The Astrophysical Journal, 936:6 (17pp), 2022 September 1 https://doi.org/10.3847/1538-4357/ac8478

An Improved Method for Coupling Hydrodynamics with Astrophysical Reaction Networks

M. Zingale[1], M. P. Katz[1], A. Nonaka[2], and M. Rasmussen[3]
[1] Dept. of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11794-3800, USA; michael.zingale@stonybrook.edu
[2] Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[3] Utah State University, Logan, UT 84322, USA
Received 2022 June 1; revised 2022 July 22; accepted 2022 July 25; published 2022 August 25

Abstract

Reacting astrophysical flows can be challenging to model, because of the difficulty in accurately coupling hydrodynamics and reactions. This can be particularly acute during explosive burning or at high temperatures where nuclear statistical equilibrium is established. We develop a new approach, based on the ideas of spectral deferred corrections (SDC) coupling of explicit hydrodynamics and stiff reaction sources as an alternative to operator splitting, that is simpler than the more comprehensive SDC approach we demonstrated previously. We apply the new method to a double-detonation problem with a moderately sized astrophysical nuclear reaction network and explore the time step size and reaction network tolerances, to show that the simplified-SDC approach provides improved coupling with decreased computational expense compared to traditional Strang operator splitting. This is all done in the framework of the Castro hydrodynamics code, and all algorithm implementations are freely available.

Unified Astronomy Thesaurus concepts: Nucleosynthesis (1131); Computational methods (1965); Hydrodynamical simulations (767)

1. Introduction

Modeling astrophysical reacting flows can be challenging because of the disparity between the nuclear and hydrodynamics timescales. Reaction networks tend to be stiff, requiring implicit integration techniques to stably integrate the system (Byrne & Hindmarsh 1987). In contrast, compressible hydrodynamics flows are limited by the (often much longer) sound-crossing time over a computational cell, and can be solved using explicit time integration. Traditional methods of coupling hydrodynamics and reactions used in astrophysics use operator splitting—each physical process acts on the output of the previous process in alternating fashion. This makes it easy to use different time-integration methods for the different physics, and to build a simulation code in a modular way. However, competition between the different physical processes can cause the coupling to break down, since the reactions do not directly incorporate the effects of the hydrodynamics and vice versa. These splitting errors can lead to loss of accuracy and further time step limitations.

A particularly difficult phase of evolution to model is the nuclear statistical equilibrium that sets in for temperatures in excess of few $\times 10^9$ K. Physically, the forward and reverse rates of reaction should balance, leading to an equilibrium. With operator splitting, an NSE region will have a large positive flow through the network in a zone in one step, followed by a large negative flow over the next time step, as the code struggles to produce an equilibrium. These large changes in abundances (and large alternately positive and negative energy generation rates) can be a challenge for the integration method—it may take more steps than allowed by the ODE integrator, require a time step below floating point accuracy, or fail to meet the tolerances. The easiest way to improve the coupling is to cut the time step (see, e.g., Couch et al. 2015; Rivas et al. 2022), but this can make simulations prohibitively expensive. Sometimes the burning is simply halted on a zone-by-zone basis when NSE conditions are reached (e.g., as in Zingale et al. 2001). The present work focuses on networks alone, but in a follow-on paper we will explore hybrid burning models consisting of networks and tables for nuclear statistical equilibrium.

The Castro hydrodynamics code (Almgren et al. 2010, 2020) is used for all of our numerical experiments. Castro is a compressible (magneto-, radiation-) hydrodynamics code built on the AMReX adaptive mesh refinement (AMR) framework (Zhang et al. 2019). Castro has been designed to be performance-portable and runs on massively parallel CPU, multicore, and GPU architectures (Katz et al. 2020). For hydrodynamics, the corner transport upwind (CTU; Colella 1990) method with the piecewise parabolic method (PPM; Colella & Woodward 1984; Miller & Colella 2002) is used. Castro includes self-gravity, rotation, arbitrary equations of state, and reaction networks, and has been used for modeling X-ray bursts and different models of thermonuclear, core-collapse, and pair-instability supernovae. Recently, in Zingale et al. (2019), we developed second- and fourth-order accurate in space and time method-of-lines approaches for coupling hydrodynamics and nuclear reaction networks based on spectral deferred corrections (SDC), and demonstrated these methods using a variety of test problems.

The time-integration approach presented here is considerably simpler than the SDC method of Zingale et al. (2019), but allows us to reuse the piecewise-linear or piecewise-parabolic CTU hydrodynamics construction (Saltzman 1994; Miller & Colella 2002) used in the the original Castro paper, as well as a largely similar ODE integration scheme, making this method easier to add to existing simulation codes. Furthermore, it also extends to adaptive mesh refinement with subcycling in a

straightforward manner, avoiding the complications described in McCorquodale & Colella (2011) needed to fill ghost cells when using method-of-lines integration. However, it is restricted to second-order accuracy in time overall. We term this algorithm the "simplified-SDC method." In this paper, we describe the overall method and demonstrate it on a test problem and astrophysical simulation using Castro. All of the code to reproduce the results in this paper is freely available in the Castro GitHub repository.[4]

2. Numerical Methodology

We solve the Euler equations for compressible, reacting flow. For ease of exposition, we describe the one-dimensional case; multidimensional extensions are a straightforward modification to include in the CTU hydrodynamics scheme. Our conserved variables are

$$\boldsymbol{\mathcal{U}} = \begin{pmatrix} \rho \\ \rho X_k \\ \rho u \\ \rho E \\ \rho e \end{pmatrix}, \tag{1}$$

where ρ is the mass density, u is velocity, E is specific total energy, p is the pressure, and we carry nuclear species mass fractions, X_k. The specific total energy relates to the specific internal energy, e, as $E = e + u^2/2$, and we also separately evolve e as part of a dual-energy formulation (see Bryan et al. 1995; Katz et al. 2016). The mass fractions are constrained to sum to 1, $\sum_k X_k = 1$. Defining the hydrodynamical fluxes,

$$\boldsymbol{F}(\boldsymbol{\mathcal{U}}) = \begin{pmatrix} \rho u \\ \rho X_k u \\ \rho u^2 p \\ (\rho E + p) u \\ \rho e u \end{pmatrix}, \tag{2}$$

we can write the system in conservative form for all state variables aside from (ρe) as

$$\frac{\partial \boldsymbol{\mathcal{U}}}{\partial t} + \frac{\partial \boldsymbol{F}(\boldsymbol{\mathcal{U}})}{\partial x} = \boldsymbol{S}(\boldsymbol{\mathcal{U}}), \tag{3}$$

where for the special case of (ρe), we have an additional "$p\,dV$" term:

$$\frac{\partial (\rho e)}{\partial t} + \frac{\partial F_{\rho e}}{\partial x} + p \frac{\partial u}{\partial x} = S_{\rho e}. \tag{4}$$

We split the source term into gravitational and reactive parts, $\boldsymbol{S}(\boldsymbol{\mathcal{U}}) = \boldsymbol{G}(\boldsymbol{\mathcal{U}}) + \boldsymbol{R}(\boldsymbol{\mathcal{U}})$, with

$$\boldsymbol{G}(\boldsymbol{\mathcal{U}}) = \begin{pmatrix} 0 \\ 0 \\ \rho g \\ \rho u g \\ 0 \end{pmatrix}, \quad \boldsymbol{R}(\boldsymbol{\mathcal{U}}) = \begin{pmatrix} 0 \\ \rho \dot{\omega}_{X_k} \\ 0 \\ \rho \dot{S} \\ \rho \dot{S} \end{pmatrix}. \tag{5}$$

From reactions, $\dot{\omega}_{X_k}$ is the creation rate for species k and $\dot{S}$ is the energy generation rate per unit mass. We note that the internal energy $p \partial u/\partial x$ ("$p\,dV$") work is not treated as a source term, but is instead constructed with the hydrodynamical fluxes that are computed in the CTU method. Aside from the "$p\,dV$" term in the internal energy equation, this system is in conservative form with source terms.

We define the advective terms with gravitational sources, $\boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{U}})$, as

$$\boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{U}}) = -\frac{\partial \boldsymbol{F}(\boldsymbol{\mathcal{U}})}{\partial x} + \boldsymbol{G}(\boldsymbol{\mathcal{U}}) \tag{6}$$

for the general case, with the (ρe) component again having the extra "$p\,dV$" term:

$$\mathcal{A}_{\rho e} = -\frac{\partial F_{\rho e}}{\partial x} - p \frac{\partial u}{\partial x} + G_{\rho e}. \tag{7}$$

To close the system, we need an equation of state of the form:

$$p = p(\rho, \boldsymbol{X}, e). \tag{8}$$

Sometimes, it is preferable to work with the primitive variables:

$$\boldsymbol{q} = \begin{pmatrix} \rho \\ X_k \\ u \\ p \\ (\rho e) \end{pmatrix}. \tag{9}$$

Here, the system appears as

$$\boldsymbol{q}_t + \boldsymbol{A}^{(x)}(\boldsymbol{q}) \boldsymbol{q}_x = \boldsymbol{S}(\boldsymbol{q}), \tag{10}$$

with the matrix $\boldsymbol{A}^{(x)}$ giving the coefficients of the spatial derivatives of the primitive variables:

$$\boldsymbol{A}^{(x)}(\boldsymbol{q}) = \begin{pmatrix} u & 0 & \rho & 0 & 0 \\ 0 & u & 0 & 0 & 0 \\ 0 & 0 & u & 1/\rho & 0 \\ 0 & 0 & \Gamma_1 p & u & 0 \\ 0 & 0 & \rho h & 0 & u \end{pmatrix}, \tag{11}$$

where h is the specific enthalpy and Γ_1 is an adiabatic index, $\Gamma_1 = d \log p / d \log \rho|_s$, at constant entropy. The CTU+PPM algorithm uses the characteristic wave structure of $\boldsymbol{A}$ to collect the information that makes it to an interface over a time step in order to compute the fluxes through the interface. Note that the primitive state has two thermodynamic quantities, p and (ρe), to more efficiently handle the general equation of state in the Riemann solver, as described in Almgren et al. (2010), but alternate formulations are possible (Colella & Glaz 1985). The source term vector, $\boldsymbol{S}(\boldsymbol{q})$, can again be decomposed into gravitational sources (now in terms of the primitive variables) and reaction terms,

$$\boldsymbol{S}(\boldsymbol{q}) = \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{R}(\boldsymbol{q}), \tag{12}$$

with

$$\boldsymbol{G}(\boldsymbol{q}) = \begin{pmatrix} 0 \\ 0 \\ g \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{R}(\boldsymbol{q}) = \begin{pmatrix} 0 \\ \dot{\omega}_{X_k} \\ 0 \\ \Gamma_1 p \sigma \dot{S} \\ \rho \dot{S} \end{pmatrix}, \tag{13}$$

[4] https://github.com/amrex-astro/Castro/

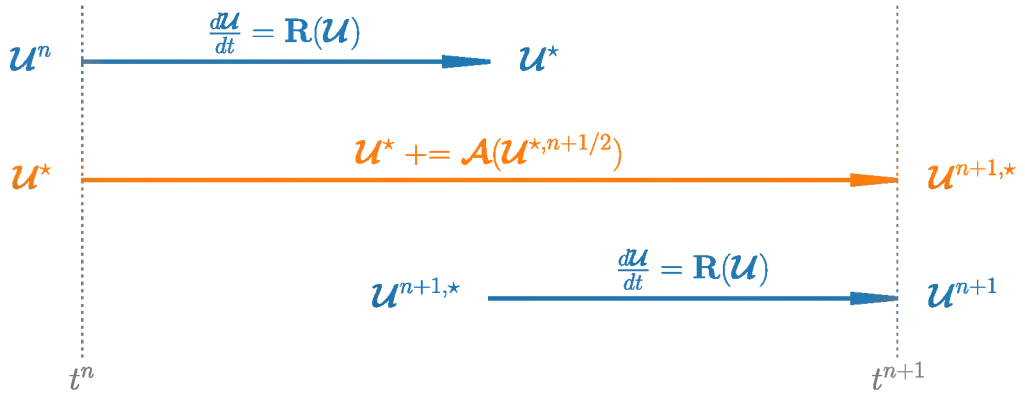

Figure 1. Flowchart of the operator split (Strang) time integration. We first react for $\Delta t/2$, then advect for Δt, and finally react for $\Delta t/2$. Each process uses the state left behind from the previous process.

where

$$\sigma \equiv \frac{\partial p/\partial T|_\rho}{\rho c_p \partial p/\partial \rho|_T} \tag{14}$$

and c_p is the specific heat at constant pressure, $c_p = \partial h/\partial T|_p$. A derivation of this source for the pressure equation can be found in Almgren et al. (2008). We note that this source is algebraically identical to that shown in Equation (25) of Almgren et al. (2010).

The CTU+PPM method for hydrodynamics is second-order accurate in space and time. We want to couple the reactive sources to the hydrodynamics to be second-order in time as well. As discussed above, nuclear reaction sources are stiff, and need to be integrated using implicit methods for stability. Operator splitting (e.g., Strang) is traditionally employed here, and is used as a benchmark for comparison in this paper. We discuss this traditional approach next, before moving on to our new time-coupling method.

2.1. Strang Splitting

In the Strang splitting (Strang 1968) flavor of operator splitting, we first integrate the system with reactions terms only (no advection) over $\Delta t/2$, then integrate the advection terms only (no reactions) over Δt, and finally integrate the reaction terms only over $\Delta t/2$. The staggering of the reactive update means the hydro implicitly sees a time-centered reactive source, making the update second-order accurate in time. This flow is illustrated in Figure 1.

In the absence of advective terms, our reaction system appears as just $d\mathcal{U}/dt = \boldsymbol{R}(\mathcal{U})$, or

$$\frac{d\rho}{dt} = 0 \tag{15}$$

$$\frac{d(\rho X_k)}{dt} = \rho \dot{\omega}_{X_k} \tag{16}$$

$$\frac{d(\rho u)}{dt} = 0 \tag{17}$$

$$\frac{d(\rho E)}{dt} = \rho \dot{S}. \tag{18}$$

Notice that, in the Strang formulation, density is held constant when reacting. We can write the energy equation as

$$\frac{d(\rho E)}{dt} = \frac{d(\rho e)}{dt} + \frac{dK}{dt} = \rho \dot{S}, \tag{19}$$

where K is the kinetic energy, $K = \rho|u|^2/2$. Since the density and velocity are unchanged by reactions (when Strang splitting), our energy equation becomes

$$\frac{d(\rho e)}{dt} = \rho \frac{de}{dt} = \rho \dot{S}. \tag{20}$$

The reaction rates are typically expressed as $\dot{\omega}_k(\rho, T, \boldsymbol{X})$ when we evolve this system, which requires us to get the temperature from the equation of state each time we need to evaluate the reactive terms.

We also typically integrate mass fractions, instead of partial densities:

$$\frac{dX_k}{dt} = \dot{\omega}_k. \tag{21}$$

We integrate Equations (20) and (21) using an implicit ODE solver described below.

We explored this and other approaches (including not evolving an energy/temperature equation during the reaction step) in Zingale et al. (2021). In that work, we showed that the above formulation achieved second-order convergence, and in particular, that integrating an energy/temperature equation is needed to get second-order accuracy (see also Müller 1986 for a discussion on stability). We note, however, that some astrophysical simulation codes only evolve the species equations. For very strong reactions, when using Strang splitting the state can drift significantly off of the smooth solution to the coupled reactive hydrodynamics equations, as shown graphically in Zingale et al. (2019) (using an earlier version of the present algorithm).

2.2. Time Step Limiters and Retry Mechanism

Since this method is based off of the CTU hydrodynamics scheme, it benefits from the larger time step that method can take (when done with full corner coupling, the advective CFL condition is unity) as compared to a method-of-lines approach; see Colella (1990). In addition to the standard CFL time step limiter for explicit hydrodynamics, time step limiters based on the energy generation or abundance changes over a time step,

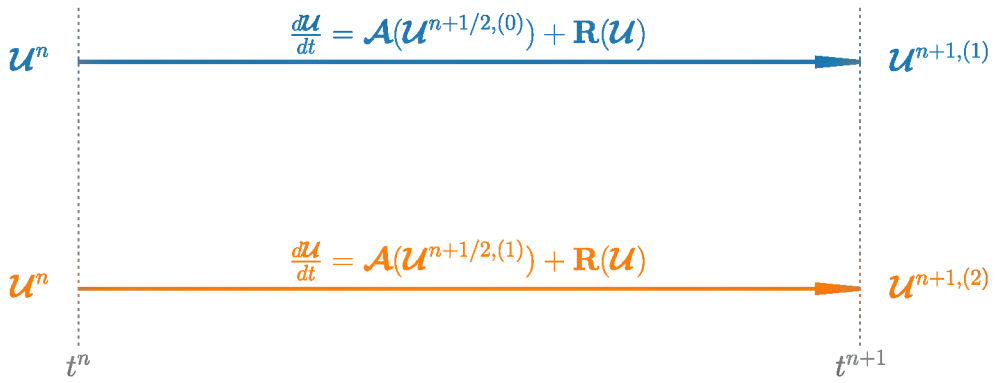

Figure 2. Flowchart of the simplified-SDC approach. An iteratively lagged advection term is used to update the system to the new time, with each iteration improving the coupling.

such as those introduced in Fryxell et al. (1989), are often used. An energy limiter takes the form of either

$$\Delta t \leqslant f_e \min_i \left\{ \frac{e_i^n}{|\dot{S}_i^n|} \right\} \tag{22}$$

or

$$\Delta t \leqslant f_e \min_i \left\{ \frac{e_i^n}{\left| \int \dot{S}_i(t)\, dt \right|} \right\}, \tag{23}$$

where i is the zone index and f_e is a parameter used to control the allowed change (e.g., $f_e = 0.5$ will limit the step such that reactions can only increase the internal energy by 50%). The two formulations differ in that Equation (22) uses the instantaneous evaluation of the energy generation rate at the start of the time step, while Equation (23) uses the integral of the energy generation rate over the entire step (or last Strang half). Usually, these limiters are used in a reactive manner—the next step will see a smaller time step based on what happened during the current step, but the current step is still accepted even if it violated these conditions. Our goal is to avoid the need for these limiters, by improving the coupling of hydrodynamics and reactions.

Castro has the ability to reject a time step, if it detects a failure, and retry with smaller time steps (subcycling to make up the original required time step). Among the conditions that can trigger this are density falling below zero during advection, the ODE integration failing to converge in the implicit solve (due to too many steps or the internal time step falling below a minimum), or violation of one of the time step limiters during the step. This means that the time step constraints are proactive instead of reactive. The retry mechanism in Castro works with both the Strang and simplified-SDC integration scheme. Retrying the step is much more stringent than simply adjusting the next step, so we put a cap on the number of retries allowed, aborting if we exceed the cap.

2.3. Spectral Deferred Corrections

The basic idea of SDC (Dutt et al. 2000) is to divide each time step into substeps defined by high-order quadrature points (e.g., Gauss–Lobatto) and iteratively correct the solution at each temporal node in order to reduce the integration error. Each correction equation can be formulated using low-order integration techniques, such as forward- or backward-Euler, with additional terms on each right-hand side arising from integrating the residual over the substep from the previous iteration using high-order temporal integration. Each iteration over all substeps increases the overall order of accuracy of the method by one, up to the underlying order of accuracy of the quadrature over the entire time step. Over the years, a number of SDC approaches targeting tighter multiphysics coupling have been developed, such as Bourlioux et al. (2003) and Nonaka et al. (2012). In these approaches, different physical processes can be treated with different approaches (such as forward-Euler for advection and backward-Euler for reactions), and furthermore, different physical processes can be substepped between temporal nodes.

The approach that we implemented in Zingale et al. (2019) is based off the multi-implicit method-of-lines approach in Bourlioux et al. (2003). In contrast to those works, the focus of this paper is not higher-order integration, but tighter coupling between advection and reactions in a second-order framework. Thus, our simplified-SDC approach is based on the method described in Nonaka et al. (2012) and a similar (but unpublished) implementation in the MAESTROeX simulation code (Fan et al. 2019). The key point in this approach is that we are reusing the same numerical kernels for advection and reactions that are used in a Strang splitting approach, but we are able to include, rather than exclude, the effects of the other physical processes in each correction equation solve.

We begin with the general form of a multi-implicit SDC correction equation with two physical processes. Using (k) to denote the iterate, we have

$$\begin{aligned} \mathcal{U}^{(k)}(t) = \mathcal{U}^n &+ \int_{t^n}^{t} [\mathcal{A}(\mathcal{U}^{(k)}) + \boldsymbol{R}(\mathcal{U}^{(k)}) \\ &- \mathcal{A}(\mathcal{U}^{(k-1)}) - \boldsymbol{R}(\mathcal{U}^{(k-1)})]\, dt \\ &+ \int_{t^n}^{t} [\mathcal{A}(\mathcal{U}^{(k-1)}) + \boldsymbol{R}(\mathcal{U}^{(k-1)})]\, dt. \end{aligned} \tag{24}$$

In the spirit of Nonaka et al. (2012), we treat each advection term as piecewise constant over the time step, and equal to a time-centered advection term evaluated with the Godunov integrator. We note that this is a departure from the approach in Bourlioux et al. (2003), which uses a method-of-lines approach where the advection terms are computed at each node, rather than the midpoint between nodes, using spatial extrapolation

only. Using $\mathcal{A}(\mathcal{U}^{n+1/2})$ to denote the advection term, we have

$$\mathcal{U}^{(k)}(t) = \mathcal{U}^n + \int_{t^n}^{t} [\mathcal{A}(\mathcal{U}^{n+1/2,(k)}) + \boldsymbol{R}(\mathcal{U}^{(k)}) - \mathcal{A}(\mathcal{U}^{n+1/2,(k-1)}) - \boldsymbol{R}(\mathcal{U}^{(k-1)})]dt + \int_{t^n}^{t} [\mathcal{A}(\mathcal{U}^{n+1/2,(k-1)}) + \boldsymbol{R}(\mathcal{U}^{(k-1)})]dt. \quad (25)$$

Also following Nonaka et al. (2012), we differentiate this equation in time:

$$\frac{\partial \mathcal{U}^{(k)}}{\partial t} = [\mathcal{A}(\mathcal{U}^{n+1/2,(k)}) + \boldsymbol{R}(\mathcal{U}^{(k)}) - \mathcal{A}(\mathcal{U}^{n+1/2,(k-1)}) - \boldsymbol{R}(\mathcal{U}^{(k-1)})] + [\mathcal{A}(\mathcal{U}^{n+1/2,(k-1)}) + \boldsymbol{R}(\mathcal{U}^{(k-1)})]. \quad (26)$$

We combine terms, and arrive at a system of ODEs at each cell:

$$\frac{\partial \mathcal{U}^{(k)}}{\partial t} = [\mathcal{A}(\mathcal{U}^{n+1/2,(k)}) + \boldsymbol{R}(\mathcal{U}^{(k)})]. \quad (27)$$

We integrate this system from t^n to t^{n+1} using a modified version of the VODE (Brown et al. 1989) integrator, as described in Section 2.5. Thus, we are effectively using two temporal nodes (t^n and t^{n+1}) and the overall integration is second-order accurate. In order to properly couple reactions into the computation of the advection terms, we use an iteratively lagged reaction source term, denoted $\mathcal{I}_q$ (this is basically an approximation of $\boldsymbol{R}(\boldsymbol{q})$). The overall flow of the simplified-SDC algorithm is shown in Figure 2.

2.4. Time Advancement Scheme

The basic time update algorithm proceeds as follows. We begin with the state at t^n, denoted $\mathcal{U}^n$, and proceed as follows:

1. *Iterate*—Iterate from $k = 1,\ldots,k_{\max}$. For second-order accuracy, $k_{\max} = 2$ is sufficient; further SDC iterations will continue to decrease the splitting error, but they will not increase the formal order of accuracy of the method. In addition to denoting the time level with a superscript (e.g., n or $n+1$), we use a second superscript in parentheses to keep track of the iteration. A single iteration, k, starts with $\mathcal{U}^n$ and results in the new time-level state for that iteration, $\mathcal{U}^{n+1,(k)}$.
   (a) *Step 1: Create the advective update term,* $[\mathcal{A}(\mathcal{U})]^{n+1/2,(k)}$
      i. Convert $\mathcal{U} \rightarrow \boldsymbol{q}$. This is an algebraic transformation that utilizes the equation of state.
      ii. Predict $\boldsymbol{q}$ to the interfaces at $t^{n+1/2}$ using the CTU PPM method—this involves taking the cell-centered primitive state and tracking all of the information that can reach the interface over a time step, following the procedure from Colella (1990) and Miller & Colella (2002). The source terms, $\boldsymbol{S}_q$, used in the prediction are

$$\boldsymbol{S}(\boldsymbol{q}) = \boldsymbol{G}(\boldsymbol{q}) + \mathcal{I}_q^{n+1/2,(k-1)}. \quad (28)$$

$\mathcal{I}_q^{n+1/2,(k-1)}$ is a numerical approximation for the effect of reactions from the previous iterate, which is computed at the end of each iterate as described below. The use of this term contrasts with Strang splitting, where no reactive source terms are included in the hydrodynamics update. If $k = 1$, we instead use the value from the last iteration of the previous time step, $\mathcal{I}_q^{n-1/2,(k_{\max})}$.

In the unsplit CTU method (Colella 1990), the interface states used for the final Riemann problem through the zone interface consist of a normal predictor and a transverse flux correction. We can add the source terms either to the normal predictor (for example, doing characteristic tracing as described in Colella & Woodward 1984) or after all of the transverse flux corrections are made. Both are second-order accurate. For the gravitational sources, we do those in the normal predictor, consistent with the formulation in Colella & Woodward (1984). However, for the reactive sources, we found that it is most reliable to add them at the end of the interface state construction, after the transverse flux corrections. This is because we want to ensure that the sum over species of $\mathcal{I}_q$ is zero, and characteristic tracing or the various flux corrections in general do not preserve this. We enforce that the species interface states remain in [0, 1] after adding the reactive source.

      iii. Solve the Riemann problem at each interface to get a unique conserved state on each interface, $\mathcal{U}_{i+1/2}^{n+1/2,(k)}$.
      iv. Construct the advective update terms, $[\tilde{\mathcal{A}}(\mathcal{U})]_i^{n+1/2,(k)}$, first without the gravitational

Table 1
Convergence (L_1 Norm) for the Reacting Convergence Problem Using Strang Splitting

Field	$\epsilon_{64\rightarrow128}$	Rate	$\epsilon_{128\rightarrow256}$	Rate	$\epsilon_{256\rightarrow512}$
ρ	2.794×10^{18}	2.044	6.777×10^{17}	2.554	1.154×10^{17}
ρu	6.796×10^{26}	2.448	1.245×10^{26}	2.889	1.681×10^{25}
ρv	6.796×10^{26}	2.448	1.245×10^{26}	2.889	1.681×10^{25}
ρE	2.451×10^{35}	2.351	4.803×10^{34}	2.742	7.179×10^{33}
ρe	2.261×10^{35}	2.320	4.526×10^{34}	2.821	6.403×10^{33}
T	2.237×10^{21}	1.691	6.927×10^{20}	2.482	1.240×10^{20}
$\rho X(^4\mathrm{He})$	2.878×10^{18}	2.020	7.096×10^{17}	2.529	1.229×10^{17}
$\rho X(^{12}\mathrm{C})$	1.698×10^{17}	1.950	4.393×10^{16}	2.232	9.353×10^{15}
$\rho X(^{16}\mathrm{O})$	1.687×10^{14}	1.660	5.338×10^{13}	1.957	1.375×10^{13}
$\rho X(^{56}\mathrm{Fe})$	2.794×10^{8}	2.044	6.777×10^{7}	2.554	1.154×10^{7}

Table 2
Convergence (L_1 Norm) for the Reacting Convergence Problem Using SDC Integration (Two Iterations)

Field	$\epsilon_{64\rightarrow128}$	Rate	$\epsilon_{128\rightarrow256}$	Rate	$\epsilon_{256\rightarrow512}$
ρ	2.784×10^{18}	2.048	6.734×10^{17}	2.558	1.143×10^{17}
ρu	6.779×10^{26}	2.447	1.243×10^{26}	2.895	1.671×10^{25}
ρv	6.779×10^{26}	2.447	1.243×10^{26}	2.895	1.671×10^{25}
ρE	2.450×10^{35}	2.353	4.795×10^{34}	2.737	7.193×10^{33}
ρe	2.256×10^{35}	2.320	4.518×10^{34}	2.817	6.414×10^{33}
T	2.232×10^{21}	1.695	6.893×10^{20}	2.484	1.233×10^{20}
$\rho X(^4\mathrm{He})$	2.866×10^{18}	2.024	7.049×10^{17}	2.534	1.217×10^{17}
$\rho X(^{12}\mathrm{C})$	1.695×10^{17}	1.959	4.357×10^{16}	2.236	9.250×10^{15}
$\rho X(^{16}\mathrm{O})$	1.681×10^{14}	1.662	5.313×10^{13}	1.963	1.363×10^{13}
$\rho X(^{56}\mathrm{Fe})$	2.784×10^{8}	2.048	6.734×10^{7}	2.558	1.143×10^{7}

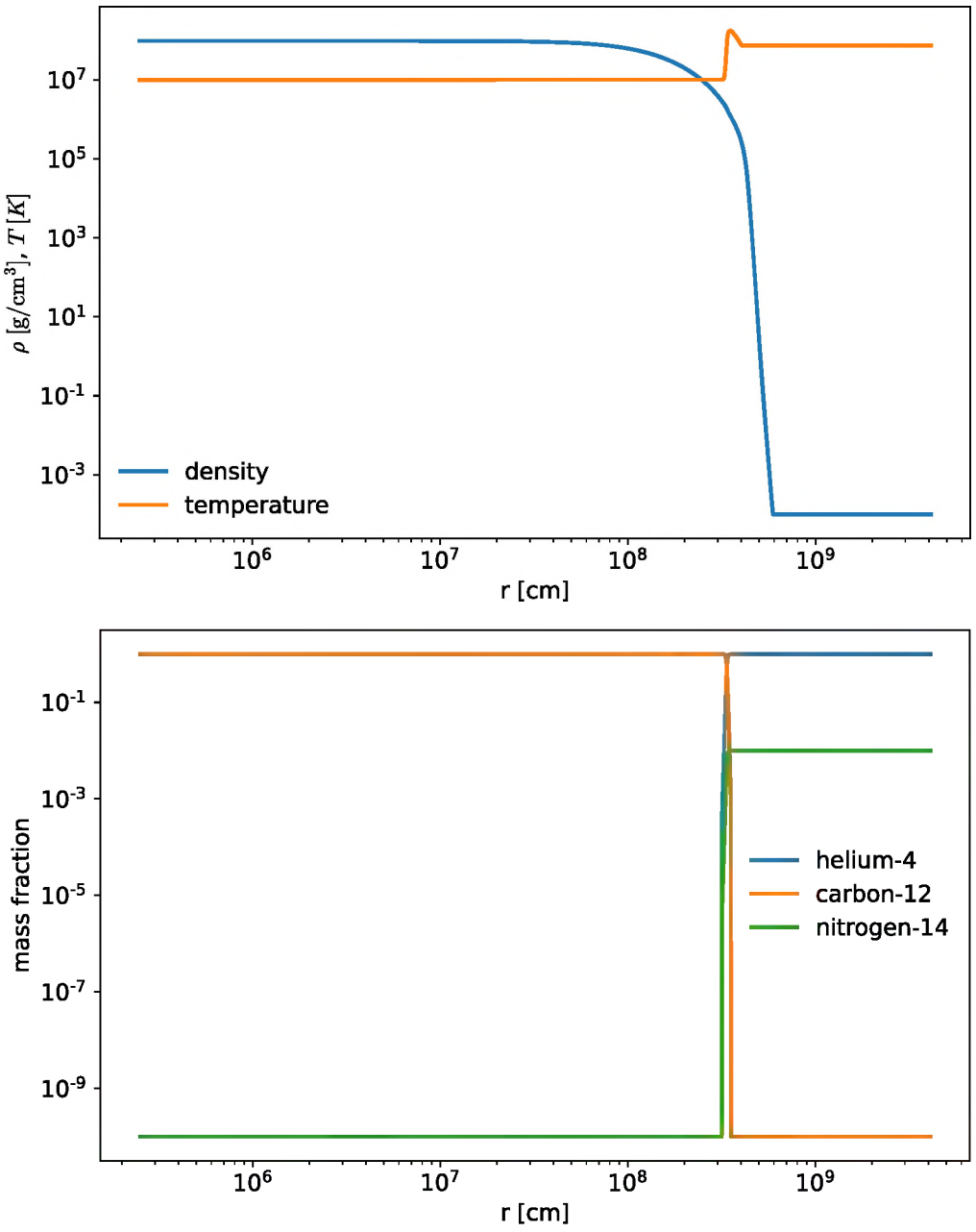

Figure 3. Initial model for the double-detonation test problem.

sources,

$$[\tilde{\mathcal{A}}(\mathcal{U})]_i^{n+1/2,(k)} = -\frac{\boldsymbol{F}^{(x)}(\mathcal{U}_{i+1/2}^{n+1/2,(k)}) - \boldsymbol{F}^{(x)}(\mathcal{U}_{i-1/2}^{n+1/2,(k)})}{\Delta x}, \tag{29}$$

for the general case, and

$$(\tilde{\mathcal{A}}_{\rho e}^{n+1/2,(k)})_i = -\frac{(F_{\rho e}^{n+1/2,(k)})_{i+1/2} - (F_{\rho e}^{n+1/2,(k)})_{i-1/2}}{\Delta x} - \left(\frac{p_{i+1/2}^{n+1/2,(k)} + p_{i-1/2}^{n+1/2,(k)}}{2}\right)\left(\frac{u_{i+1/2}^{n+1/2,(k)} - u_{i-1/2}^{n+1/2,(k)}}{\Delta x}\right) \tag{30}$$

for (ρe).

Now the gravitational source terms, $\boldsymbol{G}(\mathcal{U})$, are computed by first updating to the new state with advection and the old-time source term applied for the full Δt as

$$\mathcal{U}^{\star\star} = \mathcal{U}^n + \Delta t[\tilde{\mathcal{A}}(\mathcal{U})]^{n+1/2,(k)} + \Delta t\boldsymbol{G}(\mathcal{U}^n). \tag{31}$$

We then evaluate the source terms with $\mathcal{U}^{\star\star}$ and correct the advective term so that we have a time-centered source. The final advective update term is then[5]

$$[\mathcal{A}(\mathcal{U})]_i^{n+1/2,(k)} = [\tilde{\mathcal{A}}(\mathcal{U})]_i^{n+1/2,(k)} + \frac{\Delta t}{2}[\boldsymbol{G}(\mathcal{U}^n) + \boldsymbol{G}(\mathcal{U}^{\star\star})]. \tag{32}$$

Formally, per Nonaka et al. (2012), an alternative strategy

[5] For a source like gravity, this update can be done first for ρ and then the new momentum source can be defined using ρ, and likewise for energy.

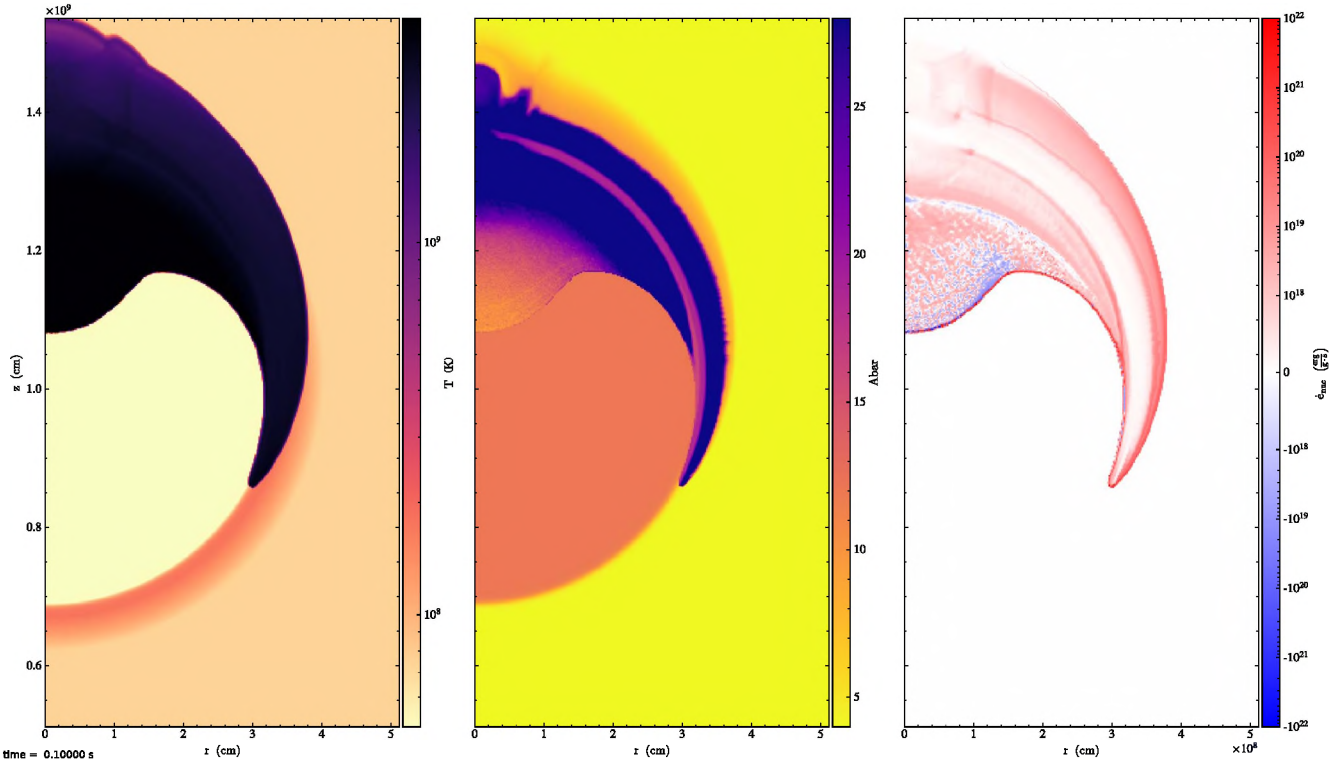

Figure 4. Temperature, mean molecular weight, and energy generation rate for the Strang CFL = 0.2 run with normal tolerances (`strang_subch2_cfl0.2`) at 0.1 s.

would be to skip Equation (31) and instead perform only Equation (32) but with $\boldsymbol{G}(\boldsymbol{\mathcal{U}}^{n+1,(k-1)})$ replacing $\boldsymbol{G}(\boldsymbol{\mathcal{U}}^{\star\star})$. In the future, we will explore the efficacy of both approaches.

(b) *Step 2: Integrate Using VODE*

We update the state by integrating Equation (27) over the full time step. Since we are approximating $\boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{U}})$ as piecewise constant in time, we can use an ODE integrator to integrate this, just as we do with the reaction system in Strang splitting. The difference here is that we are integrating the conserved variables and the state sees the effect of advection and gravity as we integrate the reactions. So, rather than use $d\boldsymbol{\mathcal{U}}/dt = \boldsymbol{R}(\boldsymbol{\mathcal{U}})$ as in the Strang case, the ODE form we use is

$$\frac{d\boldsymbol{\mathcal{U}}}{dt} = [\boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{U}})]^{n+1/2,(k)} + \boldsymbol{R}(\boldsymbol{\mathcal{U}}). \tag{33}$$

The details of the VODE (Brown et al. 1989) integrator we use are described in Section 2.5.

Looking at the form of Equation (5), we see that only the species and energies have nonzero terms in $\boldsymbol{R}$. The total and internal energies both provide the same information, and integrating both overconstrains the system, so we just integrate the internal energy. We define the subset of variables that are directly integrated as

$$\boldsymbol{\mathcal{U}}' = \begin{pmatrix} \rho X_k \\ \rho e \end{pmatrix}, \tag{34}$$

and we integrate

$$\frac{d\boldsymbol{\mathcal{U}}'}{dt} = [\boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{U}}')]^{n+1/2,(k)} + \boldsymbol{R}(\boldsymbol{\mathcal{U}}'). \tag{35}$$

This integration begins with $\boldsymbol{\mathcal{U}}'^{,n}$ and results in $\boldsymbol{\mathcal{U}}'^{,n+1,(k)}$.

We will need the density at times during the integration, which we construct as

$$\rho(t) = \rho^n + \mathcal{A}_\rho^{n+1/2,(k)}(t - t^n). \tag{36}$$

As we are integrating this system, we need to get the temperature, T, for the rate evaluations. We obtain this directly from internal energy, composition, and density using the equation of state.

Our integrator also needs the Jacobian of the system, in terms of $\boldsymbol{\mathcal{U}}'$. This is different than the form of the Jacobian usually used in reaction networks (we depend on e instead of T). We describe the form of the Jacobian in Appendix A.

At the end of the integration, we can do the conservative update of momentum and energy. Momentum is straightforward, since there are no reactive sources:

$$(\rho u)^{n+1,(k)} = (\rho u)^n + \Delta t \mathcal{A}_{\rho u}^{n+1/2,(k)}. \tag{37}$$

For total energy, we first need to isolate the reactive source for energy:

$$(\rho \dot{S})^{n+1/2,(k)} = \frac{(\rho e)^{n+1,(k)} - (\rho e)^n}{\Delta t} - \mathcal{A}_{\rho e}^{n+1/2,(k)}. \tag{38}$$

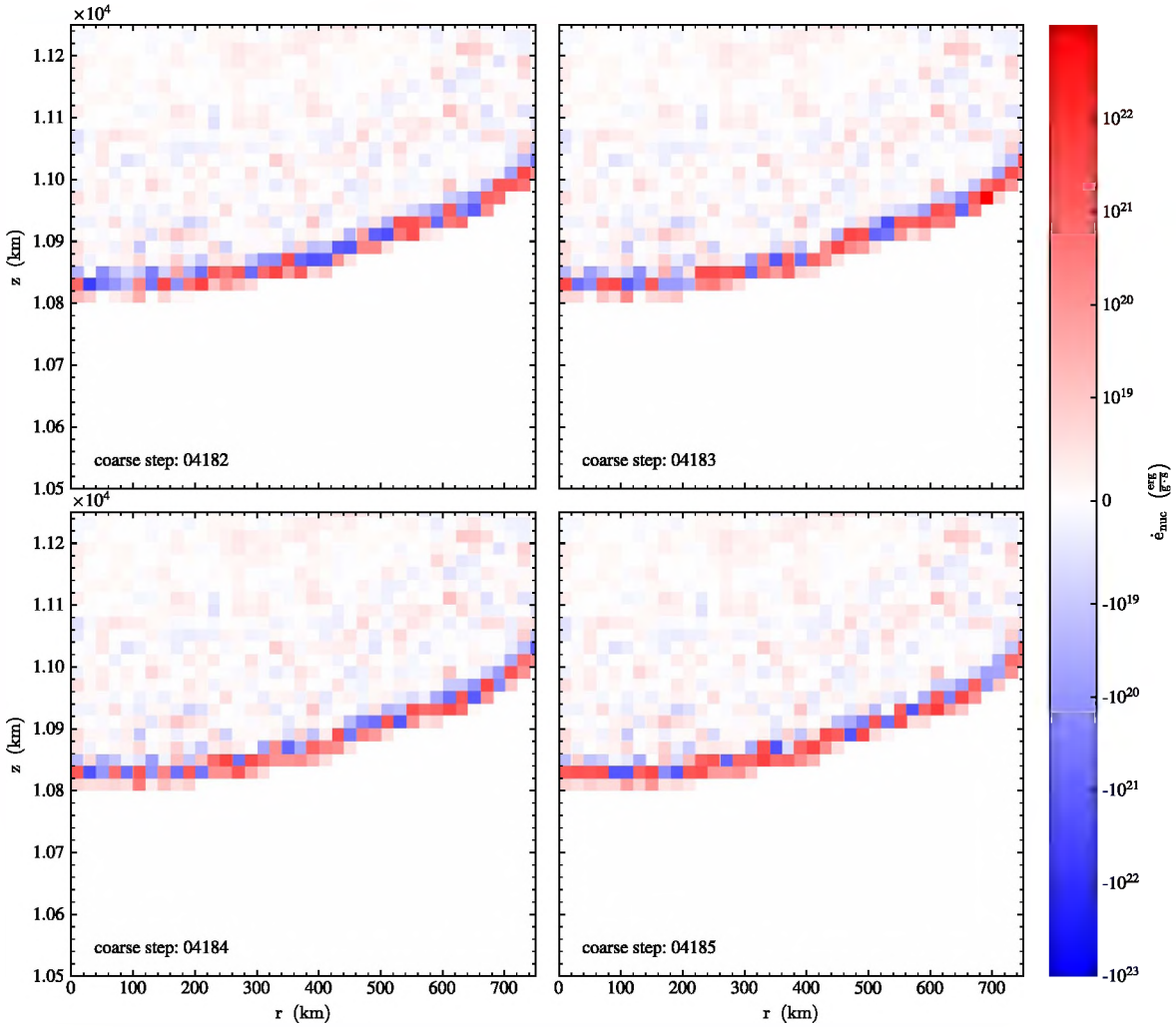

Figure 5. Energy generation rate for the Strang CFL = 0.2 run with normal tolerances (`strang_subch2_cfl0.2`) shown at four consecutive time steps around $t = 0.1$ s, zoomed-in at the inward-propagating detonation.

Then the update is

$$(\rho E)^{n+1,(k)} = (\rho E)^n + \Delta t \mathcal{A}_{\rho E}^{n+1/2,(k)} + \Delta t (\rho \dot{S})^{n+1/2,(k)}. \quad (39)$$

(c) *Step 3: Compute the Reactive Source Terms, $\mathcal{I}^{n+1/2,(k)}$*

We now seek the $\mathcal{I}$s that capture the effect of just the reaction sources on the state variables for the next iteration. For the conserved quantities, these would simply be

$$\mathcal{I}_{\mathcal{U}}^{n+1/2,(k)} = \frac{\mathcal{U}^{n+1,(k)} - \mathcal{U}^n}{\Delta t} - [\mathcal{A}(\mathcal{U})]^{n+1/2,(k)}. \quad (40)$$

However, for our primitive variables, which are used in the interface state prediction, we need to construct the required source terms more carefully. We want

$$\mathcal{I}_q^{n+1/2,(k)} = \frac{q^{n+1,(k)} - q^n}{\Delta t} - [\mathcal{A}(q)]^{n+1/2,(k)}, \quad (41)$$

but we need the advective update for q, which we have not constructed. We note that $\mathcal{I}_q^{n+1/2,(k)}$ is an approximation to the integral of Equation (13) over the time step. Additionally, we cannot simply use the equation of state on $\mathcal{I}_{\mathcal{U}}^{n+1/2,(k)}$, because this is a time derivative and does not represent a well-defined state in itself. Instead, we derive $\mathcal{I}_q^{n+1/2,(k)}$ via a multistep process. We first find the conservative state as if it were updated only with advection:

$$\mathcal{U}^\star = \mathcal{U}^n + \Delta t [\mathcal{A}(\mathcal{U})]^{n+1/2,(k)}, \quad (42)$$

and then construct the corresponding primitive variable state via an algebraic transform, $\mathcal{U}^\star \rightarrow q^\star$. This allows us to define the advective update for q as

$$[\mathcal{A}(q)]^{n+1/2,(k)} = \frac{q^\star - q^n}{\Delta t}. \quad (43)$$

Defining the primitive state corresponding to the fully updated conserved state via an algebraic transform, $\mathcal{U}^{n+1,(k)} \rightarrow q^{n+1,(k)}$, we can construct $\mathcal{I}_q^{n+1/2,(k)}$ by combining Equations (41) and (43):

$$\mathcal{I}_q^{n+1/2,(k)} = \frac{q^{n+1,(k)} - q^\star}{\Delta t}. \quad (44)$$

This completes the update of a single iteration.

2.5. Stiff ODE Integrator

We use a modified version of the VODE (Brown et al. 1989) integrator, designed for stiff ODEs. Our version has been

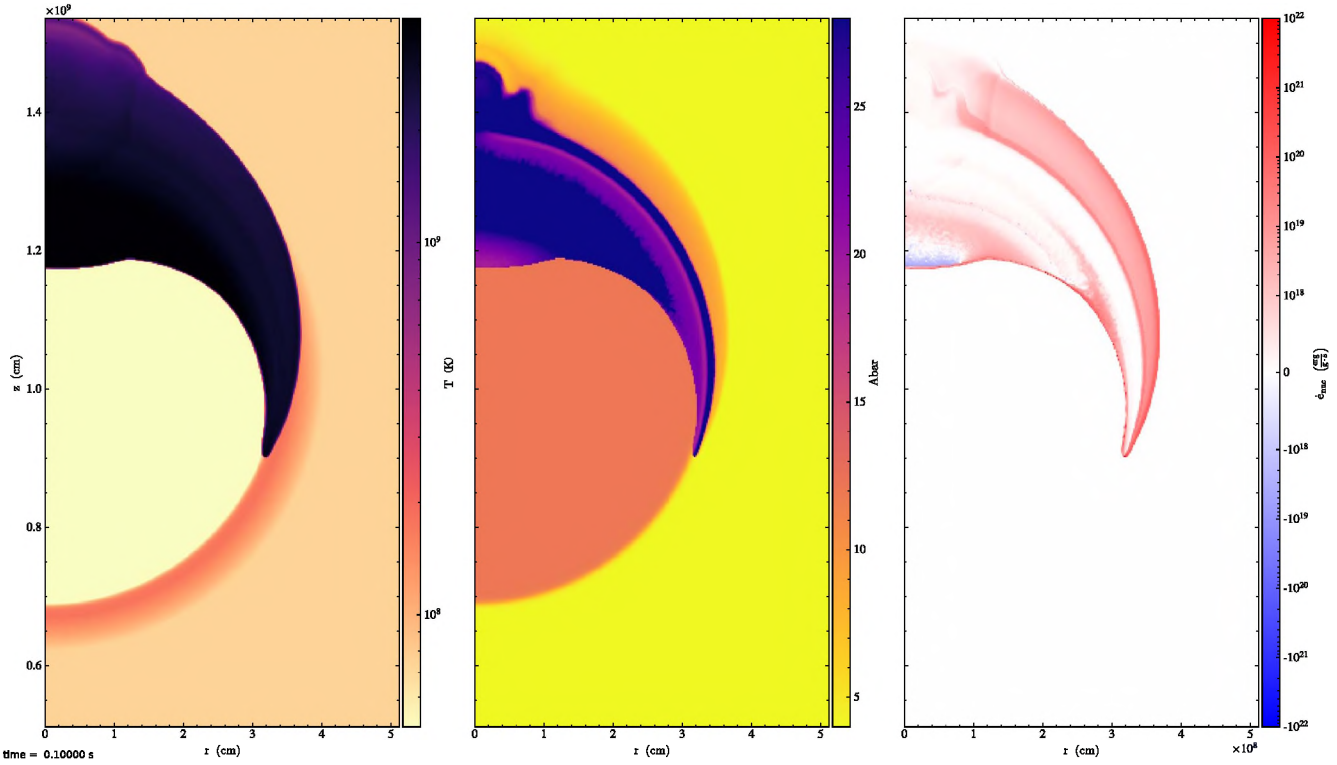

Figure 6. Temperature, mean molecular weight, and energy generation rate for the simplified-SDC CFL = 0.2 run with normal tolerances (`sdc_subch2_cfl0.2`) at 0.1 s.

ported to modern C++ and designed to run on GPUs. We have also added checks to the time step rejection logic that help prevent VODE from wandering too far off the solution.[6] We have found empirically that these checks improve the performance of VODE.

VODE allows us to specify both relative and absolute tolerances for the species and energy during the integration. They are combined into a weight that VODE uses to assess convergence of the step,

$$w_i = \epsilon_{\rm rel}|y_i| + \epsilon_{\rm abs}, \tag{45}$$

for integration variable y_i. We will denote the species tolerances as $\epsilon_{\rm rel,spec}$ and $\epsilon_{\rm abs,spec}$, and the energy tolerances as $\epsilon_{\rm rel,ener}$ and $\epsilon_{\rm abs,ener}$.

For Strang splitting, we require that an individual mass fraction not change too much over a step:

$$\begin{aligned} &|X_k^{m+1}| < f|X_k^m| \\ &|X_k^{m+1}| > \frac{1}{f}|X_k^m| \ \text{if } |X_k^m| \\ &\quad > \eta\epsilon_{\rm abs,spec} \text{ and } |X_k^{m+1}| > \eta\epsilon_{\rm abs,spec}. \end{aligned} \tag{46}$$

Here, m is the current VODE solution and $m+1$ is the potential new-time solution. The parameter f is the factor a mass fraction is allowed to change per VODE step. We use $f=4$ for the results here. The parameter η allows us to only use this condition for species with mass fractions above a threshold. We use $\eta=1$ by default, but $\eta=f$ sometimes helps prevent VODE from worrying about species that dip below the threshold during integration, since VODE uses a norm over all of the integrated variables to compute the total error. If these conditions are violated, then VODE will reject the time step and retry with a smaller step. We also require that the mass fractions are contained in [0, 1] to a tolerance of 10^{-2}. For SDC, we enforce these same constraints, but for Equation (46), we only use the change from m to $m+1$, due to reactions, by subtracting off the advective contribution over that substep. Finally, we enforce that

$$\rho^{m+1} > 0 \tag{47}$$

$$(\rho e)^{m+1} > 0. \tag{48}$$

Some of the rates in Cyburt et al. (2010) increase by hundreds of orders of magnitude at low temperature, presumably because the interpolant fit was done at higher temperatures. Since VODE can take exploratory right-hand side evaluations outside of the nominal time range, it can encounter these poorly behaved rates, and this can cause the integration to fail (by generating `infs`). To prevent this behavior, we set a lower temperature limit for the reaction rates, below which we zero them out. For the simulations in Section 3.2, we choose a cutoff of 5×10^7 K.

For both Strang splitting and simplified-SDC, we use a numerical Jacobian that VODE computes internally (following the algorithm in Radhakrishnan & Hindmarsh 1993). For completeness, we derive the analytic form of the Jacobian in Appendix A for the simplified-SDC method. VODE employs Jacobian-caching, so the Jacobian does not need to be re-evaluated each time it is used.

[6] This modified version of VODE is available in our Microphysics repo: https://github.com/AMReX-Astro/Microphysics.

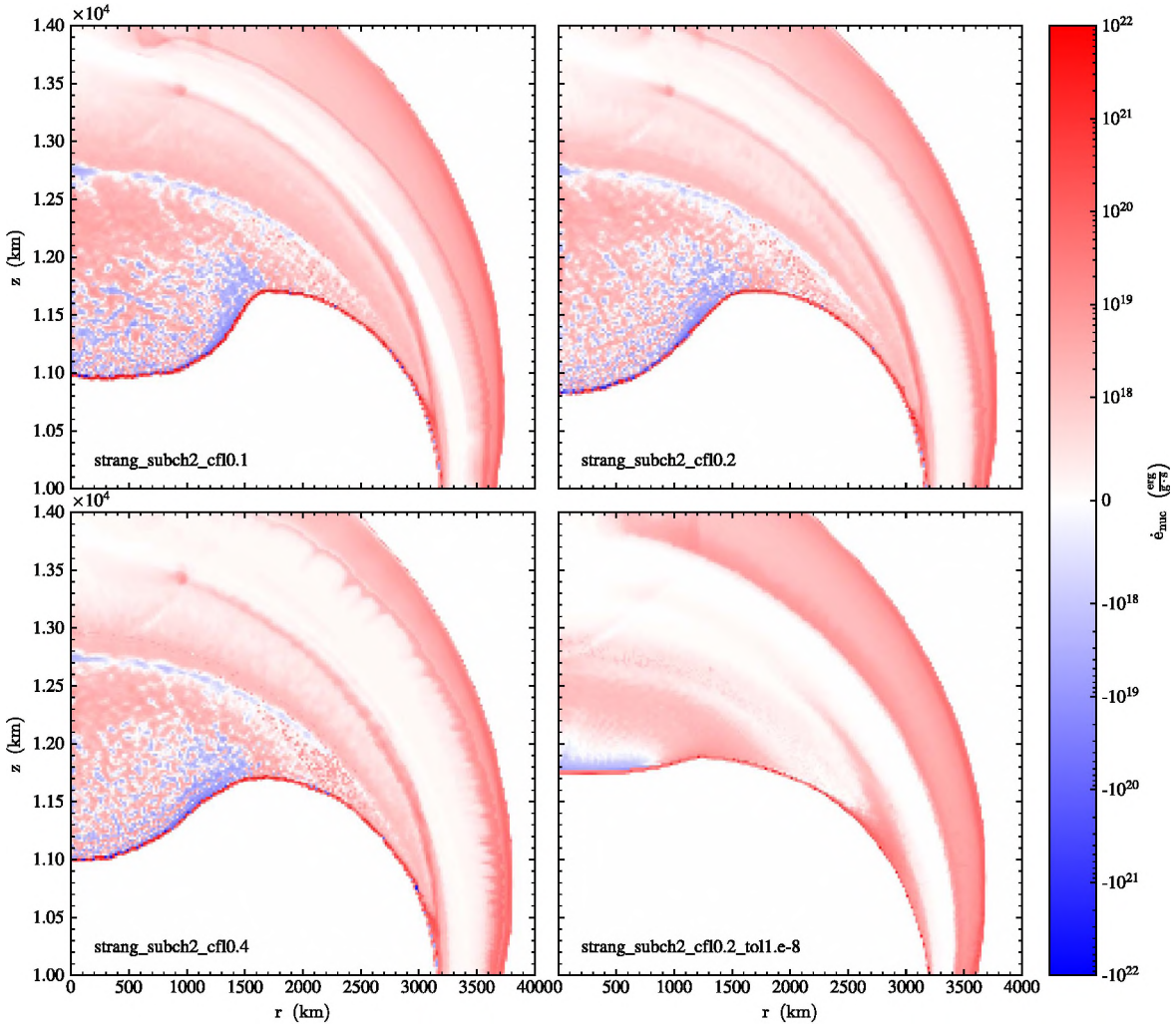

Figure 7. Comparison of Strang runs showing the nuclear energy generation rate behind the inward-propagating shock.

3. Tests

3.1. Reacting Convergence Test Problem

Zingale et al. (2019) introduced a test problem for measuring convergence of a reacting hydrodynamic algorithm. This is based on the acoustic pulse problem from McCorquodale & Colella (2011), and was also used in Zingale et al. (2021). A periodic domain with a uniform entropy is initialized with a pressure profile with a small perturbation. This drives a low-amplitude acoustic wave radially outward from the center. A simple reaction network with the triple-alpha and $^{12}C(\alpha, \gamma)^{16}O$ rates releases energy. By running at different spatial resolutions with a fixed CFL number, so that the time step scales with resolution, we can compute the convergence rate in space and time. In order to demonstrate second-order convergence, the problem must be smooth (so the slope limiters do not kick in severely), which means that the energy release cannot be so vigorous as to drive shocks.

Tables 1 and 2 show the Strang splitting and simplified-SDC convergence, respectively. We ran each integration method on a domain with 64^2, 128^2, 256^2, and 512^2 zones. An error is defined between successive resolutions by coarsening the higher-resolution run and taking the L1 norm of the zone-by-zone difference of the two simulations—for example, $\epsilon_{64\to128}$ is the error between the 64^2 and 128^2 simulations. The convergence rate from $\epsilon_{64\to128}$ to $\epsilon_{128\to256}$ is computed as $\log_2 \epsilon_{64\to128}/\epsilon_{128\to256}$ and is given in the column between those errors, and likewise for the higher-resolution case. A rate of 2.0 indicates second-order convergence. We see that both methods exhibit convergence at roughly second-order and agree well. This demonstrates that both methods are working as designed.

3.2. Sub-Chandra Double Detonation

We next model a double-detonation Type Ia supernova. In this model, a detonation begins in the accreted helium layer on a sub-Chandrasekhar mass white dwarf, and that detonation can either drive a compression wave into the core, igniting a second detonation in the carbon/oxygen core, or burn through the interface at the base of the accreted layer, producing an inward-propagating carbon detonation (see, e.g., Fink et al. 2007). Our goal here is not to explore the feasibility of the model or understand whether a detonation propagating through the He-C interface is physical, but rather to look at the coupling of the hydrodynamics and reactions. We want to drive vigorous burning and directly compare Strang splitting and simplified-SDC time integration on this problem. For that reason, we explore only a single model and start off with a rather large temperature perturbation in the He layer.

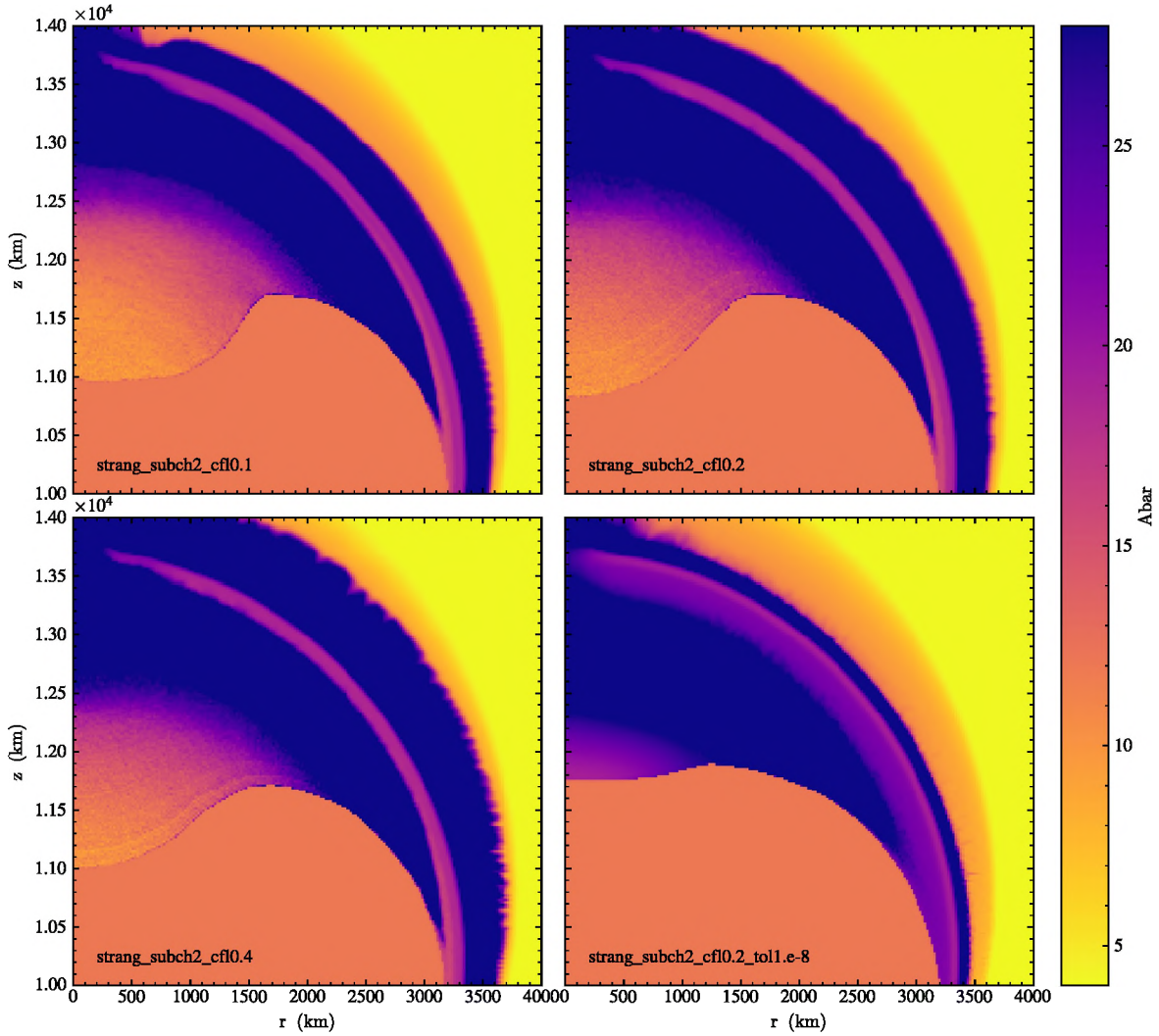

Figure 8. Comparison of Strang runs, showing the mean molecular weight behind the inward-propagating shock.

We use a network with 28 nuclei and 107 rates that captures He burning with links for $(\alpha, p)(p, \gamma)$ reactions and the rates involving ^{14}N to bypass $^{12}\mathrm{C}(\alpha, \gamma)^{16}\mathrm{O}$ as described in Shen & Bildsten (2009). The rates are taken from the ReacLib library (Cyburt et al. 2010) and the network is written by pynucastro (Willcox & Zingale 2018) directly in the C++ format that our code requires. The details of the network are given in Appendix B. This moderate-sized network has many reaction pathways (forward and reverse), making it a good test for the coupling between hydro and reactions.

The initial model is generated following the procedure described in Zingale et al. (2013). We use a $1.1\,M_\odot$ carbon white dwarf with a $0.05\,M_\odot$ He layer. We choose a pure ^{12}C white dwarf to make the reactions more vigorous, consistent with our focus on the coupling of hydrodynamics and reactions. We seed the He layer with 1% (by mass) of ^{14}N. The white dwarf is isothermal and the temperature ramps up at the base of the He layer and is then isentropic until the surface. Figure 3 shows the structure of the initial model. We use 2D axisymmetric coordinates with a domain size of 1.024×10^9 cm by 2.048×10^9 cm, a base grid of 256×512 zones, and a single level of refinement (jumping by 2×), giving a fine-grid resolution of 20 km. Gravity is modeled as a monopole. Our choice of initial perturbation has the detonation propagate inward into the carbon white dwarf. We run the simulation for 0.1 s—this is enough time for the carbon detonation to be well-developed.

An initial perturbation is placed on the symmetry axis, of the form

$$T = T_0[1 + X(^4\mathrm{He}) f (1 + \tanh(2 - r_1))], \tag{49}$$

where

$$r_1 = [x^2 + (y - r_0)^2]^{1/2}/\lambda \tag{50}$$

and

$$r_0 = r_{\mathrm{pert}} + r_{\mathrm{base}}. \tag{51}$$

Here, r_{base} is the radius at which the helium layer begins and r_{pert} is the distance above the base to put the perturbation. We choose $r_{\mathrm{pert}} = 100$ km. The temperature is perturbed above the initial model value, denoted as T_0 here. The amplitude of the perturbation is $f = 12$, and the scale of the perturbation is $\lambda = 200$ km.

We do a suite of runs with both Strang splitting and the simplified-SDC integration, using CFL numbers of 0.1, 0.2, and 0.4, and reaction network tolerances of $\epsilon_{\mathrm{abs,spec}} = \epsilon_{\mathrm{rel,spec}} = 10^{-5}$ and $\epsilon_{\mathrm{abs,ener}} = \epsilon_{\mathrm{rel,ener}} = 10^{-5}$. We also run the

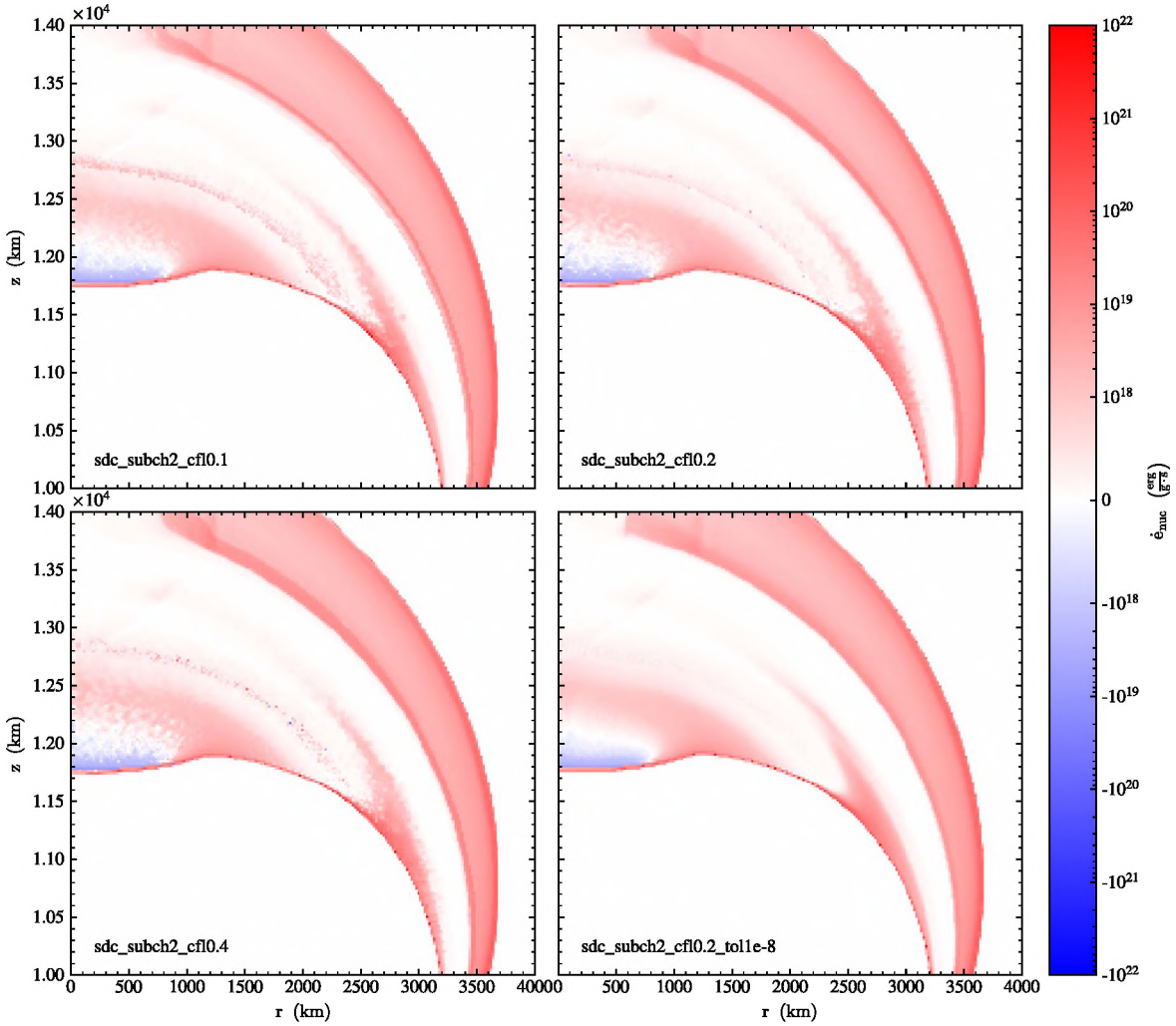

Figure 9. Comparison of simplified-SDC runs, showing the nuclear energy generation rate behind the inward-propagating shock.

CFL = 0.2 case with a tighter tolerance, using $\epsilon_{abs,spec} = 10^{-8}$. For the step rejection part of VODE, we use $f = 4$ and $\eta = 4$. The runs are labeled with the prefix `sdc_` or `strang_` to denote the integrator, and also include the CFL number in the name. Finally, the runs with the tighter tolerances have the suffix `_tol1.e-8`. We note that both the Strang and the SDC runs encounter VODE integration failures occasionally and utilize the Castro step retry functionality.

Figure 4 shows the temperature, mean molecular weight, and nuclear energy generation rate for the Strang splitting CFL = 0.2 simulation (`strang_subch2_cfl0.2`). At this point in the evolution, the helium detonation has wrapped more than halfway across the surface and the ingoing carbon detonation is approaching the center of the star. We see that the mean molecular weight appears mottled behind the inward-propagating detonation in the white dwarf. Comparing this to the energy generation rate, we see that region is characterized by large energy releases of alternating signs checkerboarding throughout the region. This is characteristic of the nucleosynthesis struggling to come into equilibrium at high temperatures, where the reverse rates can be important. To highlight this, Figure 5 shows the nuclear energy generation rate in a highly zoomed-in region around the detonation over four consecutive coarse time steps. We see that, on the detonation front itself, the sign of the energy generation rate changes from one step to the next in many zones.

We next look at the simplified-SDC version of this same case, `sdc_subch2_cfl0.2`. Figure 6 shows the temperature, mean molecular weight, and energy generation rate. Compared to Figure 4, we see that the composition and energy generation rates are smooth, without any of the checkerboarding that plagued the Strang simulation. This suggests that the simplified-SDC algorithm more accurately attains the correct equilibrium in these conditions. More strikingly, we see that the inward-propagating carbon detonation has not progressed as far into the white dwarf at this time. The two integration methods are giving very different results here.

We next look at how the structure seen in Figures 4 and 6 varies with CFL number and network tolerance. We focus just on the region behind the inward-propagating shock. Figures 7 and 8 show the nuclear energy generation rate and mean molecular weight for all of the Strang runs. We see that changing the CFL number does little to affect the state behind the shock—it continues to take on the mottled appearance characteristic of not reaching a proper equilibrium. For the case with tight tolerances, we see that the state is greatly improved with a smoother gradient. Furthermore, the inward-propagating carbon detonation reaches the same position as in the SDC runs.

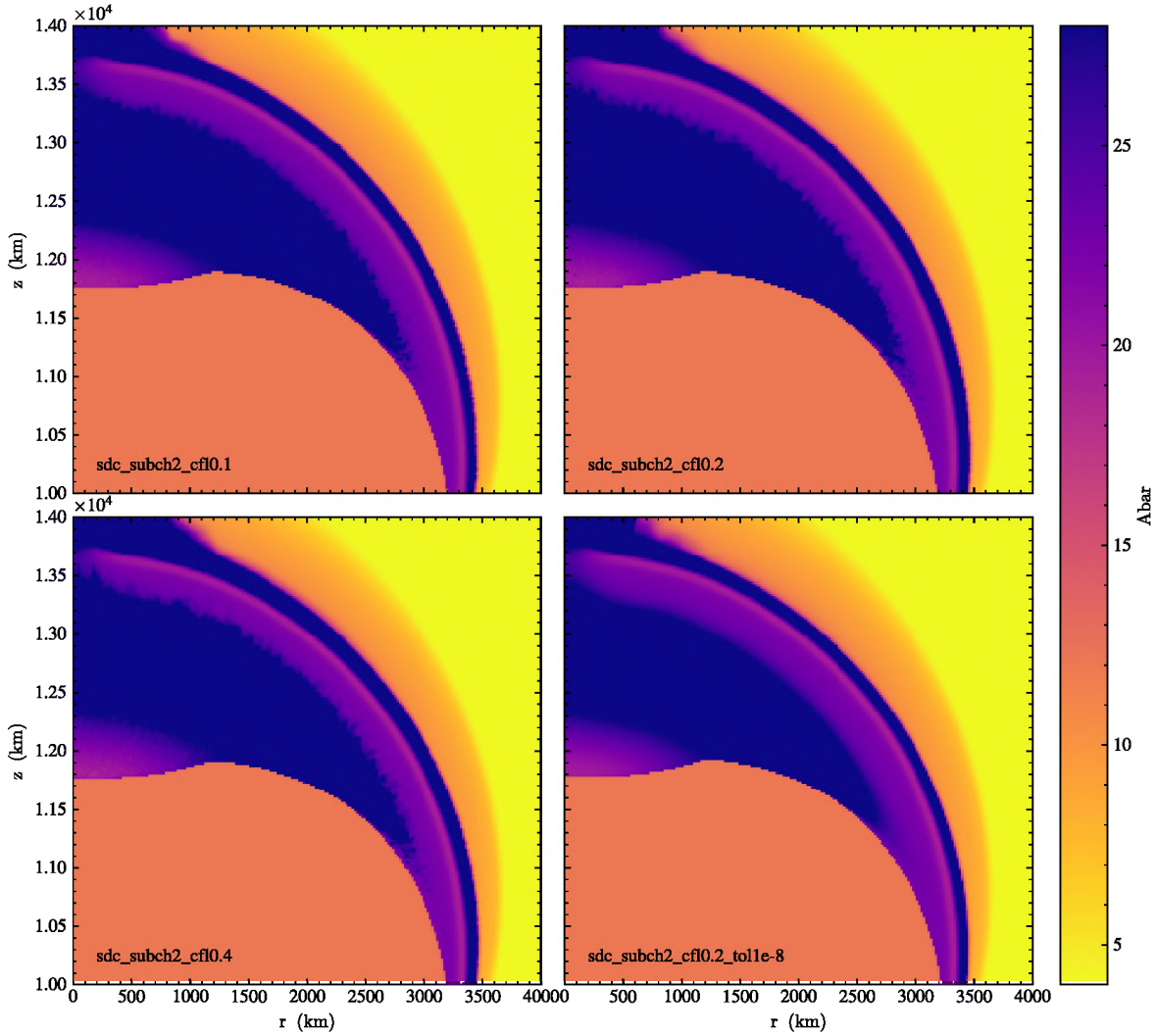

Figure 10. Comparison of simplified-SDC runs, showing the mean molecular weight behind the inward-propagating shock.

Table 3
Species Masses

Simulation	$M_{^4\mathrm{He}}$	$M_{^{28}\mathrm{Si}}$	$M_{^{56}\mathrm{Ni}}$
	Strang runs		
strang_subch2_cfl0.1	1.059×10^{32}	3.804×10^{31}	1.092×10^{32}
strang_subch2_cfl0.2	9.114×10^{31}	4.174×10^{31}	1.078×10^{32}
strang_subch2_cfl0.2_tol1.e-8	7.222×10^{31}	8.623×10^{31}	4.041×10^{31}
strang_subch2_cfl0.4	7.819×10^{31}	4.208×10^{31}	1.045×10^{32}
	SDC runs		
sdc_subch2_cfl0.1	7.319×10^{31}	9.147×10^{31}	3.765×10^{31}
sdc_subch2_cfl0.2	7.208×10^{31}	9.151×10^{31}	3.783×10^{31}
sdc_subch2_cfl0.2_tol1.e-8	7.376×10^{31}	9.220×10^{31}	3.513×10^{31}
sdc_subch2_cfl0.4	7.203×10^{31}	8.843×10^{31}	3.754×10^{31}

The SDC comparisons are shown in Figures 9 and 10. We see that the state behind the shock looks smoother and more consistent regardless of the CFL number or integrator tolerances. This demonstrates that the simplified-SDC integration algorithm is more robust than Strang integration for this problem.

To quantify this effect, we next look at the total mass of some key species. This is often a desired diagnostic for supernova calculations, since ^{56}Ni powers the lightcurve. Table 3 shows the total mass of ^{4}He, ^{28}Si, and ^{56}Ni for each of the simulations. For the Strang runs, there are large variations between the different simulations, while the SDC

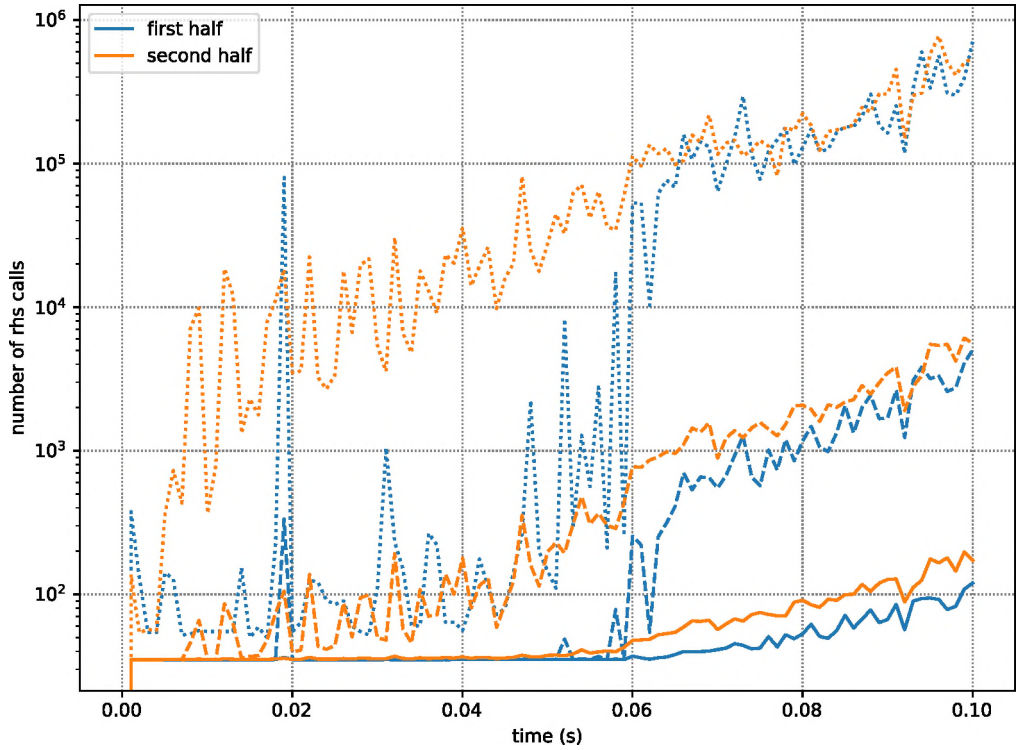

Figure 11. Number of right-hand side evaluations for `strang_subch2_cfl0.2`, showing the L1 norm/average (solid), L2 norm (dashed), and L-inf/maximum (dotted) for both halves of the Strang integration.

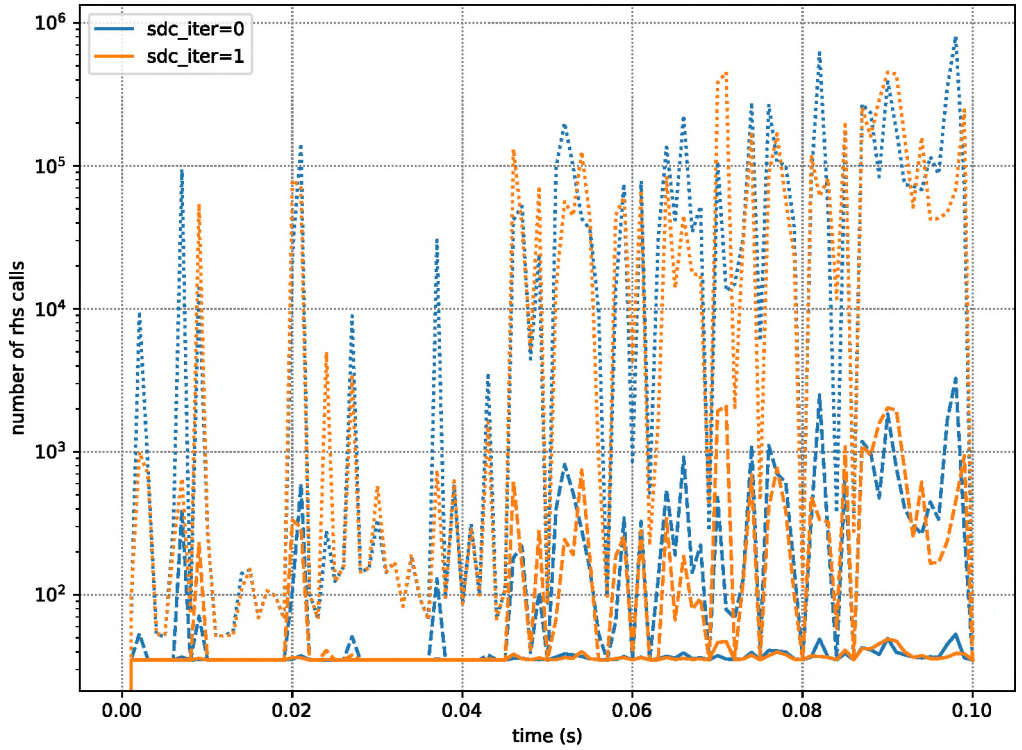

Figure 12. Number of right-hand side evaluations for `sdc_subch2_cfl0.2`, showing the L1 norm/average (solid), L2 norm (dashed), and L-inf/maximum (dotted) for two iterations of the simplified-SDC integration.

simulations all agree to within a few percent. Furthermore, the Strang run with the tighter tolerances is in good agreement with the masses found by the SDC runs. This supports the ideas described above that all of the SDC runs agree while only the Strang run with the tighter tolerances is reliable.

Finally, we can ask how many right-hand side evaluations each integration method required, as a measure of computational cost; when an integration uses many RHS evaluations, it often implies difficulty around equilibrium. We stored this in the plotfiles every 0.001 s, and on the finest level, for zones that

were burning, we compute the L1, L2, and L-inf norms of the number of right-hand side calls (here, the L1 norm is the same as the average and the L-inf is the same as the maximum). This includes the calls that were made to evaluate the numerical Jacobian. Figure 11 shows the results for the `strang_subch2_cfl0.2` run. We see a steady increase in the cost of the reactions as time progresses. Note that the L-inf norm curve is much higher than the other curves, but this is dominated by only a few zones right at the detonation. The other norms give a better sense of the total work in reactions. Figure 12 shows the corresponding plot for the simplified-SDC case (`sdc_subch2_cfl0.2`). While the L-inf norm is of the same magnitude, it fluctuates much more. However, both the L1 and L2 norms are smaller, indicating that the integration does not need to work as hard with the simplified-SDC integration as compared to Strang.

4. Summary

We presented a simplified SDC scheme for coupling hydrodynamics and reactions. We demonstrated that, for a moderate-sized reaction network, the simplified-SDC method compares well against Strang for the double-detonation problem—it finds a better solution at looser reaction tolerances, returns consistent results across a range of CFL numbers, and is more computationally efficient. The benefits of this new integration scheme are likely problem- and network-dependent, and we will explore it on other science problems in future papers.

We have not focused on performance- or load-balancing the reactions in the tests that we ran, but for the simulations shown here, the simplified-SDC method runs as fast (or faster) than the Strang splitting method for the same tolerances and CFL number. We would expect similar behavior for any simulations that are dominated by the cost of reactions. If the results above are robust across different problems and networks, then this suggests that we can run the SDC simulation with larger CFL number and/or less strict reaction tolerances, which would translate into a large performance boost.

This new time-integration method is freely available in Castro, and we will continue to explore it on other problems. Although these simulations were run on CPUs, the entire simplified-SDC algorithm also runs on GPUs. We are interested in applying this method to study massive star evolution, where the vigorous shell burning leading up to core collapse would be an ideal application for the simplified-SDC method. In a follow-on paper, we will show how to couple this new integration method to a table for evaluating an NSE distribution in the iron core of a massive star.

Future work is to extend this methodology to MHD. This is very straightforward because the simplified-SDC method operates solely on the thermodynamic state. We will also explore how to couple radiation hydrodynamics, where we need to do an implicit update for the radiation energy.

Castro is freely available at https://github.com/AMReX-Astro/Castro. All of the code and problem setups used here are available in the git repo. The double-detonation problem is in `Castro/Exec/science/subchandra`. The reaction network used here is available at https://github.com/AMReX-Astro/Microphysics as the `subch2` network. The work at Stony Brook was supported by DOE/Office of Nuclear Physics grant DE-FG02-87ER40317. We thank Alice Harpole for her contributions to the AMReX Astrophysics suite. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of Nuclear Physics, Scientific Discovery through the Advanced Computing (SciDAC) program under Award Number DE-SC0017955. This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. M.R. was supported via an NSF REU grant to Stony Brook, NSF 1852143. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award NP-ERCAP0020354.

Software: AMReX (Zhang et al. 2019), Castro (Almgren et al. 2010), GNU Compiler Collection (https://gcc.gnu.org/), Linux (https://www.kernel.org), matplotlib (Hunter 2007, http://matplotlib.org/), NetworkX (Hagberg et al. 2008), NumPy (Oliphant 2007; van der Walt et al. 2011), python (https://www.python.org/), pynucastro (Willcox & Zingale 2018), pytest (Krekel et al. 2004), SymPy (Meurer et al. 2017), yt (Turk et al. 2011).

Appendix A
Jacobian

To solve the reaction system implicitly, the ODE solver needs the Jacobian, $\partial \boldsymbol{R}/\partial \boldsymbol{\mathcal{U}}'$, where $\boldsymbol{\mathcal{U}}' = (\rho X_k, \rho e)^\top$ is the subset of the conserved variables we are integrating. We follow the method of Zingale et al. (2019) and factor this into two pieces,

$$\boldsymbol{J} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{w}} \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{\mathcal{U}}'}. \tag{A1}$$

where the state $\boldsymbol{w}$ is chosen to match the set of variables used to evaluate the reaction rates. Writing this out for two species, X_α and X_β, we have

$$\boldsymbol{\mathcal{U}}' = \begin{pmatrix} \rho X_\alpha \\ \rho X_\beta \\ \rho e \end{pmatrix}. \tag{A2}$$

For interfacing with the reaction network, we use

$$\boldsymbol{w} = \begin{pmatrix} X_\alpha \\ X_\beta \\ T \end{pmatrix}. \tag{A3}$$

Note: even though we are using T here instead of e, we still do the overall ODE integration in terms of (ρe), consistent with the Strang method described in Zingale et al. (2021).

The Jacobian transformation $\partial \boldsymbol{\mathcal{U}}'/\partial \boldsymbol{w}$ is

$$\frac{\partial \boldsymbol{\mathcal{U}}'}{\partial \boldsymbol{w}} = \begin{pmatrix} \rho & 0 & 0 \\ 0 & \rho & 0 \\ \rho e_{X_\alpha} & \rho e_{X_\beta} & \rho c_v \end{pmatrix}, \tag{A4}$$

where we use the following notation for compactness:

$$e_{X_k} = \left.\frac{\partial e}{\partial X_k}\right|_{\rho, T, X_{j, j\neq k}}, \tag{A5}$$

and the specific heat at constant volume is

$$c_v = \left.\frac{\partial e}{\partial T}\right|_{\rho,X_k}. \tag{A6}$$

We get the inverse (computed via SymPy) as

$$\frac{\partial \boldsymbol{w}}{\partial \mathcal{U}'} = \begin{pmatrix} \frac{1}{\rho} & 0 & 0 \\ 0 & \frac{1}{\rho} & 0 \\ -\frac{e_{X_\alpha}}{\rho c_v} & -\frac{e_{X_\beta}}{\rho c_v} & \frac{1}{\rho c_v} \end{pmatrix}. \tag{A7}$$

The reaction vector is

$$\boldsymbol{R}(\mathcal{U}') = \begin{pmatrix} \rho\dot{\omega}_\alpha \\ \rho\dot{\omega}_\beta \\ \rho\dot{S} \end{pmatrix}. \tag{A8}$$

We take all the quantities to be functions of ρ, e, and X_k, but since ρ does not change from the reactions (the reactive source of the continuity equation is zero), it is held constant in these derivatives. The Jacobian is computed as $\partial \boldsymbol{R}/\partial \boldsymbol{w}$:

$$\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{w}} = \begin{pmatrix} \rho\frac{\partial \dot{\omega}_\alpha}{\partial X_\alpha} & \rho\frac{\partial \dot{\omega}_\alpha}{\partial X_\beta} & \rho\frac{\partial \dot{\omega}_\alpha}{\partial T} \\ \rho\frac{\partial \dot{\omega}_\beta}{\partial X_\alpha} & \rho\frac{\partial \dot{\omega}_\beta}{\partial X_\beta} & \rho\frac{\partial \dot{\omega}_\beta}{\partial T} \\ \rho\frac{\partial \dot{S}}{\partial X_\alpha} & \rho\frac{\partial \dot{S}}{\partial X_\beta} & \rho\frac{\partial \dot{S}}{\partial T} \end{pmatrix}. \tag{A9}$$

The final Jacobian is found by multiplying these two:

$$\frac{\partial \boldsymbol{R}}{\partial \mathcal{U}'} = \begin{pmatrix} \frac{\partial \dot{\omega}_\alpha}{\partial X_\alpha} - \frac{e_{X_\alpha}}{c_v}\frac{\partial \dot{\omega}_\alpha}{\partial T} & \frac{\partial \dot{\omega}_\alpha}{\partial X_\beta} - \frac{e_{X_\beta}}{c_v}\frac{\partial \dot{\omega}_\alpha}{\partial T} & \frac{1}{c_v}\frac{\partial \dot{\omega}_\alpha}{\partial T} \\ \frac{\partial \dot{\omega}_\beta}{\partial X_\alpha} - \frac{e_{X_\alpha}}{c_v}\frac{\partial \dot{\omega}_\beta}{\partial T} & \frac{\partial \dot{\omega}_\beta}{\partial X_\beta} - \frac{e_{X_\beta}}{c_v}\frac{\partial \dot{\omega}_\beta}{\partial T} & \frac{1}{c_v}\frac{\partial \dot{\omega}_\beta}{\partial T} \\ \frac{\partial \dot{S}}{\partial X_\alpha} - \frac{e_{X_\alpha}}{c_v}\frac{\partial \dot{S}}{\partial T} & \frac{\partial \dot{S}}{\partial X_\beta} - \frac{e_{X_\beta}}{c_v}\frac{\partial \dot{S}}{\partial T} & \frac{1}{c_v}\frac{\partial \dot{S}}{\partial T} \end{pmatrix}. \tag{A10}$$

We note that the form of these entries is the same as one would arrive at if one started with the rates expressed as $\omega_k(\rho, T(\rho, X_j, e), X_j)$ and recognized that constant e implies that

$$\left.\frac{\partial T}{\partial X_k}\right|_{\rho,e,X_{j,j\neq k}} = -\frac{e_{X_k}}{c_v}. \tag{A11}$$

Finally, we note that one implication of this formulation is that the Jacobian is no longer sparse. Future work will explore alternate formulations and approximate Jacobians.

Appendix B
Reaction Network

We build a reaction network that approximates an α-chain, including all ^{4}He, and all of the α nuclei from ^{12}C to ^{56}Ni. We want to capture both the (α, γ) and $(\alpha, p)(p, \gamma)$ links, so we include the intermediate nuclei from the (α, p) rate instead of approximating these links assuming proton equilibriation. Following Shen & Bildsten (2009), we include the sequence ^{14}N$(\alpha, \gamma)^{18}$F$(\alpha, p)^{21}$Ne, which produce the protons that can allow for ^{12}C$(p, \gamma)^{13}$N$(\alpha, p)^{16}$O to compete with ^{12}C$(\alpha, \gamma)^{16}$O. We add ^{22}Na to link the ^{21}Ne produced back to the α-chain. We grab all of the ReacLib (Cyburt et al. 2010) reaction rates linking these nuclei using pynucastro (Willcox & Zingale 2018). We are missing one rate each for ^{12}C + ^{12}C, ^{12}C + ^{16}O, and ^{16}O + ^{16}O. These are the sequences ^{12}C$(^{12}$C, $n)^{23}$Mg$(n, \gamma)^{24}$Mg, ^{16}O$(^{12}$C, $n)^{27}$Si$(n, \gamma)^{28}$Si, and ^{16}O$(^{16}$O, $n)^{31}$S$(n, \gamma)^{32}$S, involving neutron production followed by a capture back to one of our α-chain nuclei. These are the only rates we approximate here, assuming that the subsequent neutron capture is instantaneous.

Our final set of 28 nuclei is comprised of ^{1}H, ^{4}He, ^{12}C, ^{13}N, ^{14}N, ^{16}O, ^{18}F, ^{20}Ne, ^{21}Ne, ^{22}Na, ^{23}Na, ^{24}Mg, ^{27}Al, ^{28}Si, ^{31}P, ^{32}S, ^{35}Cl, ^{36}Ar, ^{39}K, ^{40}Ca, ^{43}Sc, ^{44}Ti, ^{47}V, ^{48}Cr, ^{51}Mn, ^{52}Fe, ^{55}Co, and ^{56}Ni. The 107 rates linking them are given in Table 4, with the forward rates having a positive Q value and the reverse having a negative Q value. We do not recompute the reverse rates via detailed balance or correct them with high-temperature partition functions—this will be considered in a future science-focused paper. We add screening to the rates using the prescription from Graboske et al. (1973), Alastuey & Jancovici (1978), and Itoh et al. (1979). The C++ code for the network is output by pynucastro directly for Castro. A graphical overview of the network showing the links between the nuclei is shown in Figure 13.

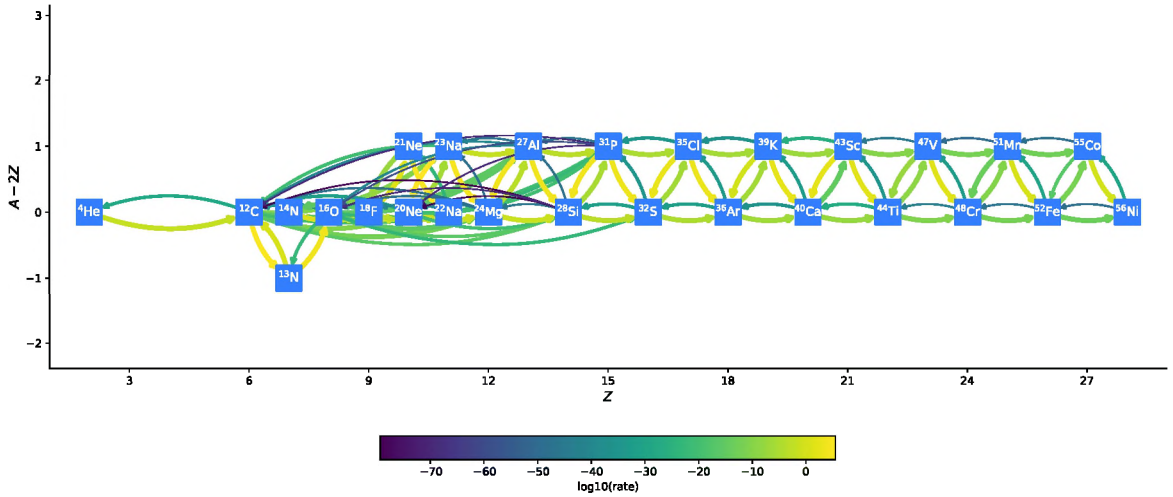

Figure 13. Reaction network for the double-detonation problem.

Table 4
Reaction Rates Included in Our Network

Forward Reaction	Reverse Reaction
$^4\mathrm{He} + {}^4\mathrm{He} + {}^4\mathrm{He} \rightarrow {}^{12}\mathrm{C}$	$^{12}\mathrm{C} \rightarrow {}^4\mathrm{He} + {}^4\mathrm{He} + {}^4\mathrm{He}$
$^{12}\mathrm{C} + {}^1\mathrm{H} \rightarrow {}^{13}\mathrm{N}$	$^{13}\mathrm{N} \rightarrow {}^1\mathrm{H} + {}^{12}\mathrm{C}$
$^{12}\mathrm{C} + {}^4\mathrm{He} \rightarrow {}^{16}\mathrm{O}$	$^{16}\mathrm{O} \rightarrow {}^4\mathrm{He} + {}^{12}\mathrm{C}$
$^{12}\mathrm{C} + {}^{12}\mathrm{C} \rightarrow {}^4\mathrm{He} + {}^{20}\mathrm{Ne}$	$^{20}\mathrm{Ne} + {}^4\mathrm{He} \rightarrow {}^{12}\mathrm{C} + {}^{12}\mathrm{C}$
$^{12}\mathrm{C} + {}^{12}\mathrm{C} \rightarrow {}^1\mathrm{H} + {}^{23}\mathrm{Na}$	$^{23}\mathrm{Na} + {}^1\mathrm{H} \rightarrow {}^{12}\mathrm{C} + {}^{12}\mathrm{C}$
$^{12}\mathrm{C} + {}^{12}\mathrm{C} \rightarrow {}^{24}\mathrm{Mg}$	
$^{13}\mathrm{N} + {}^4\mathrm{He} \rightarrow {}^1\mathrm{H} + {}^{16}\mathrm{O}$	$^{16}\mathrm{O} + {}^1\mathrm{H} \rightarrow {}^4\mathrm{He} + {}^{13}\mathrm{N}$
$^{14}\mathrm{N} + {}^4\mathrm{He} \rightarrow {}^{18}\mathrm{F}$	$^{18}\mathrm{F} \rightarrow {}^4\mathrm{He} + {}^{14}\mathrm{N}$
$^{16}\mathrm{O} + {}^4\mathrm{He} \rightarrow {}^{20}\mathrm{Ne}$	$^{20}\mathrm{Ne} \rightarrow {}^4\mathrm{He} + {}^{16}\mathrm{O}$
$^{16}\mathrm{O} + {}^{12}\mathrm{C} \rightarrow {}^4\mathrm{He} + {}^{24}\mathrm{Mg}$	$^{24}\mathrm{Mg} + {}^4\mathrm{He} \rightarrow {}^{12}\mathrm{C} + {}^{16}\mathrm{O}$
$^{16}\mathrm{O} + {}^{12}\mathrm{C} \rightarrow {}^1\mathrm{H} + {}^{27}\mathrm{Al}$	$^{27}\mathrm{Al} + {}^1\mathrm{H} \rightarrow {}^{12}\mathrm{C} + {}^{16}\mathrm{O}$
$^{16}\mathrm{O} + {}^{12}\mathrm{C} \rightarrow {}^{28}\mathrm{Si}$	
$^{16}\mathrm{O} + {}^{16}\mathrm{O} \rightarrow {}^4\mathrm{He} + {}^{28}\mathrm{Si}$	$^{28}\mathrm{Si} + {}^4\mathrm{He} \rightarrow {}^{16}\mathrm{O} + {}^{16}\mathrm{O}$
$^{16}\mathrm{O} + {}^{16}\mathrm{O} \rightarrow {}^1\mathrm{H} + {}^{31}\mathrm{P}$	$^{31}\mathrm{P} + {}^1\mathrm{H} \rightarrow {}^{16}\mathrm{O} + {}^{16}\mathrm{O}$
$^{16}\mathrm{O} + {}^{16}\mathrm{O} \rightarrow {}^{32}\mathrm{S}$	
$^{18}\mathrm{F} + {}^4\mathrm{He} \rightarrow {}^1\mathrm{H} + {}^{21}\mathrm{Ne}$	$^{21}\mathrm{Ne} + {}^1\mathrm{H} \rightarrow {}^4\mathrm{He} + {}^{18}\mathrm{F}$
$^{18}\mathrm{F} + {}^4\mathrm{He} \rightarrow {}^{22}\mathrm{Na}$	$^{22}\mathrm{Na} \rightarrow {}^4\mathrm{He} + {}^{18}\mathrm{F}$
$^{20}\mathrm{Ne} + {}^4\mathrm{He} \rightarrow {}^{24}\mathrm{Mg}$	$^{24}\mathrm{Mg} \rightarrow {}^4\mathrm{He} + {}^{20}\mathrm{Ne}$
$^{20}\mathrm{Ne} + {}^{12}\mathrm{C} \rightarrow {}^4\mathrm{He} + {}^{28}\mathrm{Si}$	$^{28}\mathrm{Si} + {}^4\mathrm{He} \rightarrow {}^{12}\mathrm{C} + {}^{20}\mathrm{Ne}$
$^{20}\mathrm{Ne} + {}^{12}\mathrm{C} \rightarrow {}^1\mathrm{H} + {}^{31}\mathrm{P}$	$^{31}\mathrm{P} + {}^1\mathrm{H} \rightarrow {}^{12}\mathrm{C} + {}^{20}\mathrm{Ne}$
$^{21}\mathrm{Ne} + {}^1\mathrm{H} \rightarrow {}^{22}\mathrm{Na}$	$^{22}\mathrm{Na} \rightarrow {}^1\mathrm{H} + {}^{21}\mathrm{Ne}$
$^{23}\mathrm{Na} + {}^1\mathrm{H} \rightarrow {}^4\mathrm{He} + {}^{20}\mathrm{Ne}$	$^{20}\mathrm{Ne} + {}^4\mathrm{He} \rightarrow {}^1\mathrm{H} + {}^{23}\mathrm{Na}$
$^{23}\mathrm{Na} + {}^1\mathrm{H} \rightarrow {}^{24}\mathrm{Mg}$	$^{24}\mathrm{Mg} \rightarrow {}^1\mathrm{H} + {}^{23}\mathrm{Na}$
$^{23}\mathrm{Na} + {}^4\mathrm{He} \rightarrow {}^{27}\mathrm{Al}$	$^{27}\mathrm{Al} \rightarrow {}^4\mathrm{He} + {}^{23}\mathrm{Na}$
$^{24}\mathrm{Mg} + {}^4\mathrm{He} \rightarrow {}^{28}\mathrm{Si}$	$^{28}\mathrm{Si} \rightarrow {}^4\mathrm{He} + {}^{24}\mathrm{Mg}$
$^{27}\mathrm{Al} + {}^1\mathrm{H} \rightarrow {}^4\mathrm{He} + {}^{24}\mathrm{Mg}$	$^{24}\mathrm{Mg} + {}^4\mathrm{He} \rightarrow {}^1\mathrm{H} + {}^{27}\mathrm{Al}$
$^{27}\mathrm{Al} + {}^1\mathrm{H} \rightarrow {}^{28}\mathrm{Si}$	$^{28}\mathrm{Si} \rightarrow {}^1\mathrm{H} + {}^{27}\mathrm{Al}$
$^{27}\mathrm{Al} + {}^4\mathrm{He} \rightarrow {}^{31}\mathrm{P}$	$^{31}\mathrm{P} \rightarrow {}^4\mathrm{He} + {}^{27}\mathrm{Al}$
$^{28}\mathrm{Si} + {}^4\mathrm{He} \rightarrow {}^{32}\mathrm{S}$	$^{32}\mathrm{S} \rightarrow {}^4\mathrm{He} + {}^{28}\mathrm{Si}$
$^{31}\mathrm{P} + {}^1\mathrm{H} \rightarrow {}^4\mathrm{He} + {}^{28}\mathrm{Si}$	$^{28}\mathrm{Si} + {}^4\mathrm{He} \rightarrow {}^1\mathrm{H} + {}^{31}\mathrm{P}$
$^{31}\mathrm{P} + {}^1\mathrm{H} \rightarrow {}^{32}\mathrm{S}$	$^{32}\mathrm{S} \rightarrow {}^1\mathrm{H} + {}^{31}\mathrm{P}$
$^{31}\mathrm{P} + {}^4\mathrm{He} \rightarrow {}^{35}\mathrm{Cl}$	$^{35}\mathrm{Cl} \rightarrow {}^4\mathrm{He} + {}^{31}\mathrm{P}$
$^{32}\mathrm{S} + {}^4\mathrm{He} \rightarrow {}^{36}\mathrm{Ar}$	$^{36}\mathrm{Ar} \rightarrow {}^4\mathrm{He} + {}^{32}\mathrm{S}$
$^{35}\mathrm{Cl} + {}^1\mathrm{H} \rightarrow {}^4\mathrm{He} + {}^{32}\mathrm{S}$	$^{32}\mathrm{S} + {}^4\mathrm{He} \rightarrow {}^1\mathrm{H} + {}^{35}\mathrm{Cl}$
$^{35}\mathrm{Cl} + {}^1\mathrm{H} \rightarrow {}^{36}\mathrm{Ar}$	$^{36}\mathrm{Ar} \rightarrow {}^1\mathrm{H} + {}^{35}\mathrm{Cl}$
$^{35}\mathrm{Cl} + {}^4\mathrm{He} \rightarrow {}^{39}\mathrm{K}$	$^{39}\mathrm{K} \rightarrow {}^4\mathrm{He} + {}^{35}\mathrm{Cl}$
$^{36}\mathrm{Ar} + {}^4\mathrm{He} \rightarrow {}^{40}\mathrm{Ca}$	$^{40}\mathrm{Ca} \rightarrow {}^4\mathrm{He} + {}^{36}\mathrm{Ar}$
$^{39}\mathrm{K} + {}^1\mathrm{H} \rightarrow {}^4\mathrm{He} + {}^{36}\mathrm{Ar}$	$^{36}\mathrm{Ar} + {}^4\mathrm{He} \rightarrow {}^1\mathrm{H} + {}^{39}\mathrm{K}$
$^{39}\mathrm{K} + {}^1\mathrm{H} \rightarrow {}^{40}\mathrm{Ca}$	$^{40}\mathrm{Ca} \rightarrow {}^1\mathrm{H} + {}^{39}\mathrm{K}$
$^{39}\mathrm{K} + {}^4\mathrm{He} \rightarrow {}^{43}\mathrm{Sc}$	$^{43}\mathrm{Sc} \rightarrow {}^4\mathrm{He} + {}^{39}\mathrm{K}$
$^{40}\mathrm{Ca} + {}^4\mathrm{He} \rightarrow {}^{44}\mathrm{Ti}$	$^{44}\mathrm{Ti} \rightarrow {}^4\mathrm{He} + {}^{40}\mathrm{Ca}$
$^{43}\mathrm{Sc} + {}^1\mathrm{H} \rightarrow {}^4\mathrm{He} + {}^{40}\mathrm{Ca}$	$^{40}\mathrm{Ca} + {}^4\mathrm{He} \rightarrow {}^1\mathrm{H} + {}^{43}\mathrm{Sc}$
$^{43}\mathrm{Sc} + {}^1\mathrm{H} \rightarrow {}^{44}\mathrm{Ti}$	$^{44}\mathrm{Ti} \rightarrow {}^1\mathrm{H} + {}^{43}\mathrm{Sc}$
$^{43}\mathrm{Sc} + {}^4\mathrm{He} \rightarrow {}^{47}\mathrm{V}$	$^{47}\mathrm{V} \rightarrow {}^4\mathrm{He} + {}^{43}\mathrm{Sc}$
$^{44}\mathrm{Ti} + {}^4\mathrm{He} \rightarrow {}^{48}\mathrm{Cr}$	$^{48}\mathrm{Cr} \rightarrow {}^4\mathrm{He} + {}^{44}\mathrm{Ti}$
$^{47}\mathrm{V} + {}^1\mathrm{H} \rightarrow {}^4\mathrm{He} + {}^{44}\mathrm{Ti}$	$^{44}\mathrm{Ti} + {}^4\mathrm{He} \rightarrow {}^1\mathrm{H} + {}^{47}\mathrm{V}$
$^{47}\mathrm{V} + {}^1\mathrm{H} \rightarrow {}^{48}\mathrm{Cr}$	$^{48}\mathrm{Cr} \rightarrow {}^1\mathrm{H} + {}^{47}\mathrm{V}$
$^{47}\mathrm{V} + {}^4\mathrm{He} \rightarrow {}^{51}\mathrm{Mn}$	$^{51}\mathrm{Mn} \rightarrow {}^4\mathrm{He} + {}^{47}\mathrm{V}$
$^{48}\mathrm{Cr} + {}^4\mathrm{He} \rightarrow {}^1\mathrm{H} + {}^{51}\mathrm{Mn}$	$^{51}\mathrm{Mn} + {}^1\mathrm{H} \rightarrow {}^4\mathrm{He} + {}^{48}\mathrm{Cr}$
$^{48}\mathrm{Cr} + {}^4\mathrm{He} \rightarrow {}^{52}\mathrm{Fe}$	$^{52}\mathrm{Fe} \rightarrow {}^4\mathrm{He} + {}^{48}\mathrm{Cr}$
$^{51}\mathrm{Mn} + {}^1\mathrm{H} \rightarrow {}^{52}\mathrm{Fe}$	$^{52}\mathrm{Fe} \rightarrow {}^1\mathrm{H} + {}^{51}\mathrm{Mn}$
$^{51}\mathrm{Mn} + {}^4\mathrm{He} \rightarrow {}^{55}\mathrm{Co}$	$^{55}\mathrm{Co} \rightarrow {}^4\mathrm{He} + {}^{51}\mathrm{Mn}$
$^{52}\mathrm{Fe} + {}^4\mathrm{He} \rightarrow {}^1\mathrm{H} + {}^{55}\mathrm{Co}$	$^{55}\mathrm{Co} + {}^1\mathrm{H} \rightarrow {}^4\mathrm{He} + {}^{52}\mathrm{Fe}$
$^{52}\mathrm{Fe} + {}^4\mathrm{He} \rightarrow {}^{56}\mathrm{Ni}$	$^{56}\mathrm{Ni} \rightarrow {}^4\mathrm{He} + {}^{52}\mathrm{Fe}$
$^{55}\mathrm{Co} + {}^1\mathrm{H} \rightarrow {}^{56}\mathrm{Ni}$	$^{56}\mathrm{Ni} \rightarrow {}^1\mathrm{H} + {}^{55}\mathrm{Co}$

ORCID iDs

M. Zingale https://orcid.org/0000-0001-8401-030X
M. P. Katz https://orcid.org/0000-0003-0439-4556
A. Nonaka https://orcid.org/0000-0003-1791-0265
M. Rasmussen https://orcid.org/0000-0002-0297-0313

References

Alastuey, A., & Jancovici, B. 1978, ApJ, 226, 1034
Almgren, A., Sazo, M. B., Bell, J., et al. 2020, JOSS, 5, 2513
Almgren, A. S., Beckner, V. E., Bell, J. B., et al. 2010, ApJ, 715, 1221
Almgren, A. S., Bell, J. B., Nonaka, A., & Zingale, M. 2008, ApJ, 684, 449
Bourlioux, A., Layton, A. T., & Minion, M. L. 2003, JCoPh, 189, 651
Brown, P. N., Byrne, G. D., & Hindmarsh, A. C. 1989, SIAM J. Sci. Stat. Comput., 10, 1038
Bryan, G. L., Norman, M. L., Stone, J. M., Cen, R., & Ostriker, J. P. 1995, CoPhC, 89, 149
Byrne, G. D., & Hindmarsh, A. C. 1987, JCoPh, 70, 1
Colella, P. 1990, JCoPh, 87, 171
Colella, P., & Glaz, H. M. 1985, JCoPh, 59, 264
Colella, P., & Woodward, P. R. 1984, JCoPh, 54, 174
Couch, S. M., Chatzopoulos, E., Arnett, W. D., & Timmes, F. X. 2015, ApJL, 808, L21
Cyburt, R. H., Amthor, A. M., Ferguson, R., et al. 2010, ApJS, 189, 240
Dutt, A., Greengard, L., & Rokhlin, V. 2000, BIT Numer. Math., 40, 241
Fan, D., Nonaka, A., Almgren, A. S., Harpole, A., & Zingale, M. 2019, ApJ, 887, 212
Fink, M., Hillebrandt, W., & Röpke, F. K. 2007, A&A, 476, 1133
Fryxell, B. A., Müller, E., & Arnett, D. 1989, MPA Preprint 449
Graboske, H. C., Dewitt, H. E., Grossman, A. S., & Cooper, M. S. 1973, ApJ, 181, 457
Hagberg, A. A., Schult, D. A., & Swart, P. J. 2008, in Proc. 7th Python in Science Conf., ed. G. Varoquaux, T. Vaught, & J. Millman (Pasadena, CA) 11
Hunter, J. D. 2007, CSE, 9, 90
Itoh, N., Totsuji, H., Ichimaru, S., & Dewitt, H. E. 1979, ApJ, 234, 1079
Katz, M. P., Almgren, A., Sazo, M. B., et al. 2020, in Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis, SC '20 (Piscataway, NJ: IEEE)
Katz, M. P., Zingale, M., Calder, A. C., et al. 2016, ApJ, 819, 94
Krekel, H., Oliveira, B., Pfannschmidt, R., et al. 2004, pytest 7.0, https://github.com/pytest-dev/pytest
McCorquodale, P., & Colella, P. 2011, Commun. Appl. Math. Comput. Sci., 6, 1
Meurer, A., Smith, C. P., Paprocki, M., et al. 2017, PeerJ Comp. Sci., 3, e103
Miller, G. H., & Colella, P. 2002, JCoPh, 183, 26
Müller, E. 1986, A&A, 162, 103
Nonaka, A., Bell, J. B., Day, M. S., et al. 2012, CTM, 16, 1053
Oliphant, T. E. 2007, CSE, 9, 10
Radhakrishnan, K., & Hindmarsh, A. C. 1993, Lawrence Livermore National Laboratory Report UCRL-ID-113855, 124
Rivas, F., Harris, A., Hix, R., & Messer, O. E. B. 2022, arXiv:2205.12370
Saltzman, J. 1994, JCoPh, 115, 153
Shen, K. J., & Bildsten, L. 2009, ApJ, 699, 1365
Strang, G. 1968, SJNA, 5, 506
Turk, M. J., Smith, B. D., Oishi, J. S., et al. 2011, ApJS, 192, 9
van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, CSE, 13, 22
Willcox, D. E., & Zingale, M. 2018, JOSS, 3, 588
Zhang, W., Almgren, A., Beckner, V., et al. 2019, JOSS, 4, 1370
Zingale, M., Eiden, K., Cavecchi, Y., et al. 2019, JPhCS, 1225, 012005
Zingale, M., Katz, M. P., Bell, J. B., et al. 2019, ApJ, 886, 105
Zingale, M., Katz, M. P., Willcox, D. E., & Harpole, A. 2021, RNAAS, 5, 71
Zingale, M., Nonaka, A., Almgren, A. S., et al. 2013, ApJ, 764, 97
Zingale, M., Timmes, F. X., Fryxell, B., et al. 2001, ApJS, 133, 195