhere. We use the reaction network described in Section 4.2 in Paper III. We use cutoff densities of $\rho_{\text{cutoff}} = \rho_{\text{anelastic}} = 3 \times 10^6$ g cm^{-3}, and a sponge with r_{sp} equal to the radius where $\rho_0 = 10\rho_{\text{cutoff}}$, r_{md} equal to the radius where $\rho_0 = \rho_{\text{cutoff}}$, and $\kappa = 10$ s^{-1}. We specify periodic boundary conditions on the side walls, outflow at the top, and a solid wall at the bottom of the domain.

Since we do not have an exact analytical solution, we consider a single-level simulation run with 1024×4096 cells and $\Delta t = 3.125 \times 10^{-3}$ s to be the exact solution. We perform three single-level simulations using resolutions of 64×256, 128×512, and 256×1024 grid cells using fixed time steps of $\Delta t = 0.05$ s, 0.025 s, and 0.0125 s, respectively. We also perform two simulations with a single level of refinement with effective resolutions of 128×512 and 256×1024 grid cells with fixed time steps of $\Delta t = 0.025$ s and 0.0125 s, respectively. These fixed time steps correspond to a CFL of 0.9. We refine all cells in the range $r \in [1.8 \times 10^7, 5.4 \times 10^7]$ cm, so effectively we have refined 1/8th of the domain, making sure the hot spot is contained within the refined region. We run each simulation to $t = 1$ s. For this test, whenever we call **Enforce HSE** to compute p_0 from ρ_0, we use $r = 1.8 \times 10^7$ cm as the starting point for integration rather than the location of the cutoff density to ensure that numerical
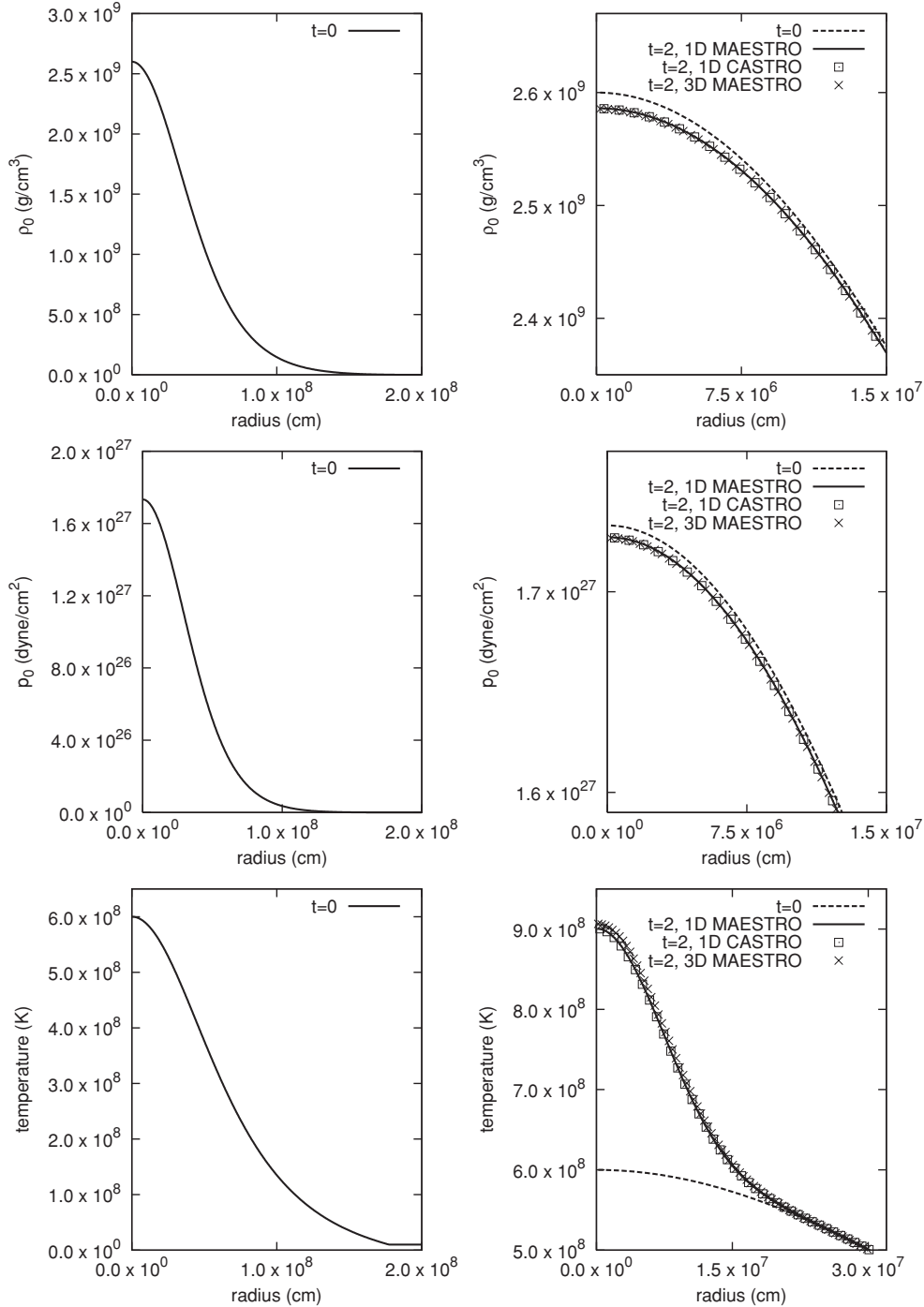
Figure 7. Plots of ρ_0 (top), p_0 (middle), and $\overline{T}$ (bottom) vs. radius for a white-dwarf star subject to heating. The initial profiles are on the left. Close-up views of the initial profiles and final solutions are on the right. We use three test codes: a one-dimensional version of MAESTRO in spherical coordinates, a one-dimensional version of the compressible code, CASTRO, in spherical coordinates, and a full-star three-dimensional version of MAESTRO.

errors due to integrating the equation of hydrostatic equilibrium across simulations with different resolutions are minimized.

In order to compute the L_1 error norm for each simulation, we average the data from the exact solution onto a grid with corresponding resolution. We measure the L_1 error norm in the physical space corresponding to the refined region using

$$L_1 = \frac{1}{n_{\text{cell}}} \sum_{i,j} |\phi_{ij} - \phi_{ij,\text{exact}}|, \qquad (25)$$

where n_{cell} is the number of cells we sum over. This form of the L_1 error norm gives us the average error per cell. We compute

the convergence rate, p, between a coarser and finer simulation using

$$p = \log_2\left(\frac{L_{1,\text{coarser}}}{L_{1,\text{finer}}}\right). \qquad (26)$$

Tables 1 and 2 show the L_1 error norms and convergence rates for the single-level and multi-level solutions, respectively. The convergence rates correspond to the two columns on either side of the reported value. We note second-order convergence in each variable. Additionally, the magnitude of the L_1 error norms for the multi-level simulations is comparable to the corresponding resolution error norms for the single-level simulations. This

Table 1
L_1 Error Norms and Convergence Rates for the Single-level Simulations

Variable	64×256 Error	Rate, p	128×512 Error	Rate, p	256×1024 Error
ρ	2.23×10^4	2.02	5.51×10^3	2.30	1.12×10^3
u	1.40×10^4	2.02	3.44×10^3	2.13	7.90×10^2
v	1.82×10^4	2.03	4.45×10^3	2.24	9.40×10^2
h	3.14×10^{13}	1.97	8.03×10^{12}	2.09	1.89×10^{12}
$X(^{24}Mg)$	5.06×10^{-9}	2.14	1.15×10^{-9}	2.01	2.86×10^{-10}
T	1.38×10^6	1.94	3.59×10^5	2.04	8.72×10^4

Table 2
L_1 Error Norms and Convergence Rates for the Multi-level Simulations

| Variable	128×512 Error	Rate, p	256×1024 Error
ρ	5.83×10^3	2.25	1.23×10^3
u	4.30×10^3	2.09	1.01×10^3
v	4.99×10^3	2.20	1.09×10^3
h	8.20×10^{12}	2.08	1.94×10^{12}
$X(^{24}Mg)$	1.15×10^{-9}	2.01	2.86×10^{-10}
T	3.66×10^5	2.04	8.03×10^4

means that the multi-level simulations are accurately capturing the finer-scale features, as compared to the single-level simulations, i.e., the presence of coarse grid data and/or coarse-fine interfaces is not harming the solution in the refined region.

6.4. Adaptive Bubble Rise

To test the ability for an adaptive, three-dimensional planar simulation to track a localized feature, we examine a hot bubble rising in a white-dwarf environment. The problem setup is exactly the same as in Section 6.3, except that we now compute in three dimensions and allow the grid structure to change with time. We choose a domain size of 7.2×10^7 cm by 7.2×10^7 cm 2.88×10^8 cm and for each simulation, we generate an initial model with $\Delta r = 5.625 \times 10^5$ cm (which is equal to the effective Δx for both of the simulations in this test) using the method described in Appendix C with $r_{base} = 0$. We add a temperature perturbation of the form given in Equation (24), but in three dimensions with the hot spot centered at location $(3.6 \times 10^7$ cm, 3.6×10^7 cm, 3.2×10^7 cm$)$. We will show that the adaptive simulation captures the same dynamics as the single-level simulation in a more computationally efficient manner.

We compute a single-level simulation with $128 \times 128 \times 512$ grid cells, and an adaptive simulation with two levels of refinement at the same effective resolution. For each cell, if the $T - \overline{T} > 3 \times 10^7$ K, we tag all cells at that height to ensure they are computed at the finest refinement level. We run each simulation to $t = 2.5$ s using a CFL number of 0.9.

Figure 8 shows the initial profile of $T - \overline{T}$ and the initial grid structure of the multi-level run. The single-level simulation has 8,388,608 grid cells and takes approximately 32 s per time step. The adaptive simulation initially has 131,072 grid cells at the coarsest level, 114,688 cells at the first level of refinement, and 688,128 cells at the finest level of refinement (the number of grid cells at the finer levels changes slightly with time as the grid structure changes to track the bubble) and takes approximately 12 s per time step, for a factor of 2.7 speedup. Both computations were performed using 32 processors on the Franklin XT4 machine at NERSC. Figure 9 shows a series of planar slices of the simulations at 0.5 s intervals in order to show that the adaptive simulation captures the same dynamics
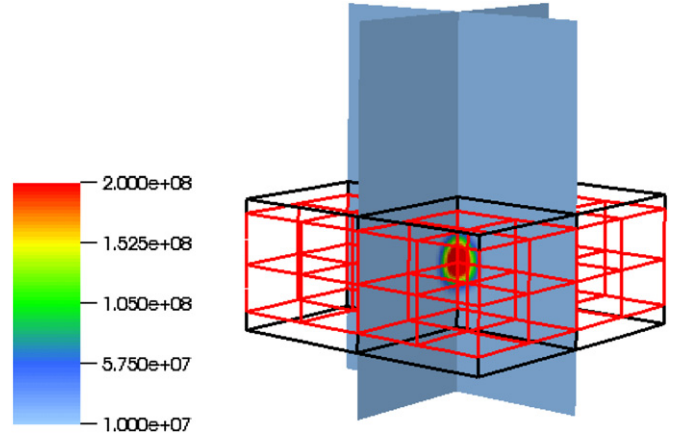
Figure 8. Profile of $T - \overline{T}$ for a hot bubble in a white dwarf environment. The black and red lines represent grids of increasing refinement. The vertical distance shown is from $z = 0$ to 9.2×10^7 cm.

as the single-level simulation. The vertical distance shown is from $z = 0$ to 9.2×10^7 cm.

6.5. Forced Convection

To test the expansion of the base state in a multi-level, two-dimensional planar simulation, we simulate a white-dwarf environment with an external heating layer. We also show that the multi-level simulation captures the same fine-scale structure as a single-level simulation at the same effective resolution for a short time. We performed a similar test in Paper III, but without refinement.

We choose a domain size of 2.5×10^8 cm by 4×10^8 cm and for each simulation, we generate an initial model with Δr equal to the effective Δx for that simulation using the method described in Appendix C with $r_{base} = 5 \times 10^7$ cm. The low entropy layer beneath our model atmosphere prevents the convective flow from reaching our lower boundary.

We apply an external heating layer of the form

$$H_{\text{ext},ij} = H_0 e^{(r_j - r_{\text{layer}})^2} \left[1 + \sum_{m=1}^{3} b_m \sin \left(\frac{k_m \pi x_i}{L_x} + \Psi_m \right) \right] \quad (27)$$

with $r_{\text{layer}} = 1.25 \times 10^8$ cm, $H_0 = 2.5 \times 10^{16}$ erg g^{-1} s^{-1}, $L_x = 2.5 \times 10^8$ cm, and r_j and x_i being the radial and horizontal physical coordinates of cell (i, j). The perturbation parameters are $b = (0.00625, 0.01875, 0.0125)$, $k = (2, 6, 8)$, and $\Psi = (0, \pi/3, \pi/5)$. We disable reactions for this test, since the heating layer was chosen to expand the base state over a very short time period, rather than accurately model reactions.

We use cutoff densities of $\rho_{\text{cutoff}} = \rho_{\text{anelastic}} = 3 \times 10^6$ g cm^{-3} and a sponge with r_{sp} equal to the radius where $\rho_0 = 10^8$ g cm^{-3}, r_{md} equal to the radius where $\rho_0 = \rho_{\text{cutoff}}$, and
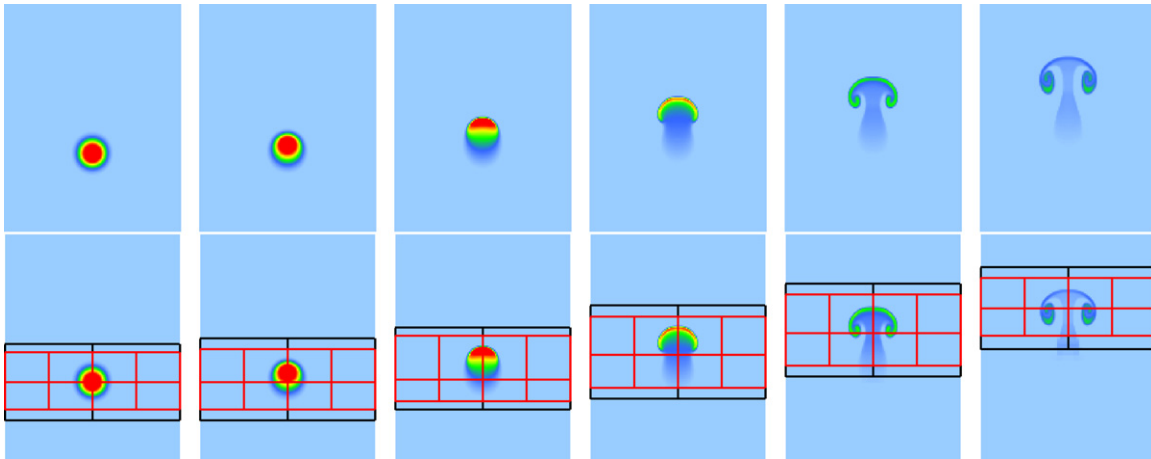
Figure 9. Time-lapse cross section of Figure 8 at $t = 0, 0.5, 1.0, 1.5, 2.0$, and 2.5 s for a single-level simulation (above) an adaptive simulation with two levels of refinement at the same effective resolution (below).

$\kappa = 100$ s^{-1}. We specify periodic boundary conditions on the side walls, outflow at the top, and a solid wall at the bottom of the domain.

In this test, we use a CFL number of 0.9. We perform two single-level simulations using 80×128 and 320×512 cells, and a simulation with two levels of refinement and an effective resolution of 320×512 cells. For the multi-level simulation, we fix the refined grids, ensuring that $r \in [9.375 \times 10^7, 1.5626 \times 10^8]$ cm (which contains the external heating layer) is at the finest level of refinement. We run each simulation to $t = 30$ s.

Figure 10 shows ρ_0, p_0, and $\overline{T}$ after 30 s of convection for the single-level fine grid simulation and the multi-level simulation. There is an excellent agreement between these two simulations, except in the temperature profiles at the top of the domain. However, this corresponds to a region with density below the cutoff densities where the temperature is extremely sensitive to small density perturbations, and furthermore, is not fully refined in this test, so this is an acceptable difference. Both simulations were performed using four processors; the single-level fine grid run required approximately 1.9 s per time step and the multi-level run required approximately 0.6 s per time step, for a factor of 3 speedup.

Figure 11 shows the temperature profile after 3 and 4 s for each of the three simulations. The vertical distance shown is from $z = 5 \times 10^7$ cm to 2.2×10^8 cm. At $t = 3$ s, the multi-level simulation is able to capture the finer-scale structure visible in the single-level fine grid simulation, which is not captured in the single-level coarse grid simulation. At $t = 4$ s, a finer-scale structure is still visible in the multi-level simulation, but the solution begins to drift from the single-level fine grid simulation, which is expected since we are deliberately refining only a part of the convective region.

6.6. Full-star AMR

We now compute three-dimensional, full-star calculations of convection in a white dwarf. This problem models the convection and energetics of a white dwarf that is a few hours from reaching ignition. We performed similar simulations using an earlier version of the algorithm in Paper IV.

We begin with the one-dimensional white-dwarf model described in Section 2.4 in Paper IV. We map this one-dimensional model into the center of a computational domain of 5×10^8 cm on a side. The first simulation is a single-level simulation with 384^3 cells. The second simulation is adaptive with two levels
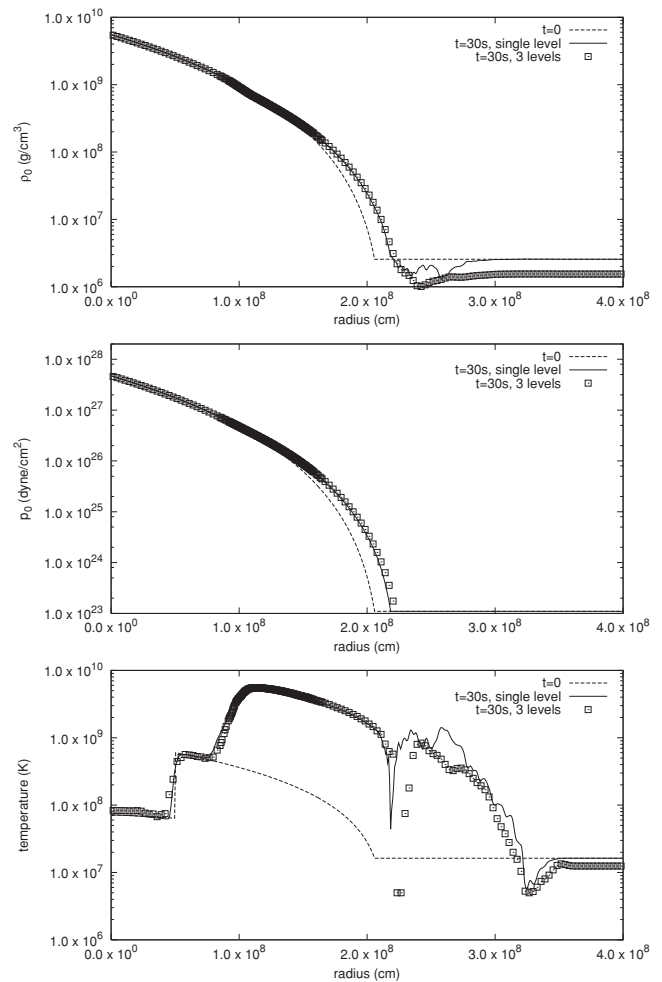
Figure 10. Comparison between ρ_0 (top), p_0 (middle), and $\overline{T}$ (bottom) at $t = 0$, and $t = 30$ s in a white dwarf environment with a heating layer for the single-level fine grid and multi-level simulations. The multi-level simulation captures the same expansion of the base state as the single-level simulation.

of refinement and an effective resolution of 384^3 cells. We use the reaction network strategy from Chamulak et al. (2008) to compute the energetics from the carbon burning. This modified network differs from the one used in Paper IV in the ash composition (we now burn to an ash with $A = 18$ and $Z = 8.8$) and the
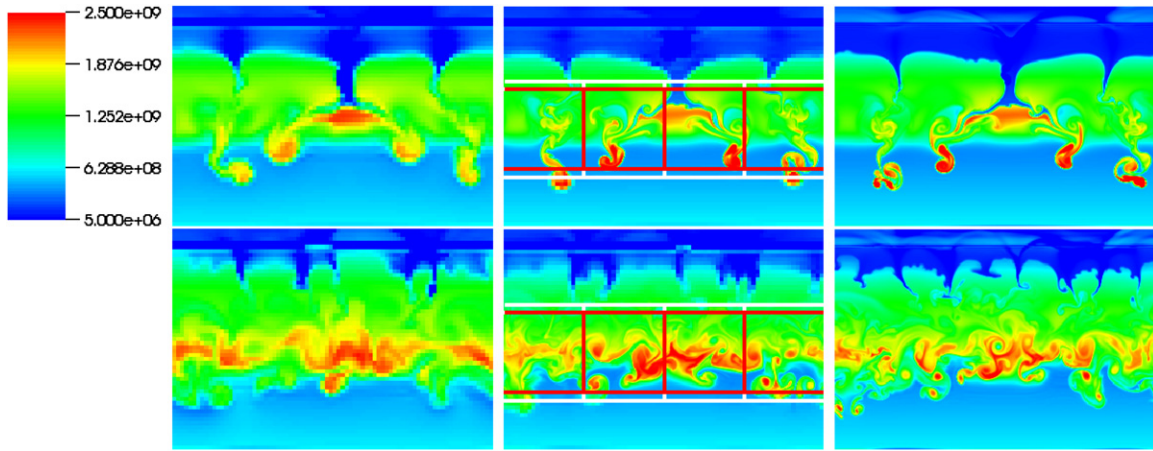
Figure 11. Temperature plots at $t = 3$ s (above) and $t = 4$ s (below) of a white dwarf environment with a heating layer. The single-level coarse grid solution is on the left, the multi-level solution is in the center, and the single-level fine grid solution as on the right. The colored boxes indicate the grid structure at each level of refinement. The vertical distance shown is from $z = 5 \times 10^7$ cm to 2.2×10^8 cm. At $t = 3$ s, the multi-level simulation is able to capture the finer-scale structure visible in the single-level fine grid simulation, which is not captured as well in the single-level coarse grid simulation. At $t = 4$ s, a finer-scale structure is still visible in the multi-level simulation, but the solution begins to drift from the single-level fine grid simulation, which is expected since we are deliberately refining only a part of the convective region.

energy release (we use a quadratic fit to the q-values tabulated on page 164 of Chamulak et al. 2008). Finally, instead of destroying two carbon nuclei for each reaction, we use the M_{12} value of 2.93 described in that paper. We initialize the simulation with a velocity perturbation described exactly as in Section 2.4 in Paper IV. We use the same cutoff densities, sponge parameters, and boundary conditions as in Section 6.2.

Figure 12 shows the initial grid structure of the adaptive simulation. Based on our work in Paper IV, we choose to fully refine all cells where $\rho > 10^8$ g cm^{-3}, since at early times, the dynamics of the star are driven by the reactions and convection in this inner region. We wish to examine whether the adaptive simulation can give the same result as the single-level simulation, and the computational efficiency of each simulation.

We use a CFL number of 0.7 and compute to $t = 900$ s. We choose two diagnostics used in Paper IV to compare the simulations. Peak temperature is a useful diagnostic since the reaction rates are extremely sensitive to temperature, and thus peak temperature serves as a good guide for observing the progression toward ignition. Peak radial velocity is another useful diagnostic as it is a simple measure of the strength of the convection within the star. Since the solution of our equation is highly nonlinear, and the reaction rates scale with temperature as $\sim T^{23}$ (Woosley et al. 2004), we expect that errors from the coarse grid will perturb the solution at the finest level, eventually causing significant differences in the exact flow field. However, as shown in Paper IV, when we run our simulation to the point of ignition, we require upward of hundreds of thousands of time steps. Therefore, in the comparison diagnostics in this test, it is sufficient to compare the overall qualitative solution. An exact quantitative comparison is impossible over long times.

Figure 13 shows the evolution of the peak temperature and peak radial velocity over the first 900 s for both simulations. The adaptive simulation gives the same qualitative result as the single-level simulation. As mentioned before, the curves do not match up more closely because the equations are highly nonlinear, and slight differences in the solution caused by solver tolerance and discretization error change details of the results, but not the overall qualitative results. The single-level simulation has 56,623,760 grid cells and takes approximately 36 s per time
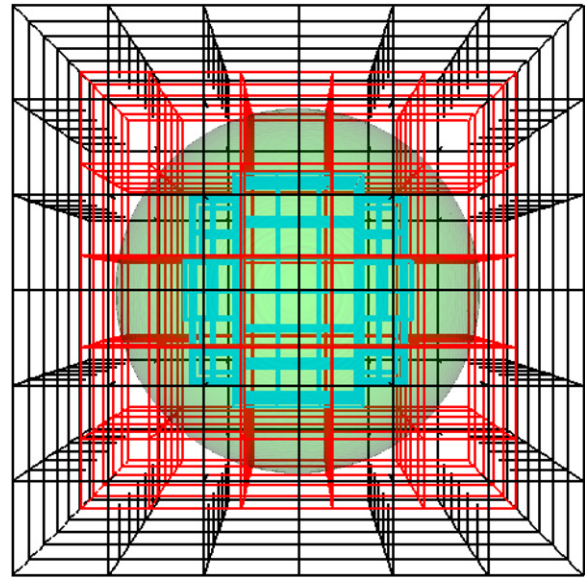
Figure 12. Initial grid structure for a full white dwarf star simulation with two levels of refinement. The colored boxes indicated the grid structure at each level of refinement, each grid containing up to 32^3 cells. The finest grids have effective resolution of 384^3. The shaded sphere indicates where the density is 10^5 g cm^{-3} or greater. In this simulation, we have chosen to include all points where $\rho > 10^5$ g cm^{-3} at the first level of refinement and all points where $\rho > 10^8$ g cm^{-3} at the second level of refinement. There are 216 black grids at the coarse level, 125 red grids at the next level, and 212 blue grids at the finest level.

step. The adaptive simulation initially has 884,736 grid cells at the coarsest level, 3,511,808 cells at the first level of refinement, and 4,282,048 cells at the finest level of refinement (the number of grid cells at the finer levels changes slightly with time) and takes approximately 18 s per time step, for a factor of 2 speedup. Also note that the overall memory requirements are significantly less for the adaptive simulation, as can be seen by the reduced total cell count. Each simulation was run using the Franklin XT4 machine at NERSC with 216 processors.

We note that in the future, when we perform longer calculations up to the point of ignition, we may have to refine
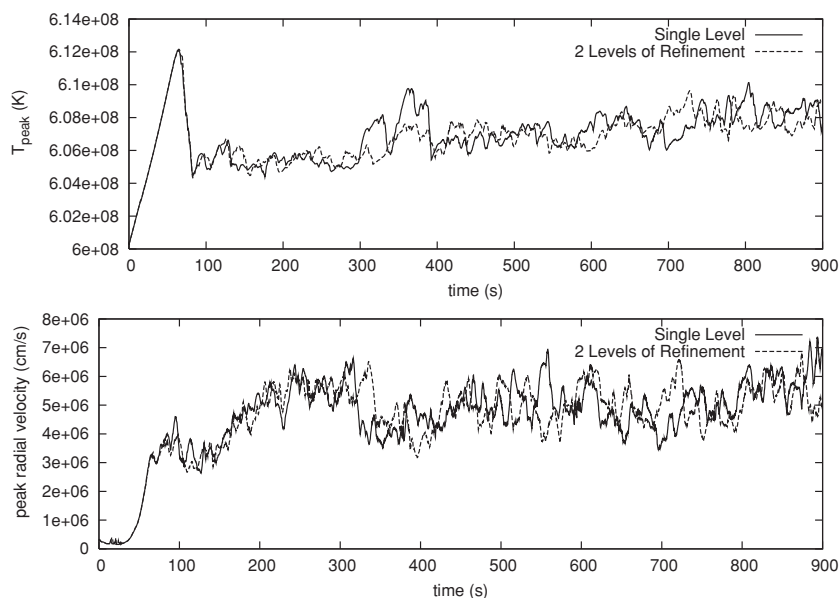
Figure 13. Peak temperature (top) and peak radial velocity (bottom) as a function of time for a single-level and adaptive simulation, each with an effective 384^3 resolution.

a greater portion of the star in order to properly capture the overall dynamics as the convective region expands. In a forthcoming paper, we plan on performing higher resolution studies, where AMR will save us both time and computational resources.

7. CONCLUSIONS AND FUTURE WORK

We have developed a low Mach number hydrodynamics algorithm suitable for full-star flows and local atmospheric regions with a time-evolving base state within an AMR framework. In forthcoming papers, we will use MAESTRO to further our scientific investigation of the convective phase of SNe Ia. Our previous simulations in Paper IV were at modest resolution and assumed a constant base state. We are now performing simulations at higher effective resolutions with the use of AMR along with a time-varying base state. As part of this study, we are examining the tagging conditions necessary to model a full star up to the point of ignition. We are also studying Type I X-ray bursts (Strohmayer & Bildsten 2006; Lin et al. 2006), which are believed to be caused by the thermonuclear explosive burning of hydrogen and/or helium gas accreted into a thin shell on the surface of neutron stars. We pose this problem in planar geometry, model a small patch of the star, and refine near the base of the accreted layer.

APPENDIX A

TIME ADVANCEMENT ALGORITHM

A.1. Single-level Algorithm Changes From Papers III and IV

The single-level algorithm has gone through numerous changes since Papers III and IV. The current single-level algorithm is presented in Sections A.2–A.5; we summarize here the changes since Papers III and IV:

1. We have extended the base state evolution to spherical problems by defining **Advect Base Density Spherical** (Section A.3.2), **Advect Base Enthalpy Spherical** (Section A.3.4), and **Compute** w_0 **Spherical** (Section A.3.5). We have also defined spherical discretizations for ψ and η_ρ (**Steps 4C, 4F, 8C,** and **8F** in Section A.4).

2. For spherical problems, we have improved the accuracy of the mapping of data between the one-dimensional radial array and the three-dimensional Cartesian grid (Section 4).

3. We have upgraded our unsplit piecewise-linear Godunov method for time-centered edge state prediction for the base state and Cartesian grid data to use the unsplit piecewise-parabolic method (PPM; Colella & Woodward 1984) with full corner coupling in three dimensions (Miller & Colella 2002; Saltzman 1994). We shall refer to this procedure as "computing the time-centered edge states" (Section A.3.2, Section A.3.4, and **Steps 3, 4, 7, 8,** and **11** in Section A.4).

4. As introduced in Paper IV, we update T after the call to **React State** (Section A.3.1).

Start of Timestep
ρ^n, $(\rho h)^n$, $(\rho X_k)^n$, T^n, $\tilde{U}^n$
$\rho_0{}^n$, $p_0{}^n$, $w_0{}^{n-\frac{1}{2}}$

Step 1. *React for $\Delta t/2$*
$\rho^{(1)}$, $(\rho h)^{(1)}$, $(\rho X_k)^{(1)}$, $T^{(1)}$, $\tilde{U}^n$
$\rho_0{}^n$, $p_0{}^n$, $w_0{}^{n-\frac{1}{2}}$

corrector

Step 2. *Compute $S^{n+\frac{1}{2},\star\star}$, $w_0{}^{n+\frac{1}{2},\star}$*
$\rho^{(1)}$, $(\rho h)^{(1)}$, $(\rho X_k)^{(1)}$, $T^{(1)}$, $\tilde{U}^n$
$\rho_0{}^n$, $p_0{}^n$, $w_0{}^{n+\frac{1}{2},\star}$

Step 6. *Compute $S^{n+\frac{1}{2},\star}$, $w_0{}^{n+\frac{1}{2}}$*
$\rho^{(1)}$, $(\rho h)^{(1)}$, $(\rho X_k)^{(1)}$, $T^{(1)}$, $\tilde{U}^n$
$\rho_0{}^n$, $p_0{}^n$, $w_0{}^{n+\frac{1}{2}}$

Step 3. *Construct $U^{ADV,\star}$*
$\rho^{(1)}$, $(\rho h)^{(1)}$, $(\rho X_k)^{(1)}$, $T^{(1)}$, $\tilde{U}^n$
$\rho_0{}^n$, $p_0{}^n$, $w_0{}^{n+\frac{1}{2},\star}$

Step 7. *Construct U^{ADV}*
$\rho^{(1)}$, $(\rho h)^{(1)}$, $(\rho X_k)^{(1)}$, $T^{(1)}$, $\tilde{U}^n$
$\rho_0{}^n$, $p_0{}^n$, $w_0{}^{n+\frac{1}{2}}$

Step 4. *Advect for Δt*
$\rho^{(2,\star)}$, $(\rho h)^{(2,\star)}$, $(\rho X_k)^{(2,\star)}$, $T^{(2,\star)}$, $\tilde{U}^n$
$\rho_0{}^{n+1,\star}$, $p_0{}^{n+1,\star}$, $w_0{}^{n+\frac{1}{2},\star}$

Step 8. *Advect for Δt*
$\rho^{(2)}$, $(\rho h)^{(2)}$, $(\rho X_k)^{(2)}$, $T^{(2)}$, $\tilde{U}^n$
$\rho_0{}^{n+1}$, $p_0{}^{n+1}$, $w_0{}^{n+\frac{1}{2}}$

Step 5. *React for $\Delta t/2$*
$\rho^{n+1,\star}$, $(\rho h)^{n+1,\star}$, $(\rho X_k)^{n+1,\star}$, $T^{n+1,\star}$, $\tilde{U}^n$
$\rho_0{}^{n+1,\star}$, $p_0{}^{n+1,\star}$, $w_0{}^{n+\frac{1}{2},\star}$

Step 9. *React for $\Delta t/2$*
ρ^{n+1}, $(\rho h)^{n+1}$, $(\rho X_k)^{n+1}$, T^{n+1}, $\tilde{U}^n$
$\rho_0{}^{n+1}$, $p_0{}^{n+1}$, $w_0{}^{n+\frac{1}{2}}$

predictor

Step 10. *Define S^{n+1}*
ρ^{n+1}, $(\rho h)^{n+1}$, $(\rho X_k)^{n+1}$, T^{n+1}, $\tilde{U}^n$
$\rho_0{}^{n+1}$, $p_0{}^{n+1}$, $w_0{}^{n+\frac{1}{2}}$

Step 11. *Update U*
ρ^{n+1}, $(\rho h)^{n+1}$, $(\rho X_k)^{n+1}$, T^{n+1}, $\tilde{U}^{n+1}$
$\rho_0{}^{n+1}$, $p_0{}^{n+1}$, $w_0{}^{n+\frac{1}{2}}$

Step 12. *Compute new Δt*
ρ^{n+1}, $(\rho h)^{n+1}$, $(\rho X_k)^{n+1}$, T^{n+1}, $\tilde{U}^{n+1}$
$\rho_0{}^{n+1}$, $p_0{}^{n+1}$, $w_0{}^{n+\frac{1}{2}}$

Figure 14. Flowchart of the algorithm. The thermodynamic state variables, base state variables, and local velocity are indicated in each step. Red text indicates that quantity was updated during that step. The predictor–corrector steps are outlined by the dotted box. The blue text indicates state variables that are the same in **Step 6** as they are in **Step 2**, i.e., they are unchanged by the predictor steps.

5. Rather than evolve p_0 using an evolution equation, we simply update p_0 using the hydrostatic equilibrium equation (Section A.3.3). These two methods are analytically equivalent, but in our experience, the numerical drift associated with evolving p_0 using an evolution equation causes the entire solution to drift from thermodynamic equilibrium over time.

6. As explained in Section 2, in the advection step, we predict $(\rho h)'$ to edges instead of T (**Steps 4Hi** and **8Hi** in Section A.4). Thus, a base state enthalpy, $(\rho h)_0$ is required in order to construct the enthalpy fluxes for the conservative update. Unlike ρ_0, we do not need to carry the complete evolution of $(\rho h)_0$. In practice, we set $(\rho h)_0$ equal to the lateral average of the full state enthalpy after the first call to **React State**, i.e., $(\rho h)_0^n = \overline{(\rho h)^{(1)}}$. We then advect $(\rho h)_0$ without reactions or heating to mirror the treatment of (ρh) in the advection step (**Steps 4G** and **8G** in Section A.4).

7. In the advection step, rather than simultaneously updating each of the base state quantities, followed by an update of all the full state quantities, we have interwoven these updates in order to obtain better accuracy. In order: we advect ρ_0, advect ρ, correct ρ_0, advance p_0, advect $(\rho h)_0$, and advect (ρh) (**Steps 4** and **8** in Section A.4). This enables us to use a time-centered ψ rather than a time-lagged ψ for the enthalpy update.

8. Rather than compute $\nabla \cdot \eta_\rho$ and use it to adjust ρ_0 to account for convecting overturning, as was done in **Correct Base** in Paper III, we simply use the lateral average operator to enforce $\rho_0 = \overline{\rho}$, which is simpler and analytically equivalent (**Steps 4D** and **8D** in Section A.4). We use the improved averaging procedure described in Section 4.1, which greatly improves the accuracy of this mapping for spherical problems.

9. We have moved the first reaction step (formerly **Step 3** in Section A.4) to occur before **Steps 1** and **2** in Section A.4. This is a non-functional change; it was only made so that **Steps 2–5** mirror **Steps 6–9** in the overall predictor–corrector scheme.

10. For planar problems, we evolve the enthalpy for the sole purpose of updating the temperature, which is subsequently fed into
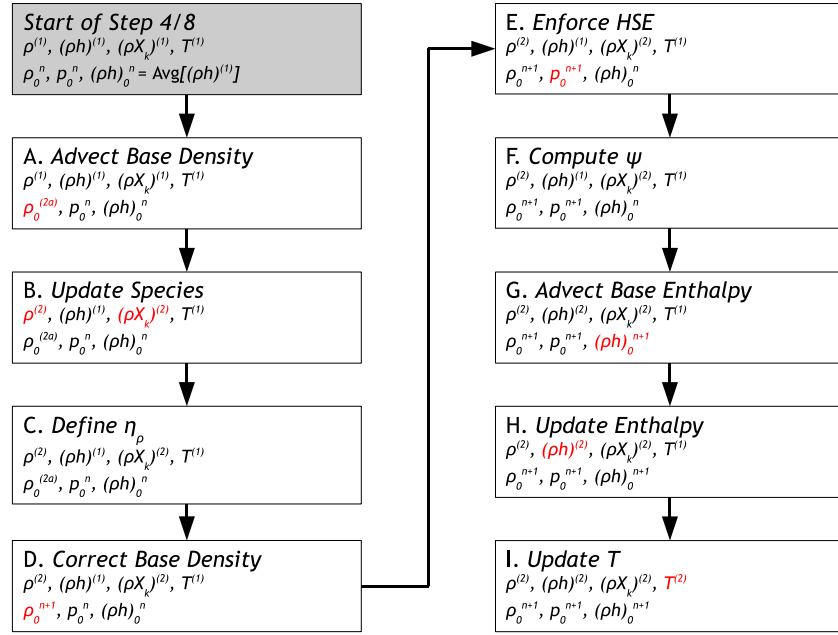
Figure 15. Flowchart for **Steps 4** and **8**. The thermodynamic state variables and base state variables are indicated in each step. Red text indicates that quantity was updated during that step. Note, for **Step 4**, the updated quantities should also have a $\star$ superscript, e.g., **Step 8I** defines $T^{(2)}$ while **Step 4I** defines $T^{(2),\star}$.

the reaction network and used in computing thermodynamic derivatives. For spherical problems, as described in Paper IV, we instead define T from ρ, p_0, and X_k. This completely decouples the enthalpy from the rest of the solution. Our experience has shown that, with spherical geometry, the discretization errors are minimized by using the hydrostatic, radial base state pressure to define temperature. We still evolve the enthalpy, but we use it as a diagnostic to examine to what extent our numerical method keeps the solution in thermodynamic equilibrium.

A.2. Gravity

For planar problems, we treat gravity as constant in space and time. For spherical problems, gravity is computed directly from ρ_0 in the following manner. First, we define the enclosed mass at radial edges and cell centers:

$$m_{\text{encl},j-1/2} = \sum_{k=1}^{j} \frac{4}{3}\pi\left(r_{k-1/2}^3 - r_{k-3/2}^3\right)\rho_{0,k-1}; \qquad m_{\text{encl},j} = m_{\text{encl},j-1/2} + \frac{4}{3}\pi\left(r_j^3 - r_{j-1/2}^3\right)\rho_{0,j}. \tag{A1}$$

In practice, we compute $(r_{k-1/2}^3 - r_{k-3/2}^3)$ as $\Delta r(r_{k-1/2}^2 + r_{k-1/2}r_{k-3/2} + r_{k-3/2}^2)$, in order to minimize roundoff error at large radii and use a similar formula for the radial difference term in the equation for $m_{\text{encl},j}$. Next, we define the gravity at both radial edges and cell centers:

$$g_{j-1/2} = \frac{Gm_{\text{encl},j-1/2}}{r_{j-1/2}^2}; \qquad g_j = \frac{Gm_{\text{encl},j}}{r_j^2}. \tag{A2}$$

We indicate which base state density is used to compute gravity by using a shorthand notation with superscripts on g, e.g., $g^n \equiv g(\rho_0^n)$.

A.3. Main Algorithm Notation

We make use of the following shorthand notations in outlining the algorithm:

A.3.1. Reactions

React State $[\rho^{\text{in}}, (\rho h)^{\text{in}}, X_k^{\text{in}}, T^{\text{in}}, (\rho H_{\text{ext}})^{\text{in}}, p_0^{\text{in}}] \rightarrow [\rho^{\text{out}}, (\rho h)^{\text{out}}, X_k^{\text{out}}, T^{\text{out}}, (\rho\dot{\omega}_k)^{\text{out}}, (\rho H_{\text{nuc}})^{\text{out}}]$ evolves the species and enthalpy due to reactions through $\Delta t/2$ according to

$$\frac{dX_k}{dt} = \dot{\omega}_k(\rho, X_k, T); \qquad \frac{dT}{dt} = \frac{1}{c_p}\left(-\sum_k \xi_k\dot{\omega}_k + H_{\text{nuc}}\right). \tag{A3}$$

Full details of the solution procedure can be found in Paper III. We then define

$$(\rho\dot{\omega}_k)^{\text{out}} = \frac{\rho^{\text{out}}\left(X_k^{\text{out}} - X_k^{\text{in}}\right)}{\Delta t/2}, \tag{A4}$$

$$(\rho h)^{\text{out}} = (\rho h)^{\text{in}} + \frac{\Delta t}{2}(\rho H_{\text{nuc}})^{\text{out}} + \frac{\Delta t}{2}(\rho H_{\text{ext}})^{\text{in}}, \tag{A5}$$

where the enthalpy update includes external heat sources $(\rho H_{\text{ext}})^{\text{in}}$. As introduced in Paper IV, we update the temperature using $T^{\text{out}} = T(\rho^{\text{out}}, h^{\text{out}}, X_k^{\text{out}})$ for planar geometry or $T^{\text{out}} = T(\rho^{\text{out}}, p_0^{\text{in}}, X_k^{\text{out}})$ for spherical geometry. Note that the density remains unchanged within **React State**, i.e., $\rho^{\text{out}} = \rho^{\text{in}}$.

A.3.2. ρ_0 Advection

Advect Base Density $[\rho_0^{\text{in}}, w_0^{\text{in}}] \rightarrow [\rho_0^{\text{out}}, \rho_0^{\text{out},n+1/2}]$ is the process by which we update the base state density through Δt in time. We keep the time-centered edge states, $\rho_0^{\text{out},n+1/2}$, since they are used later in discretization of η_ρ for planar problems.
Planar. We discretize Equation (14) to compute the new base state density,

$$\rho_{0,j}^{\text{out}} = \rho_{0,j}^{\text{in}} - \frac{\Delta t}{\Delta r}\big[\big(\rho_0^{\text{out},n+1/2} w_0^{\text{in}}\big)_{j+1/2} - \big(\rho_0^{\text{out},n+1/2} w_0^{\text{in}}\big)_{j-1/2}\big]. \tag{A6}$$

We compute the time-centered edge states, $\rho_0^{\text{out},n+1/2}$, by discretizing an expanded form of Equation (14):

$$\frac{\partial \rho_0}{\partial t} + w_0 \frac{\partial \rho_0}{\partial r} = -\rho_0 \frac{\partial w_0}{\partial r}, \tag{A7}$$

where the right-hand side is used as the force term.
Spherical. The base state density update now includes the area factors in the divergences:

$$\rho_{0,j}^{\text{out}} = \rho_{0,j}^{\text{in}} - \frac{1}{r_j^2}\frac{\Delta t}{\Delta r}\big[\big(r^2 \rho_0^{\text{out},n+1/2} w_0^{\text{in}}\big)_{j+1/2} - \big(r^2 \rho_0^{\text{out},n+1/2} w_0^{\text{in}}\big)_{j-1/2}\big]. \tag{A8}$$

In order to compute the time-centered edge states, an additional geometric term is added to the forcing, due to the spherical discretization of (14):

$$\frac{\partial \rho_0}{\partial t} + w_0 \frac{\partial \rho_0}{\partial r} = -\rho_0 \frac{\partial w_0}{\partial r} - \frac{2\rho_0 w_0}{r}. \tag{A9}$$

A.3.3. p_0 Update

Enforce HSE $[p_0^{\text{in}}, \rho_0^{\text{in}}] \rightarrow [p_0^{\text{out}}]$ has replaced **Advect Base Pressure** from Paper III as the process by which we update the base state pressure. Rather than discretizing the evolution equation for p_0, we enforce hydrostatic equilibrium directly, which is numerically simpler and analytically equivalent. We first set $p_{0,j=0}^{\text{out}} = p_{0,j=0}^{\text{in}}$ and then update p_0^{out} using

$$p_{0,j+1}^{\text{out}} = p_{0,j}^{\text{out}} + \Delta r\, g_{j+1/2}\frac{\big(\rho_{0,j+1}^{\text{in}} + \rho_{0,j}^{\text{in}}\big)}{2}, \tag{A10}$$

where $g = g(\rho_0^{\text{in}})$. As soon as $\rho_{0,j}^{\text{in}} < \rho_{\text{cutoff}}$, we set $p_{0,j+1}^{\text{out}} = p_{0,j}^{\text{out}}$ for all remaining values of j. Then we compare $p_{0,j_{\max}}^{\text{out}}$ with $p_{0,j_{\max}}^{\text{in}}$ and offset every element in p_0^{out} so that $p_{0,j_{\max}}^{\text{out}} = p_{0,j_{\max}}^{\text{in}}$. We are effectively using the location where the ρ_0^{in} drops below ρ_{cutoff} as the starting point for integration.

A.3.4. $(\rho h)_0$ Advection

Advect Base Enthalpy $[(\rho h)_0^{\text{in}}, w_0^{\text{in}}, \psi^{\text{in}}] \rightarrow [(\rho h)_0^{\text{out}}]$ is the process by which we update the base state enthalpy through Δt in time.
Planar. We discretize Equation (15), neglecting reaction source terms, to compute the new base state enthalpy,

$$(\rho h)_{0,j}^{\text{out}} = (\rho h)_{0,j}^{\text{in}} - \frac{\Delta t}{\Delta r}\big\{\big[(\rho h)_0^{n+1/2} w_0^{\text{in}}\big]_{j+1/2} - \big[(\rho h)_0^{n+1/2} w_0^{\text{in}}\big]_{j-1/2}\big\} + \Delta t\, \psi_j^{\text{in}}. \tag{A11}$$

We compute the time-centered edge states, $(\rho h)_0^{n+1/2}$, by discretizing an expanded form of Equation (15):

$$\frac{\partial (\rho h)_0}{\partial t} + w_0 \frac{\partial (\rho h)_0}{\partial r} = -(\rho h)_0 \frac{\partial w_0}{\partial r} + \psi. \tag{A12}$$

Spherical. The base state enthalpy update now includes the area factors in the divergences:

$$(\rho h)_{0,j}^{\text{out}} = (\rho h)_{0,j}^{\text{in}} - \frac{1}{r_j^2}\frac{\Delta t}{\Delta r}\big\{\big[r^2(\rho h)_0^{n+1/2} w_0^{\text{in}}\big]_{j+1/2} - \big[r^2(\rho h)_0^{n+1/2} w_0^{\text{in}}\big]_{j-1/2}\big\} + \Delta t\, \psi^{\text{in},n+1/2}. \tag{A13}$$

In order to compute the time-centered edge states, an additional geometric term is added to the forcing, due to the spherical discretization of (15):

$$\frac{\partial (\rho h)_0}{\partial t} + w_0 \frac{\partial (\rho h)_0}{\partial r} = -(\rho h)_0 \frac{\partial w_0}{\partial r} - \frac{2(\rho h)_0 w_0}{r} + \psi. \tag{A14}$$

A.3.5. Computing w_0

Here we describe the process by which we compute w_0. The arguments are different for planar and spherical geometries.

Compute w_0 Planar$[\overline{S}^{\text{in}}, \overline{\Gamma}_1^{\text{in}}, p_0^{\text{in}}, \psi^{\text{in}}] \rightarrow [w_0^{\text{out}}]$:
In Paper III, we showed that $\psi = \eta_\rho g$ for planar geometries, and derived an alternate expression for Equation (11). We compute w_0 using Equation (35) in Paper III using the following discretization:

$$\frac{w_{0,j+1/2}^{\text{out}} - w_{0,j-1/2}^{\text{out}}}{\Delta r} = \left(\overline{S}^{\text{in}} - \frac{1}{\overline{\Gamma}_1^{\text{in}} p_0^{\text{in}}} \psi^{\text{in}} \right)_j, \tag{A15}$$

with $w_{0,-1/2} = 0$.
Compute w_0 Spherical $[\overline{S}^{\text{in}}, \overline{\Gamma}_1^{\text{in}}, \rho_0^{\text{in}}, p_0^{\text{in}}, \eta_\rho^{\text{in}}] \rightarrow [w_0^{\text{out}}]$:
We begin with Equation (11) written in spherical coordinates:

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \beta_0 w_0 \right) = \beta_0 \left(\overline{S} - \frac{1}{\overline{\Gamma}_1 p_0} \frac{\partial p_0}{\partial t} \right). \tag{A16}$$

We expand the spatial derivative and recall from Paper I that

$$\frac{1}{\overline{\Gamma}_1 p_0} \frac{\partial p_0}{\partial r} = \frac{1}{\beta_0} \frac{\partial \beta_0}{\partial r}, \tag{A17}$$

giving

$$\frac{1}{r^2} \frac{\partial}{\partial r}(r^2 w_0) = \overline{S} - \frac{1}{\overline{\Gamma}_1 p_0} \underbrace{\left(\frac{\partial p_0}{\partial t} + w_0 \frac{\partial p_0}{\partial r} \right)}_{\psi}. \tag{A18}$$

We solve this equation for w_0 as described in Appendix B.

A.4. Main Algorithm Description

We now describe the main algorithm, making frequent use of the shorthand developed above. In summary, in the predictor step (**Steps 2–5**) we use an estimate of the expansion term, S, to compute a preliminary solution at the new time level, denoted with an "$n + 1, \star$" superscript. In the corrector step (**Steps 6–9**), we use the results from the predictor step to compute a more accurate expansion term, and compute the final solution at the new time level, denoted with an "$n + 1$" superscript. We use Strang splitting to achieve second-order accuracy in time. See Figure 14 for a flow chart of the algorithm, including the notation used as we advance the solution by Δt. Figure 15 is a flow chart of the advection steps (**Steps 4** and **8**), which includes the notation we use as we advect the solution through a time interval of Δt.
The discussion that follows mirrors closely that in Paper III, but has been updated to reflect all the changes throughout the algorithm. The advance of the state through a single time step appears as follows:

Step 0. *Initialization*
   This step remains unchanged from Paper III. The initialization step only occurs at the beginning of the simulation. The initial values for $\mathbf{U}^0$, ρ^0, $(\rho h)^0$, X_k^0, T^0, ρ_0^0, p_0^0, and $\overline{\Gamma}_1^0$ are specified from the problem-dependent initial conditions. The initial time step, Δt^0, is computed as in Paper III. Finally, initial values for $w_0^{-1/2}$, $\eta_\rho^{-1/2}$, $\psi^{-1/2}$, $\pi^{-1/2}$, S^0, and S^1 come from a preliminary pass through the algorithm.

Step 1. *React the full state through the first time interval of $\Delta t/2$.*
   Call **React State**$[\rho^n, (\rho h)^n, X_k^n, T^n, (\rho H_{\text{ext}})^n, p_0^n] \rightarrow [\rho^{(1)}, (\rho h)^{(1)}, X_k^{(1)}, T^{(1)}, (\rho \dot{\omega}_k)^{(1)}, (\rho H_{\text{nuc}})^{(1)}]$.

Step 2. *Compute the provisional time-centered expansion, $S^{n+1/2, \star\star}$, provisional base state velocity, $w_0^{n+1/2, \star}$, and provisional base state velocity forcing.*

   A. Compute $S^{n+1/2, \star\star}$. We compute an estimate for the time-centered expansion term in the velocity divergence constraint (Equation (12)). For the first time step ($n = 0$), we set

$$S^{n+1/2, \star\star} = \frac{S^0 + S^1}{2}, \tag{A19}$$

   where S^1 is found during initialization. For other time steps ($n \neq 0$), following Bell et al. (2004), we extrapolate to the half-time using S at the previous and current time levels

$$S^{n+1/2, \star\star} = S^n + \frac{\Delta t^n}{2} \frac{S^n - S^{n-1}}{\Delta t^{n-1}}. \tag{A20}$$

Next, compute

$$\overline{S^{n+1/2,\star\star}} = \mathbf{Avg}(S^{n+1/2,\star\star}). \tag{A21}$$

B. Compute $w_0^{n+1/2,\star}$.
For planar geometry, call
Compute w_0 Planar$[\overline{S^{n+1/2,\star\star}}, \overline{\Gamma_1^n}, p_0^n, \psi^{n-1/2}] \rightarrow [w_0^{n+1/2,\star}]$.
For spherical geometry, call
Compute w_0 Spherical$[\overline{S^{n+1/2,\star\star}}, \overline{\Gamma_1^n}, \rho_0^n, p_0^n, \eta_\rho^{n-1/2}] \rightarrow [w_0^{n+1/2,\star}]$.

C. Compute the provisional base state velocity forcing. Rearrange Equation (9),

$$-\frac{1}{\rho_0}\frac{\partial \pi_0}{\partial r} = \frac{\partial w_0}{\partial t} + w_0 \frac{\partial w_0}{\partial r}, \tag{A22}$$

with the following discretization:

$$\left(\frac{1}{\rho_0}\frac{\partial \pi_0}{\partial r}\right)^{n,\star} = -\frac{w_0^{n+1/2,\star} - w_0^{n-1/2}}{(\Delta t^n + \Delta t^{n-1})/2} - w_0^{n,\star}\left(\frac{\partial w_0}{\partial r}\right)^{n,\star}, \tag{A23}$$

where $w_0^{n,\star}$ and $(\partial w_0/\partial r)^{n,\star}$ are defined as

$$w_0^{n,\star} = \frac{\Delta t^n w_0^{n-1/2} + \Delta t^{n-1} w_0^{n+1/2,\star}}{\Delta t^n + \Delta t^{n-1}}, \tag{A24}$$

$$\left(\frac{\partial w_0}{\partial r}\right)^{n,\star} = \frac{1}{\Delta t^n + \Delta t^{n-1}}\left[\Delta t^n \left(\frac{\partial w_0}{\partial r}\right)^{n-1/2} + \Delta t^{n-1}\left(\frac{\partial w_0}{\partial r}\right)^{n+1/2,\star}\right]. \tag{A25}$$

If $n = 0$, we use $\Delta t^{-1} = \Delta t^0$.

Step 3. *Construct the provisional time-centered advective velocity on edges, $\widetilde{\mathbf{U}}^{\mathrm{ADV},\star}$.*
Using Equation (10), we compute time-centered edge velocities, $\widetilde{\mathbf{U}}^{\mathrm{ADV},\dagger,\star}$, using $\mathbf{U} = \widetilde{\mathbf{U}}^n + w_0^{n+1/2,\star}$. The $\dagger$ superscript refers to the fact that the predicted velocity field does not satisfy the divergence constraint. We then construct $\widetilde{\mathbf{U}}^{\mathrm{ADV},\star}$ from $\widetilde{\mathbf{U}}^{\mathrm{ADV},\dagger,\star}$ using a MAC projection, as described in detail in Appendix B of Paper III. We note that $\widetilde{\mathbf{U}}^{\mathrm{ADV},\star}$ satisfies the discrete version of $\overline{(\widetilde{\mathbf{U}}^{\mathrm{ADV},\star} \cdot \mathbf{e}_r)} = 0$ as well as

$$\nabla \cdot \left(\beta_0^n \widetilde{\mathbf{U}}^{\mathrm{ADV},\star}\right) = \beta_0^n \left(S^{n+1/2,\star\star} - \overline{S^{n+1/2,\star\star}}\right), \tag{A26}$$

$$\beta_0^n = \beta_0\left(\rho_0^n, p_0^n, \overline{\Gamma_1^n}\right), \tag{A27}$$

where β_0 is computed as described in Appendix C of Paper III.

Step. 4. *Advect the base state and full state through a time interval of Δt.*

A. Update ρ_0, saving the time-centered density at radial edges by calling
Advect Base Density $\left[\rho_0^n, w_0^{n+1/2,\star}\right] \rightarrow \left[\rho_0^{(2a),\star}, \rho_0^{n+1/2,\star,\mathrm{pred}}\right]$.

B. Update (ρX_k) using a discretized version of Equation (1) omitting the reaction terms, which were already accounted for in **React State**. The update consists of two steps:

i. Compute the time-centered species edge states, $(\rho X_k)^{n+1/2,\star,\mathrm{pred}}$, for the conservative update of $(\rho X_k)^{(1)}$. We use Equations (17) and (13) to predict $\rho'^{(1)} = \rho^{(1)} - \rho_0^n$ and $X_k^{(1)} = (\rho X_k)^{(1)}/\rho^{(1)}$ to time-centered edges using $\mathbf{U} = \widetilde{\mathbf{U}}^{\mathrm{ADV},\star} + w_0^{n+1/2,\star}\mathbf{e}_r$, yielding $\rho'^{n+1/2,\star,\mathrm{pred}}$ and $X_k^{n+1/2,\star,\mathrm{pred}}$. We convert the perturbational density full state density using

$$\rho^{n+1/2,\star,\mathrm{pred}} = \rho'^{n+1/2,\star,\mathrm{pred}} + \frac{\rho_0^n + \rho_0^{(2a),\star}}{2}, \tag{A28}$$

where the base state density terms are mapped to Cartesian edges. Then,
$(\rho X_k)^{n+1/2,\star,\mathrm{pred}} = (\rho^{n+1/2,\star,\mathrm{pred}} X_k^{n+1/2,\star,\mathrm{pred}})$.

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

$$(\rho X_k)^{(2),\star} = (\rho X_k)^{(1)} - \Delta t\left\{\nabla \cdot \left[(\widetilde{\mathbf{U}}^{\mathrm{ADV},\star} + w_0^{n+1/2,\star}\mathbf{e}_r)(\rho X_k)^{n+1/2,\star,\mathrm{pred}}\right]\right\}, \tag{A29}$$

$$\rho^{(2),\star} = \sum_k (\rho X_k)^{(2),\star}, \qquad X_k^{(2),\star} = (\rho X_k)^{(2),\star}/\rho^{(2),\star}. \tag{A30}$$

C. Define a radial edge-centered $\eta_\rho^{n+1/2,\star}$.
For planar geometry, since $\eta_\rho = \overline{\rho'(\mathbf{U} \cdot \mathbf{e}_r)} = \overline{\rho(\mathbf{U} \cdot \mathbf{e}_r)} - \overline{\rho_0(\mathbf{U} \cdot \mathbf{e}_r)} = \overline{\rho(\mathbf{U} \cdot \mathbf{e}_r)} - \rho_0 w_0$,

$$\eta_\rho^{n+1/2,\star} = \mathbf{Avg} \sum_k \left[\left(\widetilde{\mathbf{U}}^{\mathrm{ADV},\star} \cdot \mathbf{e}_r + w_0^{n+1/2,\star} \right) (\rho X_k)^{n+1/2,\star,\mathrm{pred}} \right] - w_0^{n+1/2,\star} \rho_0^{n+1/2,\star,\mathrm{pred}}. \tag{A31}$$

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

$$\eta_\rho^{\mathrm{cart},n+1/2,\star} = \left[\left(\frac{\rho^{(1)} + \rho^{(2),\star}}{2} \right) - \left(\frac{\rho_0^n + \rho_0^{(2a),\star}}{2} \right) \right] \cdot \left(\widetilde{\mathbf{U}}^{\mathrm{ADV},\star} \cdot \mathbf{e}_r + w_0^{n+1/2,\star} \right). \tag{A32}$$

Then,

$$\eta_\rho^{n+1/2,\star} = \mathbf{Avg}\left(\eta_\rho^{\mathrm{cart},n+1/2,\star} \right). \tag{A33}$$

This gives a radial cell-centered $\eta_\rho^{n+1/2,\star}$. To get $\eta_\rho^{n+1/2,\star}$ at radial edges, average the two neighboring radial cell-centered values.

D. Correct ρ_0 by setting $\rho_0^{n+1,\star} = \mathbf{Avg}(\rho^{(2),\star})$.

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

F. Compute $\psi^{n+1/2,\star}$.
For planar geometry,
$$\psi_j^{n+1/2,\star} = \tfrac{1}{2} \left(\eta_{\rho,j-1/2}^{n+1/2,\star} + \eta_{\rho,j+1/2}^{n+1/2,\star} \right) g. \tag{A34}$$

For spherical geometry, first compute

$$\overline{\Gamma_1^{(1)}} = \mathbf{Avg}\left[\Gamma_1\left(\rho^{(1)}, p_0^n, X_k^{(1)} \right) \right], \tag{A35}$$

$$\overline{\Gamma_1^{(2),\star}} = \mathbf{Avg}\left[\Gamma_1\left(\rho^{(2),\star}, p_0^{n+1,\star}, X_k^{(2),\star} \right) \right]. \tag{A36}$$

Then, define $\psi^{n+1/2,\star}$ using Equation (A18)

$$\psi_j^{n+1/2,\star} = \left(\frac{\overline{\Gamma_1^{(1)}} + \overline{\Gamma_1^{(2),\star}}}{2} \right)_j \left(\frac{p_0^n + p_0^{n+1,\star}}{2} \right)_j \left\{ \overline{S_j^{n+1/2,\star}} - \frac{1}{r_j^2} \left[\left(r^2 w_0^{n+1/2,\star} \right)_{j+1/2} - \left(r^2 w_0^{n+1/2,\star} \right)_{j-1/2} \right] \right\}. \tag{A37}$$

G. Update $(\rho h)_0$. First, compute $(\rho h)_0^n = \mathbf{Avg}[(\rho h)^{(1)}]$. Then, call
Advect Base Enthalpy $[(\rho h)_0^n, w_0^{n+1/2,\star}, \psi^{n+1/2,\star}] \to [(\rho h)_0^{n+1,\star}]$.

H. Update the enthalpy using a discretized version of Equation (3), again omitting the reaction and heating terms since we already accounted for them in **React State**. This equation takes the form

$$\frac{\partial(\rho h)}{\partial t} = -\nabla \cdot (\mathbf{U}\rho h) + \psi + (\widetilde{\mathbf{U}} \cdot \mathbf{e}_r)\frac{\partial p_0}{\partial r}. \tag{A38}$$

For spherical geometry, we solve the analytically equivalent form

$$\frac{\partial(\rho h)}{\partial t} = -\nabla \cdot (\mathbf{U}\rho h) + \psi + \nabla \cdot (\widetilde{\mathbf{U}} p_0) - p_0 \nabla \cdot \widetilde{\mathbf{U}}, \tag{A39}$$

which experience has shown to minimize the drift from thermodynamic equilibrium. The update consists of two steps:

i. Compute the time-centered enthalpy edge state, $(\rho h)^{n+1/2,\star,\mathrm{pred}}$, for the conservative update of $(\rho h)^{(1)}$. We use Equation (18) to predict $(\rho h)' = (\rho h)^{(1)} - (\rho h)_0^n$ to time-centered edges, using $\mathbf{U} = \widetilde{\mathbf{U}}^{\mathrm{ADV},\star} + w_0^{n+1/2,\star}\mathbf{e}_r$, yielding $(\rho h)'^{n+1/2,\star,\mathrm{pred}}$. We convert the perturbational enthalpy to a full state enthalpy using

$$(\rho h)^{n+1/2,\star,\mathrm{pred}} = (\rho h)'^{n+1/2,\star,\mathrm{pred}} + \frac{(\rho h)_0^n + (\rho h)_0^{n+1,\star}}{2}. \tag{A40}$$

For planar geometry, we map $(\rho h)_0$ directly to Cartesian edges. In spherical geometry, our experience has shown that a slightly different approach leads to reduced discretization errors. We first map $h_0 \equiv (\rho h)_0/\rho_0$ and ρ_0 to Cartesian edges separately, and then multiply these terms to get $(\rho h)_0$.

ii. Evolve $(\rho h)^{(1)} \to (\rho h)^{(2),\star}$.
For planar geometry,

$$(\rho h)^{(2),\star} = (\rho h)^{(1)} - \Delta t \left\{ \nabla \cdot \left[(\widetilde{\mathbf{U}}^{\mathrm{ADV},\star} + w_0^{n+1/2,\star} \mathbf{e}_r)(\rho h)^{n+1/2,\star,\mathrm{pred}} \right] \right\} + \Delta t \left(\widetilde{\mathbf{U}}^{\mathrm{ADV},\star} \cdot \mathbf{e}_r \right) \left(\frac{\partial p_0}{\partial r} \right)^n + \Delta t \, \psi^{n+1/2,\star}.$$
(A41)

For spherical geometry,

$$(\rho h)^{(2),\star} = (\rho h)^{(1)} - \Delta t \left\{ \nabla \cdot \left[(\widetilde{\mathbf{U}}^{\mathrm{ADV},\star} + w_0^{n+1/2,\star} \mathbf{e}_r)(\rho h)^{n+1/2,\star,\mathrm{pred}} \right] \right\}$$
$$+ \Delta t \left\{ \nabla \cdot \left(\widetilde{\mathbf{U}}^{\mathrm{ADV},\star} p_0^n \right) - p_0^n \nabla \cdot \widetilde{\mathbf{U}}^{\mathrm{ADV},\star} \right\} + \Delta t \, \psi^{n+1/2,\star}.$$
(A42)

Then, for each Cartesian cell where $\rho^{(2),\star} < \rho_{\mathrm{cutoff}}$, we recompute enthalpy using

$$(\rho h)^{(2),\star} = \rho^{(2),\star} h\left(\rho^{(2),\star}, p_0^{n+1,\star}, X_k^{(2),\star} \right).$$
(A43)

I. Update the temperature using the equation of state: $T^{(2),\star} = T(\rho^{(2),\star}, h^{(2),\star}, X_k^{(2),\star})$ (planar geometry) or $T^{(2),\star} = T(\rho^{(2),\star}, p_0^{n+1,\star}, X_k^{(2),\star})$ (spherical geometry).

Step 5. *React the full state through a second time interval of $\Delta t/2$.*
    Call **React State**$\left[\rho^{(2),\star}, (\rho h)^{(2),\star}, X_k^{(2),\star}, T^{(2),\star}, (\rho H_{\mathrm{ext}})^{(2),\star}, p_0^{n+1,\star} \right] \to \left[\rho^{n+1,\star}, (\rho h)^{n+1,\star}, X_k^{n+1,\star}, T^{n+1,\star}, (\rho \dot{\omega}_k)^{(2),\star}, (\rho H_{\mathrm{nuc}})^{(2),\star} \right]$.

Step 6. *Compute the time-centered expansion, $S^{n+1/2,\star}$, base state velocity, $w_0^{n+1/2}$, and base state velocity forcing.*

A. Compute $S^{n+1/2,\star}$. First, compute $S^{n+1,\star}$ with

$$S^{n+1,\star} = -\sigma \sum_k \xi_k (\dot{\omega}_k)^{(2),\star} + \frac{1}{\rho^{n+1,\star} p_\rho} \sum_k p_{X_k} (\dot{\omega}_k)^{(2),\star} + \sigma H_{\mathrm{nuc}}^{(2),\star} + \sigma H_{\mathrm{ext}}^{(2),\star},$$
(A44)

where $(\dot{\omega}_k)^{(2),\star} = (\rho \dot{\omega}_k)^{(2),\star} / \rho^{(2),\star}$ and the thermodynamic quantities are defined using $\rho^{n+1,\star}$, $X_k^{n+1,\star}$, and $T^{n+1,\star}$ as inputs to the equation of state. Then, define

$$\overline{S^{n+1/2,\star}} = \mathbf{Avg}(S^{n+1/2,\star}), \qquad S^{n+1/2,\star} = \frac{S^n + S^{n+1,\star}}{2}.$$
(A45)

B. Compute $w_0^{n+1/2}$. First, define

$$\overline{\Gamma_1^{n+1/2,\star}} = \frac{\overline{\Gamma_1^n} + \overline{\Gamma_1^{n+1,\star}}}{2}, \qquad \rho_0^{n+1/2,\star} = \frac{\rho_0^n + \rho_0^{n+1,\star}}{2}, \qquad p_0^{n+1/2,\star} = \frac{p_0^n + p_0^{n+1,\star}}{2},$$
(A46)

with

$$\overline{\Gamma_1^{n+1,\star}} = \mathbf{Avg}\left[\Gamma_1 \left(\rho^{n+1,\star}, p_0^{n+1,\star}, X_k^{n+1,\star} \right) \right].$$
(A47)

For planar geometry, call
    Compute w_0 Planar$\left[\overline{S^{n+1/2,\star}}, \overline{\Gamma_1^{n+1/2,\star}}, p_0^{n+1/2,\star}, \psi^{n+1/2,\star} \right] \to \left[w_0^{n+1/2} \right]$.
For spherical geometry, call
    Compute w_0 Spherical$\left[\overline{S^{n+1/2,\star}}, \overline{\Gamma_1^{n+1/2,\star}}, \rho_0^{n+1/2,\star}, p_0^{n+1/2,\star}, \eta_\rho^{n+1/2,\star} \right] \to \left[w_0^{n+1/2} \right]$.

C. Compute the base state velocity forcing. Rearrange Equation (A22),

$$\left(\frac{1}{\rho_0} \frac{\partial \pi_0}{\partial r} \right)^n = -\frac{w_0^{n+1/2} - w_0^{n-1/2}}{1/2(\Delta t^n + \Delta t^{n-1})} - w_0^n \left(\frac{\partial w_0}{\partial r} \right)^n,$$
(A48)

where w_0^n and $(\partial w_0 / \partial r)^n$ are defined as

$$w_0^n = \frac{\Delta t^n w_0^{n-1/2} + \Delta t^{n-1} w_0^{n+1/2}}{\Delta t^n + \Delta t^{n-1}},$$
(A49)

$$\left(\frac{\partial w_0}{\partial r} \right)^n = \frac{1}{\Delta t^n + \Delta t^{n-1}} \left[\Delta t^n \left(\frac{\partial w_0}{\partial r} \right)^{n-1/2} + \Delta t^{n-1} \left(\frac{\partial w_0}{\partial r} \right)^{n+1/2} \right].$$
(A50)

If $n = 0$, we use $\Delta t^{-1} = \Delta t^0$.

Step 7. *Construct the time-centered advective velocity on edges, $\widetilde{\mathbf{U}}^{\mathrm{ADV}}$.*

The procedure to construct $\widetilde{\mathbf{U}}^{\mathrm{ADV},\dagger}$ is identical to the procedure for computing $\widetilde{\mathbf{U}}^{\mathrm{ADV},\dagger,\star}$ in **Step 3**, but uses the updated values $w_0^{n+1/2}$ and π_0^n rather than $w_0^{n+1/2,\star}$ and $\pi_0^{n,\star}$. We note that $\widetilde{\mathbf{U}}^{\mathrm{ADV}}$ satisfies the discrete version of $\overline{(\widetilde{\mathbf{U}}^{\mathrm{ADV}} \cdot \mathbf{e}_r)} = 0$ as well as

$$\nabla \cdot \left(\beta_0^{n+1/2,\star} \widetilde{\mathbf{U}}^{\mathrm{ADV}} \right) = \beta_0^{n+1/2,\star} \left(S^{n+1/2,\star} - \overline{S^{n+1/2,\star}} \right), \tag{A51}$$

$$\beta_0^{n+1/2,\star} = \frac{\beta_0^n + \beta_0^{n+1,\star}}{2}; \qquad \beta_0^{n+1,\star} = \beta_0 \left(\rho_0^{n+1,\star}, p_0^{n+1,\star}, \overline{\Gamma_1^{n+1,\star}} \right). \tag{A52}$$

Step 8. *Advect the base state and full state through a time interval of Δt.*

    A. Update ρ_0, saving the time-centered density at radial edges by calling

        Advect Base Density $\left[\rho_0^n, w_0^{n+1/2} \right] \to \left[\rho_0^{(2a)}, \rho_0^{n+1/2,\mathrm{pred}} \right]$.

    B. Update (ρX_k). This step is identical to **Step 4B** except we use the updated values $w_0^{n+1/2}$, $\widetilde{\mathbf{U}}^{\mathrm{ADV}}$, and $\rho_0^{(2a)}$ rather than $w_0^{n+1/2,\star}$, $\widetilde{\mathbf{U}}^{\mathrm{ADV},\star}$, and $\rho_0^{(2a),\star}$. In particular,

        i. Compute the time-centered species edge states, $(\rho X_k)^{n+1/2,\mathrm{pred}}$, for the conservative update of $(\rho X_k)^{(1)}$. We use equations (17) and (13) to predict $\rho'^{(1)} = \rho^{(1)} - \rho_0^n$ and $X_k^{(1)} = (\rho X_k)^{(1)}/\rho^{(1)}$ to time-centered edges with $\mathbf{U} = \widetilde{\mathbf{U}}^{\mathrm{ADV}} + w_0^{n+1/2}\mathbf{e}_r$, yielding $\rho'^{n+1/2,\mathrm{pred}}$ and $X_k^{n+1/2,\mathrm{pred}}$. We convert the perturbational density to a full state density using

$$\rho^{n+1/2,\mathrm{pred}} = \rho'^{n+1/2,\mathrm{pred}} + \frac{\rho_0^n + \rho_0^{(2a)}}{2}. \tag{A53}$$

        Then, $(\rho X_k)^{n+1/2,\mathrm{pred}} = (\rho^{n+1/2,\mathrm{pred}} X_k^{n+1/2,\mathrm{pred}})$.

        ii. Evolve $(\rho X_k)^{(1)} \to (\rho X_k)^{(2)}$ using

$$(\rho X_k)^{(2)} = (\rho X_k)^{(1)} - \Delta t \left\{ \nabla \cdot \left[\left(\widetilde{\mathbf{U}}^{\mathrm{ADV}} + w_0^{n+1/2}\mathbf{e}_r \right) (\rho X_k)^{n+1/2,\mathrm{pred}} \right] \right\}, \tag{A54}$$

$$\rho^{(2)} = \sum_k (\rho X_k)^{(2)}, \qquad X_k^{(2)} = (\rho X_k)^{(2)}/\rho^{(2)}. \tag{A55}$$

    C. Define a radial edge-centered $\eta_\rho^{n+1/2}$.

        For planar geometry,

$$\eta_\rho^{n+1/2} = \mathbf{Avg} \sum_k \left[\left(\widetilde{\mathbf{U}}^{\mathrm{ADV}} \cdot \mathbf{e}_r + w_0^{n+1/2} \right) (\rho X_k)^{n+1/2,\mathrm{pred}} \right] - w_0^{n+1/2} \rho_0^{n+1/2,\mathrm{pred}}. \tag{A56}$$

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

$$\eta_\rho^{\mathrm{cart},n+1/2} = \left[\left(\frac{\rho^{(1)} + \rho^{(2)}}{2} \right) - \left(\frac{\rho_0^n + \rho_0^{(2a)}}{2} \right) \right] \left(\widetilde{\mathbf{U}}^{\mathrm{ADV}} \cdot \mathbf{e}_r + w_0^{n+1/2} \right). \tag{A57}$$

        Then,

$$\eta_\rho^{n+1/2} = \mathbf{Avg} \left(\eta_\rho^{\mathrm{cart},n+1/2} \right). \tag{A58}$$

        This gives a radial cell-centered $\eta_\rho^{n+1/2}$. To get $\eta_\rho^{n+1/2}$ at radial edges, average the two neighboring cell-centered values.

    D. Correct ρ_0 by setting $\rho_0^{n+1} = \mathbf{Avg}(\rho^{(2)})$.

    E. Update p_0 by calling **Enforce HSE** $\left[p_0^n, \rho_0^{n+1} \right] \to \left[p_0^{n+1} \right]$.

    F. Compute $\psi^{n+1/2}$.

        For planar geometry,

$$\psi_j^{n+1/2} = \tfrac{1}{2} \left(\eta_{\rho,j-1/2}^{n+1/2} + \eta_{\rho,j+1/2}^{n+1/2} \right) g. \tag{A59}$$

        For spherical geometry, first compute

$$\overline{\Gamma_1^{(2)}} = \mathbf{Avg}\left[\Gamma_1 \left(\rho^{(2)}, p_0^{n+1}, X_k^{(2)} \right) \right]. \tag{A60}$$

    Then, define $\psi^{n+1/2}$ using Equation (A18):

$$\psi_j^{n+1/2} = \left(\frac{\overline{\Gamma_1^{(1)}} + \overline{\Gamma_1^{(2)}}}{2} \right)_j \left(\frac{p_0^n + p_0^{n+1}}{2} \right)_j \left\{ \overline{S_j^{n+1/2}} - \frac{1}{r_j^2} \left[\left(r^2 w_0^{n+1/2} \right)_{j+1/2} - \left(r^2 w_0^{n+1/2} \right)_{j-1/2} \right] \right\}. \tag{A61}$$

G. Update $(\rho h)_0$ by calling **Advect Base Enthalpy**$\left[(\rho h)_0^n, w_0^{n+1/2}, \psi^{n+1/2}\right] \rightarrow \left[(\rho h)_0^{n+1}\right]$.

H. Update the enthalpy. This step is identical to **Step 4H** except we use the updated values $w_0^{n+1/2}, \widetilde{\mathbf{U}}^{\mathrm{ADV}}, \rho_0^{n+1}, (\rho h)_0^{n+1}, p_0^{n+1/2}$, and $\psi^{n+1/2}$ rather than
$w_0^{n+1/2,\star}, \widetilde{\mathbf{U}}^{\mathrm{ADV},\star}, \rho_0^{n+1,\star}, (\rho h)_0^{n+1,\star}, p_0^n$, and $\psi^{n+1/2,\star}$. In particular,

    i. Compute the time-centered enthalpy edge state, $(\rho h)^{n+1/2,\mathrm{pred}}$, for the conservative update of $(\rho h)^{(1)}$. We use Equation (18) to predict $(\rho h)' = (\rho h)^{(1)} - (\rho h)_0^n$ to time-centered edges with $\mathbf{U} = \widetilde{\mathbf{U}}^{\mathrm{ADV}} + w_0^{n+1/2}\mathbf{e}_r$, yielding $(\rho h)'^{n+1/2,\mathrm{pred}}$. We convert the perturbational enthalpy to a full state enthalpy using

$$(\rho h)^{n+1/2,\mathrm{pred}} = (\rho h)'^{n+1/2,\mathrm{pred}} + \frac{(\rho h)_0^n + (\rho h)_0^{n+1}}{2}. \tag{A62}$$

    ii. Evolve $(\rho h)^{(1)} \rightarrow (\rho h)^{(2)}$.
For planar geometry,

$$(\rho h)^{(2)} = (\rho h)^{(1)} - \Delta t\left\{\nabla \cdot \left[\left(\widetilde{\mathbf{U}}^{\mathrm{ADV}} + w_0^{n+1/2}\mathbf{e}_r\right)(\rho h)^{n+1/2,\mathrm{pred}}\right]\right\} + \Delta t\left(\widetilde{\mathbf{U}}^{\mathrm{ADV}} \cdot \mathbf{e}_r\right)\left(\frac{\partial p_0}{\partial r}\right)^{n+1/2} + \Delta t\,\psi^{n+1/2}. \tag{A63}$$

For spherical geometry,

$$\begin{aligned}(\rho h)^{(2)} = (\rho h)^{(1)} &- \Delta t\left\{\nabla \cdot \left[\left(\widetilde{\mathbf{U}}^{\mathrm{ADV}} + w_0^{n+1/2}\mathbf{e}_r\right)(\rho h)^{n+1/2,\mathrm{pred}}\right]\right\} \\ &+ \Delta t\left[\nabla \cdot \left(\widetilde{\mathbf{U}}^{\mathrm{ADV}} p_0^{n+1/2}\right) - p_0^{n+1/2}\nabla \cdot \widetilde{\mathbf{U}}^{\mathrm{ADV}}\right] + \Delta t\,\psi^{n+1/2},\end{aligned} \tag{A64}$$

where $p_0^{n+1/2}$ is defined as $p_0^{n+1/2} = (p_0^n + p_0^{n+1})/2$.

Then, for each Cartesian cell where $\rho^{(2)} < \rho_{\mathrm{cutoff}}$, we recompute enthalpy using

$$(\rho h)^{(2)} = \rho^{(2)}h\left(\rho^{(2)}, p_0^{n+1}, X_k^{(2)}\right). \tag{A65}$$

I. Update the temperature using the equation of state: $T^{(2)} = T(\rho^{(2)}, h^{(2)}, X_k^{(2)})$ (planar geometry) or $T^{(2)} = T(\rho^{(2)}, p_0^{n+1}, X_k^{(2)})$ (spherical geometry).

Step 9. *React the full state through a second time interval of $\Delta t/2$.*
    Call **React State**$[\rho^{(2)}, (\rho h)^{(2)}, X_k^{(2)}, T^{(2)}, (\rho H_{\mathrm{ext}})^{(2)}, p_0^{n+1}] \rightarrow [\rho^{n+1}, (\rho h)^{n+1}, X_k^{n+1}, T^{n+1}, (\rho\dot\omega_k)^{(2)}, (\rho H_{\mathrm{nuc}})^{(2)}]$.

Step 10. *Define the new time expansion, S^{n+1}, and $\overline{\Gamma_1^{n+1}}$.*

A. Define

$$S^{n+1} = -\sigma\sum_k \xi_k(\dot\omega_k)^{(2)} + \sigma H_{\mathrm{nuc}}^{(2)} + \frac{1}{\rho^{n+1}p_\rho}\sum_k p_{X_k}(\dot\omega_k)^{(2)} + \sigma H_{\mathrm{ext}}^{(2)}, \tag{A66}$$

where $(\dot\omega_k)^{(2)} = (\rho\dot\omega_k)^{(2)}/\rho^{(2)}$ and the thermodynamic quantities are defined using ρ^{n+1}, X_k^{n+1}, and T^{n+1} as inputs to the equation of state. Then, compute

$$\overline{S^{n+1}} = \mathbf{Avg}(S^{n+1}). \tag{A67}$$

B. Define

$$\overline{\Gamma_1^{n+1}} = \mathbf{Avg}\left[\Gamma_1\left(\rho^{n+1}, p_0^{n+1}, X_k^{n+1}\right)\right]. \tag{A68}$$

Step 11. *Update the velocity.*
    First, we compute the time-centered edge velocities, $\widetilde{\mathbf{U}}^{n+1/2,\mathrm{pred}}$. Then, we define

$$\rho^{n+1/2} = \frac{\rho^n + \rho^{n+1}}{2}, \qquad \rho_0^{n+1/2} = \frac{\rho_0^n + \rho_0^{n+1}}{2}. \tag{A69}$$

We update the velocity field $\widetilde{\mathbf{U}}^n$ to $\widetilde{\mathbf{U}}^{n+1,\dagger}$ by discretizing Equation (10) as

$$\begin{aligned}\widetilde{\mathbf{U}}^{n+1,\dagger} = \widetilde{\mathbf{U}}^n &- \Delta t\left[\left(\widetilde{\mathbf{U}}^{\mathrm{ADV}} + w_0^{n+1/2}\mathbf{e}_r\right) \cdot \nabla\widetilde{\mathbf{U}}^{n+1/2,\mathrm{pred}}\right] - \Delta t\left(\widetilde{\mathbf{U}}^{\mathrm{ADV}} \cdot \mathbf{e}_r\right)\left(\frac{\partial w_0}{\partial r}\right)^{n+1/2}\mathbf{e}_r \\ &+ \Delta t\left[-\frac{1}{\rho^{n+1/2}}\mathbf{G}\pi^{n-1/2} + \left(\frac{1}{\rho_0}\frac{\partial \pi_0}{\partial r}\right)^n\mathbf{e}_r - \frac{\left(\rho^{n+1/2} - \rho_0^{n+1/2}\right)}{\rho^{n+1/2}}g^{n+1/2}\mathbf{e}_r\right],\end{aligned} \tag{A70}$$

where $\mathbf{G}$ approximates a cell-centered gradient from nodal data. Again, the $\dagger$ superscript refers to the fact that the updated velocity does not satisfy the divergence constraint.

Finally, we use an approximate nodal projection to define $\widetilde{\mathbf{U}}^{n+1}$ from $\widetilde{\mathbf{U}}^{n+1,\dagger}$, such that $\widetilde{\mathbf{U}}^{n+1}$ approximately satisfies

$$\nabla \cdot \left(\beta_0^{n+1/2}\widetilde{\mathbf{U}}^{n+1}\right) = \beta_0^{n+1/2}\left(S^{n+1} - \overline{S^{n+1}}\right), \tag{A71}$$

where $\beta_0^{n+1/2}$ is defined as

$$\beta_0^{n+1/2} = \frac{\beta_0^n + \beta_0^{n+1}}{2}; \qquad \beta_0^{n+1} = \beta\left(\rho_0^{n+1}, p_0^{n+1}, \overline{\Gamma_1^{n+1}}, g^{n+1}\right). \tag{A72}$$

As part of the projection we also define the new-time perturbational pressure, $\pi^{n+1/2}$. This projection necessarily differs from the MAC projection used in **Step 3** and **Step 7** because the velocities in those steps are defined on edges and $\widetilde{\mathbf{U}}^{n+1}$ is defined at cell centers, requiring different divergence and gradient operators. Details of the approximate projection are given in Paper III.

Step 12. *Compute a new Δt*.

Compute Δt for the next time step with the procedure described in Section 3.4 of Paper III using w_0 as computed in **Step 6** and $\widetilde{\mathbf{U}}^{n+1}$ as computed in **Step 11**.

This completes one step of the algorithm.

Figure 14 is a flow chart summarizing the 12 step algorithm, including the notation used as we advance the solution by Δt. Figure 15 is a flow chart of the advection steps (**Steps 4** and **8**), which includes the notation we use as we advect the solution through a time interval of Δt.

A.5. Numerical Cutoffs

As discussed in Paper IV, in order to prevent the velocity from becoming too large in low density regions far from the center of the star, we impose a cutoff at a moderately small density, ρ_{cutoff}, and hold the density at this constant value outside of the star. The cutoff affects the evolution in the following ways:

1. After advancing enthalpy in the advection step (**Steps 4Hii** and **8Hii** in Section A.4), we recompute the enthalpy using the equation of state if $\rho \leqslant \rho_{\text{cutoff}}$.
2. When computing gravity (Section A.2) we only add ρ_0 to m_{encl} if $\rho_0 > \rho_{\text{cutoff}}$ in order to prevent an unphysical amount of mass from contributing to the calculation.
3. When computing p_0 in **Enforce HSE** (Section A.3.3), we hold p_0 constant once $\rho_0 \leqslant \rho_{\text{cutoff}}$.
4. In **React State** (Section A.3.1), we set $\dot{\omega}_k = 0$ and $\rho H_{\text{nuc}} = 0$ if $\rho \leqslant \rho_{\text{cutoff}}$.
5. When computing ψ (**Steps 4F** and **8F** in Section A.4), we set $\psi = 0$ if $\rho_0 \leqslant \rho_{\text{cutoff}}$.
6. When we compute the velocity forcing in **Steps 3**, **7**, **11**, and **12**, we set the buoyancy term (the term proportional to $\rho - \rho_0$) to zero if $\rho < 5\rho_{\text{cutoff}}$.

Additionally, we use an anelastic cutoff density, $\rho_{\text{anelastic}}$, in the computation of β_0 (**Steps 3**, **7**, and **11** in Section A.4). When $\rho_{0,j} \leqslant \rho_{\text{anelastic}}$, we set $\beta_{0,j} = (\rho_{0,j}/\rho_{0,j-1})\beta_{0,j-1}$.

APPENDIX B

COMPUTING w_0 FOR SPHERICAL PROBLEMS

Recall that we want to solve

$$\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 w_0) = \overline{S} - \frac{1}{\overline{\Gamma_1}p_0}\underbrace{\left(\frac{\partial p_0}{\partial t} + w_0\frac{\partial p_0}{\partial r}\right)}_{\psi}. \tag{B1}$$

for the base state velocity, w_0. We first decompose w_0 by setting $w_0 = \overline{w_0} + \delta w_0$, where the $\overline{w_0}$ term is the contribution to w_0 due to the expansion term:

$$\frac{1}{r^2}\frac{\partial}{\partial r}(r^2\overline{w_0}) = \overline{S}^{\text{in}}. \tag{B2}$$

Then we can write an equation for the remaining term, δw_0:

$$\frac{1}{r^2}\frac{\partial}{\partial r}(r^2\delta w_0) = -\frac{1}{\overline{\Gamma_1}p_0}\left[\frac{\partial p_0}{\partial t} + (\overline{w_0} + \delta w_0)\frac{\partial p_0}{\partial r}\right]. \tag{B3}$$

Multiplying Equation (B3) through by $\overline{\Gamma_1}p_0$, taking another derivative with respect to r, and switching the order of temporal and spatial derivatives, we obtain

$$\frac{\partial}{\partial r}\left[\frac{\overline{\Gamma_1}p_0}{r^2}\frac{\partial}{\partial r}(r^2\delta w_0)\right] = -\frac{\partial}{\partial t}\frac{\partial p_0}{\partial r} - \frac{\partial}{\partial r}\left[(\overline{w_0} + \delta w_0)\frac{\partial p_0}{\partial r}\right]. \tag{B4}$$

To solve for δw_0 we will need to substitute for the derivatives of p_0. To do so we start with the hydrostatic equilibrium equation,

$$\frac{\partial p_0}{\partial r} = -\rho_0 g; \qquad g = \frac{Gm_{\mathrm{encl}}}{r^2}, \tag{B5}$$

where $m_{\mathrm{encl}}(r)$ is the mass enclosed at radius r and G is the gravitational constant. Using this, we can then write Equation (B4) as

$$\frac{\partial}{\partial r}\left[\frac{\overline{\Gamma_1} p_0}{r^2}\frac{\partial}{\partial r}(r^2 \delta w_0)\right] = \frac{\partial}{\partial t}(\rho_0 g) + \frac{\partial}{\partial r}(w_0 \rho_0 g)$$

$$= g\left[\frac{\partial \rho_0}{\partial t} + \frac{\partial}{\partial r}(w_0 \rho_0)\right] + \rho_0\left(\frac{\partial g}{\partial t} + w_0\frac{\partial g}{\partial r}\right). \tag{B6}$$

The mass enclosed inside any radius, r, is $m_{\mathrm{encl}}(r) = 4\pi\int_0^r \rho_0(s)s^2 ds$, or alternately, $\partial m_{\mathrm{encl}}/\partial r = 4\pi r^2 \rho_0$. The Lagrangian derivative of the enclosed mass is then

$$\frac{D_0 m_{\mathrm{encl}}}{Dt} = \frac{\partial m_{\mathrm{encl}}}{\partial t} + w_0\frac{\partial m_{\mathrm{encl}}}{\partial r}$$

$$= 4\pi\left(\frac{\partial}{\partial t}\int_0^r \rho_0(s)s^2 ds + w_0 r^2 \rho_0\right) = 4\pi\left(\int_0^r \frac{\partial \rho_0}{\partial t}s^2 ds + w_0 r^2 \rho_0\right)$$

$$= 4\pi\left\{-\int_0^r\left[\frac{1}{s^2}\frac{\partial(s^2\rho_0 w_0)}{\partial s} + \frac{1}{s^2}\frac{\partial(s^2\eta_\rho)}{\partial s}\right]s^2 ds + w_0 r^2 \rho_0\right\}$$

$$= 4\pi\left(-s^2\rho_0 w_0\big|_0^r - s^2\eta_\rho\big|_0^r + w_0 r^2 \rho_0\right) = -4\pi r^2 \eta_\rho, \tag{B7}$$

where we used the spherical form of Equation (29) in Paper III,

$$\frac{\partial \rho_0}{\partial t} + \frac{1}{r^2}\frac{\partial(r^2\rho_0 w_0)}{\partial r} + \frac{1}{r^2}\frac{\partial(r^2\eta_\rho)}{\partial r} = 0, \tag{B8}$$

to eliminate $\partial \rho_0/\partial t$. We note that in the absence of any mixing, $\eta_\rho = 0$, and $D_0 m_{\mathrm{encl}}/Dt = 0$. Equation (B7) allows us to write the Lagrangian change in the gravitational acceleration as

$$\frac{D_0 g}{Dt} = \frac{\partial g}{\partial t} + w_0\frac{\partial g}{\partial r} = \frac{D_0}{Dt}\left(\frac{Gm_{\mathrm{encl}}}{r^2}\right) = Gm_{\mathrm{encl}}\frac{D_0}{Dt}\left(\frac{1}{r^2}\right) + \frac{G}{r^2}\frac{D_0 m_{\mathrm{encl}}}{Dt}$$

$$= -\frac{2w_0 Gm_{\mathrm{encl}}}{r^3} - 4\pi G\eta_\rho = -\frac{2w_0 g}{r} - 4\pi G\eta_\rho. \tag{B9}$$

Putting it all together, Equation (B6) becomes

$$\frac{\partial}{\partial r}\left[\frac{\overline{\Gamma_1} p_0}{r^2}\frac{\partial}{\partial r}(r^2\delta w_0)\right] = g\left[\frac{\partial \rho_0}{\partial t} + \frac{\partial}{\partial r}(w_0\rho_0)\right] + \rho_0\left(\frac{-2w_0 g}{r} - 4\pi G\eta_\rho\right). \tag{B10}$$

Finally, we can use Equation (B8) to write Equation (B10) as

$$\frac{\partial}{\partial r}\left[\frac{\overline{\Gamma_1} p_0}{r^2}\frac{\partial}{\partial r}(r^2\delta w_0)\right] = g\left[-\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 w_0\rho_0) - \frac{1}{r^2}\frac{\partial}{\partial r}(r^2\eta_\rho) + \frac{\partial}{\partial r}(w_0\rho_0)\right] + \rho_0\left(\frac{-2w_0 g}{r} - 4\pi G\eta_\rho\right)$$

$$= -\frac{g}{r^2}\frac{\partial(r^2\eta_\rho)}{\partial r} - \frac{4(\overline{w_0} + \delta w_0)\rho_0 g}{r} - 4\pi G\rho_0\eta_\rho. \tag{B11}$$

We discretize this elliptic equation in the radial dimension as

$$\frac{1}{\Delta r}\left\{\left[\frac{\overline{\Gamma_1} p_0}{r^2}\frac{\partial(r^2\delta w_0)}{\partial r}\right]_j - \left[\frac{\overline{\Gamma_1} p_0}{r^2}\frac{\partial(r^2\delta w_0)}{\partial r}\right]_{j-1}\right\} + \left[\frac{4(r^2\delta w_0)\rho_0 g}{r^3}\right]_{j-1/2} = -\frac{g_{j-1/2}}{r_{j-1/2}^2\Delta r}\left[(r^2\eta_\rho)_j - (r^2\eta_\rho)_{j-1}\right]$$

$$-\left(\frac{4\overline{w_0}\rho_0 g}{r}\right)_{j-1/2} - \left(4\pi G\rho_0\eta_\rho\right)_{j-1/2}, \tag{B12}$$

where we choose to solve for $(r^2\delta w_0)$ rather than δw_0 so that we can easily enforce $\partial(r^2\delta w_0)/\partial r = 0$ at the upper boundary. Then, using hydrostatic equilibrium, we expand this to

$$\frac{1}{\Delta r}\left\{\left(\frac{\overline{\Gamma_1}p_0}{r^2}\right)_j \frac{[(r^2\delta w_0)_{j+1/2} - (r^2\delta w_0)_{j-1/2}]}{\Delta r} - \left(\frac{\overline{\Gamma_1}p_0}{r^2}\right)_{j-1} \frac{[(r^2\delta w_0)_{j-1/2} - (r^2\delta w_0)_{j-3/2}]}{\Delta r}\right\}$$

$$-\left(\frac{4}{r_{j-1/2}^3}\frac{p_{0,j} - p_{0,j-1}}{\Delta r}\right)(r^2\delta w_0)_{j-1/2} = \left(\frac{4}{r_{j-1/2}^3}\frac{p_{0,j} - p_{0,j-1}}{\Delta r}\right)(r^2\overline{w_0})_{j-1/2}$$

$$-\frac{g_{j-1/2}}{r_{j-1/2}^2\Delta r}\left[(r^2\eta_\rho)_j - (r^2\eta_\rho)_{j-1}\right] - (4\pi G\rho_0\eta_\rho)_{j-1/2}. \tag{B13}$$

If we write this in matrix form, so that

$$A_j(r^2\delta w_0)_{j-3/2} + B_j(r^2\delta w_0)_{j-1/2} + C_j(r^2\delta w_0)_{j+1/2} = F_j, \tag{B14}$$

then

$$A_j = \frac{1}{\Delta r^2}\left(\frac{\overline{\Gamma_1}^{\text{in}}p_0^{\text{in}}}{r^2}\right)_{j-1}, \tag{B15}$$

$$B_j = -\frac{1}{\Delta r^2}\left[\left(\frac{\overline{\Gamma_1}^{\text{in}}p_0^{\text{in}}}{r^2}\right)_j + \left(\frac{\overline{\Gamma_1}^{\text{in}}p_0^{\text{in}}}{r^2}\right)_{j-1}\right] - \left(\frac{4}{r_{j-1/2}^3}\frac{p_{0,j}^{\text{in}} - p_{0,j-1}^{\text{in}}}{\Delta r}\right), \tag{B16}$$

$$C_j = \frac{1}{\Delta r^2}\left(\frac{\overline{\Gamma_1}^{\text{in}}p_0^{\text{in}}}{r^2}\right)_j, \tag{B17}$$

$$F_j = \left(\frac{4}{r_{j-1/2}^3}\frac{p_{0,j}^{\text{in}} - p_{0,j-1}^{\text{in}}}{\Delta r}\right)(r^2\overline{w_0})_{j-1/2} - \frac{g_{j-1/2}^{\text{in}}}{r_{j-1/2}^2\Delta r}\left[(r^2\eta_\rho^{\text{in}})_j - (r^2\eta_\rho^{\text{in}})_{j-1}\right] - (4\pi G\rho_0^{\text{in}}\eta_\rho^{\text{in}})_{j-1/2}. \tag{B18}$$

We define the lower boundary condition, $\delta w_0 = 0$ at $r = 0$, which corresponds to $j = 0$, by setting

$$A_0 = C_0 = F_0 = 0; \quad B_0 = 1. \tag{B19}$$

We also specify $\partial(r^2\delta w_0)/\partial r = 0$ at the upper boundary, which corresponds to the location where ρ_0 falls below ρ_{cutoff}, by setting

$$A_N = -1; \quad B_N = 1; \quad C_N = F_N = 0. \tag{B20}$$

Finally, $w_0^{\text{out}} = \overline{w_0} + \delta w_0$. Once ρ_0 falls below ρ_{cutoff}, we hold $r^2 w_0^{\text{out}}$ constant.

APPENDIX C

TEST PROBLEM INITIAL MODEL

We use the same general initial model for the convergence test (Section 6.3), adaptive bubble rise test (Section 6.4) and forced convection test (Section 6.5). We define a base temperature, $T_{\text{base}} = 6 \times 10^8$ K, and density, $\rho_{\text{base}} = 2.6 \times 10^9$ g cm^{-3}, at some height r_{base} above the bottom of the domain (r_{base} varies in each problem, and can be equal to zero). This defines a base entropy, $s_{\text{base}} = s(\rho_{\text{base}}, T_{\text{base}})$. The gravitational acceleration, $g = -1.5 \times 10^{10}$ cm s^{-2}, is constant. The composition is uniform everywhere with $X(^{12}\text{C}) = 0.3$ and $X(^{16}\text{O}) = 0.7$. For $r > r_{\text{base}}$, we integrate the equation of hydrostatic equilibrium along with the constraint that entropy is constant:

$$p_{0,j+1} = p_{0,j} + \tfrac{1}{2}\Delta r(\rho_{0,j} + \rho_{0,j+1})g. \tag{C1}$$
$$s_{0,j+1} = s_{\text{base}} \tag{C2}$$

upward from r_{base}. We define a temperature cutoff, $T_{\text{cutoff}} = 10^7$ K, and when $T_{0,j+1} < T_{\text{cutoff}}$, we replace Equation (C2) with $T_{0,j+1} = T_{\text{cutoff}}$. If we choose $r_{\text{base}} > 0$, then we must also define the model for $0 < r < r_{\text{base}}$. In this case, we define a desired linear entropy profile with a discontinuous jump from s_{base} as

$$s_{\text{want}} = \frac{1}{3}s_{\text{base}} + \frac{r - r_{\text{base}}}{r_{\text{base}}}\frac{s_{\text{base}}}{12}. \tag{C3}$$

This entropy profile creates a convectively stable layer below the atmosphere to prevent any motions generated from heating above from interfering with the lower boundary. The initial model for this region is then computed by integrating

$$p_{0,j} = p_{0,j+1} - \tfrac{1}{2}\Delta r(\rho_{0,j} + \rho_{0,j+1})g, \qquad (C4)$$

$$s_{0,j} = s_{\text{want}}(r_j), \qquad (C5)$$

downward from $r = r_{\text{base}}$.

REFERENCES

Almgren, A. S., Bell, J. B., Colella, P., Howell, L. H., & Welcome, M. 1998, J. Comput. Phys., 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
Almgren, A. S., Bell, J. B., & Zingale, M. 2007, J. Phys. Conf. Ser., 78, 2085
Almgren, A. S., et al. 2010, ApJS, in press
Bell, J. B., Berger, M. J., Saltzman, J. S., & Welcome, M. 1994, SIAM J. Sci. Stat. Comput., 15, 127
Bell, J. B., Day, M. S., Rendleman, C. A., Woosley, S. E., & Zingale, M. A. 2004, J. Comput. Phys., 195, 677
Berger, M. J., & Colella, P. 1989, J. Comput. Phys., 82, 64
Berger, M. J., & Rigoutsos, J. 1991, IEEE Trans. Syst. Man Cybern., 21, 1278
Chamulak, D. A., Brown, E. F., Timmes, F. X., & Dupczak, K. 2008, ApJ, 677, 160
Colella, P., & Woodward, P. R. 1984, J. Comput. Phys., 54, 174
Day, M. S., & Bell, J. B. 2000, Combust. Theor. Model., 4, 535
Durran, D. 1990, Meteorol. Monogr., 23, 59
Fryxell, B., et al. 2000, ApJS, 131, 273
Glasner, S. A., Livne, E., & Truran, J. W. 2007, ApJ, 665, 1321
Höflich, P., & Stein, J. 2002, ApJ, 568, 779
Kuhlen, M., Woosley, S. E., & Glatzmaier, G. A. 2006, ApJ, 640, 407
Lin, D. J., Bayliss, A., & Taam, R. E. 2006, ApJ, 653, 545
Meakin, C. A., & Arnett, D. 2007, ApJ, 665, 690
Miller, G. H., & Colella, P. 2002, J. Comput. Phys., 183, 26
Pember, R. B., Howell, L. H., Bell, J. B., Colella, P., Crutchfield, W. Y., Fiveland, W. A., & Jessee, J. P. 1998, Combust. Sci. Technol., 140, 123
Saltzman, J. 1994, J. Comput. Phys., 115, 153
Strohmayer, T., & Bildsten, L. 2006, in Compact Stellar X-Ray Sources, ed. W. H. G. Lewin & M. van der Klis (Cambridge: Cambridge Univ. Press), 113
Timmes, F. X., & Swesty, F. D. 2000, ApJS, 126, 501
Woosley, S. E., Wunsch, S., & Kuhlen, M. 2004, ApJ, 607, 921
Zingale, M., Almgren, A. S., Bell, J. B., Nonaka, A., & Woosley, S. E. 2009, ApJ, 704, 196