ON THE INTERPRETATION OF CORE-COLLAPSE SUPERNOVAE LIGHT CURVES AND
DEVELOPMENT OF PERFORMANCE PORTABLE SIMULATIONS

By

Brandon Barker

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Astronomy & Astrophysics and Computational Mathematics, Science, & Engineering—Doctor of
Philosophy

2024

ABSTRACT

Core-collapse supernovae (CCSNe) are the tumultuous explosions that accompany the ends of lives of massive stars. After millions of years being seemingly idle, laboriously creating increasingly heavy elements, the star exhausts its fuel supply and, in an instant, is ripped apart. Their innards, consisting of millions of years of nucleosynthesis products, are spread throughout the interstellar medium as fertilizer for the next generation of stars. Left in their wake is a stellar mass compact object – a black hole or neutron star. CCSNe are vital to understanding our own origins. Our understanding of CCSNe is driven by the union of observation and theory. Computational models, constantly leveraging the most advanced supercomputers of the time, provide insights into the central engines powering CCSNe and connect to observations of CCSNe. Observations, providing a goal post and validation for computational models, require a theoretical framework to be interpreted. The work presented in this Dissertation seeks to provide novel approaches to interpreting CCSN observables and develops new computational models for studying the explosion mechanisms of CCSNe.

I produce synthetic supernova light curves from high fidelity, neutrino-driven supernova models – the largest such study. Using these light curves, I demonstrate the improved ability of neutrino-driven models to constrain observations. I demonstrate how the imprint from the core structure of the star on the explosion can be seen in observed photometry. In followup work, I build on this and investigate the core structures of a population of observed supernovae. Using a novel Bayesian analysis, I use these inferences to constrain the mass distribution of the stellar population. To demonstrate the ineffectiveness of simplified models to constrain observations, I produce a grid of roughly 2000 light curves and demonstrate that, with these simplified models, the results are degenerate and ill-constraining.

I also report on the development of several open source software projects to further investigate the CCSN explosion mechanism. First, I present the THORNADO hydrodynamics algorithms. THORNADOuses a novel high order discontinuous Galerkin approach to modeling the underlying partial differential equations and is posed to power the next generation of models. Next, I present

Singularity-Eos, an open source microphysics library for fluid dynamics that is capable of leveraging modern heterogeneous hardware. Finally, I close with a description of Phoebus, a new simulation software for supernovae, compact object accretion, and mergers set to make use of exascale computing resources.

To my family, both born and found, without whom none of this would have been possible. In loving memory of my father, who lost his battle with cancer before the completion of this work.

Richard Prince, I am grateful for your friendship through the years.

I am grateful for the countless hours of work provided by developers of open source scientific software, without with much of the work here would have been made more difficult. Of note, I thank the maintainers of numpy, scipy, matplotlib, astropy, yt, snec, flash, and emcee. Global scientific productivity is owed to efforts such as these.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

MSU   Michigan State University

ZAMS   Zero-age main sequence

RSG   Red supergiant

CCSN   Core-collapse supernova

SN   Supernova

SNEC   SuperNova Explosion Code

STIR   Supernova Turbulence In Reduced dimensionality

GRMHD   General-relativistic magnetohydrodynamics

PDE   Partial differential equation

DG   Discontinuous Galerkin

CG   Continuous Galerkin

MCMC   Markov Chain Monte Carlo

CHAPTER 1

INTRODUCTION

Splendors of elemental strife;

Smit suns that startle back the
gloom;

New light whose tale of stellar
doom

Fares to uncomprehending life;

George Stirling, *The Testimony
of the Suns*

1.1 Overview

Massive stars with zero-age main sequence (ZAMS) masses greater than about ten solar masses are doomed to suffer an inevitable fate – robbed of pressure support against gravity, their cores collapse and produce a subsequent terminal explosion or implosion. These tumultuous explosions, known as core-collapse supernovae (CCSNe), play a leading role in the cosmic drama. Upon their deaths, their innards – newly synthesized elements such as carbon, oxygen, and nitrogen – are scattered. These elements, acting as fertilizer, provide for the next generation of stars and planets and drive the evolution of their host galaxies. Through their deaths, these stars provide the birth channels for stellar mass compact objects – black holes and neutron stars. They are laboratories for fundamental nuclear physics, probing matter in environments unattainable in Terrestrial experiments. However, understanding these explosions and their astrophysical impact requires modeling a rich amount of tightly coupled physics and connecting to observed supernovae: a deeply difficult task.

In this Dissertation, I explore ways to improve our ability to interpret observations of CCSNe, implications for populations of CCSNe, and develop open source software for modeling CCSNe. I begin by producing synthetic light curves from 136 solar metallicity stellar models, simulated

1

with a realistic, neutrino-driven mechanism and evolved for 300 days post bounce. In this work I demonstrate the ability of neutrino-driven models to constrain observations and provide insights into their core structures. In followup work, I use this suite of explosion models to constrain CCSNe at the population level, applying Bayesian Markov Chain Monte Carlo methods to infer the mass distribution of progenitors in a sample of observed CCSNe. Next, I produce a large grid of approximately 2000 CCSN light curves from parametric explosion models spanning densely a range of progenitor masses and explosion energies. Using this grid of models I demonstrate the failures of simplified explosion models to constrain observations, highlighting the degeneracies inherent in these simplified models and the need for realistic, neutrino-driven simulations. In the final sections I describe the development of new software for high fidelity simulations of CCSNe and other phenomena in relativistic astrophysics. First, I present the hydrodynamics methods for THORNADO, the toolkit for high order neutrino radiation hydrodynamics, utilizing novel discontinuous Galerkin (DG) methods for high order accurate solutions. I briefly describe SINGULARITY-EOS, a new open source software for performance-portable equations of state in fluid and continuum dynamics simulations. Finally, I conclude by presenting PHOEBUS, a new performance portable general relativistic radiation magnetohydrodynamics code for CCSNe, compact object mergers, and black hole accretion. PHOEBUS includes a novel treatment of general relativistic gravity and state-of-the-art physics. I present a suite of test problems, including production-scale simulations of neutrino-cooled black hole accretion, and discuss its future as a tool for the community. The resultant data products from this Dissertation are publicly available online.

1.2 Explosion Mechanism

Core-collapse supernovae (CCSNe) are the explosions that accompany the deaths of massive stars. These stars with zero age main sequence (ZAMS) masses greater than about ten solar masses inevitably form electron degenerate iron cores. What follows has emerged as one of the most complex multiphysics problems and remains a grand challenge in astrophysics. The moments preceding the observable transient are filled with such a range of physics that few phenomena can boast: general relativistic magnetohydrodynamics, both weak and strong particle interactions,

neutrino radiation transport and interactions with matter, photon transport, hot dense matter up to and beyond nuclear saturation, and the potential for the existence of exotic matter. The supernova explosion, prior to shock breakout and an observable electromagnetic transient, can be divided into a few phases: collapse, bounce, stalled accretion shock, and explosion. These phases together comprise the CCSN *explosion mechanism*.

Stars with ZAMS masses (M_{ZAMS}) $\gtrsim 10M_\odot$ will develop degenerate iron cores that become gravitationally unstable at the Chandrasekhar limit. There is an upper limit to this mass range for iron core collapse occurring when their helium cores exceed about $65M_\odot$– the lower limit for pair-instability supernovae (Heger & Woosley, 2002; Heger et al., 2003). These cores are pressure supported primarily by degeneracy pressure from electrons. What follows is a consequence of this degeneracy. These electron degenerate cores, having reached the effective Chandrasekhar mass and, aided by electron captures and photo-disintegration of heavy nuclei, collapse. The collapse proceeds homologously, with infall velocity increasing linearly with radius. The local sound speed, however, decreases outwards with density and radius. Thus at some radius – the sonic point – the infall velocity exceeds the sound speed. As a result, the collapsing core is split in two: an inner core collapsing homologously and subsonically, and an outer core in supersonic free fall. During collapse, electron captures on (primarily) nuclei further reduce degeneracy pressure and accelerate the collapse.

Eventually the collapse proceeds through nuclear densities, the inner core undergoes a phase transition to bulk nuclear matter, the equation of state stiffens due to reaching the repulsive regime of the strong force, and the inner core rebounds – commonly referred to as core *bounce*. Information from this rebound propagates outwards in the form of sound waves until reaching the sonic point delineating the subsonic inner core and supersonic outer core, and stops, falling inwards as quickly as it can move outwards. The result: the production of an outward propagating shock wave. The prevailing idea, for a time, was that this shock wave would fully disrupt the entire star in a supernova explosion – referred to now as the *prompt* explosion mechanism. This, it turns out, is not the case. The shock produced from the rebounding inner core is rapidly enervated, owing primarily to the

dissociation of iron group nuclei. For every $0.1 M_\odot$ of material the shock traverses, dissociation of these nuclei robs it of around 10^{51} erg of energy – the characteristic energy of a strong explosion. If that were not enough, electron captures on the newly freed protons results in the production of massive amounts of electron neutrinos. These neutrinos, initially trapped in the dense matter beneath the shock, are eventually released in the neutrino burst as the shock passes into sufficiently low densities. The result: the shock, robbed of energy, stalls before it can escape the iron core. As supernovae indeed occur (see, e.g., Baade & Zwicky, 1934, for confirmation), the shock must be rejuvenated. The means for this rejuvenation – often referred to as shock revival – has been the focus of decades of theoretical work. Figure 1.1 shows radial velocity profiles at various points from collapse through stall.

Figure 1.2 shows a schematic at the point of shock stall. Now, the core is separated into a cooler, inner core, and a hot, shocked mantle that together form the proto-neutron star (PNS). Around the surface of the PNS is a collection of radii known as *neutrinospheres*. Analogous to the photosphere of a star, these are surfaces of last scatter for each neutrino flavor and energy as they diffuse out of the dense PNS. Above these neutrinospheres, the matter is cooled by neutrino emission. Further out, still beneath the shock, net neutrino heating occurs, driven by charged-current neutrino absorption. Approximately 10% of the early radiated neutrino flux is sufficient to revive the shock, leading to the so-called neutrino-driven delayed explosion mechanism (Bethe & Wilson, 1985).

Experience has shown, however, that neutrino heating beneath the shock alone is insufficient to revive the explosion. Spherically symmetric explosion models, even with the most robust treatment of neutrino transport, fail to explode. It was not until the era of multi-D simulations that this was fully understood: absent in spherical symmetry are hydrodynamic instabilities. These instabilities, such as convection and turbulence, are now known to play a leading role in CCSN dynamics, with turbulent pressure potentially reaching around 50% of the thermal pressure beneath the shock (Couch & Ott (2015), but see also, e.g., Murphy & Meakin (2011); Murphy et al. (2013)). All of these effects work together to determine whether or not a star can revive its shock and produce a supernova explosion. It seems that for some progenitors, neutrino heating and hydrodynamic effects
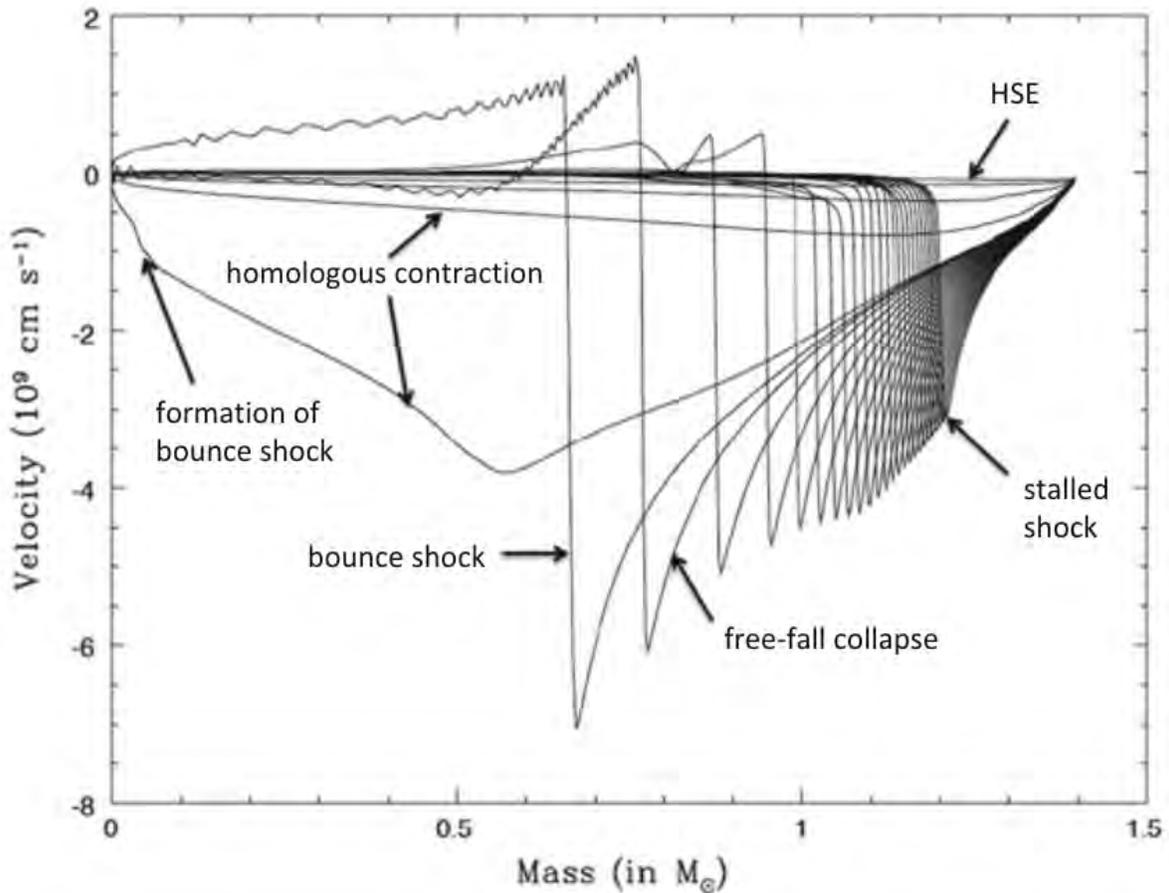
Figure 1.1 Radial velocity evolution at various epochs. Initially at rest in hydrostatic equilibrium (HSE), the material begins to collapse and separates into a homologously collapsing inner core and freefalling outer core. Bounce occurs and shortly afterwards the shock stalls. Figure from Branch & Wheeler (2017).

fail to reinvigorate the stalled shock and the full star will collapse into the compact object, now doomed to black hole formation. These cases are referred to as *failed* supernovae. Observational searches for these events are difficult, but ongoing with some success (see, e.g., Neustadt et al., 2021, and references therein). There are other potential explosion mechanisms for rare classes of supernovae, such as magnetorotational supernovae, in which rapidly rotating, highly magnetized stars may form accretion disks and magnetically-driven jets which drive an explosion (LeBlanc & Wilson, 1970).

The importance of hydrodynamic instabilities in the supernova mechanism has cemented the

5

Rapidly falling material

Stalled shock front

Hot neutrinos

Proto-neutron star
approximately 40-km radius

Quasi-static layer

Region cooled by neutrinos
40- to 50-km radius;
neutrinos are produced
by the following reactions:

$$p + e^- \rightarrow n + \nu_e$$
$$n + e^+ \rightarrow p + \bar{\nu}_e$$
$$e^+ + e^- \rightarrow \nu + \bar{\nu}$$

Region heated by neutrinos
50- to 100-km radius;
neutrinos are absorbed
by the following reactions:

$$n + \nu_e \rightarrow p + e^-$$
$$p + \bar{\nu}_e \rightarrow n + e^+$$
$$\nu + e^{+/-} \rightarrow \nu + e^{+/-}$$
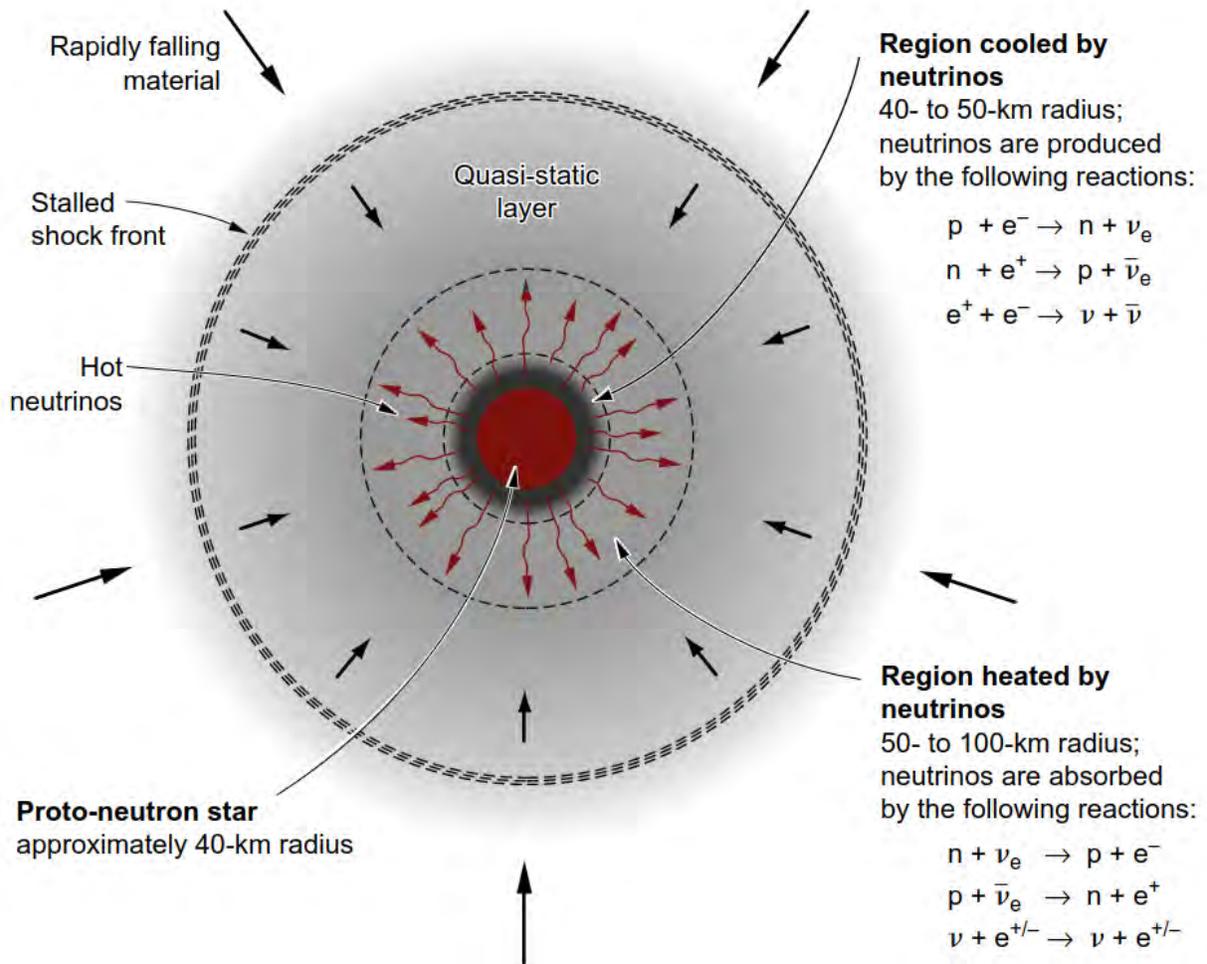
Figure 1.2 Schematic at time of shock stall. The core now consists of two regions: a cool, inner core that is unshocked material, and a hot, shocked, outer core. The outer core is cooled by neutrino emission. Further out, beneath the stalled shock, net neutrino heating occurs (the gain region) due to charged-current neutrino absorption. Figure from Herant et al. (1997).

idea that CCSNe are fundamentally multidimensional phenomena. Therein lies the problem: multidimensional CCSN simulations with all the bells and whistles (see Section 1.5) are computationally expensive. These facts have lead to the development of so-called *effective* supernova models – treatments meant to, in spherical symmetry, drive an explosion that might mimic some aspects of the full supernova problem. One such effective model, STIR (Supernova Turbulence In Reduced dimensionality, Couch et al. (2020)) is featured heavily in this Dissertation. In STIR, the impacts of turbulence and convection are included in spherical symmetry in a parameterized way. By including terms in the evolution equations from a modified mixing length theory, the effects of turbulent and convection motions can be captured with only a few free parameters. These free parameters are fit to full 3D simulations, and the result is a spherically symmetric model capable of producing explosions that are quite similar to 3D.

There are other effective explosion models, such as PUSH (Ebinger et al., 2017, 2019), which modifies heavy lepton neutrino energetics to emulate the impact of convection on neutrino heating, and PHOTB (Ugliano et al., 2012; Sukhbold et al., 2016, and references therein) which parameterized the neutrino luminosity. While the results from these works are sensitive to the effective model, and comparison made harder by complications from stellar evolutionary modeling, one key point has been made clear: the fate of a star undergoing core-collapse is not a monotonic function of its ZAMS mass. That is, there exists no mass cut separating neutron star and black hole formation. Instead, there exist so-termed *islands of explodability*, where failed supernovae (and so black hole formation) are dispersed between successful explosions, like dark seas amongst the landscape of explosions. Figure 1.3 shows one such result obtained with the STIR model. Plotted are explosion results – green for a successful explosion, black for failure and black hole formation. The different horizontal bands are results for differing values of the mixing length parameters α_Λ which scales the strength of convection in the model (1.25 is the fiducial value fit to 3D simulations). Notably, black hole formation is not cleanly separated from successful supernova explosions. The question of exactly which stars explode, and how that might be predicted, remains one of the largest open questions in the field.
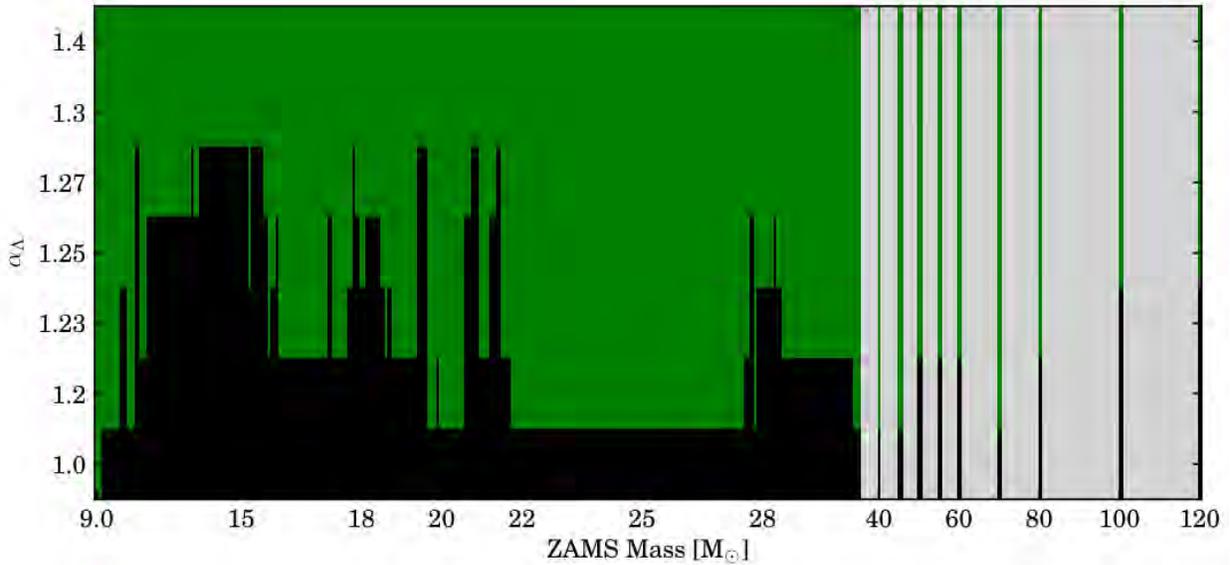
Figure 1.3 Landscape of explodability for progenitors with masses from 9-120$M_\odot$. Green represents a successful explosion and black a failed explosion. Vertical bars (values of α_Λ) are different values of the mixing length parameters. Figure from Couch et al. (2020).

For in-depth reviews of the CCSN mechanism, there are a host of review articles, e.g., Mezzacappa (2001, 2005); Janka et al. (2012, 2016); Burrows (2013); Hix et al. (2014); Müller et al. (2016); Couch (2017); Pejcha (2020); Müller (2020); Mezzacappa et al. (2020); Burrows & Vartanyan (2021); Mezzacappa (2022).

1.3 Observational Characteristics

Core-collapse supernovae are fundamentally multimessenger events. Their emission spans all parts of the electromagnetic spectrum and includes both neutrino and gravitational wave (GW) emission. Each of these emission channels provides key insights into the life and death of the star.

1.3.1 Electromagnetic Emission

Observed electromagnetic emission falls intro two camps: photometry and spectra. The former is energy integrated measurements, often constructing bolometric (total luminosity) and broadband light curves whereas spectra yield flux as a function of energy or wavelength. Photometry provide constraints on bulk properties: ejected mass, explosion energy, synthesized ^{56}Ni, and the stellar radius (however, as I will discuss in Section 1.4, such constraints are difficult and degenerate).
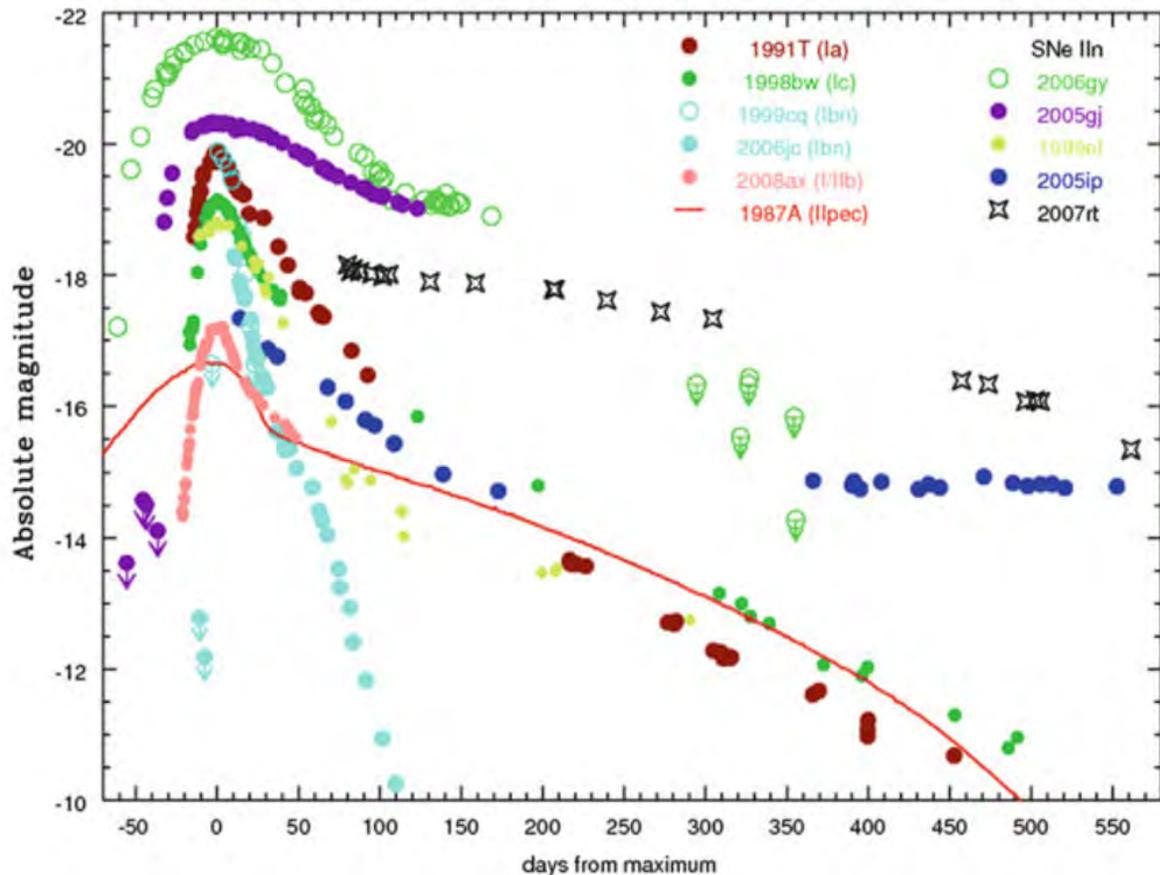
Figure 1.4 Comparison of supernova light curves. Figure from Trundle et al. (2009).

Spectra provide fine tuned details about the stellar composition and velocity of ejected matter. Spectra give more information than photometry at the cost of being more difficult to measure. Both are used to construct supernova classifications, with the primary classification indicating the presence (Type II) or absence (Type I) of hydrogen in the ejecta. In Figure 1.4, I show a sample of SN light curves including both Type I and Type II as well as subclasses. Notably, there is an enormous diversity even in the energy integrated measurements, providing insights into the diversity of supernova progenitors. For all observed supernova throughout history, all but one have been detected only through electromagnetic emission, highlighting its importance as a messenger.

Most of the work in this Dissertation is focused on a particular subclass of Type II SNe known as Type IIP SNe, where "P" here denotes "plateau." These are the most common class of CCSNe,

representing approximately half of all observed CCSNe (Li et al., 2011). These SNe are now known to originate from red supergiant (RSG) progenitors with extended hydrogen rich envelopes, ZAMS masses less than about thirty solar masses, and radii between 100 and 1000 solar radii (Smartt, 2009). The actual upper mass limit for Type IIP SNe is highly contended (see, e.g., Smartt, 2009; Davies & Beasor, 2018, 2020; Morozova et al., 2018; Martinez et al., 2022, and references therein), with the ambiguity termed the "red supergiant problem", and is the subject of Chapter 3. Even amongst this most common subclass of CCSNe, there is tremendous diversity of observed transients Anderson et al. (2014); Valenti et al. (2016); Gutiérrez et al. (2017a,b). Owing to their commonality and observed diversity, understanding SNe IIP is a crucial step to understanding SNe as a whole.

The physics of their light curves is well understood. The shock launched from the initial explosion will breakout from the stellar surface in roughly 24 hours. This breakout produces a bright flash (the shock-breakout signal) in the ultraviolet (UV) and X-ray, whose properties depend mainly on the stellar radius and shock temperature. Only a handful of shock-breakout events have been detected for any class of CCSNe, and none so far for SNe IIP. Following shock-breakout, the shock-heated ejecta expands and cools. As the outer layers cool below 10,000K, shock-ionized hydrogen begins to recombine into neutral hydrogen. A recombination waves being to move inward, with the recombined, neutral material being optically thin and the inner ionized material optically thick. As a result, the photosphere follows the recombination wave, moving inward in mass but staying roughly constant in radius. As hydrogen recombination at fixed composition occurs at fixed temperature, the photospheric temperature remains roughly constant. As a result, the bolometric luminosity remains constant, i.e.,

$$L = 4\pi R^2 \sigma T^4 \tag{1.1}$$

for luminosity L, stellar radius R, Stefan-Boltzmann constant σ, and surface temperature T. Thus, this recombination wave produces a flat "plateau" in the observed light curve lasting around 100 days. The optically thick plateau phase ends as the recombination wave reaches the inner boundary of the hydrogen envelope. Here, the luminosity rapidly drops and falls into linear decline, now
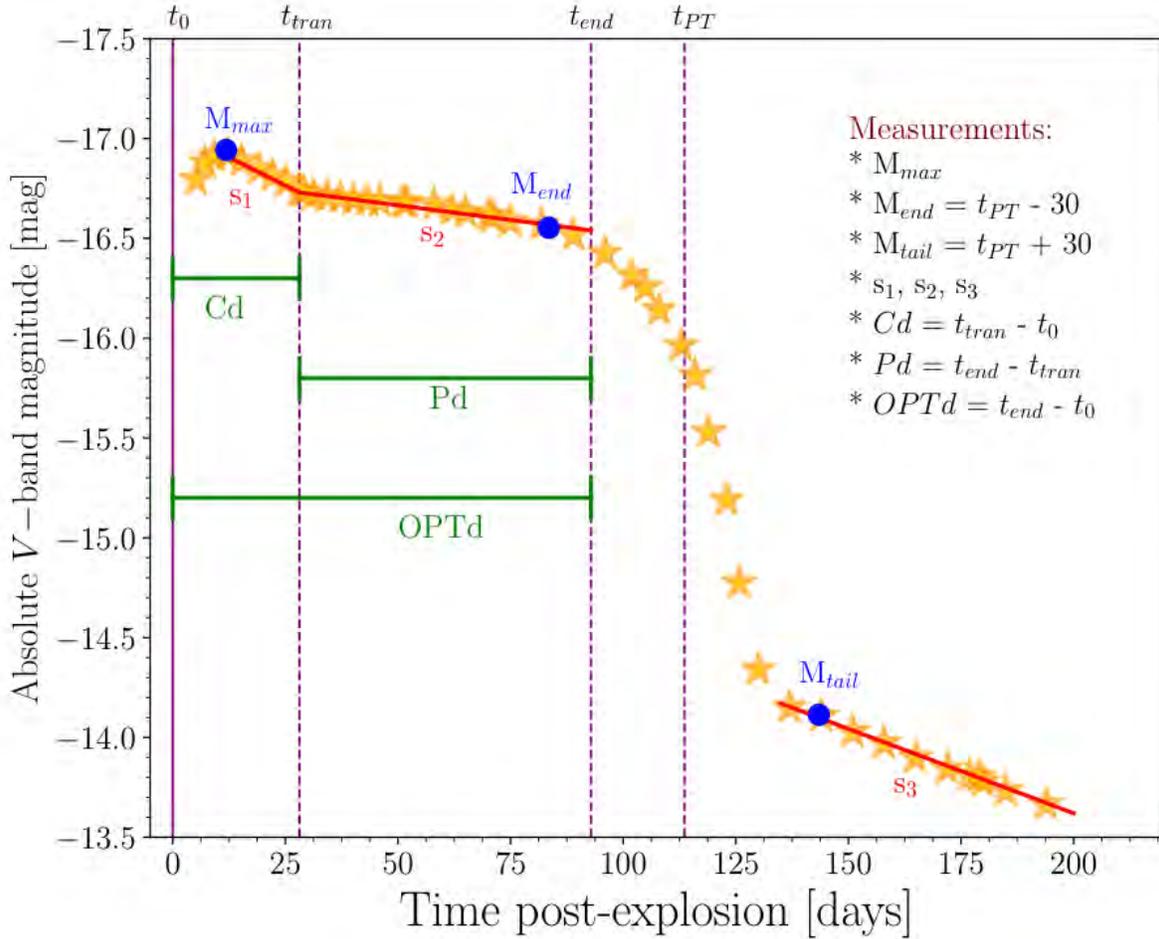
Figure 1.5 Schematic of a Type IIP SN light curve. Figure from Gutiérrez et al. (2017b).

powered by radioactive decay of ^{56}Ni. Figure 1.5 shows a schematic of a Type IIP SN.

1.3.2 Neutrino and Gravitational Wave Emission

Core-collapse supernovae are copious emitters of neutrinos. Owing to electron captures on both free protons and nuclei during core collapse, neutrinos are produced in vast numbers. As collapse proceeds and central densities rise beyond about 10^{12} g cm^{-3}, neutrinos become trapped in the core. Throughout the following seconds during which the supernova matures, neutrinos diffuse out of the core until reaching the neutrinospheres around 10^{11} g cm^{-3} and can move mostly uninhibited through the star. These neutrinos, emitted from deep in the stellar interior, carry information about the environment there.

To date there has been only one detection of supernova neutrinos: SN 1987A (Arnett et al., 1989). The serendipitous detection of roughly 20 neutrinos from this event confirmed the results of early supernova models and set the stage for the next decades of work. The low mean energies and long time span confirmed that the observed neutrinos diffused out of a dense inner core.

From core bounce onward, the proto-neutron star (PNS) rings with gravitational waves (Kotake, 2013; Mezzacappa & Zanolin, 2024). At core bounce, rotating stars (that is, all stars) produce a loud burst as the bounce deforms the PNS and creates a time dependent mass quadrupole moment. Later, accretion onto the PNS excites smaller amplitude, stochastic modes that emit gravitational waves. These waves, emitted by vibrations of the PNS, carry information about the size and stiffness of the PNS and would potentially help constrain the equation of state of dense, nuclear matter. However, such GW sources are much quieter than those produced by merging compact objects and already detected by LIGO. Much worse than the case for CCSN neutrinos, the current suite of detectors (aLIGO, VIRGO, and KAGRA) can only detect GWs from a CCSN if it occurs within a distance of about 100 kpc (Abbott et al., 2016). As such, there have been no detected GWs from CCSNe.

The community waits tirelessly for the next galactic core-collapse supernova.

1.4 Interpreting Observations

Our understanding of CCSNe is driven by the union of theory and observation. For any CCSN observation, be it photometry, spectra, neutrino, or GW, an underlying model is required to discern any information besides what is directly measured. For supernova light curves, these models fall into two categories: analytic models and numerical models.

1.4.1 Analytic Models

Analytic models for interpreting CCSN light curves rely on making a set of simplifying assumptions that allow one to relate fundamental properties such as progenitor mass and explosion energy to easily observable properties such as a characteristic luminosity, timescale, and expansion velocity. To construct such as model, the number of simplifications required is practically uncountable.

The earliest successful analytic models for Type II SNe are owed to (Arnett, 1980; Arnett & Fu,

1989). The closely related model of Popov (1993) has become commonly adopted. In this model they describe the plateau phase of a Type II SN using the following primary assumptions:

- the matter is spherically symmetric

- the envelope conforms to a two zone opacity model – an optically thick, inner region and a thin, inner region

- the density profile is uniform in space

- the envelope is expanding homologously (with velocity proportional to radius)

- all opacity comes from electron scattering

- the envelope is pure hydrogen (zero metallicity)

- heating due to radioactive decay is negligible.

In practice, none of these assumptions are satisfied perfectly, and it is unclear under what conditions the model works or fails. Under these assumptions, the ejecta velocity can be approximated as

$$v \approx \left(\frac{10 E_{\text{expl}}}{3M} \right)^{1/2} \tag{1.2}$$

for explosion energy E_{expl} and ejecta mass M, where it is implicitly assumed that the kinetic energy of the ejecta is approximately the energy generated in the initial explosion. From here, the analysis continues until the explosion energy and ejecta mass can be expressed in terms of key observables

$$\log_{10} \left(\frac{E_{\text{expl}}}{10^{51} \text{erg}} \right) = \boldsymbol{\theta}_{\text{e}} \cdot \boldsymbol{O} + \theta_{\text{c,e}} \tag{1.3}$$

and

$$\log_{10} \left(\frac{M_{\text{ej}}}{M_{\odot}} \right) = \boldsymbol{\theta}_{\text{m}} \cdot \boldsymbol{O} + \theta_{\text{c,m}} \tag{1.4}$$

where $\boldsymbol{O} = \{\log_{10}(L_{50}), \log_{10}(t_p), \log_{10}(v_{50})\}$ are the observed plateau luminosity, duration, and ejecta velocity, and $\boldsymbol{\theta}_e$, $\boldsymbol{\theta}_m$, $\theta_{c,e}$, and $\theta_{c,m}$ are power law coefficients. From Popov (1993), the parameters $\boldsymbol{\theta}_o$ and $\theta_{c,e}$ are $\{0.4, 4.0, 5.0\}$ and -4.311, respectively (e.g., Pejcha & Prieto, 2015,

13
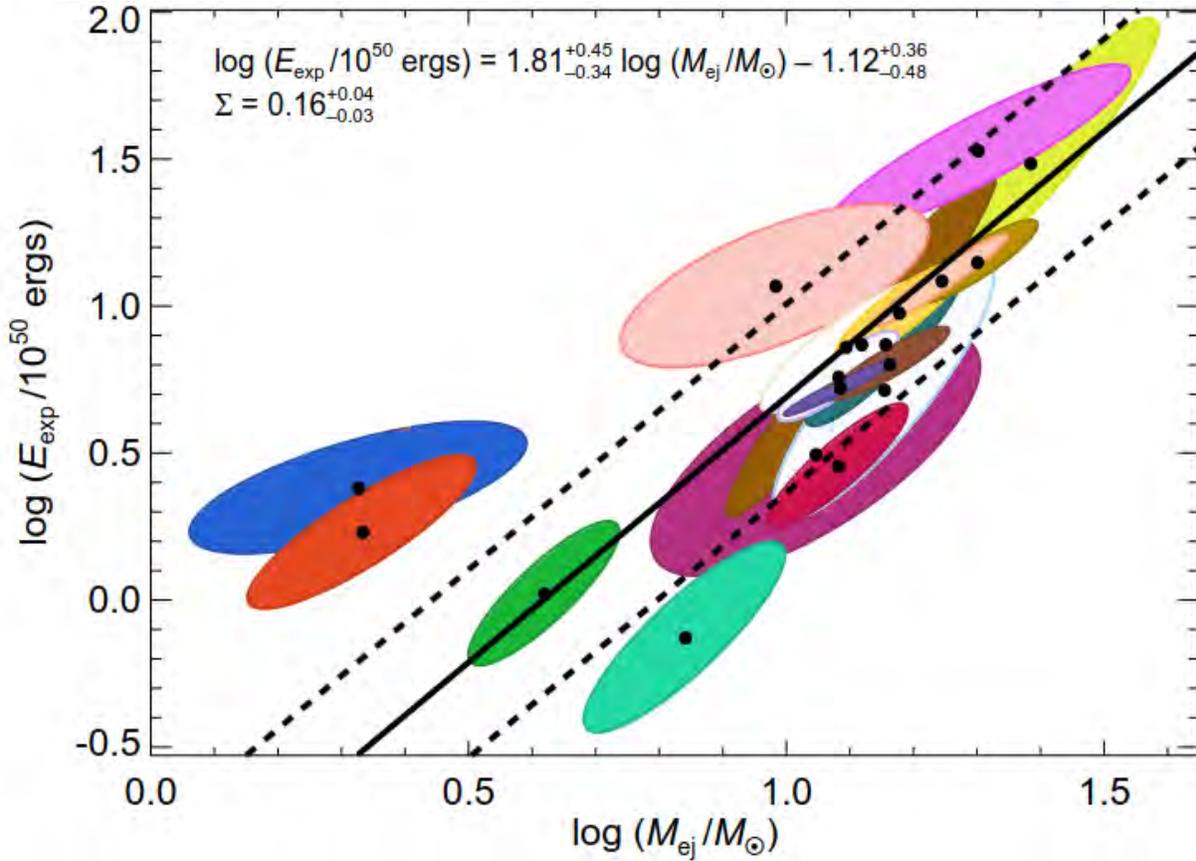
Figure 1.6 Estimated explosion energy vs estimated ejecta mass for a suite of observed supernovae with error estimates (shaded regions). Figure from Pejcha & Prieto (2015).

where these values assume the luminosity is instead a magnitude). There is some ambiguity in the above, as it is not clear from the original work of Popov (1993) exactly *what* the ejecta mass refers to. Some works have considered it the full ejected mass (roughly defined as the mass above the iron core) and others considered it the mass of the hydrogen envelope. In some cases, the two may differ substantially. The literature has yet to reach a consensus on this question. More modern approaches have been taken to improve upon this approach, e.g., Kasen & Woosley (2009).

Such a model could, ideally, provide explosion energy estimates for virtually every observed supernova. This result has been used broadly to estimate explosion energies of observed SNe. An example of such a landscape is shown in Figure 1.6.

1.4.2 Numerical Models

Numerical models, in contrast to analytic models, rely on fewer assumptions. The trade off is that now the relationships remain partial differential equations and require numerical solution. Historically, such models are often called "hydrodynamic models" in the literature[1]. Early examples include Falk & Arnett (1973, 1977); Litvinova & Nadezhin (1985).

These models include a treatment of spherically symmetric hydrodynamics, typically using a Lagrangian (comoving with mass elements) method. The hydrodynamics are coupled to radiation transport, most often using a flux-limited diffusion approach. A simple, analytic equation of state appropriate for stellar plasmas is adopted to close the hydrodynamic equations (e.g., that of Paczynski (1983) is a common choice). A set of radiative photon opacities is chosen, often gray (energy integrated). Finally, a prescription for heating due to radioactive decay of $^{56}Ni \rightarrow {}^{56}Co \rightarrow {}^{56}Fe$ is needed. A common method is that of Swartz et al. (1995), which treats the gamma-ray radiative transfer in the gray diffusion limit.

This physical model is insufficient to capture the explosion mechanism as laid out in Section 1.2. Lacking an equation of state appropriate for dense nuclear matter, the core will never reach the repulsive regime of the strong nuclear force, and core bounce will never be realized. Worse still, without neutrino radiation transport, there is no heating source to power an explosion. In lieu of a self consistently driven explosion, these models parameterize away the source of the explosion, placing a user defined amount of energy in the core to initiate the explosion. In this way, one has control over the explosion energetics and may explore light curves in a controlled fashion. A common workflow is then: given one or more progenitor models, select a set of explosion energies to to use with them, as well as masses of synthesized radioactive ^{56}Ni (as this requires a nuclear reaction network to create self consistently). The result is a grid of artificial explosion models in progenitor mass, explosion energy, and nickel mass, each with a resultant light curve which may be compared to observations. The method is not without cost: there is no way to determine if a given progenitor may explode with a given energy (or at all) without more physics.

---

[1]Ironically, such models treat coupled radiation hydrodynamics.
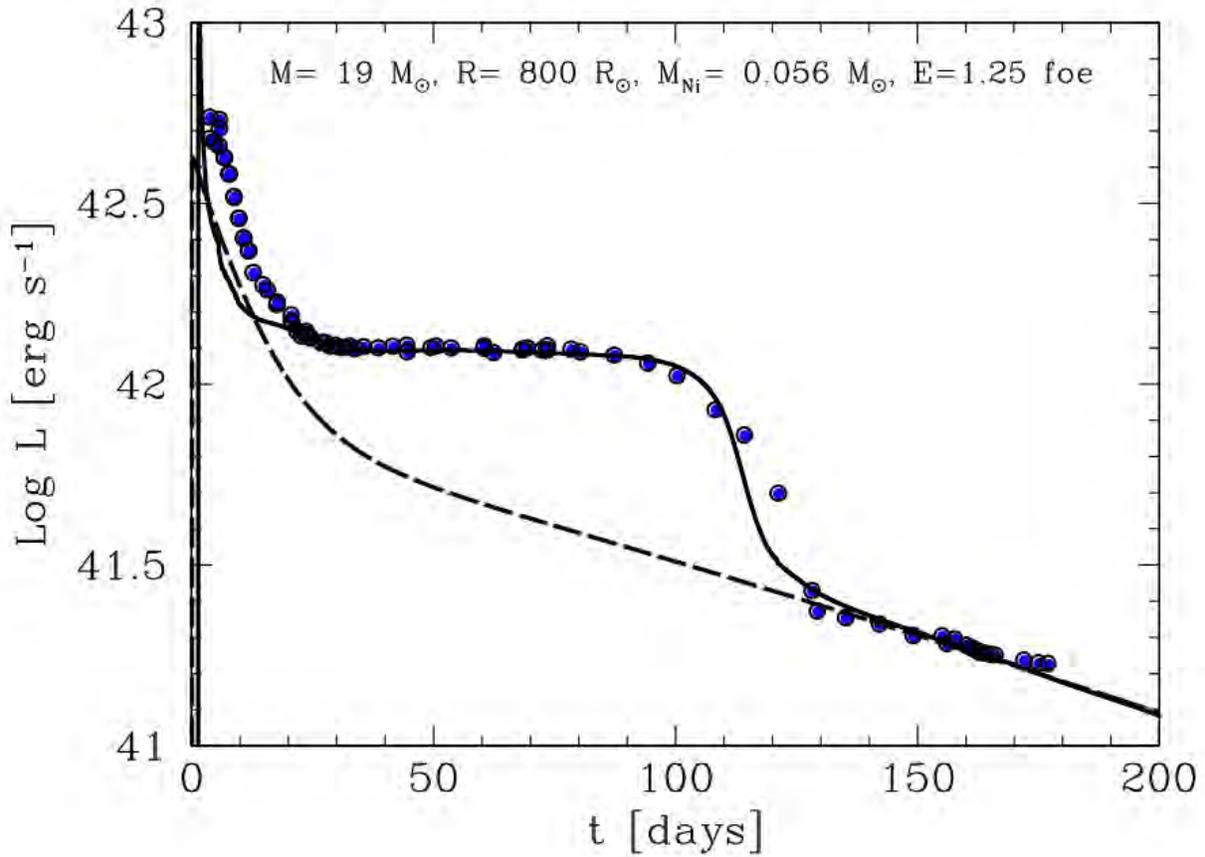
Figure 1.7 Bolometric light curve for a radiation hydrodynamic model (solid line) compared to observational data for SN 1999em (blue dots). Dashed line shows luminosity due to radioactive decay. Figure from Bersten et al. (2011).

Figure 1.7 shows an example of such a calculation. The synthetic bolometric light curve model found to best match SN 1999em (blue dots) is shown with the black line. This particular work (Bersten et al., 2011) finds a 19M$_\odot$ model with 1.25 foe (1 foe = 10^{51} erg) explosion energy model best reproduces the well-observed SN 1999em.

One such numerical model for SN light curves which is featured heavily in this Dissertation is SNEC[2], the SuperNova Explosion Code (Morozova et al., 2015). SNEC models spherically symmetric photon radiation hydrodynamics with a gray, flux limited diffusion approach. SNEC includes Newtonian hydrodynamics using a finite differencing scheme with artificial viscosity to

---

[2]SNEC is publicly available https://stellarcollapse.org/SNEC.html

captures shocks (Mezzacappa & Bruenn, 1993). It includes the stellar equation of state of Paczynski (1983) which includes contributions from radiation, ions, and electrons with approximate electron degeneracy. The equation of state is coupled with a Saha ionization solver for the ionization state. Hydrodynamic mixing of material is approximated with a boxcar smoothing algorithm applied at the beginning of the simulation. Gray radiative transfer for the gamma-rays produced from ^{56}Ni and ^{56}Co decay is followed using the approach of Swartz et al. (1995). Positrons, which may sometimes be produced from the decay of ^{56}Co are not included – an error of perhaps a few percent is incurred. Finally, SNEC has the option to initiate explosions using a thermal bomb. This Dissertation uses SNEC heavily, often coupled to more sophisticated simulations to eliminate the need for a thermal bomb.

1.4.3 Degeneracies

The synthetic light curve models of the preceding section provide enormous freedom for exploring explosion models and connecting to observations. With that freedom, however, has crept a snake. In Nature, a given star will explode in a specified, deterministic way. By arbitrarily specifying the explosion energy as a free parameter we have allowed for potentially unphysical explosions that may never be realized in Nature.

Worse still, these synthetic light curves are degenerate with each other. Different combinations of explosion energy and progenitor mass produce identical light curves. For example, the plateau luminosity and duration scale as

$$L_{pl} \propto M_{\rm ej}^{-1/2} E_{\rm expl}^{5/6} R^{2/3} X_H \tag{1.5}$$

$$t_{pl} \propto M_{\rm ej}^{1/2} E_{\rm expl}^{-1/4} R^{1/6} X_H^{1/2} \tag{1.6}$$

for ejecta mass $M_{\rm ej}$, explosion energy $E_{\rm expl}$, stellar radius R, and envelope hydrogen mass fraction X_H (Kasen & Woosley, 2009). The seeds of the problem are shown here: treating progenitor and explosion energy as independent, a given light curve plateau may be achieved by adjusting the stellar progenitor (ejecta mass and stellar radius) *or* the explosion energy.

Only recently have these degeneracies been explored (Goldberg et al., 2019; Dessart & Hillier,
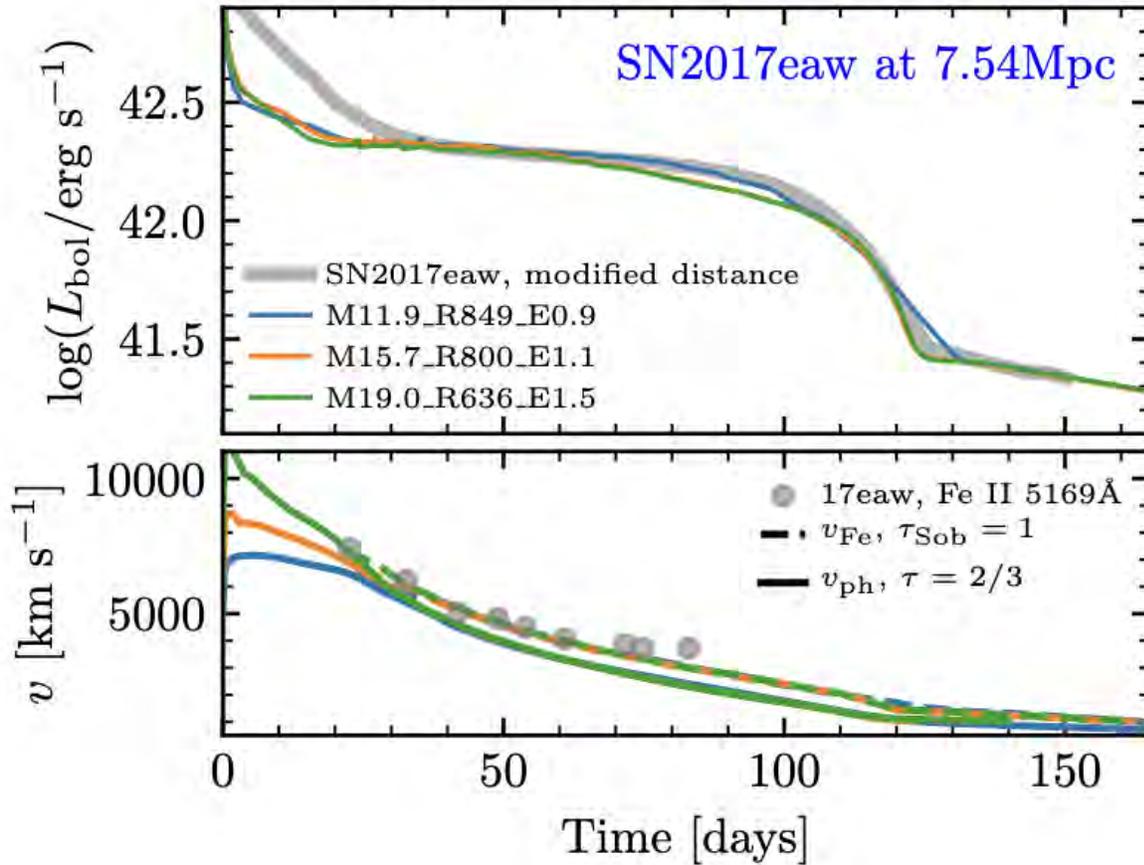
Figure 1.8 Light curves and iron line velocities for observations (gray) and best fit models (colored lines) for SN 2017eaw. Figure adapted from Goldberg & Bildsten (2020).

2019). Here, it was shown that there exist *families* of light curve models that match a given observation. Artificial explosion models with progenitor ZAMS masses spanning 10 solar masses or more can easily reproduce a given observation. The ability of thermal bomb models to constrain observations is severely limited. Figure 1.8 shows an example of such a set of models for SN 2017eaw. Bolometric light curves (top) and iron line velocities (bottom) are shown for observational data of SN 2017eaw (gray) and best fit models (colored lines). The three models presented here span approximately $10M_\odot$ in ZAMS mass and 0.6 foe in explosion energy.

Additional measurements might help to constrain some relevant stellar properties and thus reduce the scale of the parameter space and lessen the degeneracies. One such option involves constraining the stellar radius. Such constraints limit the range of progenitors and reduce the size of

the parameter space. This was demonstrated in Goldberg & Bildsten (2020) to successfully reduce degeneracies, but far from completely. The prospects are made more difficult when uncertainties on the radius inferences are pulled in, which in practice can be quite large. It is an area of active work to identify other measurements which might help to reduce these degeneracies.

At the core of this degeneracy issue is the treating of the explosion itself as independent of the stellar structure. In reality, the explosion energy is fully determined by the stellar profile $E_{\text{expl}} = f(\mathcal{P}(M_{\text{ej}}, R))$ where $f(\mathcal{P}(M_{\text{ej}}, R))$ is some function of the stellar progenitor $\mathcal{P}$[3]. The exact relationship between the explosion energy and the progenitor properties (the so-called explosion landscape) remains an area of active research (Pejcha & Thompson, 2015; Perego et al., 2015; Ebinger et al., 2017; Sukhbold et al., 2016; Couch et al., 2020). There is some hope, then, that by treating the explosion self consistently, instead of artificially generating it, that this issue might be resolved, at least in part. That hope lies at the core of this Dissertation.

In summary, there is *no* model-independent way to interpret supernova observations. Moreover, with the current suite of models, matching an observation is a necessary, but not sufficient condition for inferring explosion and progenitor properties.

1.5 Modeling Supernovae

Modeling core-collapse supernovae from first principles requires following a huge range of tightly coupled physics across a range of spatio-temporal scales. The choice of what physics is included as well as how that physics is discretized determines the scope of the resulting model.

Despite the complex, multiphysical nature of CCSNe, the recipe for modeling them is relatively straightforward. One needs to include a treatment of (magneto)hydrodynamics, neutrino radiation transport, general relativistic gravity, and a suite of microphysics including a dense matter equation of state and neutrino opacities.

For hydrodynamics we solve the Euler equations, given below in the non-relativistic limit:

---

[3]In reality the ejecta mass and stellar radius do not directly impact the explosion energy, instead they are products of stellar evolution. Indeed, the picture may be further complicated by the fact that the exact details of the ejected mass depend on knowledge of the explosion dynamics. Stellar evolution sets the *ejactable* mass and the details of the resulting explosion determine exactly what is ejected. In spherical symmetry such complications are unimportant.

$$\partial_t \rho + \frac{1}{\sqrt{\gamma}} \, \partial_i \left(\sqrt{\gamma} \, \rho \, v^i \right) = 0, \tag{1.7}$$

$$\partial_t (\rho \, v_j) + \frac{1}{\sqrt{\gamma}} \, \partial_i \left(\sqrt{\gamma} \, \Pi^i_{\ j} \right) = \frac{1}{2} \, \Pi^{ik} \, \partial_j \gamma_{ik} - \rho \, \partial_j \Phi, \tag{1.8}$$

$$\partial_t E + \frac{1}{\sqrt{\gamma}} \, \partial_i \left(\sqrt{\gamma} \, [\, E + p \,] \, v^i \right) = -\rho \, v^i \, \partial_i \Phi, \tag{1.9}$$

$$\partial_t (\rho y_e) + \frac{1}{\sqrt{\gamma}} \, \partial_i \left(\sqrt{\gamma} \, \rho y_e \, v^i \right) = 0, \tag{1.10}$$

for density ρ, three velocity v^i, energy E, pressure p, gravitational potential Φ, electron fraction y_e, coordinate metric γ, and stress tensor $\Pi^i_{\ j} = \rho \, v^i \, v_j + p \, \delta^i_{\ j}$. The metric γ, in this form, encapsulates coordinate systems (Cartesian, spherical, and so on). These equations describe conservation of mass, momentum, energy, and lepton number. This set of equations is closed by the equation of state $p = p(\rho, T, y_e)$. Depending on the scientific questions one wants to answer, magnetic field evolution may be included. Additional complications are gained in the case of special or general relativistic hydrodynamics (for example, the so-called Valencia formulation Banyuls et al., 1997).

Neutrinos are implemented by assuming that they are massless and thus behave according to traditional radiation transport approaches. The assumption of masslessness is justified as the typical neutrino energy in CCSNe of $\mathcal{O}(10 \text{ MeV})$ is much larger than the neutrino rest mass which, while unknown, is known to be small. The species-dependent neutrino distribution function $f_\nu(x^\alpha, p^\alpha)$, for 4-position and 4-momentum, x^α and p^α, evolves according to the 6+1 Boltzmann equation

$$p^\alpha \left[\frac{\partial f_\nu}{\partial x^\alpha} - \Gamma^\beta_{\alpha\gamma} p^\gamma \frac{\partial f_\nu}{\partial p^\beta} \right] = \left[\frac{df_\nu}{d\tau} \right]_{\text{coll}} \tag{1.11}$$

where $\Gamma^\beta_{\alpha\gamma} p^\gamma$ are the Christoffel symbols and the right hand side is the collision term including neutrino-matter interactions. This must be solved for the distribution function f_ν of *each* neutrino species. Full solution of the 6+1 Boltzmann equation in dynamical environments remains computationally intractable and simplifications must be made. For the case of CCSNe, where the dynamics are so sensitive to the fidelity of neutrino transport, there are several considerations when choosing an appropriate approximation to the transport. Firstly, supernovae are not spherical and the transport implementation must be multidimensional as well. Furthermore, neutrinos emitted

20

from the PNS are not in equilibrium with matter necessitating a multigroup transport method. As both neutrino-matter absorption opacities and the neutrino heating rate scale as the square of the neutrino energy E_ν^2, the emitted neutrino spectrum must be captured. With these needs in mind, the state of the art for neutrino transport is a two-moment approach (Thorne, 1981; Shibata et al., 2011). In this approach, one models the evolution of the (frequency dependent) energy density E and flux F_i, obtained by taking angular moments of the neutrino distribution function:

$$\partial_t(\sqrt{\gamma}E) + \partial_i[\sqrt{\gamma}(\alpha F^i - \beta^i E)] + \partial_\epsilon[\epsilon(R_t + O_t)] = G_t + C_t, \qquad (1.12)$$

$$\partial_t(\sqrt{\gamma}F_i) + \partial_j[\sqrt{\gamma}(\alpha P_i^j - \beta^j F_i)] + \partial_\epsilon[\epsilon(R_i + O_i)] = G_i + C_i, \qquad (1.13)$$

where α is the lapse, β is the shift, γ is the determinant of the three-metric, G_μ, C_μ are the source terms due to geometric and matter effects, and R_μ, O_μ are gravitational redshifting and observer correction terms.

The radiation pressure tensor P_{ij} is required to close the truncated moments. Unlike the hydrodynamics, where the pressure used to close the system of equations is given through the equation of state, the pressure tensor here must be more arbitrarily prescribed. While there are a number of so-called closures for two moment radiation transport, the general approach is to construct a scheme that shows the correct behavior in the diffusive and free streaming regimes and interpolates between them in intermediate optical depths (see, e.g., Murchikova et al., 2017; Richers et al., 2017, for reviews of these closure methods).

1.5.1 Numerical Methods

With a physical model in hands, the equations must be discretized to allow for numerical solution. For supernova modeling, most of the physics takes the form of hyperbolic partial differential equations (PDEs), with gravity (elliptic) and diffusion (parabolic), such as for diffusive radiation, being exceptions. This is the case for the hydrodynamics and radiation transport as described previously. Here I lay out some of the basic properties of these hyperbolic PDEs and common

ways of solving them numerically. Focus is given to hyperbolic PDEs as these are ubiquitous in computational astrophysics and, in some sense, have the most strict requirements of discretization.

Hyperbolic PDEs describe, amongst other things (but most importantly), conservation laws and are widely used to model, for example, wave motion and transport. These equations take the form

$$\partial_t \mathbf{U} + \boldsymbol{\nabla} \cdot \mathbf{F}(\mathbf{U}) = \mathbf{S}(\mathbf{U}) \tag{1.14}$$

for a, in general, vector of conserved quantities $\mathbf{U}$, flux vector $\mathbf{F}(\mathbf{U})$, and source vector $\mathbf{S}(\mathbf{U})$ (noting that for strict conservation laws, the source term is $\mathbf{0}$). This form of hyperbolic PDEs, while somewhat specialized from the most general form, describes a vast amount of physics. Let $\mathbf{U} = \rho_c$ be charge density and $\mathbf{F} = \mathbf{J}$ be current density and this expresses conservation of charge. Take $\mathbf{U} = (\rho, \rho v, E)^{\mathrm{T}}$ and $\mathbf{F} = (\rho v, \rho v^2 + p, E + p)^{\mathrm{T}}$ and we obtain conservation of mass, momentum, and energy, i.e., non-relativistic equations of hydrodynamics in the absence of gravity. The mathematical theory of hyperbolic PDEs is rich and could fill the entire text of this Dissertation on its own, so I refer the interested reader to some of the great texts available that cover this topic (e.g., LeVeque, 1992, 2002; Larsson & Thomee, 2003; LeVeque, 2007; Toro, 2009a).

With a system of PDEs in hand, they must be prepared for numerical solution. As computers are inherently discrete and may only represent a finite number of states, complicated operations such as integration and differentiation must be replaced by discrete approximations. The most common such discretization is the finite difference derivative, i.e.,

$$\partial_x f(x) \approx \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{1.15}$$

for some small Δx. This simple approximation forms the basis for finite difference PDE solvers and allows us to introduce an important concept: convergence rate. Taylor expanding $f(x)$ and applying the finite difference stencil, one can see that it is accurate up to a term proportional to Δx. That is, the error scales as $O((\Delta x)^1)$, and the approximation is said to be first order accurate. In general, for an n-th order accurate numerical method, the discretization error scales as

$$E \propto (\Delta x)^n \tag{1.16}$$

such that increasing the resolution by a factor of 2 reduces the error by a factor of 2^n. What this shows, then, is that by moving to a higher-order accurate numerical method, one may use reduced resolution (larger Δx) without compromising accuracy. This is beneficial for a number of reasons: reducing the numerical resolution generally allows for faster time to solution, as there are fewer grid points to perform the update on, the memory footprint is reduced, and the simulation can take larger timesteps as the limiting timescale is the minimum sound crossing time, $\Delta t \propto \Delta x / c_s$ for sound speed c_s (the constant of proportionality here is referred to as the Courant-Friedrichs-Lewy (CFL) factor and must be less than unity). Furthermore, in applications involving turbulence – common in astrophysical settings – a very high resolution is required to capture the turbulent behavior. High-order methods can vastly reduce the required resolution to model turbulence faithfully.

With these concepts in hand, we turn to the discretization of our hyperbolic PDE, Equation 1.14. Unlike ordinary differential equations (ODEs), which contain a derivative in only one variable and may be directly integrated with, e.g., a Runge-Kutta integrator, PDEs require extra care. The common practice for integrating hyperbolic PDEs is known as the method of lines approach. In this approach, the flux term $\nabla \cdot \mathbf{F}(\mathbf{U})$ is discretized in space with a standard approach such as finite differences, or others, to be discussed below. Equation 1.14 becomes, for example,

$$\partial_t \mathbf{U} + \mathcal{L}_\nabla \mathbf{F}(\mathbf{U}) = \mathbf{S}(\mathbf{U}) \tag{1.17}$$

where $\mathcal{L}_\nabla$ is some discretization operator (for example, the finite differencing introduced previously). Equation 1.17 is called the semi-discrete form, as the spatial components are discretized while the time component remains continuous. This, then, is an ODE and can be evolved in time with standard ODE integration methods such as Runge-Kutta methods.

Most of the complexity of a PDE solver lies in the choice and implementation of $\mathcal{L}_\nabla$. PDE solvers are grouped, broadly, into a few classes. At the highest level, there are grid based methods and gridless methods. Grid based methods are, by far, most common. Gridless methods include, for example, smoothed particle hydrodynamics. Grid based methods are classified as structured or unstructured. Structured grids are most common in astrophysical applications, with unstructured grids being more prevalent in engineering disciplines, and these grids may be constructed from

triangles or tetrahedra, for example. For hyperbolic PDEs, there emerge three primary categories of grid based discretizations: finite difference, finite volume, and finite element methods (although there are some notable extensions to this list).

Finite difference schemes have more or less been covered already. Derivative operators are discretized following Equation 1.15 and integrated in time (often, even the time derivative is discretized in this way). Generally, higher-order accurate finite difference stencils are chosen, as Equation 1.15 is only first-order accurate. While finite difference methods have lost favor in the solution of hyperbolic PDEs for e.g., fluid dynamics, they remain one of the central players in parabolic and elliptic PDEs, such as for initial data solvers in numerical relativity (see some of the many great texts, e.g., Alcubierre, 2008; Baumgarte & Shapiro, 2010).

Finite volume methods, owing largely to Godunov (1959), are the primary class of hyperbolic PDE solver in use in computational astrophysics. These methods, in contrast to finite difference methods, are based on the integral form of the PDEs, instead of the differential forms. In what follows, I specialize to one spatial dimension. The computational domain $\mathcal{D}$ is decomposed into N uniform zones of width Δx. The full domain $\mathcal{D} = [x_L, x_R]$ is then the union of N non-overlapping computational cells (or zones), each indexed as x_i. The left and right boundaries of each of these cells is denoted as half integer indices, e.g., $x_{i-1/2}$, $x_{i+1/2}$, respectively. On each of these finite volumes, the solution $\mathbf{U}$ is considered to be constant. Figure 1.9 shows an example of a finite volume grid.

Integrating Equation 1.14 over a computational cell and normalizing by cell volume, an integral form of the conservation laws is obtained (assuming for the moment that there are no sources)

$$\frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{1+1/2}} \partial_t \mathbf{U} \, dx = - \int_{x_{i-1/2}}^{x_{i+1/2}} \partial_x \mathbf{F} \left(\mathbf{U} \right) dx \tag{1.18}$$

Pull the time derivative out of the integral on the left, yielding the time rate of change of the cell average, and apply the divergence theorem to the right hand side, giving:

$$\partial_t \langle \mathbf{U} \rangle_i = -\frac{1}{\Delta x} \left[\mathbf{F}^*(\mathbf{U})|_{x_{i+1/2}} - \mathbf{F}^*(\mathbf{U})|_{x_{i-1/2}} \right] \tag{1.19}$$
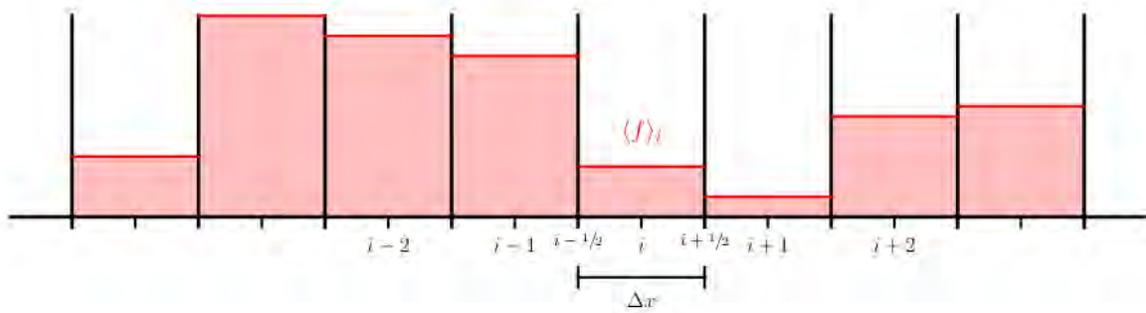
Figure 1.9 Example of a finite volume grid. Each cell stores the average value $\langle f_i \rangle$. Figure adapted from *Introduction to Computational Astrophysical Hydrodynamics*, Open Astrophysics Bookshelf.

Notice that the right hand side is simply a surface integral, and so the time rate of change of the cell average of a quantity **U** changes only by flux into or out of the finite volume. This conservation property is what makes finite volume schemes so popular. Additionally, building the scheme using integral forms of conservative equations allows for weak solution of the underlying PDEs without the need to explicitly introduce artificial viscosity. Notice that above, the fluxes $\mathbf{F}(\mathbf{U})$ were replaced with "numerical flux functions" $\mathbf{F}^*(\mathbf{U})$ evaluated at the cell boundaries. This is due to the fact that in the finite volume picture, quantities are often discontinuous between cells (refer back to Figure 1.9), leading to am ambiguity in evaluating the fluxes there. The practice of evaluating the numerical flux at the cell boundaries is that of solving a so-called Riemann problem which captures various wave families that propagate from discontinuous interfaces. This is one of the two primary freedoms in constructing a finite volume scheme and is the topic of a mountain of literature (Toro, 2009a). The other freedom in constructing a finite volume scheme is the so called reconstruction method. In the finite volume picture, quantities are piecewise constant on computational cells. This turns out to be second order accurate at cell centers, but very inaccurate when evaluating quantities at cell boundaries. Thus, it is common to "reconstruct" the interface values by interpolating to these positions using neighboring cells. Different reconstruction methods have different properties, such as higher orders of convergence or oscillation controlling. A common choice is the piecewise parabolic method that is third order accurate in space (Colella & Woodward,

25

1984). The reconstruction method is part of what determines the (spatial) order of accuracy of a finite volume method[4]. One ill property of high order finite volume reconstructions is that they have a large stencil, that is, they require several neighboring cells, similar to high order finite differences. This has implications for parallel finite volume algorithms.

The final[5] class of PDE solver is known as finite element methods. While uncommon in astrophysical settings, for reasons to be shown, they provide important concepts for a final, hybrid class of methods. In the finite element method, the computational domain $\mathcal{D}$ is divided into N finite elements (conceptually identical to the finite volumes previously used, but with a key difference). On each element, the solution $\mathbf{U}$ is approximated as a basis expansion in some set of basis functions such as Legendre polynomials. The benefit to this is that high order information is trivially obtained without need for reconstruction or fuss with cell averages. Unlike finite volume methods, where the solution is discontinuous across cells, the finite element solutions are required to be continuous across elements. This enforced continuity, while very advantageous for parabolic and elliptic problems, is disastrous for hyperbolic PDEs which, notably, allow for waves and discontinuities. For these reasons, pure finite element methods are seldom used in astrophysical settings where wave phenomena and shocks are common. They have, however, found widespread use in engineering disciplines in part for their straightforward use with unstructured meshes which can conform to material boundaries, for example.

The benefits of finite volume and finite element methods have lead to the development of a hybrid method, so-called discontinuous Galerkin (DG) methods (Reed & Hill, 1973; Cockburn & Shu, 1989; Cockburn et al., 1990, 1989; Cockburn & Shu, 1998). DG methods follow closely to finite element methods, with the distinction that the solution may be discontinuous between elements, much like finite volume methods. Then, similar to finite volume methods, a numerical flux is used to solve the Riemann problem between cells. This union of finite element and volume

---

[4]The formal order of accuracy of a finite volume scheme requires more than just a high order reconstruction, gaining complications from the fact that the cell average and the cell center are not the same (McCorquodale & Colella, 2011).

[5]There are, of course, other classes of methods than the three presented here. These are the fundamentals, and many others combine features of them. Closely related to finite element methods, for example, are spectral methods, which also use globally smooth basis functions.

methods provides local high order accuracy while also allowing for shock capturing schemes (infact, a second order classical DG scheme is nearly identical to a finite volume scheme with a particular reconstruction). They are, potentially, very flexible methods, allowing for refinement in the spatial grid as well as the local order of accuracy. DG schemes are relatively new in the astrophysics setting and have a lot of potential and are the focus of Chapter 5.

While a number of details remain, such as coupling between radiation and hydrodynamics, the discussion here is a fairly complete picture of the essential numerical methods require for CCSN modeling.

1.5.2 Moving Forward

Supernova simulations have become increasingly mature, with the ability to successfully explode now a generic feature and the validity of the neutrino-driven explosion mechanism firmly established. There is, of course, much room for improvement.

On the physics side, the community is only just beginning to move towards proper treatments of general relativistic gravity. There is an enormous amount of work to be done exploring the impacts of magnetic fields in the context of CCSNe in different regimes. Perhaps most importantly, there are many approximations remaining in the neutrino transport sector that been shown, in simplified cases, to be violated. One such case is the presence of muons in the equation of state and their feedback into heavy lepton type neutrino populations. A much harder, and potentially more impactful, problem is that of neutrino fast flavor oscillations. This neutrino flavor instability, which operates on nanometer scales, could drastically alter every aspect of the supernova problem (or none at all), but remains computationally intractable with current methods.

Numerically, there is much interest in moving towards high-order methods (see Chapter 5). Providing a more cost effective treatment of turbulence, high-order methods will be a key tool in future studies of CCSNe. Additionally, as more physics is implemented, there is increasing need for more efficient methods to handle the added physical complexity. The computational cost is managed, in part, by increasingly powerful computational resources. These resources, however, are becoming increasingly heterogeneous accelerators such as GPUs becoming commonplace. This

trend is here to stay, as Moore's law – our ability to pack transistors into chips – is nearing its end. Software must be developed accordingly to take full advantage of these computational resources (such is the focus of Chapter 7). The latter chapters of this Dissertation seek to provide resources to address these points.

1.6 Outline

This Dissertation seeks to improve the ability to interpret CCSN observations and provide open source scientific tools to better model CCSNe. Chapter 2 produces novel light curves from neutrino-driven supernovae and investigates how these light might be used to better constrain observed SNe. Chapter 3 explores how the previous methods, in tandem with Bayesian inference, can be used to constrain entire populations. Chapter 5 presents the hydrodynamics methods for a new code, THORNADO, for modeling supernovae with high-order numerical methods. Chapter 6 presents an open-source tool for performance portable microphysics in fluid and continuum dynamics codes. Finally, Chapter 7 presents PHOEBUS, a new open-source, performance portable, GPU accelerated simulation software for supernovae, accretion disks, and mergers.

CHAPTER 2

CONNECTING THE LIGHT CURVES OF TYPE IIP SUPERNOVAE TO THE PROPERTIES OF THEIR PROGENITORS

It's tempting to linger in this moment, while every possibility still exists. But unless they are collapsed by an observer, they will never be more than possibilities.

Solanum, *Outer Wilds*

This chapter is based on the published work of B. L. Barker, *et al.* 2022 ApJ **934** 1.

ABSTRACT

Observations of core-collapse supernovae (CCSNe) reveal a wealth of information about the dynamics of the supernova ejecta and its composition but very little direct information about the progenitor. Constraining properties of the progenitor and the explosion requires coupling the observations with a theoretical model of the explosion. Here, we begin with the CCSN simulations of Couch et al. (2020), which use a non-parametric treatment of the neutrino transport while also accounting for turbulence and convection. In this work we use the SuperNova Explosion Code to evolve the CCSN hydrodynamics to later times and compute bolometric light curves. Focusing on SNe IIP, we then (1) directly compare the theoretical STIR explosions to observations and (2) assess how properties of the *progenitor's core* can be estimated *from optical photometry in the plateau phase alone*. First, the distribution of plateau luminosities (L_{50}) and ejecta velocities achieved by our simulations is similar to the observed distributions. Second, we fit our models to the light curves and velocity evolution of some well-observed SNe. Third, we recover well-known correlations, as well as the difficulty of connecting any one SN property to zero-age main sequence mass. Finally, we show that there is a usable, linear correlation between iron core mass and L_{50} such that optical photometry alone of SNe IIP can give us insights into the cores of massive stars. Illustrating this by application to a few SNe, we find iron core masses of 1.3-1.5 solar masses with typical errors of 0.05 solar masses. Data are publicly available online (https://doi.org/10.5281/zenodo.6631964).

2.1 Introduction

Core-collapse supernovae (CCSNe) are the explosive deaths that result from the ends of stellar evolution for massive stars with zero-age main sequence (ZAMS) masses $M_{\mathrm{ZAMS}} \gtrsim 8M_{\odot}$. The current understanding suggests that some fraction of possible progenitors will successfully produce CCSNe while others will fail and produce a black hole (BH) (O'Connor & Ott, 2011; Lovegrove & Woosley, 2013; Ertl et al., 2016; Sukhbold et al., 2016; Adams et al., 2017; Sukhbold et al., 2018; Couch et al., 2020). The details of the explosion mechanism have been the subject of decades of work with current work favoring, for most progenitors, the delayed neutrino-driven mechanism (Bethe & Wilson, 1985). For an in-depth review of the CCSN explosion mechanism and related

problems, see recent reviews (e.g., Bethe, 1990; Janka et al., 2007, 2012, 2016; Burrows, 2013; Hix et al., 2014; Müller et al., 2016; Couch, 2017; Pejcha, 2020).

CCSNe are detectable by three primary messengers – EM waves, neutrinos, and gravitational waves (GWs). Neutrino and GW signals have the very desirable property that they are emitted directly from the core of the star at the time of collapse and may reveal information about the structure there (e.g., Pajkos et al., 2019, 2021; Warren et al., 2020; Sotani & Takiwaki, 2020), unlike photons which are emitted from the photosphere in the outer layers of the supernova ejecta until the remnant phase. However, to date there has been only one detection of neutrinos from a supernova (Arnett et al., 1989, SN1987A). With modern neutrino detectors, only CCSNe occurring within our galaxy may be detectable (Scholberg, 2012). Similarly, there have been no confirmed detections of GW emission from a CCSNe. The current suite of detectors (aLIGO, Virgo, and KAGRA) can only detect GWs from a CCSNe if it occurs within a distance of ≤ 100 kpc (Abbott et al., 2016). It is the case, however unfortunate, that the overwhelming majority CCSNe will only be observed in EM signals.

The focus of this paper is connecting EM signals to progenitor properties for SNe IIP. These events have been shown to originate from red supergiant progenitors (Van Dyk et al., 2003; Smartt, 2009; Van Dyk et al., 2019). Despite being the most common type of CCSNe, their diversity of observable features – such as light curve morphologies – is still not fully understood (e.g., Anderson et al., 2014; Valenti et al., 2016). The connection between SNe IIP and IIL supernovae, for example, still remains an open question – whether IIL's are the limit of IIP's as the H envelope is depleted or a separate class (Barbon et al., 1979; Blinnikov & Bartunov, 1993; Faran et al., 2014; Morozova et al., 2015).

Understanding the connection between SNe IIP light curves and stellar progenitors has a new urgency. Coming next-generation telescopes such as the Vera C. Rubin Observatory and its primary optical photometry survey, The Rubin Observatory Legacy Survey of Space and Time (LSST) (Ivezić et al., 2019), will allow for extremely deep imaging of the entire sky every couple of nights. The LSST will allow for statistical studies of populations of CCSNe of an unprecedented

31

scale (for recent statistical studies see, e.g., Anderson et al., 2014; Sanders et al., 2015; Gutiérrez et al., 2017a,b).

Ultimately, properly characterizing the diversity in SNe II supernova light curve morphology will require the union of observation and theory. On the theory side, this comprises realistic stellar evolution models including the core collapse, following the resulting explosion with robust physics, and calculating EM light curves (as well as neutrino and GW signals). The gold standard is full three-dimensional (3D), self-consistent simulations. Core-collapse supernovae and their progenitors are truly 3D in nature and the key to understanding the diversity of light curve morphology lies in faithfully modeling these asphericities (Wongwathanarat et al., 2013, 2015; Dessart & Audit, 2019; Stockinger et al., 2020; Sandoval et al., 2021). 3D simulations are, however, computationally expensive to perform and, as such, are limited in number and the range of parameter space that they cover. Spherically symmetric (1D) simulations remain necessary for understanding the CCSNe explosion mechanism and their observables by surveying landscapes of possible CCSNe. Great progress has been made in the last few years regarding 1D CCSN simulations (Ebinger et al., 2017; Sukhbold et al., 2016; Couch et al., 2020), allowing for successful explosions in 1D using neutrino-driven explosions across wide ranges of progenitor masses. These 1D simulations allow for large parameter studies performing potentially thousands of simulations spanning ranges of progenitor masses, equations of state, and metallicities, for example.

Light curve calculations are the final, crucial piece of the theoretical process of understanding these explosions. Commonly, calculations of synthetic bolometric light curves of CCSNe invoke a thermal bomb or piston-driven model, where energy is artificially injected into a thin region above a user-specified mass cut within the progenitor (see, e.g., Bersten et al., 2011; Morozova et al., 2015; Ricks & Dwarkadas, 2019). In these models, the explosion energy is a user-set parameter instead of being determined by the structure of the progenitor and explosion physics. The calculations cannot determine whether a given progenitor will result in a successful supernova or fail to revive its stalled shock and collapse to a black hole. The explodability has been shown to have non-trivial behavior across a large range of ZAMS mass progenitors (Sukhbold et al., 2016; Ebinger et al.,

32

2017; Sukhbold et al., 2018; Couch et al., 2020) and cannot be captured with more simplified models. The clear next step is the coupling of high fidelity CCSN simulations with bolometric light curve calculations.

Light curves contain information about their progenitor and the explosion – properties such as the composition of the ejecta, mass or radius of the progenitor, or explosion energy may be inferred (Litvinova & Nadezhin, 1985; Popov, 1993; Kasen & Woosley, 2009; Sukhbold et al., 2016). The process of inferring progenitor and explosion properties from light curves has been shown to be highly degenerate (Goldberg et al., 2019; Dessart & Hillier, 2019) with many combinations of properties being capable of producing a given light curve. Of particular interest, however, is the early time light curve dominated by radiation streaming form the shock heated outer envelope. This early time behavior can be compared to shock cooling models to put constraints on the stellar radius (Nakar & Sari, 2010; Tolstov et al., 2013; Shussman et al., 2016; Kozyreva et al., 2020). Recently Morozova et al. (2016); Rubin & Gal-Yam (2017) explored the effectiveness and temporal limitations of these models and these methods have been widely used for constraining the progenitor pre-explosion radius (e.g., Rabinak & Waxman, 2011; Gall et al., 2015; González-Gaitán et al., 2015; Sapir & Waxman, 2017; Soumagnac et al., 2020; Vallely et al., 2021). These early time observations may help to break the degeneracies between progenitor and explosion properties (Goldberg & Bildsten, 2020).

In this work, we calculate the bolometric light curves of the recent 1D simulations done with the FLASH[1] (Fryxell et al., 2000; Dubey et al., 2009) code using the new Supernova Turbulence In Reduced-dimensionality (STIR) model (Couch et al., 2020). This 1D convection scheme has the benefit of being more consistent with some properties of full physics 3D CCSN simulations – such as explosion energies and landscapes – while leaving the neutrino physics unaltered. Like any 1D method, it remains a simplification of the full picture and is not without its shortcomings (e.g., Müller, 2019). Similar 1D schemes have also been used to study Rayleigh-Taylor instabilities in supernova remnants (Duffell, 2016). The initial conditions of these models are set by the 1D

---

[1] https://flash.rochester.edu/site/

stellar evolution models of Sukhbold et al. (2016), which make up a suite of 200 solar metallicity, non-rotating massive stars between 9 and 120 $M_\odot$. We couple the final state of the STIR simulations with the SuperNova Explosion Code (SNEC) (Morozova et al., 2015), which follows the explosion through the rest of the star and through the plateau and nebular phases of the light curves. We will demonstrate that using a more sophisticated 1D explosion model to determine a distribution of explosion energies consistent with 3D simulations imparts non-trivial features to observables and thus properties inferred from them, highlighting the importance of the explosion model used. With this set of light curves, we make available a new set of theoretical predictions to compare directly with observations. Furthermore, we investigate direct correlations between progenitor properties and light curve properties. We recover known correlations, and we quantify the dependence of SNe IIP luminosity on the progenitor iron core mass at time of collapse – thus providing a way of obtaining core properties from EM signals without the need for the much rarer neutrino and GW signals.

This paper is laid out as follows: in Section 7.3 we discuss the various progenitors, codes, and statistical methods that are used in this study. Section 3.3 presents our results: 2.3.1 presents observable properties of our light curves and their trends across ZAMS mass, 2.3.2 presents preliminary comparisons to observations of SNe IIP, 2.3.3 shows correlations found between light curve and progenitor properties. In Section 3.4, we summarize our results and briefly discuss comparison to other theoretical light-curve calculations and prospects for future work.

2.2 Methods

For this work, we begin with massive stellar progenitors evolved up to the point of core collapse in Sukhbold et al. (2016) evolved using KEPLER. The core collapse and following explosion or collapse to BH are simulated using the FLASH simulation framework (Fryxell et al., 2000; Dubey et al., 2009) with the Supernova Turbulence In Reduced-dimensionality (STIR) model (Couch et al., 2020), the details of which are discussed in Section 2.2.2. The output of the STIR simulations are mapped into the SuperNova Explosion Code (SNEC) (Morozova et al., 2015, 2016, 2018) to generate bolometric light curves as discussed in Section 2.2.3. In Section 2.2.4 we present

the methods used to analyze statistical relationships between properties of the progenitor and observables.

2.2.1 Progenitors

We begin with the 200 non-rotating, solar metallicity models of Sukhbold et al. (2016). These models cover a range of ZAMS masses from 9 – 120 $M_\odot$ and were created with the KEPLER code assuming no magnetic fields or rotation and single star evolution. Progenitors with ZAMS masses above $31 M_\odot$ experienced significant mass loss during their lifetimes and did not explode as SNe IIP and this is the upper limit on our mass range (see Sukhbold et al., 2016, for details on their stellar evolution). The more massive Type I SNe progenitors are too few in number to perform a meaningful statistical analysis and we defer their analysis for future work. This leaves 136 progenitors producing SNe IIP supernovae used in this work.

These progenitors span a wide range of possible CCSNe progenitor properties. Figure 2.1 shows the mass of the H-rich envelope as a function of pre-supernova radius (top) and the stellar pre-supernova mass as a function of ZAMS mass (bottom). Here we show only models that successfully exploded in Couch et al. (2020) and are included in this work. Gaps in this figure – such as that from about 12-15$M_\odot$– represent models which failed to explode and are not included in this work. These progenitors become mass-loss dominated around $23 M_\odot$, as seen in the bottom panel of Figure 2.1. This complicates correlations between quantities of interest and tends to cause them to deviate from monotonicity. This is key to investigating observable trends in light curves across a wide range of progenitors, as we demonstrate later.

The progenitors in Sukhbold et al. (2016) were further investigated in Sukhbold et al. (2018) using a set of high resolution stellar evolutionary models. They showed that the features of these progenitors – notably the compactness landscape – was not numerical in nature and was present in their high resolution models. Similarly, other, recent works have found similar trends in the presupernova mass and compactness (e.g., Laplace et al., 2021, using MESA (Paxton et al., 2011, 2013, 2015, 2018, 2019)). We note that while general trends may be reproduced, details such as the apparent 'chaos' seen in Sukhbold et al. (2016, 2018) are sensitive to implementation details of
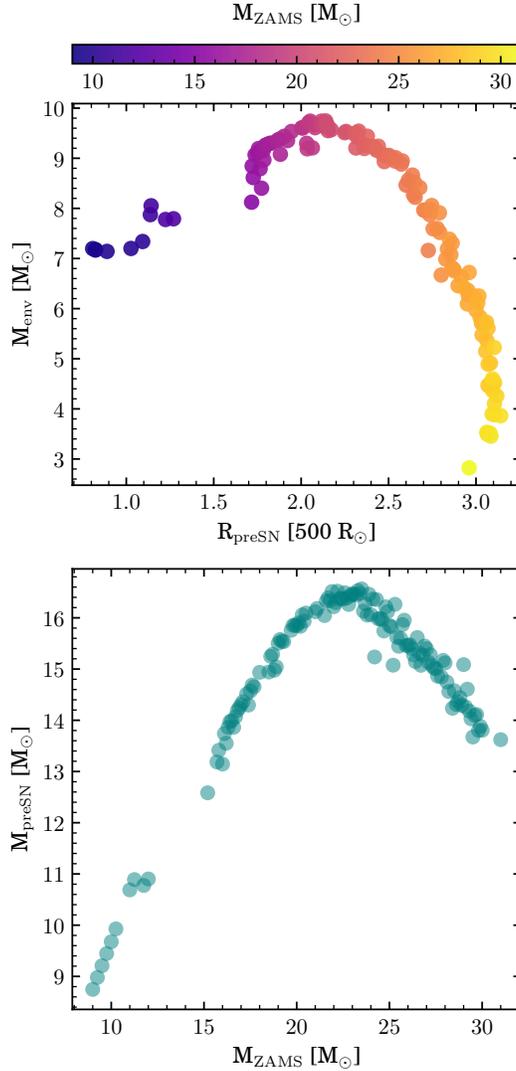
Figure 2.1 Properties of the progenitors of Sukhbold et al. (2016). (top) Mass of the H-rich envelope (M_{env}) as a function of pre-supernova radius (R_{preSN}). (bottom) Final stellar mass (M_{preSN}), after mass loss, as a function of ZAMS mass.

stellar evolution and may not appear in other studies (Chieffi & Limongi, 2020). Using a different set of progenitors with a different compactness curve would likely affect the explosion landscape but isn't expected to affect the results presented here.

A key result of systematic 1D studies of CCSNe are the so-called "islands of explodabilty" (Sukhbold et al., 2016). The final result of stellar core collapse - a successful or failed explosion - is not a monotonic function of ZAMS mass. Instead, the explodability of the progenitor is sensitive

to the core structure at the time of collapse. While the placement of these islands is sensitive to the explosion model and the progenitors used, it is a feature that has now been seen amongst many groups (O'Connor & Ott, 2011; Perego et al., 2015; Sukhbold et al., 2016; Ertl et al., 2016; Ebinger et al., 2019; Couch et al., 2020). However, studies using exclusively thermal bomb driven explosions uninformed by neutrino driven calculations cannot reproduce the explosion/implosion fate of a progenitor and are insensitive to this feature. Any systematic study of light curves from populations of SNe must capture this complex behavior.

2.2.2 FLASH

The CCSN simulations were conducted in Couch et al. (2020) using the FLASH code framework with the STIR turbulence-aided explosion model. This model is a new method for artificially driving explosions in 1D CCSN simulations.. Turbulence is key in simulating successful, realistic explosions, as turbulence may constitute 50% or more of the total pressure behind the shock (Murphy et al., 2013; Couch & Ott, 2015) and turbulent dissipation is important for post-shock heating (Mabanta & Murphy, 2018). The combined impact of these effects is to aid the explosion. The inclusion of turbulent effects allows for successful explosions in 1D simulations while reproducing the results seen in 3D simulations from various groups (Couch et al., 2020) without the need for parametrized neutrino physics.

STIR models turbulence using the Reynolds-averaged Euler equations with mixing length theory as a closure. This model has one primary scalable parameter, the mixing length parameter α_Λ, inherited from mixing length theory which scales the strength of convection. The mixing length parameter has been tuned to fit STIR simulations to full 3D simulations run with FLASH and reproduces 3D results seen in FLASH and other codes, noting particularly good agreement with the 3D works of (Burrows et al., 2020). We use the fiducial value found in Couch et al. (2020) for the mixing length parameter, $\alpha_\Lambda = 1.25$. STIR also includes four additional diffusion parameters that control the convective mixing of internal energy, turbulent kinetic energy, composition, and neutrinos. As in Couch et al. (2020), all four of these diffusion coefficients are set to 1/6, a value consistent with comparison to fully 3D simulation of convection in massive stars (Müller et al.,

2016). We note that the convective dynamics are insensitive to the choice of diffusion coefficients and, thus, impacts on the explosion are negligible (Müller et al., 2016; Boccioli et al., 2021). `FLASH` with the STIR model has the desireable benefit that there is no need to tune the model to match a specific observation. Instead, its one primary parameter is tuned to be consistent with multi-physics 3D CCSN simulations, reducing the possibility of inserting biases into the results.

STIR includes neutrino transport using a state-of-the-art two moment method with an analytic "M1" closure (Shibata et al., 2011; Cardall et al., 2013; O'Connor, 2015; O'Connor & Couch, 2018). We simulate three neutrino flavors: ν_e, $\bar{\nu}_e$ and ν_x, where ν_x combines the $\mu - \tau$ neutrino and antineutrino flavors. M1 transport requires no tuning and has no free parameters (up to the choice of a closure for the high-order radiation moments), allowing for truly physics-driven explosions. The STIR simulations use the now commonly adopted, empirically-motivated "SFHo" equation of state for dense nuclear matter (Steiner et al., 2013a) which is able to replicate observed neutron star masses.

At the end of the STIR simulations, the explosion energies for all but the highest-mass progenitors have asymptoted. It is commonplace in CCSNe work to define the explosion energy as the sum total energy, from all sources, of material that has both positive total energy and positive velocity (e.g., Bruenn et al., 2016) at the end of the simulation. This is zero during the stalled shock phase, when all of the material is still gravitationally bound, and becomes positive if/when the shock begins to move outward again due to neutrino heating and other effects. This energy, once it has reached its asymptotic value, represents the energy that is injected into the rest of the star to drive the explosion and unbind the stellar material. When discussing the combined STIR + SNEC simulations, this is the explosion energy that we will reference. It is important to note that this energy is different than the energy that would be used in hydrodynamical modelling (e.g., thermal bomb explosions). In the thermal bomb regime, a user set energy is deposited at $t = 0$ over defined temporal and spatial extent, and assumes that the energy of the shock comes directly from the core-bounce which is inconsistent with the physical picture of CCSNe. In the case of high fidelity simulations, a large amount of material has already been gravitationally unbound by the shock

when the explosion energy is measured. A thermal bomb model with the "same" energy injected into the inner zones would, by the time the same amount of material is unbound, be less energetic by exactly the binding energy of the material. Care should be taken when comparing energetics from these two approaches. While the physics of these two explosion methods are inconsistent with each other, the thermal bomb energetics can be made consistent with neutrino-driven energetics by correcting the bomb energy by the binding energy of the material between the shock and the PNS surface (this material is already unbound when the explosion energy is calculated as above, but in thermal bomb or piston-driven explosions it is not). Without this correction, a thermal bomb model using energetics from neutrino-driven simulations will have less energy available for the explosion, impacting observables.

Figure 2.2 shows the explosion energies obtained with STIR (black) alongside the explosion energy with the progenitor's overburden energy removed (blue). The progenitor's overburden energy is the (negative) total energy above the shock that the explosion must overcome to unbind the star (Bruenn et al., 2016). The total energy, which we compute as the total energy on the computational domain after the explosion has set in is closer to what will characterize the ejecta. Gaps in mass, such as from about $12M_\odot$ to $15M_\odot$, indicate regions where progenitors failed to successfully launch an explosion in STIR. The bottom panel shows the explosion energy as a function of the iron core mass. These explosion energies are set largely by the structure of the cores of their progenitors – effects which can only be seen by employing neutrino driven explosions (for recent examples of the impacts of core structures on explosions and observable signatures, see Warren et al., 2020; Burrows et al., 2020). The emerging picture from high fidelity simulations is that there is no simple relationship between explosion energy and ZAMS mass, instead requiring multi-physics simulations to determine robustly (Sukhbold et al., 2016; Ebinger et al., 2017, 2019; Sukhbold et al., 2018; Couch et al., 2020; Burrows et al., 2020; Ertl et al., 2020). The explosion energy is more closely related to the pre-supernova mass and properties of the core, such as the compactness parameter or the mass of the iron core.
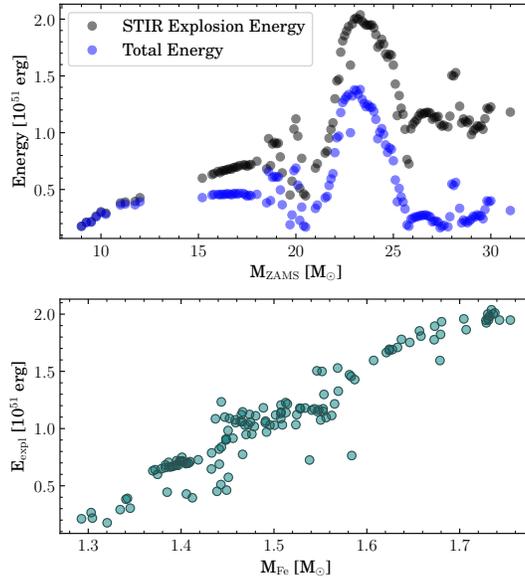
Figure 2.2 Top: Explosion energies realized in the STIR CCSN simulations of Couch et al. (2020) (black) and the final energy after removing the overburden energy of the progenitor (blue). Bottom: STIR explosion energy as a function of the progenitor's iron core mass.

2.2.3 SNEC

We simulate light curves for all of the models that successfully produced explosions in Couch et al. (2020) (see their Figure 6, middle row). This is all but about 50 of the original 200 progenitors. This limits our study to light curves obtained from progenitors that actually explode, allowing us to explore solely relationships that come from physically-driven explosions. At the end of the STIR simulations, the final states are mapped into the SuperNova Explosion Code (SNEC) (Morozova et al., 2015). SNEC is a spherically symmetric, Lagrangian, equilibrium flux-limited diffusion radiation-hydrodynamics code and is publicly available[2]. Unlike STIR, SNEC does not include any form of general relativistic gravity, neutrino transport, or dense matter EoS, which are all important for modeling the explosion but not necessarily for computing the light curve. Instead, it follows the basic physics needed for predicting bolometric supernova light curves. SNEC includes Lagrangian Newtonian hydrodynamics with artificial viscosity following the formulation in Mezzacappa & Bruenn (1993) and a stellar equation of state following Paczynski (1983) that includes contributions from radiation, ions, and electrons with approximate electron degeneracy.

---

[2]http://stellarcollapse.org/SNEC

This is used in tandem with a Saha ionization solver that can follow ionization of any number of present elements. At high temperatures SNEC uses OPAL Type II opacities (Iglesias & Rogers, 1996) suitable for solar metallicity. These opacities are supplemented by those of Ferguson et al. (2005) at low temperatures.

1D modeling cannot properly capture the mixing at compositional interfaces due to Rayleigh-Taylor and Richtmyer-Meshkov instabilities, for example. Without mixing, sharp compositional gradients appear that produce features in light curves that are not observed in nature (Utrobin, 2007). In these mixing processes, shock propagation outwards can cause light elements to mix inwards and heavy elements to mix outwards (Wongwathanarat et al., 2015). Of particular importance is the mixing of radioactive ^{56}Ni, whose mixing extent affects the light curve properties (Morozova et al., 2015). SNEC applies boxcar smoothing that smooths out compositional profiles, simulating mixing and avoiding unphysical light curve bumps. We use the fiducial parameters of Morozova et al. (2015) for our boxcar smoothing.

In the present work we follow the ionization of ^{1}H, ^{3}He, and ^{4}He, similarly to Morozova et al. (2015). H and He make up the majority of the energy contributions from recombination relevant for producing bolometric SNe IIP light curves. Our STIR simulations do not currently track detailed compositional information in their output. When mapping into SNEC, we fill the composition in the STIR part of the domain to be pure ^{4}He. This has no noticeable effect on the light curves in this study (see Appendix 9A). Figure 2.3 shows mass fractions of ^{1}H (light blue), ^{4}He (dark blue), ^{12}C (gold), ^{16}O (red), and ^{56}Ni (black, dot-dashed line). The solid lines show the unmixed profiles that are input to SNEC. Notably, the gray region shows the STIR domain where the composition, prior to mixing, is set to pure ^{4}He. The dashed lines show the compsotion after boxcar smoothing is applied. The bottom panel shows the radial density profile in the STIR domain (solid line) and in SNEC after mapping (dashed line).

The final, critical ingredient for powering a SNe light curve is radioactive heating from the ^{56}Ni $\rightarrow$ ^{56}Co $\rightarrow$ ^{56}Fe decay chain. Radioactive ^{56}Ni is produced in explosive nuclear burning during the first epochs of the explosion in the inner parts of the star. Hydrodynamic instabilities mix the ^{56}Ni

41
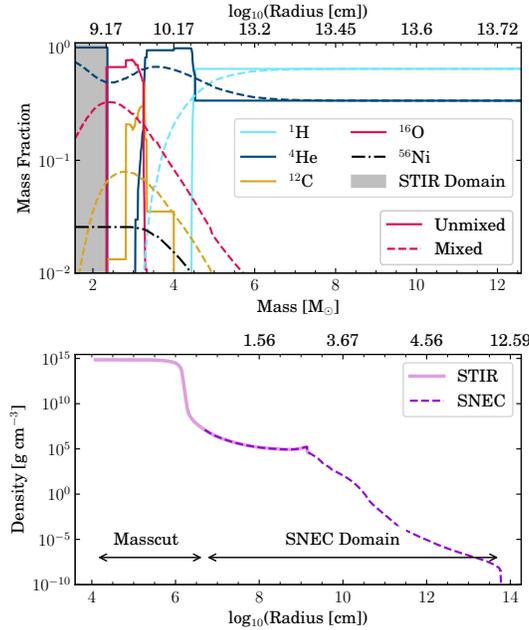
Figure 2.3 Top: Mass fraction of ^{1}H (light blue), ^{4}He (dark blue), ^{12}C (gold), ^{16}O (red), and ^{56}Ni (black, dot-dashed line). Solid lines show the unmixed profiles, dashed lines show the profiles after boxcar smoothing is applied. The gray shaded region represents the STIR domain, which is originally set to pure ^{4}He prior to smoothing. Bottom: Radial density profile for the STIR domain (solid line) and SNEC mapping (dashed line).

outward. Gamma-rays and positrons emitted from the decay process diffuse outward and provide an additional source of energy. Capturing this is crucial as, after the end of the plateau phase, the light curve is powered entirely by this radioactive decay. SNEC follows the radiative transfer of gamma-rays from the ^{56}Ni and ^{56}Co decays using the gray transfer approximation (Swartz et al., 1995) and the resulting energy release is coupled to the hydrodynamics independently from the rest of the radiation.

Currently, neither our STIR models used here nor the public version of SNEC include a nuclear reaction network. To alleviate this issue, SNEC allows for a user specified amount of ^{56}Ni to be input by hand throughout a specified mass coordinate. Sukhbold et al. (2016) simulate the explosions of these progenitors including a large nuclear reaction network, and we use the ^{56}Ni yields as a function of explosion energy from their work (see their Table 4, Figure 17) to estimate a mass of ^{56}Ni from that relationship to be distributed by SNEC. For all but the lightest progenitors,

they find around $0.07 M_\odot$ of ^{56}Ni. We disperse the ^{56}Ni up to about 75% of the way through the He shell – avoiding mixing into the H envelope. This provides control amongst the progenitors. As the mixing extent must be set by hand, any further treatment would require a large parameter study. In recent high fidelity models, mixing of radioactive ^{56}Ni into the H-rich envelope is realized (Utrobin et al., 2015, 2017; Stockinger et al., 2020; Utrobin et al., 2021; Sandoval et al., 2021) and is expected to occur in at least some of our models. In Morozova et al. (2015), they showed that variations in these distributions had little effect on the light curve, especially on the plateau (see their Figure 6). Goldberg et al. (2019) show slight variations in the light curve as it falls off the plateau depending on the extent of mixing (see their Figure 10). Kozyreva et al. (2019) explore the effects of mixing prescriptions for ^{56}Ni, such as uniform or boxcar, on light curves, showing differences on the plateau between these methods. The lack of a reaction network consistently incorporated into the calculations forms a weakness of the current work, despite being based on nucleosynthetic calculations and tuned to our explosion energies. However, the main results of this work (see Section 2.3.3) use quantities measured on the plateau where they are less sensitive to reasonable variations in ^{56}Ni mass and distribution. Future work will include nucleosynthesis calculations with the STIR input models to properly seed the SNEC calculations.

Typically, high fidelity CCSN simulations do not simulate the entire star – instead focusing on the inner 15,000 km or so necessary for launching the explosion. We must stitch the STIR simulation data, with the explosions developed on the grid, onto the progenitor pre-explosion profile outside the STIR boundary (15,000 km) in order to simulate the full star. Below the shock, mass profiles are taken from STIR. Above the shock mass coordinate, mass profiles are taken from the progenitor profiles. These smooth, combined STIR – pre-explosion progenitor profiles are used as the inputs to SNEC as detailed in (Morozova et al., 2015). One advantage of using the STIR models as the initial conditions to SNEC is that the high fidelity equation of state and neutrino transport yield a physically realistic remnant mass to motivate the mass cut – an amount of material not included in light curve simulations that should be close to the remnant mass. We place a mass cut outside the PNS at the point where the total energy becomes positive – removing both the PNS

and a small amount of still gravitationally bound material above it (of order $0.0001M_\odot$). For all of the simulations we use 1000 cells in the SNEC domain using a geometric grid, as in Morozova et al. (2015), that places higher resolution in the core around the shock and at the outer domain to resolve the photosphere. Our grid is slightly modified from that of Morozova et al. (2015) to place added resolution in the core over the already existing explosion. Simulations were run until 300 days when possible to adequately sample both the plateau and the tail for all events.

To simulate CCSNe directly from progenitors, SNEC has the ability to artificially drive an explosion with a piston or thermal bomb. One of the primary qualities of our method is to eliminate the need for this and thus eliminate user input explosion energies, which can take any range or distribution, replacing them with physically motivated energetics. However, for some of the more massive progenitors in this study, the explosion energies were still increasing by the time the shock reached the outer boundary. Eventually, energy generation from neutrino heating and other sources will slow as the shock expands and the explosions energies will asymptote. Since our computational domain is limited to 15,000 km, some progenitors do not reach their "true" explosion energies. In order to fully capture the energy of the explosion in STIR, we integrate the neutrino heating in the gain region at the end of the FLASH simulations to estimate the asymptotic explosion energy and add the difference – at most about 0.3×10^{51} erg – as a thermal bomb over the shocked region. These additions are most necessary in the region of high energy between about $21M_\odot$ and $25M_\odot$ where the final energies were still readily increasing. This energy is what is displayed in Figure 2.2.

The light curves presented in this work represent those 136 progenitors (of the suite of 200) that both successfully launch an explosion (Section 2.2.2) and have light curves that would be identified as a SNe IIP, which we find is simply a mass cut of $M_{\mathrm{ZAMS}} \leq 31M_\odot$.

2.2.4 Correlations

We are interested in uncovering correlations between observable properties of the explosion and properties of the progenitors. The size and fidelity of the sample allows us to address these connections necessary to understand light curve diversity. Our robust treatment of the explosion physics combined with large sample of progenitors makes us uniquely situated to address corre-

lations in a novel way. We proceed similarly to Warren et al. (2020), wherein the correlations between observed neutrino and GW signals with progenitor properties were addressed.

We measure correlations with the Spearman's rank correlation coefficient. The Spearman correlation coefficient measures any monotonic relationships between variables, in contrast to the Pearson coefficient which measures only linear correlations. It is important that we are able to access non-linear relationships that are seen in the data. The combined effect of a wide range of stellar progenitors with mass loss effects and non-linear, non-monotonic explosion energetics over the range of progenitors produces robust and realistic – but not necessarily linear – relationships.

The Spearman coefficient is obtained by first ranking the data by replacing the values by their indices after sorting. For example, the data ($1.5M_\odot$, $1.4M_\odot$, $1.6M_\odot$) would transform to (2, 1, 3). Then, the Spearman rank correlation coefficient is obtained by computing the Pearson correlation of the transformed data, calculated by

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}} \tag{2.1}$$

for ranked variables x and y with $\bar{x}$ and $\bar{y}$ being the mean values. This process of first ranking the data is what allows the Spearman process to produce a more robust correlation metric. We note that the above equation is the same as that for the Pearson correlation coefficient and, when used on non-ranked data, will produce the Pearson correlation coefficient.

A value of +1(-1) represents an exact monotonic correlation (anticorrelation) and a value of 0 indicates no monotonic relationship. We consider values $|\rho| \gtrsim 0.5$ to indicate strong statistical correlation, values $0.3 \lesssim |\rho| \lesssim 0.5$ to be moderate correlation, and $|\rho| \lesssim 0.3$ to be a weak correlation, as is standard practice. Correlation coefficients were calculated using Python's *scipy.stats.spearmanr* package.

We strived to limit observables considered to those reasonably detectable with current facilities – mostly photometric and early time (meaning, in this context, on the plateau but not requiring observations within days of explosion) features. Ultimately plateau duration, plateau luminosity, and ejecta velocity – all at early times – proved to be the most useful and accessible. We explored

45

numerous properties of the progenitors for correlations with observables – shell masses, density structures, core compactness, and envelope mass to name a few. Most of these parameters had weak relationships with observable properties. Ultimately, we settled on the mass of the iron core as the most meaningful and useful progenitor property, as we will see in the next section.

2.2.5 Light curve fitting

A common method for estimating CCSN progenitor properties is to construct a grid of models with varying masses, explosion energies, and ^{56}Ni masses and distributions and select the progenitor from that grid that best fits an observed light curve (see Morozova et al., 2018; Martinez & Bersten, 2019; Martinez et al., 2020, for recent examples). We accomplish this by finding the progenitor which minimizes the average relative error ε of a quantity f

$$\varepsilon(f) = \frac{1}{N} \sum_{t^*=t_1}^{t_N} \frac{|f_{t^*} - f_{t^*}^*|}{f_{t^*}^*} \tag{2.2}$$

where $f_{t^*}^*$ is the observed quantity at time t^*, f_{t^*} is the synthetic quantity at the same time, and N is the number of observational data points. We compare the synthetic and observed data only at the times where observational data is available, using the closest synthetic data to the observational data, which is always within 0.02 days with the output frequency used with SNEC. That is, we do not interpolate between observational data points. We not not consider uncertainties in the explosion epoch in the current work. We seek models that match both observed bolometric luminosity and velocity evolution, i.e., we seek a model minimizing the combined error metric $\varepsilon(L_{50}) + \varepsilon(v_{\text{Fe}})$.

Other approaches have been used, such as (historically) simply fittng by eye, χ^2 minimization (Morozova et al., 2018), and Markov chain Monte Carlo methods (Martinez et al., 2020). We implemented several minimization approaches and found that the above method worked best for the current work. This is discussed more in Section 2.3.2.2.

2.3 Results

We consider the properties of the bolometric light curves followed through the end of the plateaus and into the radioactive tails and the ejecta velocities for models with ZAMS masses 9M$_\odot$ $\leq$ M$_{\text{ZAMS}} \leq$ 31M$_\odot$ for a total of 136 progenitors. In an effort to find relationships with observables

that are easily detectable, we consider primarily the photometric and spectroscopic properties in the plateau phase. The primary quantities that we consider are the plateau luminosity at day 50 (L_{50}), the plateau duration (t_p), and the ejecta velocity at day 50 (v_{50}). These quantities are commonly used when inferring explosion properties from observations (e.g., Litvinova & Nadezhin, 1985; Popov, 1993; Pejcha & Prieto, 2015) and so their trends from realistic models are of particular interest. These quantities are easily detectable by current and next generation facilities without the need for late time observations or particularly high cadences, acknowledging that the photospheric velocity will not be as easily observable for most sources. This will allow for a relationship to be obtained between these quantities and properties of the core of the progenitor that is both robust and easily detectable with standard measurements.

2.3.1   Landscape properties across ZAMS mass

Here we present global trends in photometric properties to test the impact of our explosion calculation on light curve features. As we will see, these properties exhibit non-monatonic features as a function of ZAMS mass and thus introduce degeneracy into attempts to infer progenitor properties from direct comparisons to light curves.

Figure 2.4 shows the bolometric luminosity at day 50 (on the plateau for all progenitors) for all masses. The imprint from the distribution of explosion energies is readily seen in the plateau luminosities, with more energetic explosions yielding brighter plateaus. A consequence of this is the highly degenerate mapping between plateau luminosity and ZAMS mass following the explosion energy distribution (Figure 2.2).

Figure 2.5 shows the plateau duration for the STIR + SNEC models. We follow Valenti et al. (2016) and Goldberg et al. (2019) and compute the plateau duration by fitting part of the light curve near the end of the plateau to a combined Fermi-Dirac – linear function of the form

$$f(t) = \frac{-a_0}{1 + \exp((t - t_p)/w_0)} + (p_0\, t) + m_0. \tag{2.3}$$

This avoids biases or inconsistencies that may possibly be introduced by determining the plateau durations by eye for a large sample of light curves. The physical significance of the various fitting
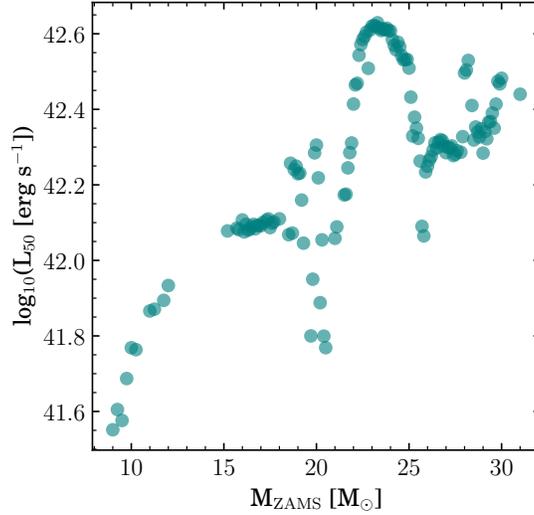
Figure 2.4 Log of the plateau luminosity at day 50 for the STIR + SNEC models.

parameters is described in detail in Valenti et al. (2016) and Goldberg et al. (2019). Importantly, the parameter t_p is taken to be the plateau duration and tends to be placed about halfway through the drop off of the plateau. Also of interest are a_0 and w_0 which describe the luminosity drop at the end of the plateau and the width of the drop, respectively. Fitting was done using Python's *scipy.optimize.curvefit* package starting shortly before the end of the plateau. For a few of the high mass models between 27 and 28$M_\odot$, timestep restrictions made it difficult to simulate the explosions into the radioactive tails. Most made it to the end of the plateau and began drop off, but two progenitors were unable to reach the end of the plateau. For the former case, the fitting is unable to work properly and the plateau duration is set by hand in a way that was consistent with the fitting routine. For the two progenitors that could not reach the end of the plateau – 27.4$M_\odot$ and 27.5$M_\odot$– we omit them in comparisons involving the plateau duration.

Clearly, the distribution of the explosion energies imparts a resulting morphology on the plateau durations that cannot be reproduced without energetics informed by neutrino-driven explosions. We note that many of the plateaus here are quite long, greater than 150 days or so, which is not very common. These plateaus originate from very massive progenitors, around 20$M_\odot$, which are rare in nature. Moreover, these models retain quite massive H-rich envelopes (see Figure 2.1) and have reduced explosion energies (see Figure 2.2). The combination of massive H-rich envelope
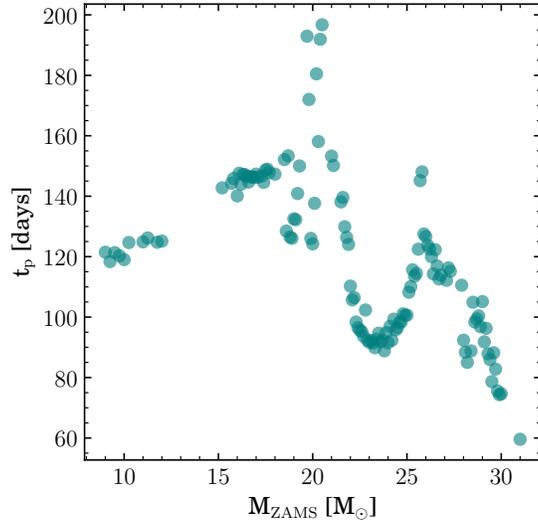
Figure 2.5 Plateau duration for the STIR + SNEC models. Two progenitors between 27 and 28$M_\odot$ have been removed for fair comparison, as some of them did not reach the radioactive tail in the simulation time.

with reduced explosion energy results in extended plateaus (Popov, 1993). Some uncertainty in the plateau duration remains through the prescription for setting the mass and mixing of radioactive ^{56}Ni, as it lengthens the plateau slightly (Kasen & Woosley, 2009; Morozova et al., 2015; Sukhbold et al., 2016; Goldberg et al., 2019; Kozyreva et al., 2019). These uncertainties, however, should be on the order of days (see, e.g., Figure 13 from Morozova et al. (2015), Figure 10 from Goldberg et al. (2019), Figure 4 from Kozyreva et al. (2019)). It is also somewhat difficult to fairly compare plateau durations to observed works, as many authors present the length of the optically thick phase duration (e.g., Gutiérrez et al., 2017b) which may be smaller than our measurement by another 5-10 days or more. For these reasons, we defer further comparisons to observational data of the plateau durations to future work.

All of this directly impacts the ability to reliably extract progenitor features from light curves. Without a distribution of explosion energies that is set by a physically realistic explosion model, any sort of arbitrary distribution of light curve properties may be recovered, even with the same diversity of progenitors used. While STIR is not a perfect or parameter free description of the explosion – no 1D model ever will be – it matches well with 3D results and provides a large set of such physically motivated explosion energies for these studies.

Another quantity of interest – albeit not a directly observable one – is the time to shock breakout. Figure 2.6 shows the time for the shock to breakout from the stellar surface for the STIR + SNEC models. This is particularly important, as the time to shock breakout sets the on source window for electromagnetic follow-ups of gravitational wave and neutrino events from core-collapse supernovae (Abbott et al., 2020). The time to shock breakout is sensitive to the structure of the progenitor and the explosion energy and may be significantly over- or under-estimated if an incorrect explosion energy is used.

With the next galactic CCSNe and prospects for detecting their gravitational wave and neutrino signals, the time to shock breakout becomes a measurable quantity through the difference between GW or neutrino detection time and first light from the SNe. The SuperNova Early Warning System (SNEWS) (Adams et al., 2013; Kharusi et al., 2021) will alert observatories to trigger an EM followup after a neutrino detection, and knowing the shock breakout time will be an important factor for the followup study. Combined with constraints from the GW detection (Abbott et al., 2020) and constraints from other EM observations, the time to shock breakout could help to place additional constraints on the SNe progenitor – provided that adequate energetics are used. Similarly, constraints on the shock breakout time after an EM signal may be used to look back at GW and neutrino data, assuming a nearby event.

The previous figures highlight the strong dependence on the distribution of explosion energies used to drive the explosion. This leads to degeneracies when mapping from observables to ZAMS mass with many progenitors of varying masses being capable of producing a given observation.

2.3.2 Comparisons with observations

In this section, we compare our light-curves to observations of SNe IIP both through global properties of many SNe and fits to the light-curves of individual SNe IIP that have M_{ZAMS} determined through pre-explosion imaging data.

2.3.2.1 Comparison with a large observational sample

Apart from photometric observations, spectroscopic observations may also be used to constrain progenitor properties. While we have not computed full synthetic spectra in this work, we can
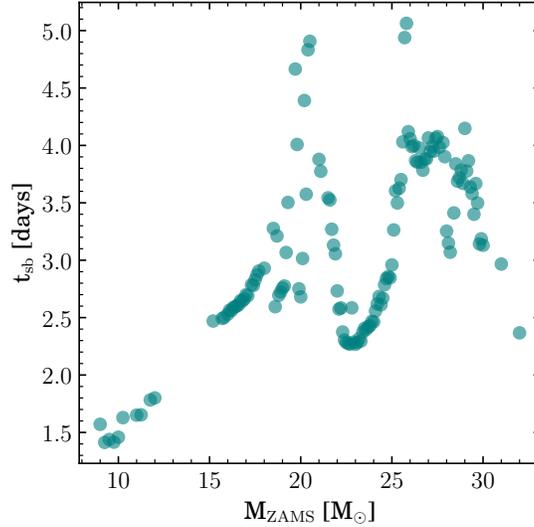
Figure 2.6 Time for shock breakout for the STIR + SNEC models.

approximate standard line velocities. Figure 2.7 shows ejecta velocity at day 50 (v_{50}) versus plateau luminosity at day 50 (L_{50}) for all progenitors that exploded as SNe IIP. Also plotted are data presented in Gutiérrez et al. (2017a,b)[3]. All ejecta velocities here are inferred from the Fe II (5169) line. In our models, this velocity is calculated in post-processing as the velocity of the ejecta at the point where the Sobolev optical depth (τ_{Sob}) is unity, with

$$\tau_{\text{Sob}} = \frac{\pi q_e^2}{m_e c} \, n_{\text{Fe}} \, \eta_i f \, t_{\text{expl}} \, \lambda_0 \tag{2.4}$$

where q_e and m_e are the electron charge and mass, n_{Fe} is the number density of iron atoms, η_i is the ionization fraction relevant for the transition of interest, $f = 0.023$ is the atomic oscillator strength, t_{expl} is the time since explosion, and λ_0 is the wavelength associated with the transition. For material in homologous expansion, this measures the strength of a particular line (Mihalas et al., 1978; Kasen et al., 2006) and the point where $\tau_{\text{Sob}} = 1$ has been shown to match better to observational measurements than the $\tau = 2/3$ electron scattering photosphere (Goldberg et al., 2019; Paxton et al., 2018). To estimate the ionization fraction, we use a table of η_i as a function of density and temperature that is now publicly available in MESA (Paxton et al., 2018).

---

[3]Bolometric luminosity data were calculated from M_V measurements at day 50 provided by C. Gutièrrez (private communication).
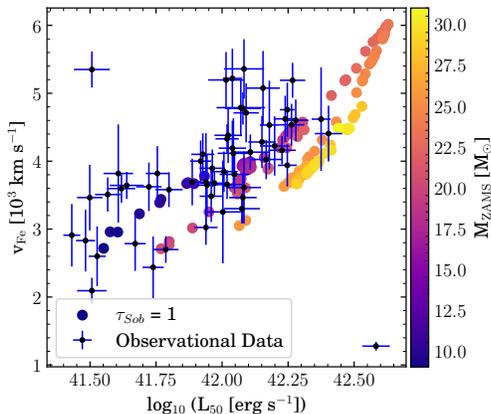
Figure 2.7 Ejecta velocity at day 50, v_{50}, versus the log of the bolometric luminosity of the plateau at day 50, L_{50} for all of the exploding progenitors. Simulated data are colored by the zero-age main sequence mass. Points with error bars are observational data from Gutiérrez et al. (2017a,b).

We choose to use this metric for the velocity evolution because ultimately we seek to compare with observables. While the standard $\tau = 2/3$ photosphere – and its velocity – are simple to compute, they are not simple to observe. On the other hand, the Fe II 5169 line is commonly measured. Therefore, we seek to estimate the location in the ejecta where this line is measured, using the Sobolev approximation that has been readily used in recent works (Paxton et al., 2018; Goldberg et al., 2019; Martinez et al., 2020). However, this approach to estimating the iron line velocity is ultimately an approximation and there are physical uncertainties associated with this method. Paxton et al. (2018) investigated the effects of the choice of the Soboloev optical depth used and found relatively small differences when compared to using the traditional photospheric velocity. In lieu of full spectral calculations this method provides an estimate of the desired velocity but more work may be needed to robustly compare to observed ejecta velocities.

The sample of luminosities and velocities from our models matches well with the observational sample, but reach higher in luminosity than the observed set. These high luminosity events are from some of the higher pre-supernova mass stars around the transition to the mass-loss dominated regime (see Figure 2.1). These high mass stars are less common than their lower mass companions. The highest ZAMS mass stars, those $\gtrsim 23M_\odot$ dominated by mass-loss, dip back down and left in luminosity- and velocity-space, obtaining similar luminosities but slightly

lower velocities than lower mass progenitors. Ultimately, we are able to reproduce observed distributions quite well without having to tune to observations, instead following the explosions from self-consistent simulations.

2.3.2.2 Determination of progenitor properties for individual events

It is commonplace to estimate supernova progenitor parameters using a grid hydrodynamical models (i.e., codes similar to SNEC using a thermal bomb) with varying initial masses, thermal bomb energies, and other parameters, and determining the best fitting model (see, e.g., Utrobin & Chugai, 2008, 2009; Pumo et al., 2017; Morozova et al., 2018; Martinez & Bersten, 2019; Martinez et al., 2020; Eldridge & Xiao, 2019). We attempt to match our set of explosions with 7 observed bolometric light curves from Martinez & Bersten (2019); Martinez et al. (2020)[4]. Bolometric luminosities are calculated using the bolometric correction method of Bersten & Hamuy (2009), which requires only BVI photometry to estimate the bolometric correction.

Figures 2.8 and 2.9 show observed bolometric light curves (left) and velocity evolution (right) for (top to bottom) SN 2004A, SN 2004et, SN 2005cs, SN 2008bk, SN 2012aw, SN 2012ec, and SN 2017eaw. Dark blue lines show bolometric luminosity and velocity evolution for best fit progenitors from our sample using the STIR + SNEC model using the fitting described in Section 2.2.5. For the velocity evolution, dashed lines show approximate Fe II $\lambda 5169$ line velocities estimated through the methods described in Section 2.3.2 and solid lines show the proper $\tau = 2/3$ photospheric velocity. Gold lines are for ZAMS mass models corresponding to estimates from pre-explosion imaging. We use the ZAMS mass estimates from Davies & Beasor (2018) for SN 2004A, SN 2004et, SN 2008bk, SN 2012aw, and SN 2012ec. Properties of these SNe are discussed in detail in Martinez & Bersten (2019) and Martinez et al. (2020). For SN 2005cs, Davies & Beasor (2018) estimated an initial mass of about 7 $M_\odot$– well below the minimum mass we consider to produce a CCSN – so we use the estimate from Smartt (2015). Finally, we use the mass estimate for SN 2017eaw from Eldridge & Xiao (2019). In all cases we use the optimal value of the initial mass when possible, or the closest value within the reported range that was both on our mass grid and produced an

---

[4]Observational data were provided by L. Martinez (private communication).

explosion.

We determine the best fit progenitor by minimizing the total relative error of both luminosity and velocity across the entire light curve after day 30 as discussed in Section 2.2.5. We also tried minimizing χ^2, as was done in Morozova et al. (2018), but found unsatisfactory performance compared to our method (see Appendix 9B for an example using SN2017eaw). We did not consider the errors associated with the observations in our fitting. The inverse variance weighting typically used in χ^2 minimization gave stronger significance to the radioactive tail, as this region has much smaller error compared to the plateau. The result was the selection of models that fit the tail nicely, but fit the plateau very badly. We do not consider data before 30 days post shock breakout, as very early time bolometric luminosities may be heavily influenced by interactions with circumstellar material (CSM) for some SNe (Morozova et al., 2018) and we have not included CSM effects in this work.

We do not expect to find close fits for all observed CCSNe. In this work, we have progenitors that cover a wide range of ZAMS masses with explosions driven by turbulence-aided neutrino radiation hydrodynamics simulations, but are limited in scope in other regards, such as rotation, metallicity, ^{56}Ni mass and distribution, and possible effects of binarity. Moreover, we do not have models with masses lower then 9M$_\odot$, which may contribute to CCSNe. For example, SN 2008bk is very underluminous with low expansion velocities as is very likely a lower mass progenitor than we have in our set (Mattila et al., 2008; Van Dyk et al., 2012; Lisakov et al., 2017; Lisakov et al., 2018; Martinez & Bersten, 2019; O'Neill et al., 2021). With these limitations in mind, we still find good fits for two observed CCSNe, notably 16.0 M$_\odot$ for SN 2012aw and 20.0 M$_\odot$ for SN 2017eaw. Our best fit progenitors tend to have larger ZAMS masses than those estimated from pre-explosion imaging, for example by about 5M$_\odot$ for SN2017eaw. This difference of about 5M$_\odot$ is not uncommon – Goldberg & Bildsten (2020), for example, find a possible ZAMS mass for SN2017eaw of 10.2M$_\odot$– also about 5M$_\odot$ from the value obtained from pre-explosion imaging. We have presented light curves for which we do not find particularly good fits for the sake of completeness and to show the strengths and weakness of the current progenitor set. As previously
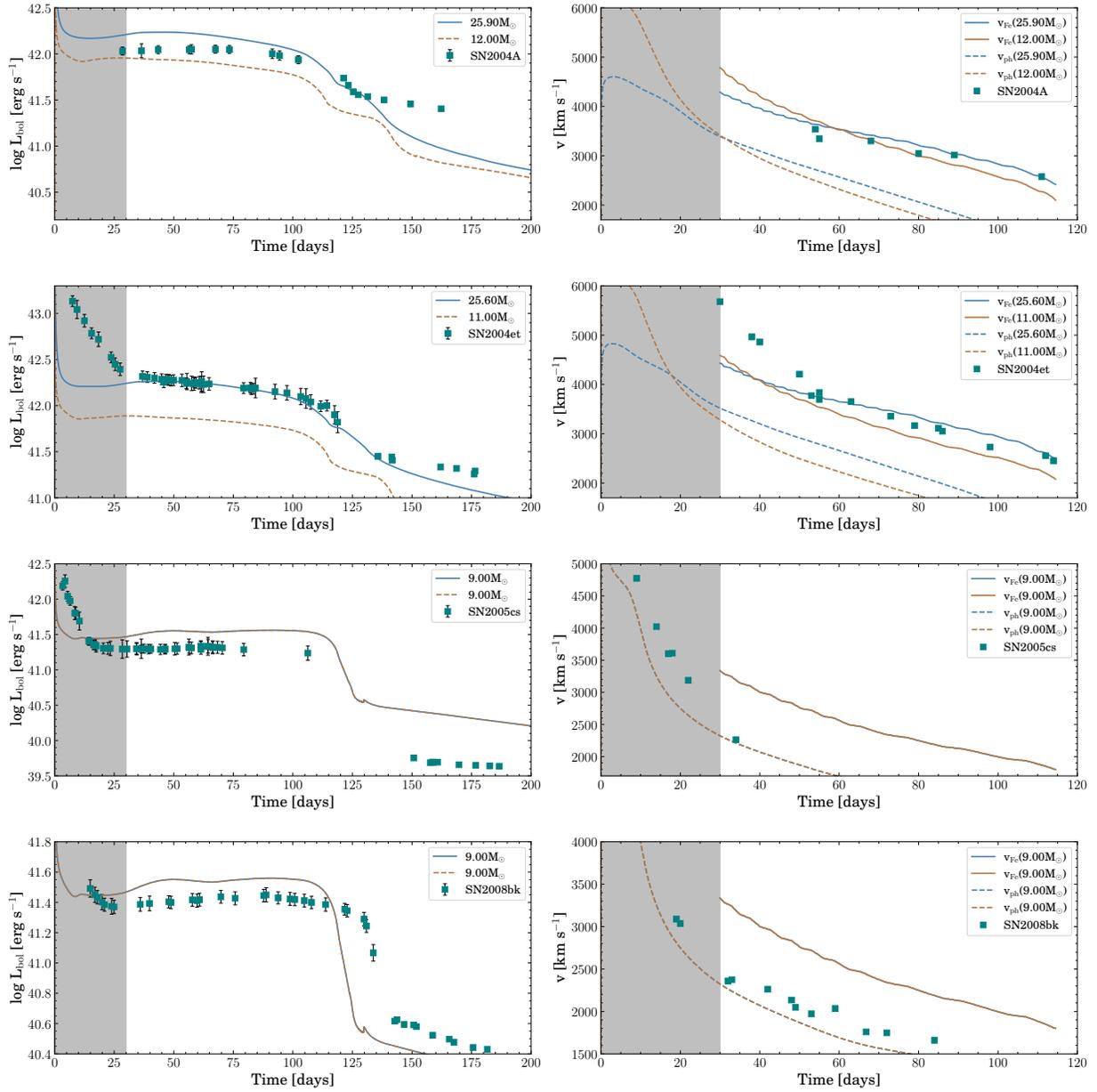
Figure 2.8 Left: Comparison between STIR + SNEC light curves (blue lines) and observations (squares). Right: Comparison between STIR + SNEC velocity evolution (lines) and Fe II λ5169 line velocity observations (squares). Solid lines show approximate Fe II λ5169 calculated in post-processing and dashed lines show the proper photospheric velocity. In both plots, blue lines show best fit STIR + SNEC models and gold lines show light curves for ZAMS masses obtained from pre-explosion imaging (Smartt, 2015; Davies & Beasor, 2018; Eldridge & Xiao, 2019). The gray shaded region shows the first 30 days that we omit from fitting. *From top to bottom*: SN 2004A, SN 2004et, SN 2005cs, and SN 2008bk.
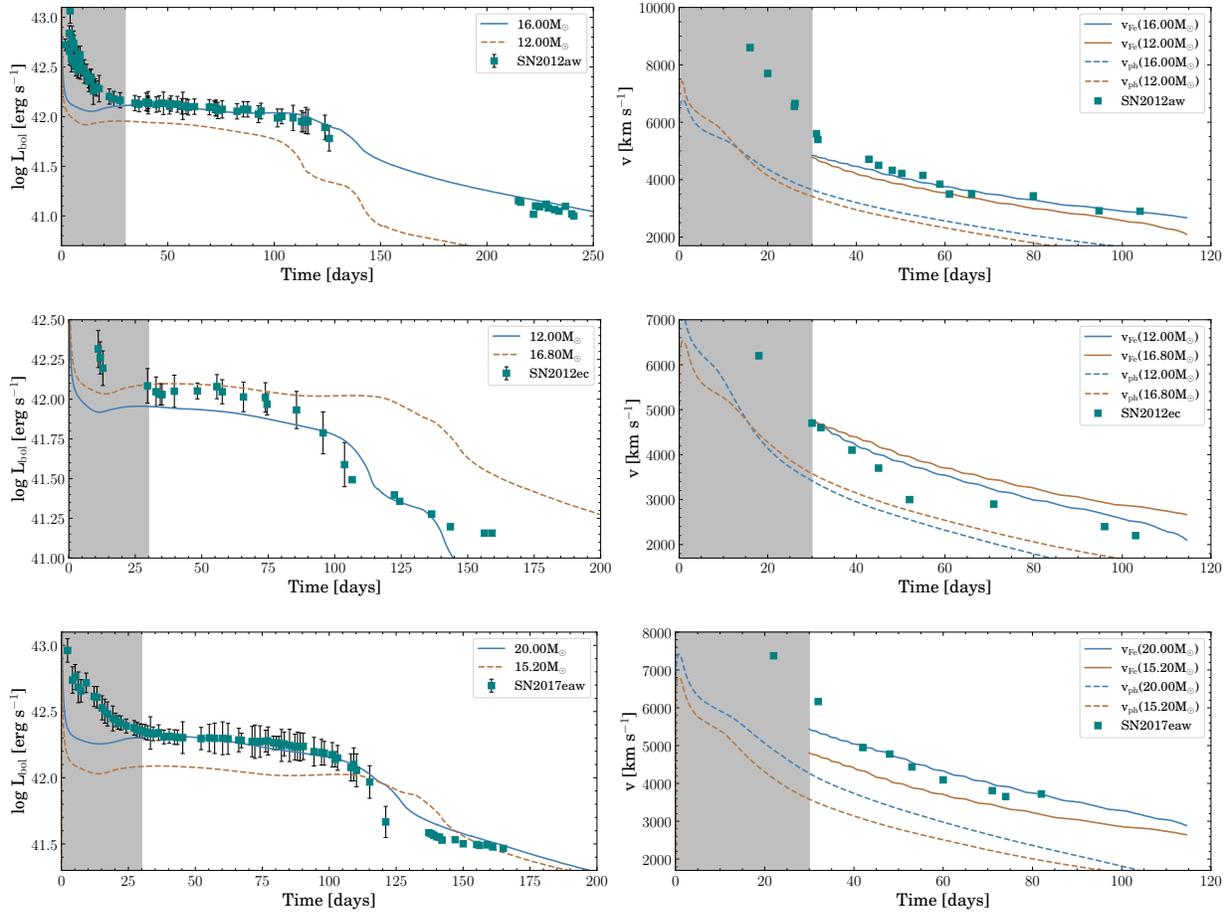
Figure 2.9 Same as Figure 2.8 but for *(from top to bottom)*: SN 2012aw, SN 2012ec, and SN 2017eaw

mentioned, there is no reason for this progenitor set to perfectly fit any specific light curve.

The differences highlighted in Figures 2.8 and 2.9 show the inherent degeneracy involved in extracting CCSNe progenitor properties. As shown in Goldberg et al. (2019); Dessart & Hillier (2019), there are familes of progenitor properties that can lead to a given light curve. This further highlights that light curve fitting is extremely degenerate – not only in the ways explored in previous works, but also in the method used to drive the explosion. Thus, we do not claim that these progenitors necessarily reflect the true progenitors, they simply match the observations given a set of neutrino-driven explosions. It has become clear that more work in needed to infer progenitor properties. Matching an observed SN is a necessary, but not sufficient, condition for

Table 2.1 Best-fit ZAMS mass and explosion energy for SN2012aw and SN2017eaw for our work and others in the literature. For all works we present the best fit model reported with the exception of 5 (Goldberg & Bildsten, 2020), where we list all presented matches.
[†] The authors only present ejected mass, so we present that as a lower bound for the ZAMS mass.

SN	Quantity	This work	1	2	3	4	5
2012aw	M_{ZAMS} [$M_\odot$]	16.0	23.0	14.35	20.0	>19.6[†]	–
	E_{expl} [10^{51} erg]	0.66	1.40	0.90	0.52	1.5	–
2017eaw	M_{ZAMS} [$M_\odot$]	21.9	–	15.47	–	–	(10.2, 12.7, 17.2, 11.9, 15.7, 19.0)
	E_{expl} [10^{51} erg]	1.09	–	1.29	–	–	(0.65, 0.84, 1.30, 0.90, 1.10, 1.50)

References: (1) Martinez & Bersten (2019); (2) Martinez et al. (2020); (3) Morozova et al. (2018); (4) Pumo et al. (2017); (5) Goldberg & Bildsten (2020).

inferring progenitor and explosion properties.

Finally, we summarize our best fitting models for those light curves for which we see good agreement (SN2012aw and SN2017eaw) in Table 2.1 alongside various other sources.

2.3.3 Correlations

In this section, we address the primary goal of this study, which is to connect light curve properties to progenitor properties using a statistically significant sample of simulations. Figure 2.10 shows the Spearman's correlation matrix for the observable quantities and progenitor properties that we consider for the STIR + SNEC models. Our goal is to assess direct correlations between individual quantities, and for this reason we do not consider correlations with ZAMS mass because it does not correlate with any single quantity. In many cases, we are simply recovering well-known correlations, which provide a sanity check on our methods. For example, relationships between ejecta velocity and luminosity have been used in SNe IIP supernova cosmology (Hamuy, 2005; Nugent et al., 2006; Poznanski et al., 2009). Relationships between photometric and spectroscopic observables, L_{50}, v_{50}, and t_p, and properties of the progenitor, such as R_{500} (the pre-supernova progenitor radius in units of $500R_\odot$) in addition to the explosion energy are used in scaling relationships, such as those in Popov (1993); Kasen & Woosley (2009); Sukhbold et al. (2016); Goldberg et al. (2019).

We first consider some typical observables of SNe IIP light curves – the plateau luminosity (L_{50}), plateau duration (t_p), and ejecta velocity measured through the Fe 5169 line during the
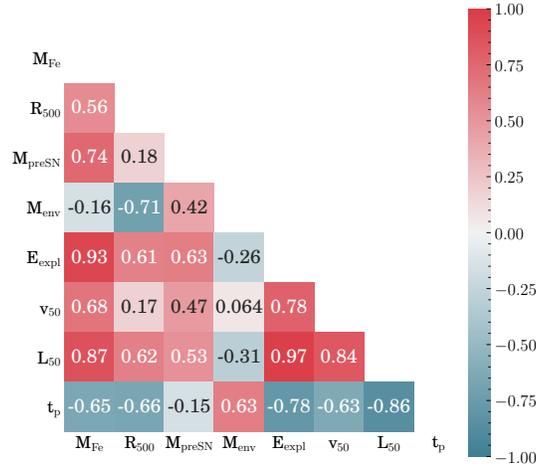
Figure 2.10 Correlation matrices for observable quantities and properties of the progenitors for STIR + SNEC. Here we consider the following quantities: iron core mass (M_{Fe}), progenitor radius (R_{500}), explosion energy (E_{expl}), ejecta velocity at day 50 (v_{50}) as determined from the Fe II (5169) line, log of the plateau luminosity at day 50 (L_{50}), and plateau duration (t_{p}). The lower left half of the matrix shows the Spearman rank correlation coefficient for each pair of quantities.

plateau phase (v_{50}). These observables correlate with each other and are expected to correlate with properties of the progenitors, such as the presupernova radius (R_{500}) and envelope mass (M_{env}). We observe significant correlations between t_p, L_{50}, and R_{500}. Correlations with R_{500} tend to be non-monotonic (see, e.g., Figure 2.1), which is why they tend to have weaker values of the correlation coefficient. There is a moderate correlation between the L_{50} and v_{50} and the presupernova mass (M_{preSN}).

The explosion energy (E_{expl}, see Section 2.2.2) is expected to correlate with both progenitor properties and observable properties. Correlations between E_{expl} and observable properties are monotonic relationships (i.e., always increasing or always decreasing, but not necessarily linear), for example with a correlation coefficient of 0.97 for $L_{50} - E_{\mathrm{expl}}$. This is because in the self-consistent STIR + SNEC models, the explosion energies are the total positive energies of unbound material as liberated by neutrino heating and is thus correlated with properties of the core (and thus, the rest of the progenitor properties through stellar evolution) of the progenitor.

Finally, we turn our attention to connections between properties of the core of the progenitor

and observable quantities. Motivated by connections between explosion energy and the compact remnant, we explore correlations with the iron core mass (M_{Fe}). Progenitors with more massive iron cores tend to liberate more gravitational binding energy, have higher neutrino luminosities, and ultimately are associated with more energetic explosions for progenitors that successfully explode. The origins of this correlation can be seen in the bottom panel of Figure 2.2 through the connection between iron core mass and explosion energy. This correlation, therefore, once again highlights the need for realistic physics in explosion models even in 1D. Equipped with this correlation, and the previously mentioned relationships between explosion energy and observables, one might expect some imprint of the iron core mass on the observables. Indeed, for the STIR + SNEC models we observe a very strong, linear relationship between iron core mass and plateau luminosity. We note that the compactness parameter $\xi_{2.5}$ (O'Connor & Ott, 2011) produces a stronger correlation. This, however, is of little practical use, as the 9-12$M_\odot$ progenitors have nearly zero values of the compactness parameter (≤ 0.02), breaking the trend for the most common progenitors, and the iron core mass is a more physical quantity (i.e., does not depend on the exact choice of mass coordinate for the measurement). The compactness parameter and iron core mass are very tightly correlated and both provide a measure of the gravitational binding energy available in the explosion.

A relationship between iron core mass and supernova observables helps constrain stellar evolution models and characterize the diversity of supernova light curves. Figure 2.11 shows iron core mass versus plateau luminosity at day 50. Higher luminosity events tend to originate from progenitors with more massive iron cores. Ultimately, more massive stellar cores collapse to form more massive proto-neutron stars, liberating more gravitational binding energy in the process and resulting in higher neutrino luminosities emanating from the PNS surface. All of this results in a more energetic explosion and a brighter supernova. In Table 3.1 we report the fits coefficient for the M_{Fe}-L_{50} relationship and the associated variances and covariances for a linear fit of the form $y = ax + b$.

This correlation, though simple, has a profound implication that we can constrain core structure from optical photometry alone. While not necessarily providing a *precise* measure of the iron core

59
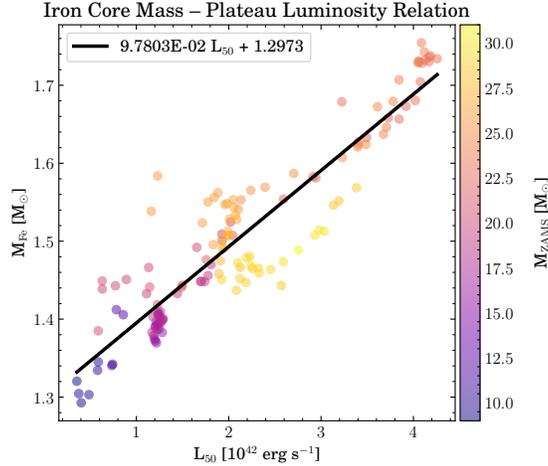
Figure 2.11 Iron core mass M_{Fe} versus plateau luminosity at day 50 L_{50}.

mass for individual events due to observational error and uncertainties on the fit parameters from scatter, which we quantify below, it provides a method for comparing the cores of *virtually all* SNe IIP simultaneously. Furthermore, these parameter estimates can be used to constrain stellar evolution models for CCSN progenitors. We find a similar, although slightly weaker, correlation between the ejecta velocity at day 50, v_{50}, as well, but most LSST sources won't have a spectroscopic follow-up so this is of limited use.

For the case of Sukhbold et al. (2016), fewer massive iron cores produced explosions, and the explosions had a tendency to be brighter. Using their data we find slope and intercept parameters of 0.033 and 1.344, respectively.

For any relationship of this type to be useful, error must be taken carefully into account. The optimal fit parameters were obtained with a least squares method. However, it is known that the covariances provided by least squares methods are not appropriate for a wide range of problems, including those with a non-Gaussian intrinsic scatter among other criteria (see, e.g., Clauset et al., 2009, and references therein). For this reason, we resort to a bootstrapping method (Efron, 1979) to obtain the errors on the fit parameters. This method has the advantage of making no assumptions about the underlying distribution of the data. Instead, bootstrapping operates by resampling the data M times with replacement. For each resampling, a new fit is made and those fit parameters stored. Then, estimates of the variance and covariance of parameters u and v are given by

$$\sigma_u^2 = \frac{1}{M} \sum_{j=1}^{M} \left(u_j - u\right)^2 \tag{2.5}$$

$$\sigma_{uv} = \frac{1}{M} \sum_{j=1}^{M} \left(u_j - u\right)\left(v_j - v\right) \tag{2.6}$$

where u and v are the optimal fit parameters and each of u_j, v_j are the fit parameters for each of the M resamples. These error estimates tend to be, for this application, somewhat smaller than parameter errors obtained through a simple least squares method. The full set of fit parameters, variances, covariances, and adjust coefficient of determination are supplied in Table 3.1. We note that the fit presented uses the non-log plateau luminosity as its independent variable, as opposed to the log luminosity presented in other parts of the paper. Then, given errors on the fit parameters it is straightforward to compute the error on an iron core mass estimate. For a linear fit, we propagate the combined observational – fit parameter uncertainty in the following way:

$$\sigma_{M_{Fe}}^2 = \sigma_a^2 L_{50}^2 + \sigma_{L_{50}}^2 a^2 + \sigma_b^2 + \sigma_{res}^2 + 2 L_{50}\sigma_{ab}, \tag{2.7}$$

where L_{50} is the luminosity at day 50 in erg s^{-1} and where we have included explicitly the covariance of the fit parameters a and b. In order to further account for intrinsic scatter in the relationship, we have included σ_{res} which is the 67% percentile on the residual distribution $r_i = |M_{Fe} - \hat{M}_{Fe}|$, where $\hat{M}_{Fe}$ is computed from the fit.

As an example, we estimate iron core masses for six well-observed SNe, shown in Table 2.3. Data for SN1999em, SN2003hl, and SN2007od are taken from Gutiérrez et al. (2017a,b). Data for SN2004et, SN2012aw, and SN2017eaw are taken from Martinez et al. (2020).

2.4 Discussion and Conclusions

We present synthetic bolometric light curves for 136 solar metallicity, non rotating CCSNe progenitors and consider statistical relationships for those with ZAMS masses ranging from 9M$_\odot$ to 31M$_\odot$. These light curves are calculated with SNEC using the CCSNe simulated in Couch et al. (2020) as the initial condition. This allows for light curves obtained without a user-set explosion energy. Our ^{56}Ni yields were fit from Sukhbold et al. (2016) who exploded the same progenitors with

Table 2.2 Linear fit paramaters for iron core mass (M_Fe) to plateau luminosity (L_{50})in units of 10^{42} erg s^{-1}. The first two rows shows the optimal fit parameters. The next two rows shows the error on each parameter. The next row shows the covariance between the parameters and the residual error accounting for intrinsic scatter. The final row shows the adjusted coefficient of determination $\bar{R}^2$ for the fit.

	$M_\text{Fe} = aL_{50} + b$
a	0.0978
b	1.29
σ_a	3.17×10^{-3}
σ_b	8.31×10^{-3}
σ_{ab}	-2.33×10^{-5}
σ_res	3.79×10^{-2}
$\bar{R}^2$	0.85

Table 2.3 Estimated iron core masses (M_Fe) and uncertainties (σ_{M_Fe}) for a sample of well-observed supernovae.

SN	M_Fe [M$_\odot$]	σ_{M_Fe} [M$_\odot$]
1999em	1.42	0.041
2003hl	1.34	0.039
2004et	1.48	0.039
2007od	1.50	0.040
2012aw	1.43	0.039
2017eaw	1.49	0.040

an expansive reaction network coupled to the evolution. This is sufficient for the current work, and future work with FLASH will include detailed nucleosynthesis calculations. These light curves, as well as the SNEC initial profiles and necessary parameters, are provided online[5]. We also include the necessary binding energy of our progenitors to correct STIR's explosion energy to produce identical results with a thermal bomb explosion. In the online resources, we furthermore provide the light curves for the $M_\text{ZAMS} > 31$M$_\odot$ models that successfully explode. For progenitors that explode with STIR, we follow the explosions in SNEC to produce bolometric light curves, forming a large, statistically significant set of CCSN light curves followed from high-fidelity explosions allowing us to address relationships between progenitor properties and properties of the explosion

---

[5]https://doi.org/10.5281/zenodo.6631964

in a statistical way. We consider the full shape of these light curves, but also reduce them to characteristic quantities such as the plateau luminosity, plateau duration, and ejecta velocity.

Next, we show that global trends in light curve properties – such as plateau duration and plateau luminosity – depend sensitively on the explosion model and require explosion energies set by robust physics. To demonstrate this, we compute bolometric light curves for the same set of progenitors using two different thermal bomb models with SNEC. The distribution of explosion energies plays a leading role in setting the distribution of observables across a large sample of progenitors. Thus, the ability to identify global trends in light curve properties and extract progenitor features from them depends sensitively on the determination of explosion energy, underscoring the need for explosions driven with high-fidelity multi-physics models.

We present a simple best-fit procedure to individual, observed CCSN light curves (Martinez et al., 2020). The usual procedure for estimating progenitor properties of observed CCSNe is to construct a large grid of "hydrodynamical models" – usually in ZAMS mass, explosion energy, and perhaps ^{56}Ni mass and distribution – and find a best fit model. This approach results in known degeneracies, for example, as shown by Goldberg et al. (2019); Dessart & Hillier (2019) wherein there are certain families of progenitor and explosion parameters (such as ejecta mass, explosion energy, and ejecta velocity) that produce a given light curve, though pre-explosion radius measurements may help to resolve this degeneracy (Goldberg & Bildsten, 2020; Kozyreva et al., 2020). Our approach differs in that we do not control the explosion properties, instead following a dense set of various ZAMS mass progenitors from neutrino driven explosions. While this does not solve the light curve degeneracy problem, it could reduce the size of the family of explosion properties for a given light curve, as some combinations of explosion energy and stellar mass are not realizable. Although the explosions are not calibrated to observed data we still find great agreement both when comparing to large samples of events and for some individual cases. Intriguingly, we find best-fit ZAMS masses that are greater by as much as $\approx 7 M_\odot$ than those estimated from pre-explosion imaging in tandem with stellar evolution modeling. The fact that hydrodynamic models have tended to find ZAMS masses in agreement with pre-explosion imaging estimates for these

CCSNe (Morozova et al., 2018; Martinez & Bersten, 2019; Martinez et al., 2020) may indicate the danger of exploring too large a parameter space instead of knowing which regions are physically realizable, though we note that some hydrodynamic models have also found noticeably higher masses in better agreement with our conclusions (e.g., Utrobin & Chugai, 2008, 2009). Ultimately, the set of solutions for matching a given observed light curve is degenerate, with many progenitors being capable of producing a given light curve.

Despite the progenitors and explosions in this study not being crafted to reproduce specific events, we find good qualitative agreement with SN2012aw and SN2017eaw. Notably, the luminosity evolution of SN 2012aw is fit by our $16.0M_\odot$ progenitor remarkably well. The best fit progenitors for the observed light curves in this study are not necessarily the progenitors that these explosions originated from – they simply reproduce the observables. We have demonstrated that beyond the now understood light curve degeneracies, there are additional degeneracies inherited from the choice of explosion model. This result is complementary to the recent findings by Farrell et al. (2020) where they showed that a star's final temperature and luminosity cannot be reliably traced back to the star's ZAMS mass – that very different mass stars may end up at the same temperature and luminosity. These results together show that much more work is needed before a SN IIP progenitor's ZAMS mass can be reliably determined – the path from stellar birth to death is not a one-to-one function.

The light curves here present avenues for future work to explore the discussion surrounding explosion energy. There is tension between explosion energies realized in 3D CCSN simulations and energies inferred from fitting hydrodynamical models to observations. The energies from these two methods differ, with those inferred from hydrodynamical modeling being significantly larger (see Murphy et al., 2019, which discusses this tension in detail). On one hand, 3D simulations of very massive progenitors have often simply not asymptoted to their final values within the simulated time. There is also still physics left to include, such as the recently demonstrated affects of magnetic fields on neutrino-matter interactions (Kuroda, 2021) and improved neutrino pair-production rates (Betranhandy & O'Connor, 2020) on the explosion mechanism, neutrino mixing, among other

affects, all of which will likely play a role in setting the final energy. On the other hand, solutions using thermal bomb models have been shown to be degenerate, and these studies access a very large area of this degenerate parameter space and may not necessarily find physically realizable solutions. The methods described here could illuminate or even weaken the tension between these energies by limiting the parameter space spanned by hydrodynamical modeling studies and by using physically-motivated explosions.

The final aim of this study is to leverage the large number of light curves to perform a statistical investigation of relationships between progenitor and explosion properties. Focusing our investigation to SNe II, 136 light curves, a number of correlations between the light curves and their progenitors are found. We find a robust relationship between the iron core mass of the progenitor and the luminosity on the plateau of the SNe. This relationship allows one to, for the first time, constrain properties of the stellar interior from photometry alone. We provide an analytic approximation to the observed correlation, including error, for future use with large survey data such as LSST.

Recently, Curtis et al. (2021) presented synthetic light curves and spectra from a sample of 62 CCSNe with the 1D PUSH model (Ebinger et al., 2017) and SNEC to obtain the light curves. Our results complement one another in several ways – notably, the size and composition of our samples differ. Our sample contains 148 light curves – 136 of which are analyzed in this work – from the same metallicity, compared to their 62 light curves from three different metallicity populations ranging from zero to solar. This allows us to more robustly survey global explosion properties of progenitors from similar origin within the nearby universe. These studies, together, survey a vast range of progenitor properties. The CCSNe simulations in our work are perfmormed with `FLASH` using the STIR model. Notably, STIR requires no tuning to observations, eliminating the potential for biases when simulating progenitors different than the one used for tuning. Importantly, the results from STIR are consistent with 3D simulations. The explosion energies, explodability, and the shape of each as a function of ZAMS mass differ non-trivially for STIR and PUSH (see Couch et al., 2020; Ebinger et al., 2019) and this could impact global trends in explosion properties. On the

other hand, Curtis et al. (2021) obtained their ^{56}Ni distributions using a nuclear reaction network in conjunction with their CCSN simulations. As aforementioned, we estimated ^{56}Ni mass from the explosion energy, informed by KEPLER yields. Curtis et al. (2021) also have a larger diversity of supernova types through their inclusion of sub-solar and zero metallicity progenitors. To keep the scope of the current work contained, we have not produced synthetic spectra for these explosions, whereas Curtis et al. (2021) calculated spectra for their supernovae.

Similarly, Sukhbold et al. (2016) present a sample of synthetic light curves of the same statistical size and originating from the same progenitors using a different parametrized, neutrino-driven explosion mechanism. Using these simulations they present scaling relations to determine explosion and progenitor properties from observables. The outcomes of these simulations – both the explosions and resulting light curves – differ from STIR and this work, having a tendency to be brighter than those produced in this work. It would be interesting, for future work, to investigate the affect of these differences in explosion mechanism when applied to populations of observed CCSNe and implications for inferred properties such as explosion energy.

This work is part of a larger context to *understand* and *predict* full multi-messenger signals from realistic CCSNe. Understanding how variations in progenitors properties tie into variations of different observables will ultimately help to constrain real populations. This work, in tandem with the work of Couch et al. (2020) and Warren et al. (2020), gives us explosion fates, energies, neutron star mass distributions, neutrino signals, approximate GW signals, and now EM signals for a massive suite of neutrino driven CCSNe. It is only through advanced methods – studying in detail all messengers from first principles simulations – used in tandem with growing observational data that we can truly understand these phenomena.

CHAPTER 3

INFERRING TYPE II-P SUPERNOVA PROGENITOR MASSES FROM PLATEAU LUMINOSITIES

> for the relief of the body
>
> and the reconstruction of the
>
> mind.
>
> ——————————————
>
> Adrienne Rich, *Planetarium*

This chapter is based on the published work of B. L. Barker, *et al.* 2023 ApJL **944** 1.

ABSTRACT

Connecting observations of core-collapse supernova explosions to the properties of their massive star progenitors is a long-sought, and challenging, goal of supernova science. Recently, Barker et al. (2022) presented bolometric light curves for a landscape of progenitors from spherically symmetric neutrino-driven core-collapse supernova (CCSN) simulations using an effective model. They find a tight relationship between the plateau luminosity of the Type II-P CCSN light curve and the terminal iron core mass of the progenitor. Remarkably, this allows us to constrain progenitor properties with photometry alone. We analyze a large observational sample of Type II-P CCSN light curves and estimate a distribution of iron core masses using the relationship of Barker et al. (2022). The inferred distribution matches extremely well with the distribution of iron core masses from stellar evolutionary models, and namely, contains high-mass iron cores that suggest contributions from very massive progenitors in the observational data. We use this distribution of iron core masses to infer minimum and maximum mass of progenitors in the observational data. Using Bayesian inference methods to locate optimal initial mass function parameters, we find $M_{\text{min}} = 9.8^{+0.37}_{-0.27}$ and $M_{\text{max}} = 24.0^{+3.9}_{-1.9}$ solar masses for the observational data.

3.1 Introduction

Core-collapse supernovae are the fate of most stars more massive than $M_{\text{ZAMS}} \gtrsim 8 M_\odot$ zero-age main sequence (ZAMS) mass. These stars, at the ends of their lives, inevitably collapse and form an outwardly moving shock that stalls due to neutrino losses and photodissociation of iron group nuclei. Some fraction of these stars will successfully revive their shocks and produce observable supernovae, while others will instead fail and form a black hole. It is certain, now, that an increasingly rich amount of physics is necessary to fully describe the CCSN explosion. For in-depth reviews of the CCSNe mechanism, we refer the reader to, e.g., Mezzacappa (2001, 2005); Janka et al. (2012, 2016); Burrows (2013); Hix et al. (2014); Müller et al. (2016); Couch (2017); Pejcha (2020); Müller (2020); Mezzacappa et al. (2020); Burrows & Vartanyan (2021); Mezzacappa (2022).

In lockstep with theoretical studies, the observational study of CCSNe has also progressed at an

ever increasing rate, with next-generation telescopes such as the Vera C. Rubin Observatory and its primary survey, The Rubin Observatory Legacy Survey of Space and Time (LSST) (Ivezić et al., 2019), posed to observe an unprecedented number of CCSNe and other transient events. Despite the growing repository and fidelity of observational data, few constraints on the cores of CCSN progenitors exist. Such constraints would bound stellar evolutionary models and guide studies of the CCSN explosion mechanism. This absence is due, in part, to the fact that photons are emitted from the photosphere which resides primarily in the original H envelope of the progenitor star, far above the core of the star in which the explosion is generated. Ideally, such constraints would come from neutrino and gravitational wave (GW) observations as they are produced directly in the core and propagate nearly unhindered through the progenitor carrying information of the inner core. To date, however, there has been only one detection of supernova neutrinos (Arnett et al., 1989, SN1987A). With modern detectors, only CCSNe occurring within the galaxy may be detected (Scholberg, 2012). There have been no confirmed detections of GWs from CCSNe. The current suite of detectors can only detect GWs from a CCSNe occurring approximately within the Galaxy (Abbott et al., 2016, 2020; Szczepańczyk et al., 2021). CCSNe are, for the vast majority of events, only detectable through electromagnetic emission.

While 3D simulations offer the most complete model of the CCSN explosion, they are computationally expensive and have limited predictive power for populations. Recently, phenomenologically modified 1D simulations have been used to great effect to simulate hundreds to thousands of CCSNe (Pejcha & Thompson, 2015; Perego et al., 2015; Ebinger et al., 2017; Sukhbold et al., 2016; Couch et al., 2020). The low computational cost of these sets of simulations allow for very powerful statistical studies. In this spirit, Meskhi et al. (2021) compared the observed neutron star (NS) and black hole (BH) mass distributions to those obtained with the PUSH method (Perego et al., 2015) to constrain the dense matter equation of state. Other works have used these methods to probe the sensitivity to the nuclear matter equation of state (e.g., Schneider et al., 2019; Yasin et al., 2020; Ghosh et al., 2022; Boccioli et al., 2022) and to electron capture rates (Johnston et al., 2022). These 1D methods also allow for the production of light curves from realistic simulations for suites of

69

progenitors (Curtis et al., 2021; Barker et al., 2022), which opens up the statistical power of these suites of simulations to electromagnetic observables.

Recently, Barker et al. (2022) (henceforth B22) simulated a landscape of 136 light curves for SNe II-P from neutrino-driven turbulence-aided explosions[1]. From this set of light curves, they identified a number of correlations between observable features and properties of the progenitor. Notably, B22 find that iron core mass is linearly correlated with the plateau luminosity to a high degree of significance – more massive cores result in more energetic and brighter explosions. This relationship provides a way to constrain properties of the cores of populations of CCSN progenitors from photometry alone. Notably, measurements of the plateau luminosity may be made robustly and cheaply for a huge swath of CCSNe, especially so as LSST comes online.

Here, we combine the relationship between iron core mass and plateau luminosity of B22 with the well studied Type II-P CCSN sample presented in Anderson et al. (2014); Gutiérrez et al. (2017a,b) (henceforth G17) in order to infer iron core masses for a large sample of observed CCSNe. We use the inferred distribution of iron core masses to constrain the minimum and maximum masses of progenitors in the sample.

In this Letter, we begin by reviewing the numerical methods and results of B22 in Section 7.3. We also briefly describe the observational sample of G17 in that section. We present the results of our Bayesian analysis for inferring CCSN progenitor iron core masses and ZAMS masses in Section 3.3, showing that observations of the Type II-P plateau luminosities alone can tightly constrain progenitor masses of populations.

3.2 Methods and Input Data

In B22, the authors simulated light curves for 136 SNe II-P starting from the progenitors of Sukhbold et al. (2016) by coupling neutrino radiation hydrodynamics calculations with a Lagrangian radiation-hydrodynamics code to simulate bolometric light curves.

These non-rotating, solar metallicity progenitor models cover a range of ZAMS masses from $9 - 31$ M$_\odot$[2] and were created with the KEPLER code assuming no magnetic fields and single

---

[1]The data may be found at https://doi.org/10.5281/zenodo.6631964

[2]Sukhbold et al. (2016) provides 200 progenitors with masses $9 - 120$M$_\odot$, but only those up to 31M$_\odot$ produced

star evolution. They span a wide, realistic range of progenitor properties making them ideal for landscape studies such as that in B22.

The collapse of the progenitors' cores and subsequent explosions were simulated with FLASH[3] (Fryxell et al., 2000; Dubey et al., 2009, 2022) in Couch et al. (2020) using the STIR turbulence-aided explosion model. Turbulence has been shown to be key in simulating successful, realistic explosions (see, e.g., Burrows et al., 1995; Murphy & Meakin, 2011; Couch & Ott, 2015; Mabanta & Murphy, 2018). The effects of turbulence and convection are included in a parametrized way with mixing length theory as a closure. These effects are parametrized by 5 free parameters – a mixing length type parameter and four diffusion parameters – the latter of which have little impact on the dynamics. The mixing length type parameter is calibrated by comparison to sets of 3D simulations of CCSNe. The inclusion of turbulence in STIR allows for successful explosions in 1D that reproduce the results of 3D simulations (Couch et al., 2020). without the need for parameterized neutrino physics or tuning to specific events.

To produce synthetic bolometric light curves, STIR is coupled with the SuperNova Explosion Code (SNEC)[4] (Morozova et al., 2015). SNEC is a Lagrangian, flux-limited diffusion radiation hydrodynamics code that allows for the calculation of bolometric light curves. It includes all of the necessary physics to model CCSN light curves beyond the initiation of the explosion, including a Saha ionization solver and radiative heating due to ^{56}Ni decay.

While SNEC alone typically requires an artificially driven explosion (e.g., a thermal bomb), STIR + SNEC together allow for the simulation of light curves from neutrino-driven explosions without user-set explosion energies that may not be realizable in nature. This allows for statistical studies that are not influenced by the user's choice of thermal bomb energetics. We refer the reader to B22 for more details on the coupling of STIR and SNEC and the results of that study.

A primary result from B22 was a linear relationship between the mass of a progenitor's iron core and its resulting plateau luminosity. Simply, more massive iron cores release more binding

---

Type II-P SNe in B22.

[3]https://flash-x.org

[4]http://stellarcollapse.org/SNEC

71

Table 3.1 Linear fit parameters for iron core mass (M_Fe) to plateau luminosity (L_{50}) from B22. The first two rows shows the optimal fit parameters. The next two rows shows the uncertainty on each parameter. The next row shows the covariance between the parameters and the residual error accounting for intrinsic scatter.

	$M_\text{Fe} = aL_{50} + b$
a	0.0978
b	1.297
σ_a	3.17×10^{-3}
σ_b	8.31×10^{-3}
σ_{ab}	-2.33×10^{-5}
σ_res	3.79×10^{-2}

energy and result in more energetic, brighter explosions. In Table 3.1 we recap the fit coefficients and their uncertainties for a linear fit of the form,

$$M_\text{Fe} = aL_{50} + b, \tag{3.1}$$

where M_Fe is in solar masses and L_{50} is in units of 10^{42} erg s^{-1}.

Here the iron core mass is defined by the mass coordinate where the Si and iron group mass fractions reach sufficient thresholds, separating the iron core from the Si shell.

Variances and covariances were calculated by bootstrapping (Efron, 1979) and we include a term σ_res calculated from the residuals that may be added in quadrature with the other sources of uncertainty to calculate the uncertainty on the iron core mass inference

$$\sigma_{\text{M}_\text{Fe}}^2 = \sigma_a^2 L_{50}^2 + \sigma_{L_{50}}^2 a^2 + \sigma_b^2 + \sigma_\text{res}^2 + 2L_{50}\sigma_{ab} \tag{3.2}$$

where $\sigma_{L_{50}}$ is the uncertainty on the plateau luminosity measurement and the other parameters are as previously defined.

We consider the observation sample of SNe II-P studied in G17 as an application of the results of B22. This sample represents a very large, well studied, statistical sample of SNe II-P, containing over 100 supernovae with both photometry and spectra. A large number of properties have been estimated for these SNe, including ^{56}Ni mass, explosion epoch, plateau duration, line velocities, various light curve slopes, and more. These observations come from a range of sources spanning
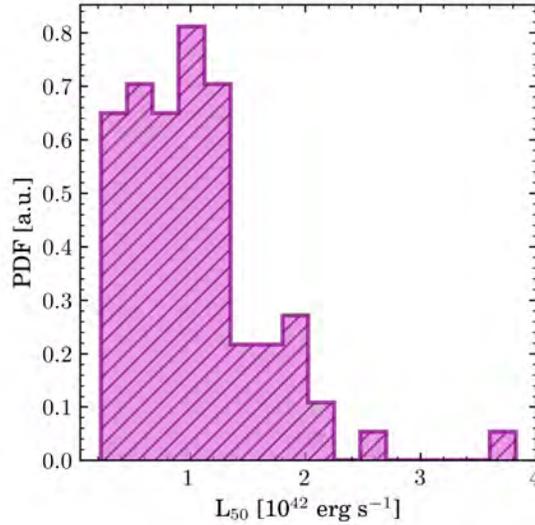
Figure 3.1 Distribution of observational plateau luminosities used in this work, taken from Gutiérrez et al. (2017a,b).

from 1986 to 2009, covering the nearby universe out to about z = 0.08. The sample contains both SNe II-P and II-L CCSNe, although for the analysis here we have excluded all Type II-L events, giving us a sample of 82 Type II-P SNe. Figure 3.1 shows the distribution of plateau luminosities from these data. For details about the data, collection, and analysis see G17 and references therein.

3.3 Analysis and Results

We begin by considering the set of observations from G17 under the lens of the iron core mass – plateau luminosity relationship of B22. When using the B22 fits, we include only a subset of the observational sample, excluding Type II-L events and events that did not have sufficient data to discern the type. We also exclude a handful of II-P events that were notably dimmer or brighter than the synthetic light curves obtained in B22 to avoid extrapolation. This gives us a sample of 82 Type II-P CCSNe.

Figure 3.2 shows (left) the iron core mass distribution inferred from the G17 sample (unfilled black histogram) using the results of B22. The data are plotted with large bins representative of the uncertainties. Also plotted is the distribution of iron core masses of the Sukhbold et al. (2016) progenitor set, convolved with the Salpeter initial mass function (IMF) (purples), for simulations that produced explosions in STIR. The histogram colors represent the ZAMS mass range of the
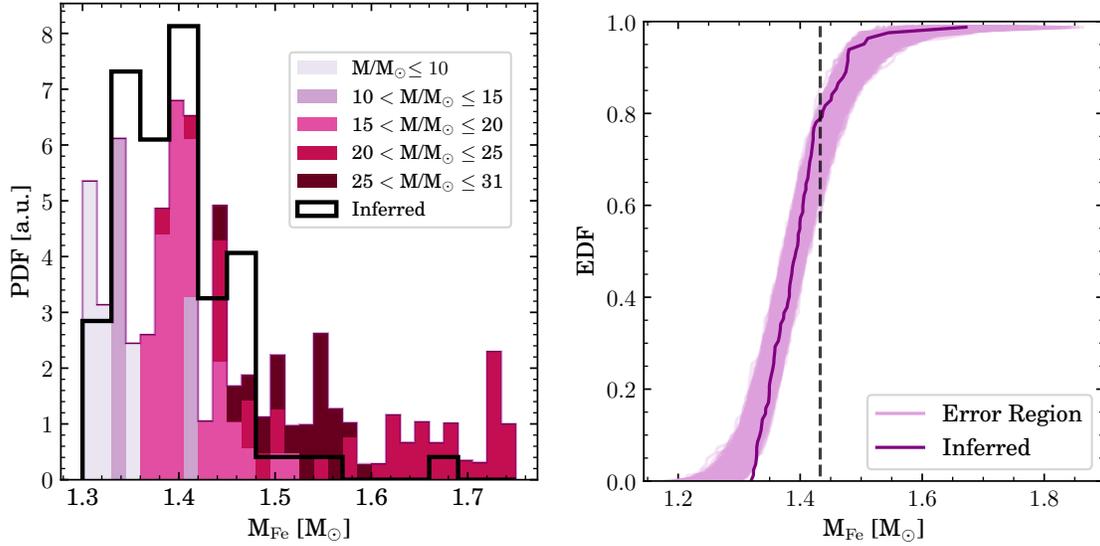
Figure 3.2 *Left:* Iron core mass distributions for the Sukhbold et al. (2016) progenitor set, convolved with the Salpeter IMF, for simulations that successfully produced explosions in STIR. Color indicates the ZAMS mass range of the progenitor in a bin. The unfilled black histogram represents the iron core mass distribution for the G17 sample determined by our M_{Fe}–L_{50} fit. Bin widths for the inferred distribution are $0.03 M_\odot$ to be comparable to iron core mass uncertainties. *Right:* Empirical distribution function (EDF) for the inferred iron core mass distribution of the G17 sample. The shaded regions represent the error region on the EDF due to the 68% uncertainties on the iron core mass inferences. The dashed black line represents the iron core mass where the primary contribution is from progenitors with ZAMS mass above $16.5 M_\odot$, which is representative of the early Smartt (2015) result.

progenitor of origin. We find remarkable agreement between the peaks of the distributions between the two samples. Most notable is the right side of the distribution, occurring around $1.5 M_\odot$, which is composed almost completely of progenitor stars with initial masses greater than or equal to about $16 M_\odot$. This provides evidence of very high mass stars in the G17 sample. The right panel shows the equivalent empirical distribution function (EDF, dark line). The light shaded area represents the error region on the EDF resulting from the uncertainties on the iron core mass inferences, obtained via Monte Carlo uncertainty propagation. The vertical dashed black line represents the iron core mass where, in the Sukhbold et al. (2016) progenitors, the primary contribution is from progenitors with ZAMS mass above $16.5 \, M_\odot$, signifying evidence of high mass progenitors in the data.

Given a distribution of iron core masses inferred from observational data (M_{Fe}^{obs}), we may begin to ask questions about the progenitor population. The distribution M_{Fe}^{obs} should encode information
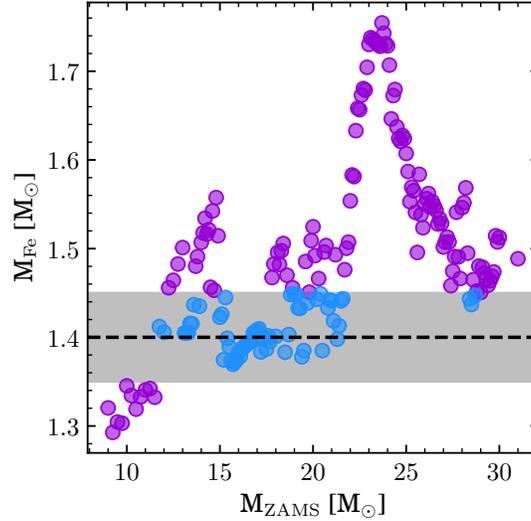
Figure 3.3 Iron core mass (M_{Fe}) distribution for the Sukhbold et al. (2016) progenitors up to $31M_\odot$. The horizontal dashed line represents a hypothetical $1.4M_\odot$ iron core with an uncertainty of $0.05M_\odot$ (shaded region). The blue points represent ZAMS mass models that could, within the uncertainty, produce such an iron core.

about, for example, the underlying distribution of progenitor masses. Unfortunately, the mapping

between iron core mass and ZAMS mass is highly degenerate and a given iron core mass could

potentially belong to one of several progenitors, disallowing a simple transformation from iron core

mass to ZAMS mass. Figure 3.3 shows the iron core masses as a function of ZAMS mass for the

Sukhbold et al. (2016) progenitor set. We show a hypothetical iron core mass inference of $1.4M_\odot$

with $0.05M_\odot$ uncertainties shown by the shaded band, highlighting the difficulty of recovering

ZAMS mass directly from iron core mass. This is a symptom of a much larger difficulty, that

determining the ZAMS mass of a given event from any one quantity is highly degenerate. The

mapping from ZAMS mass to iron core mass provided through a set of stellar evolutionary models

is, however, simple.

To alleviate this issue of retrieving the ZAMS mass, we apply Bayesian inference methods to

seek an initial mass function (IMF) whose stellar population would result in the distribution M_{Fe}^{obs}.

We begin by sampling progenitors from the cumulative distribution function (CDF) $F(m)$ of the

IMF,

$$F(m) = (m^{1-\alpha} - M_{\text{min}}^{1-\alpha})/(M_{\text{max}}^{1-\alpha} - M_{\text{min}}^{1-\alpha}). \tag{3.3}$$

Here, $M_{\text{min/max}}$ is the minimum/maximum mass of progenitors producing SNe II-P and α is the slope of the IMF. In the results presented here we take the canonical Salpeter IMF slope of 2.35. Not all of these progenitors in a given range will produce CCSNe, however, so before mapping these progenitors to a set of iron core masses, we must make an assumption about explodability. Here, we use the explodability results of Couch et al. (2020), consistent with the rest of the methods used in this study, denoting $f_E(M_{\text{ZAMS}})$ as the sampled progenitors that produce CCSNe under a given explodability result f_E. Given this filtered set of progenitors, we may then estimate the inferred set of iron core masses using the mapping $f_{M_{\text{Fe}}} : f_E(M_{\text{ZAMS}}) \mapsto M_{\text{Fe}}$ which, given a stellar evolutionary set, maps a ZAMS mass to an iron core mass. All that remains is to assess how close the inferred observational distribution $M_{\text{Fe}}^{\text{obs}}$ and the hypothetical distribution for a given minimum and maximum mass $\hat{M}_{\text{Fe}}$ are to each other. We use the Anderson-Darling statistic $A^2(F_1, F_2)$ (Anderson & Darling, 1952) to assess the closeness of the two distributions F_1 and F_2. To summarize, we sample progenitors from a given IMF, apply an explodability result, get those models' iron core masses, and compute the distance between this distribution and that inferred from observations.

Using this approach, we find the posterior distribution of IMF parameters

$$\begin{aligned} P\left(M_{\text{min}}, M_{\text{max}} | M_{\text{Fe}}^{\text{obs}}\right) \\ \propto \prod_i \mathcal{L}\left(M_{\text{Fe}}^{\text{obs}} | M_{\text{min}}, M_{\text{max}}\right) P\left(M_{\text{min}}\right) P\left(M_{\text{max}}\right) \end{aligned} \tag{3.4}$$

where $P(M_{\text{min}})$ and $P(M_{\text{max}})$ are uniform priors with $M_{\text{min}} \in [9.0, 15.0]$ and $M_{\text{max}} \in [15.0, 31.0]$. The extreme values represent the smallest and largest progenitors to produce a Type II-P SNe in B22. The cutoff of $15M_{\odot}$ is arbitrary and in practice we need only specify that $M_{\text{min}} < M_{\text{max}}$, but both parameters stayed far from $15M_{\odot}$ so this choice is justified. The likelihood function for a distribution is given by

$$\mathcal{L}\left(M_{\mathrm{Fe}}^{\mathrm{obs}}|M_{\min}, M_{\max}\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-D^2\left(M_{\mathrm{Fe}}^{\mathrm{obs}}, \hat{M}_{\mathrm{Fe}}\right)/2\sigma^2} \tag{3.5}$$

where D^2 is an appropriate distribution distance metric and for the uncertainty σ we use the 68% uncertainty on the EDF $\sqrt{ln(2/\beta)/2n}$ with $\beta = 1 - 0.68$. Here, we use the Anderson-Darling quadratic EDF statistic: $D^2 \equiv A^2 (F_1, F_2)$ for EDFs F_1 and F_2, which will become the inferred and sampled iron core mass distributions.

We use a distance measure between the EDFs as opposed to the probability density function to avoid issues with binning or kernel density estimators. The Anderson-Darling statistic has the quality of being sensitive to the tails of the distribution, which carry information about the least and most massive progenitors. There are other possible choices for the distance metric, such as the Kolmogorov-Smirnov measure or the energy distance.

To infer the posterior distributions, we use the above process with the Markov Chain Monte Carlo (MCMC) package `emcee` (Foreman-Mackey et al., 2013). In the MCMC algorithm we use 512 parallel walkers each running for a chain length of 32,768 steps – about 800 autocorrelation times – and a burn in phase of over 100 autocorrelation times. Figure 3.4 shows the resulting posterior distributions for $M_{\min}$ and $M_{\max}$. We find $M_{\min} = 9.8^{+0.37}_{-0.27}$ and $M_{\max} = 24.0^{+3.9}_{-1.9}$, where the uncertainties are the 68% percentiles of the posterior distributions.

In this analysis we incur a set of limitations from the stellar evolutionary models used. Notably, we do not consider progenitors less massive then $9M_\odot$ as this is the least massive progenitor in the Sukhbold et al. (2016) set. It is likely possible for less massive progenitors to form an iron core and result in core-collapse supernovae. In the Sukhbold et al. (2016) progenitor set, the $9M_\odot$ model forms a very light iron core and this is our lower bound. While there may be stars less massive than this in the observational sample, we do not expect these to be a significant fraction of the SNe II-P population. Similarly, we have an implicit upper limit of 31 solar masses – the most massive progenitor in the evolutionary set to retain a sufficient hydrogen envelope to produce a Type II-P SNe.

In the previous analysis we constructed sample progenitor populations from an IMF with a fixed
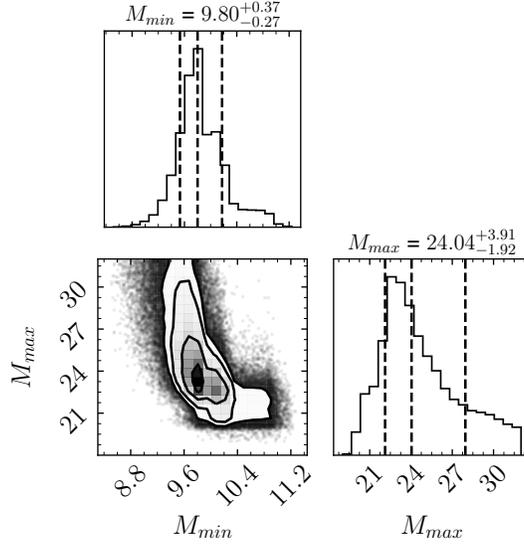
Figure 3.4 The posterior distributions for $M_{\min}$ and $M_{\max}$ in units of M$_\odot$.

canonical power-law slope of $\alpha = 2.35$ (Salpeter, 1955). Observations of young stellar clusters contest the traditional power-law slope, with many studies finding both steeper and narrower slopes (e.g., Dib, 2014; Weisz et al., 2015). Constraints from supernova remnant masses also find a spread of IMF parameters (e.g., Williams et al., 2019; Koplitz et al., 2021). Instead of fixing the slope of the IMF, then, we may allow it to vary along with the minimum and maximum masses to access the sensitivity of the results to this parameter. We find that the posterior distribution of IMF slopes is uniform when allowed to range between 2.0 and 2.5, representing some of the extremal values found from observations of young stellar clusters. This has no effect on the mass posteriors and, ultimately, this approach is not sensitive to reasonable variation of the IMF slope.

A number of studies have sought to constrain the mass limits of Type II SN progenitors. Smartt (2009), and later Smartt (2015), combined about 20 archival pre-explosion images of SN progenitors with stellar evolutionary modeling finding (in the latter work) maximum and minimum masses of $9.5^{+0.15}_{-1.0}$ and $16.5^{+1.0}_{-0.5}$, respectively. This result and the apparent lack of high mass SN progenitors became the "Red Supergiant Problem". More recently, a number of works have found larger upper mass limits using both different methods and a more thorough accounting of observational and statistical uncertainties. Davies & Beasor (2018), accounting for observational and sample

Table 3.2 Summary of IMF parameters and their reported uncertainties found in the literature.

Source	$M_{\min}$	$M_{\max}$
Smartt (2015)	$9.5^{+0.15}_{-1.0}$	$16.5^{+1.0}_{-0.5}$
Davies & Beasor (2018)	$8.7^{+0.6}_{-0.4}$	$24.0^{+3.3}_{-1.9}$
Davies & Beasor (2020)	—	$19.0^{+5.8}_{-3.6}$
Morozova et al. (2018)	$10.4^{+0.8}_{-1.0}$	$22.9^{+3.6}_{-1.9}$
Martinez et al. (2022)	$9.3^{+0.1}_{-0.1}$	$21.3^{+3.8}_{-0.4}$
This Work	$9.8^{+0.37}_{-0.27}$	$24.0^{+3.9}_{-1.9}$

size effects, found minimum and maximum masses of $8.7^{+0.6}_{-0.4}$ and $24.0^{+3.3}_{-1.9}$. In Davies & Beasor (2020), by studying the luminosity distribution of red supergiants in the Magellanic Clouds, find an upper mass limit of about $19^{+5.8}_{-3.6}$. By fitting observed light curves to parameterized light curves simulations, Morozova et al. (2018) found lower and upper limits of $10.4^{+0.8}_{-1.0}$ and $22.9^{+3.6}_{-1.9}$. Using a similar approach with a different observational sample, Martinez et al. (2022) found limits of $9.3^{+0.1}_{-0.1}$ and $21.3^{+3.8}_{-0.4}$, although with a notably steeper IMF slope. Our results, found using a distinct approach, is in good agreement with many of the recent findings. These results are summarized in Table 3.2.

The previous results were obtained by invoking an explosion model – STIR – and using the resulting relationship between iron core mass and plateau luminosity. The second and third steps, along with the relationship to infer the iron core masses from observations, are model dependent, either on the explosion model or the progenitor set. There are other models which may be used for this analysis, namely the works of Sukhbold et al. (2016) and Curtis et al. (2021). However, in the case of Sukhbold et al. (2016), their explosion model produces a large number of light curves far brighter than any in the observational sample used here and thus does not sufficiently describe the diversity of Type II SNe. As we use the same progenitors as Sukhbold et al. (2016), the difference with STIR here comes down to the explosion energies achieved in the two effective models. Sukhbold et al. (2016) achieves more explosion with energies greater than 10^{51} erg s^{-1} and at much lower mass.

On the other hand, Curtis et al. (2021) does not include progenitors less massive than about $11 M_\odot$. For these reasons, the data of B22 is the ideal starting point for this analysis.

The effective explosion model, STIR, is not without its sources of potential uncertainty either. STIR includes a primary tunable parameter that is calibrated to 3D explosions. Changes to the STIR explosion landscape could potentially alter the iron core mass – luminosity relationship and affect the results presented here. However, the explosion energies achieved with STIR are quite insensitive to variations in the mixing length like parameter (see Figures 7 and 8 of Couch et al., 2020). The main effect is that for sufficiently low (high) values, fewer (more) explosions are achieved. The exclusion (inclusion) of these explosions could potentially alter the iron core mass – luminosity relationship and thus the results presented here. However, such extreme values of the free parameter are disfavored by comparisons to 3D simulations and to observed neutron star mass distributions. Given the scale of the effect of the parameters on the STIR explosion energy, any changes to the results here would be smaller than the uncertainties on the iron core masses. For these reasons, we feel that the results presented here are insensitive to reasonable variations of the STIR model.

These results may be further sensitive to the distance metric used in the MCMC algorithm. We tested both Kolmogorov-Smirnov (KS) and the energy distance, finding that, expectedly, the KS test was less sensitive to both the low and high mass progenitors, increasing (decreasing) the minimum (maximum) mass by less than $1M_\odot$. In our tests the energy distance had a tendancy to select models that, by visual inspection, were clearly a poor fit and determined it unsuited to this problem. We conclude that the results here are not sensitive to the distance metric used, so long as a reasonable measure is sought.

3.4 Summary and Conclusions

We use the iron core mass – plateau luminosity relationship of B22 to constrain the core structures of Type II-P SNe progenitors.

The data from this work are publicly available[5].

For the first time photometry alone may give us insight into the cores of populations of CCSNe. Using this relationship alongside the observed sample of Type II CCSNe of G17, we produce a

---

[5]https://doi.org/10.5281/zenodo.7430154

distribution of inferred iron core masses for 82 observed Type II-P CCSNe. This distribution is in remarkable agreement with the IMF weighted distribution of iron core masses produced in the Sukhbold et al. (2016) progenitor set. Of note are the large fraction of events corresponding to iron cores produced by very massive stars larger than about $18M_\odot$.

By sampling massive stellar progenitor populations from a given IMF and connecting them to their resulting iron core mass distributions, we use this sample of inferred iron core masses with an MCMC analysis to infer the posteriors on the minimum and maximum ZAMS masses of progenitors of the G17 sample. We find $M_{\min} = 9.8^{+0.37}_{-0.27}$ and $M_{\max} = 24.0^{+3.9}_{-1.9}$. These results are in decent agreement with other works using both differing methods and observational samples (e.g., Morozova et al., 2018; Davies & Beasor, 2018, 2020; Martinez et al., 2022). The results presented here are not intended as evidence of an upper mass threshold for Type II-P SNe, but instead that, within this observational sample, there is evidence for massive progenitors. Determining a true upper mass limit will rely on advances in stellar evolutionary modeling, core-collapse supernova theory, and a wealth of observational data all used in tandem.

The results here rely on 1D progenitors and explosion models. Ultimately, reality is three dimensional and the results of B22 will need to be tested against suites of 3D simulations carried through their light curves. Variations in the results of B22 to higher dimensionality will potentially yield differences in the results here. Further 3D simulations may also help to tune the turbulent and convective parameters in the STIR explosion model.

This work contributes to the growing amount of evidence disfavoring the existence of the red supergiant problem. By using a novel approach, we have found evidence of very massive progenitors in this observational sample. It is becoming clear that the landscape of CCSN progenitors is complex and very likely contains contributions from very massive progenitors.

This work is the first of its kind to combine high fidelity neutrino-driven CCSN simulations – followed through their light curves – with a statistically significant sample of SNe II-P observations to infer core properties. Effective 1D core-collapse supernova models provide a means of studying not only single events, but entire populations of supernova progenitors. Future work should

explore the dependence of these results and others on the chosen effective model – ultimately a large, collaborative effort. Understanding the core-collapse explosion mechanism and inferring properties of both single observations and populations will require a union of these effective models, multidimensional modeling, and ever growing observational data.

CHAPTER 4

ON CORE-COLLAPSE SUPERNOVA LIGHT CURVE DEGENERACIES

We mortals who dwell in pain
and sorrow might, with reason,
envy the birds of heaven which
know not either!

Fyodor Dostoyevsky, *Poor*

Folks

ABSTRACT

Connecting observations of core-collapse supernovae (CCSNe) to properties of the underlying progenitor and explosion remains an ongoing challenge. The explosion dynamics are determined by a rich, and computationally expensive, set of physics that depends sensitively on the progenitor properties. Producing electromagnetic observables from high fidelity simulations is a challenging endeavor, and typically simplified "thermal bomb" calculations are used to artificially drive the explosion and produce synthetic light curves. However, these synthetic light curves from simplified explosion models have recently been shown to be highly degenerate, with many combinations of progenitor mass and explosion energy producing identical light curves, making progenitor and explosion property inference very difficult at best. In this work, we explore the severity of this degeneracy. By constructing a large, dense grid of explosion models from a wide range of masses and explosion energies, we show that fully constraining explosion and progenitor properties from parametric explosion models alone is not possible. Even when radius information is included, the progenitor's mass remains poorly constrained.

4.1 Introduction

Massive stars with zero age main sequence (ZAMS) masses greater than about $8M_\odot$ may end their lives with a tumultuous explosion. Having spent their lives evolving successive cycles of nuclear burning, they eventually develop an electron degenerate iron core. This core, nearing the effective Chandrasekhar mass (Baron & Cooperstein, 1990), inevitably collapses. What ensues is a consequence of this degeneracy in concert with a rich amount of physics yielding one of Nature's premier multiphysics problems. The collapse proceeds through supranuclear densities where the strong nuclear force becomes repulsive and the core rebounds. An outward forming shock is formed that rapidly dissociates iron group nuclei in the core. The shock is rapidly enervated and inevitably stalls, transitioning to a standing accretion shock.

Some fraction of these progenitors will successfully revive the stalled accretion shock and produce a core-collapse supernova (CCSN) while others will fail and produce a stellar mass black hole and no supernova transient (e.g., O'Connor & Ott, 2011; Lovegrove & Woosley, 2013; Pejcha

& Thompson, 2015; Ertl et al., 2016; Sukhbold et al., 2016; Adams et al., 2017; Sukhbold et al., 2018; Ebinger et al., 2019; Couch et al., 2020). The details of this revival depend sensitively on the structure of the progenitor star (and thus to the treatment of stellar evolution) as well uncertainties in fundamental physics through neutrino-matter cross section and the neutron star equation of state. For in-depth reviews of the CCSN explosion mechanism and the panoply of physics involved, we refer the reader, e.g., Mezzacappa (2001, 2005); Janka et al. (2012); Burrows (2013); Hix et al. (2014); Janka et al. (2016); Müller (2016); Couch (2017); Pejcha (2020); Müller (2020); Mezzacappa et al. (2020); Burrows & Vartanyan (2021); Mezzacappa (2023).

While our understanding of the explosion mechanism is driven primarily by the increasing fidelity of computational models, observational studies provide ever more detailed pictures of the emission of CCSNe. These observations, almost always of electromagnetic emission (see SN 1987A for the singular case of observed neutrino emission (Arnett et al., 1989)), provide insights into the dynamics and composition of the explosion (Hummer & Rybicki, 1968; Falk & Arnett, 1973, 1977; Mihalas & Mihalas, 1984; Litvinova & Nadezhin, 1985; Popov, 1993; Kasen & Woosley, 2009). These observations, however, require a theoretical framework to facilitate interpretation and infer progenitor or explosion properties. The simplest approach for inferring explosion and progenitor properties is the use of phenomenological models. These are generally some form of power-law type scaling relation owing to Litvinova & Nadezhin (1985); Popov (1993). While easy to use, these relations rely on a vast number of assumptions that are difficult to validate in realistic environments. A more sophisticated approach involves modeling simplified radiation hydrodynamics. Gray, spherically symmetric radiation hydrodynamics, with an appropriate stellar equation of state, treatment for radioactive heating from ^{56}Ni decay, and artificial means for initiating an explosion (a so-called "thermal bomb"), provides the ability to create synthetic light curves. Creating a grid of these synthetic light curves from a set of progenitor models with different explosion energies allows for the fitting of observed light curves[1] (see, e.g., Falk & Arnett, 1973;

---

[1]In reality, there are more parameters than just progenitor mass and explosion energy. Metallicity, stellar mass loss scheme, mass of synthesized ^{56}Ni, its distribution through the star, compositional mixing, circumstellar material mass and distribution, to name a few, all complicate the fitting process.

Litvinova & Nadezhin, 1985; Bersten et al., 2011; Morozova et al., 2015, 2018; Ricks & Dwarkadas, 2019; Martinez & Bersten, 2019; Goldberg et al., 2019; Goldberg & Bildsten, 2020; Martinez et al., 2020; Martinez et al., 2022, and references therein).

While these radiation hydrodynamics models (often referred to as hydrodynamical models in the literature) have been extensively to study observed CCSNe, it has recently been demonstrated that this approach is not without flaw. Goldberg et al. (2019); Dessart & Hillier (2019) recently showed that fitting light curves with artificially-driven radiation hydrodynamic models incurs a massive degeneracy: many combinations of progenitor mass and explosion energy produce an identical light curve. Indeed, without more information, progenitor and explosion properties cannot be uniquely inferred from observations. The breaking of this degeneracy remains an open problem. In followup work, **?** demonstrated that progenitor radius estimates, via pre-explosion imaging or shock cooling models, could potentially reduce the degeneracy. Even in the absence of uncertainties on the radius inference, however, the degeneracy could not be completely lifted, with a spread of roughly $10M_\odot$ in ZAMS mass for best fit models for SN 2017eaw. Taking a different approach, Barker et al. (2022); Barker et al. (2023) showed that light curves obtained from neutrino-driven explosion models, instead of thermal bomb models, could be used to constrain observations of both individual SNe and populations. By using a physically motivated, neutrino-driven explosion model, the parameter space is vastly reduced. Under this framework, a given progenitor explodes in a single deterministic way, removing the explosion energy as a free parameter.

The full severity of these degeneracies has yet to be explored. We build upon the pioneering works of Goldberg et al. (2019); Dessart & Hillier (2019). Constructing a large, dense grid of artificial explosion models, we show the full degenerate parameter landscape for a few select observed supernovae. We focus on type IIP SNe, which make up the lion's share of observed CCSNe (Li et al., 2011). The results presented here cement what has been shown before: for artificially-driven light curves, matching an observation is a *necessary* but *not sufficient* condition for inferring explosion and progenitor properties.

In Section 7.3 we lay out the methods used in this study, highlighting the construction of the

model grid, the progenitor models used, and the radiation hydrodynamic calculations used for the light curves. In Section 5.5 we lay out the results of our analysis. We conclude in Section 7.5 by reviewing our findings and next steps.

4.2 Methods

Here we lay out all of the necessary details to interpret and recreate the study. We begin with describing the construction of the model grid used. Care is then given to the details of the stellar progenitor set used in the model grid. We briefly describe the setting used with the SuperNova Explosion Code, SNEC, used for the radiation hydrodynamic modeling. We end with a discussion of light curve fitting methods.

4.2.1 Grid

Studies involving fitting observations to synthetic light curves rely on grid of parameterized models. Given the known degeneracies with this approach (Goldberg et al., 2019; Dessart & Hillier, 2019), the results of such a fitting procedure are at least as sensitive to the model grid as they are to the underlying physics. An overly sparse grid may completely miss potential solutions. As such, we construct a dense grid in progenitor mass and explosion energy

We include solar metallicity progenitors with ZAMS masses $M_{ZAMS} \in [9.0 M_\odot, 24.0 M_\odot]$. This mass range is chosen as, in single star stellar evolution, they will produce red supergiants and type II SNe. These progenitor models have a cadence

$$\Delta M = \begin{cases} 0.25 M_\odot & \text{for } M_{ZAMS} < 13.0 M_\odot \\ 0.1 M_\odot & \text{for } M_{ZAMS} > 13.0 M_\odot \end{cases} \tag{4.1}$$

resulting in 127 progenitors spanning 9 to 24 solar masses. We use an energy grid spanning $E \in$ [0.3 foe, 2.0 foe] (foe = 10^{51} erg) with $\Delta E = 0.2$ foe for a total of ten values of explosion energy per progenitor. At present we fix the mass of radioactive ^{56}Ni to be $0.07 M_\odot$. We define then the

model set

$$\mathcal{G} = \{M \mid M \in [9, 24] \text{ with } \Delta M,$$

$$E \mid E \in [0.2, 2.0] \text{ with } \Delta E, \qquad (4.2)$$

$$M_{\text{Ni}} = 0.04\}$$

This makes a total of 1270 light curve models. With a sufficiently dense grid in mass and energy, we seek to construct the degeneracy landscape for type II SNe.

4.2.2 Progenitor Models

For the progenitor grid described previously, we use the stellar evolutionary models of Sukhbold et al. (2016). These models, evolved with KEPLER (Weaver et al., 1978), are solar metallicity, non-rotating, non-magnetized progenitors evolved to core-collapse. They are single star stellar evolutionary models. We include all of the progenitors of Sukhbold et al. (2016) up to, and including, $24 M_\odot$. These progenitors are a suitable choice for this study, as they span a range of parameters relevant for type II SNe. They have a wide range of core and envelope properties that power a range of transient properties. Figure 4.2.2 shows envelope mass (top), presupernova mass (middle), and presupernova radius (bottom) for the progenitor set.

4.2.3 Synthetic Light Curve Calculations

We construct synthetic light curves using the SuperNova Explosion Code [2] (SNEC, Morozova et al., 2015). SNEC is a spherically symmetric photon radiation hydrodynamics code for modeling supernovae with artificially-driven explosions. SNEC treats Newtonian hydrodynamics in a co-moving, Lagrangian fashion using a finite difference scheme with artificial viscosity (**?**). Radiation transport is treated via gray flux-limited diffusion. As SNEC does not include the physics responsible for driving CCSNe, the explosion must be artificially-driven. SNEC uses a thermal bomb explosion method, where a user-defined amount of energy is distributed through the innermost part of the star. The stellar equation of state is that of Paczynski (1983) which includes contributions from ions, radiation, and electron with approximate degeneracy. This is coupled with a Saha ionization solver that calculates ionization fractions. In the present work we follow ionization of ^{1}H, ^{3}He, and

---

[2]https://stellarcollapse.org/SNEC.html
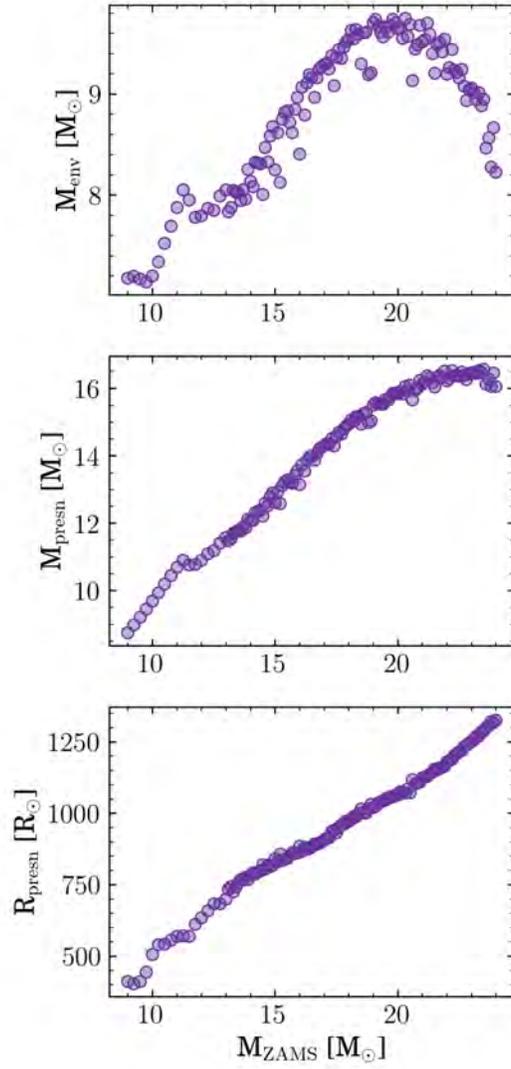
Figure 4.1 Progenitor properties of the Sukhbold et al. (2016) set used in this work, including envelope mass (top), presupernova mass (middle), and presupernova radius (bottom).

[4]He. At high temperatures, SNEC uses OPAL type II opacities (Iglesias & Rogers, 1996) which are supplemented at low temperatures by the opacities of Ferguson et al. (2005). For more details of the implementation, see Morozova et al. (2015).

Supernova light curves are determined not only by the energetics of the explosion, but by the distribution of the matter. Mixing at sharp compositional interfaces, due to Rayleigh-Taylor and Richtmyer-Meshkov instabilities, smooths out compositional boundaries and mixes lighter elements inwards and heavier elements outwards. Mixing of radioactive ^{56}Ni plays a particularly important role, as it can impact plateau brightness and duration (Kasen & Woosley, 2009; Kozyreva et al., 2019). SNEC mimics hydrodynamical mixing by applying a "boxcar" smoothing algorithm. We use the fiducial boxcar smoothing values of Morozova et al. (2015), which uses a averaging width of $\Delta M = 0.4 M_\odot$.

The late, post-plateau phase emission is dominated by heating due to the radioactive decay chain ^{56}Ni $\rightarrow$ ^{56}Co $\rightarrow$ ^{56}Fe which produces gamma emission and a small positron component. This ^{56}Ni is produced during the explosion after the initial shock revival and is mixed outwards by hydrodynamic effects. Diffusion of gamma rays and positrons from these decays provide a long term source of energy. SNEC follows the heating due to ^{56}Ni decay by a gray treatment of Swartz et al. (1995). Positron contributions are small and neglected. SNEC does not include a nuclear reaction network, and nuclear burning from 1D, artificial explosions is unreliable, so ^{56}Ni must be manually added to the stellar profile. SNEC allows for the injection of a specified amount of ^{56}Ni to be distributed over a specified amount of mass. We set the mass of synthesized ^{56}Ni as described in Section 4.2.1. The distribution of this material is, in principle, a free parameter. In order to keep the size of our model grid from growing too large, we must standardize the ^{56}Ni distribution. We mix the ^{56}Ni outwards in the stellar profile just slightly into the envelope.

SNEC -like simulations require removing the innermost part of the stellar profile – corresponding to what might become the compact object – as a "mass cut." This is necessary as this high density, gravitationally bound material would dominate the hydrodynamical timestep and greatly increase computational cost. As SNEC cannot self-consistently determine the mass of the formed compact

object, this is done manually. At present, we take a constant $1.8M_\odot$ mass cut for all of the model grid. While likely larger than the neutron star that would be produced for the lowest mass models in our grid (see, e.g., Sukhbold et al., 2016; Ebinger et al., 2019; Couch et al., 2020), it remains a reasonable choice in absence for more sophisticated calculations.

Finally, SNEC requires injecting the energy to drive the explosion by hand. This is done by adding an amount of heat over the innermost part of the ejecta nearly instantaneously (see Morozova et al., 2015, for implementation details). When injecting the explosion energies as laid out in Section 4.2.1, we inject sufficient energy such that we achieve the desired *asymptotic* explosion energy. That is, the injected energy is adjusted for the binding energy of the progenitor.

4.2.4 Light Curve Fitting

Fitting light curves requires a choice of what is fit and through what error metric. In principle, the choice of error metric can have an impact on the results (Barker et al., 2022). Here, we adopt the L_2 error metric for quantity q

$$L_2(q) = \frac{1}{N} \sum_{t^*=t_1}^{t_N} (q(t^*) - q^*(t^*))^2 \tag{4.3}$$

where a * denotes observation, i.e., $q^*(t^*)$ is an observation of q at time t^* and $q(t^*)$ is the synthetic quantity at the same time. For comparison, synthetic quantities are interpolated to the same time as observed data. In practice the quantities used are normalized.

We explore minimization of the error in the light curve as well as a combined error metric including both bolometric luminosity and ejecta velocity, i.e.,

$$\varepsilon = 0.5 \, L_2(L_{\rm bol}) + 0.5 \, L_2(v_{\rm ejecta}). \tag{4.4}$$

Simultaneously fitting both luminosity and ejecta velocity can, in principle, provide a more constrained fit as it includes more information than luminosity alone (e.g., Ricks & Dwarkadas, 2019; Goldberg et al., 2019; Goldberg & Bildsten, 2020; Martinez & Bersten, 2019; Barker et al., 2022; Martinez et al., 2020).
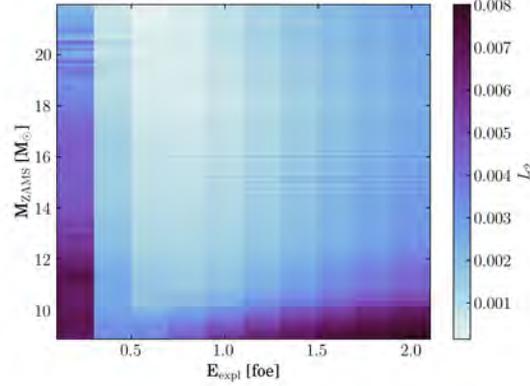
Figure 4.2 L_2 norm for SN2017eaw as a function of ZAMS mass and explosion energy. Here the L_2 is computed using only the bolometric luminosity.

4.3 Results

We begin by exploring the degeneracy landscape for SN2017eaw. Here we find the best fit model from our grid by minimizing error in the light curve, $L_2(L_{\text{bol}})$. Figure 4.3 shows the L_2 norm computed this way as a function of ZAMS mass and explosion energy. The best fit model is $(\hat{M}_{\text{ZAMS}}, \hat{E}_{\text{expl}}) = (21.9 M_\odot, 0.6 \text{ foe})$. However, it is immediately clear that the observation is not well constrained. Even at fixed explosion energy, the spread in ZAMS mass is large.

The spread in ZAMS mass in Figure 4.3 merits quantification. We define a deviation $\sigma(E)$ that is the 16% percentile of the L_2 norm taken at a given explosion energy. That is, we find the vertical deviation at a given energy in Figure 4.3. We define the set of models with norms within 1σ as

$$\mathcal{M}_\sigma = \{M \mid |\hat{L}_2 - L_2(M, \hat{E}, M_{\text{Ni}})| \leq \sigma\} \tag{4.5}$$

where a hat denotes the best fit values, i.e., $\hat{L}_2$ is the minimum norm. For the case of SN2017eaw, we find that $\mathcal{M}_\sigma = \{M \mid M \in [15.9, 24.0]\}$ – an $8 M_\odot$ spread in $|\mathcal{M}_\sigma|$.

Figure 4.3 shows observations of SN2017eaw (green squares) along with the best fit model (purple) and $\mathcal{M}_\sigma$ models (gold). The left panel shows the bolometric light curve and the right panel shows ejecta velocity. For the observed velocities, we present the common Fe II $\lambda 5169$ line velocities. For the models, on the other hand, we show $\tau = 2/3$ photospheric velocities.

We observe a few features of note. The best fit model (purple) fits the observations very well, with exception of the radioactive tail, which is simply due to there being too few values of nickel
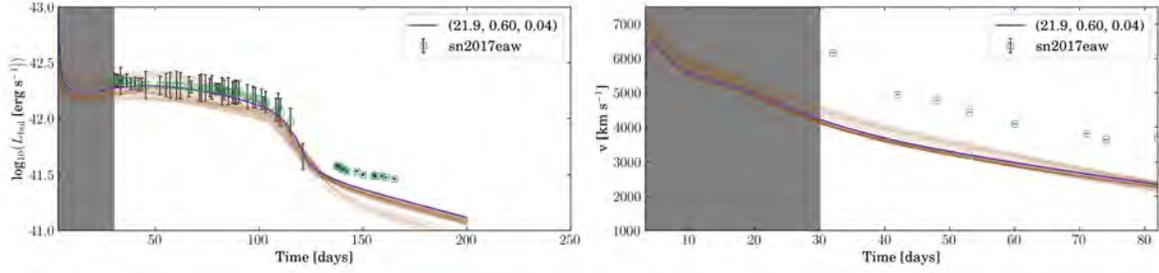
Figure 4.3 *Left*: Observational light curve data for SN2017eaw (green squares), best fit synthetic light curve model (purple), and models within $1 - \sigma$ ($M \in \mathcal{M}_\sigma$) (gold). *Right*: Observational Fe II $\lambda5169$ for SN2017eaw (green squares), best fit model (purple), and $1 - \sigma$ models (gold). Here the L_2 is computed using only the bolometric luminosity. The 1σ models span 15.9 to 21.9M$_\odot$ and have fixed explosion energy.

mass in the model grid $\mathcal{G}$. Many of the $\mathcal{M}_\sigma$ models are still very close, and those which fit visibly more poorly are still within, or close to, observational uncertainty. The photospheric velocities, on the other hand, are a poor match. This is due, perhaps, to different definitions of the velocities. The observations correspond to Fe II $\lambda5169$ line velocity, while the model velocities are $\tau = 2/3$ photospheric velocities, that is, not an observable quantity. While spectral line velocities are not immediately available from a gray radiation hydrodynamics code like SNEC, it can be approximately under the Sobolev approximation (Mihalas & Mihalas, 1984) and tends to give better agreement with observation (see, e.g., Goldberg et al., 2019; Barker et al., 2022, and references therein).

To further highlight issues with using the photospheric velocity, we seek to minimize the combined L_2 norm of both bolometric luminosity and ejecta velocity. The results are shown in Figure 4.3. Here we have a much better fit for the velocity evolution, but the light curve fit is very poor. We notice that a much larger explosion energy is required to achieve the necessary ejecta velocities. This necessitates adoption of a better synthetic velocity quantity.

4.3.1 Radius Measurements

It has been demonstrated that radius inferences might help to constrain progenitor and explosion property inferences (Goldberg & Bildsten, 2020). These stellar radius inferences typically arise from pre-explosion imaging measuring the stellar luminosity and temperature. For SN2017eaw such measurements exist with $\log(L/L_\odot) = 4.9 \pm 0.2$ and $T_{\text{eff}} = 3350^{+450}_{-250}$K (Kilpatrick & Foley,
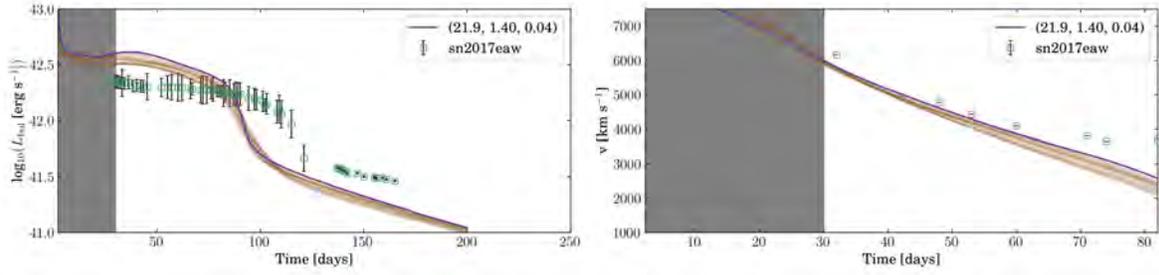
Figure 4.4 The same as Figure 4.3 except we fit both bolometric luminosity and ejecta velocity.

2018). As the propagation of asymmetric uncertainties, without knowledge of the underlying probability distribution, is not possible, we make the conservative simplifying assumption to use the larger of the two uncertainties on the stellar temperature. This will likely overestimate the propagated uncertainty on the radius. With this in mind, these values correspond to a radius inference of $892.4 R_\odot \pm 271.9 R_\odot$. For comparison, using the lower value results in a radius of $853.3 R_\odot \pm 132.7 R_\odot$. We use a Monte Carlo error propagation procedure to avoid assumptions of normality, linearity, or small errors common in traditional error propagation methods.

The impact of including a radius inference on model selection is shown in Figure 4.3.1. Presupernova radius as a function of ZAMS mass is shown with markers. The radius inference is shown with the horizontal green band with the optimal value denoted by a dashed line. The $\mathcal{M}_\sigma$ mass set is denoted with the vertical gray shaded region. The best fit model is shown with the gray dashed line. The intersection of the radius inference and $\mathcal{M}_\sigma$ regions is shaded blue, representing the additional constraints from the radius inference. There are a few features of note: first, the models closer to the optimal radius value are technically within $\mathcal{M}_\sigma$, but far from the best fit model. Additionally, the best fit model is barely within the radius uncertainties (indeed, using the lower bound radius error excludes the best fit model). Also, The radius band includes a large amount of low mass models that do not fit the observations. Radius measurements, as expected, help in constraining the model selection but should be used carefully and with their associated uncertainties.
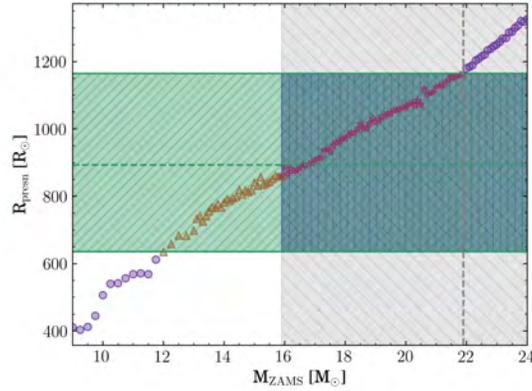
Figure 4.5 Presupernova radius (markers) as a function of ZAMS mass with several regions of interest noted. The horizontal green band (forward slash hatching) denotes the 1σ uncertainty band on the radius of SN2017eaw, with the optimal value denotes by a green dashed line. The gray vertical band (backwards slash hatching) denotes the $\mathcal{M}_\sigma$ set of masses from light curve fitting with the best fit mass denoted by dashed gray line. The intersection of these two regions is shaded blue. Data that fit into neither band are denoted with purple circles, data that falls into the radius inference band, but not the $\mathcal{M}_\sigma$ band, as gold triangles, and data in the intersection are denoted as pink stars.

4.4 Discussion and Conclusions

We construct a dense grid of artificial explosion models with corresponding synthetic light curves. These models span 9 to 24 solar masses with 127 models in that range. Each model is injected with a thermal bomb to each asymptotic explosion energies in the range 0.2 to 2.0 foe, with a cadence of 0.2 foe. This grid finely samples the range of values that might produce a type II CCSN. With this model grid we compare to SN 2017eaw in order to construct the degeneracy landscape.

We fit SN2017eaw to our model grid, finding a best fit model with $(M_{ZAMS}, E) = (21.9 M_\odot, 0.6$ foe). There is, however, large spread in the error metric, particularly as a function of ZAMS mass. Defining an uncertainty σ, we find models spanning $8 M_\odot$ within 1σ of the best fit, which reaches the upper bound of the mass grid. While acceptable agreement is found between bolometric light curves, very poor agreement is found between ejecta velocities. This highlights the difference between the photospheric velocity often used in gray radiation hydrodynamics calculations and the line velocities observed. Using a fitting procedure that seeks to fit both the light curve and ejecta

velocity finds highly energetic models that, while matching the velocity evolution well, match the light curve very poorly. This cements the need for more realistic model velocities when comparing to observations.

We explore the impact of radius measurements on model constraint. We find, as expected, that the inclusion of radius information can help to constrain the set of possible masses. However, we find that the models close to the optimal radius measurement, while still a decent fit to the observations, are quite far from the best fit explosion models. Additionally, if overly liberal assumptions about the radius uncertainty are taken, then the best fit model's stellar radius might not lie within the uncertainty on the inferred stellar radius.

This work is a step towards providing better constrains on observed supernova progenitor and explosion properties. While parametric explosions models remain unable to tightly constrain observations, the effort here helps pose the problem as a statistical, reproducible one.

CHAPTER 5

THORNADO-HYDRO: A DISCONTINUOUS GALERKIN METHOD FOR SUPERNOVA HYDRODYNAMICS WITH NUCLEAR EQUATIONS OF STATE

> To Whom the unceasing suns belong,
>
> And cause is one with consequence,
>
> To Whose divine inclusive sense
>
> The moan is blended with the song.

Ambrose Bierce, *Invocation*

This chapter is based on the published work of D. Pochik, B. L. Barker, *et al.* 2021 ApJS **253** 21.

5.1 Abstract

This paper describes algorithms for non-relativistic hydrodynamics in the **t**oolkit for **h**igh-**or**der **n**eutrino **rad**iation hydr**o**dynamics (THORNADO), which is being developed for multiphysics simulations of core-collapse supernovae (CCSNe) and related problems with Runge–Kutta discontinuous Galerkin (RKDG) methods. More specifically, THORNADO employs a spectral type nodal collocation approximation, and we have extended limiters — a slope limiter to prevent non-physical oscillations and a bound-enforcing limiter to prevent non-physical states — from the standard RKDG framework to be able to accommodate a tabulated nuclear equation of state (EoS). To demonstrate the efficacy of the algorithms with a nuclear EoS, we first present numerical results from basic test problems in idealized settings in one and two spatial dimensions, employing Cartesian, spherical-polar, and cylindrical coordinates. Then, we apply the RKDG method to the problem of adiabatic collapse, shock formation, and shock propagation in spherical symmetry, initiated with a 15 $M_\odot$ progenitor. We find that the extended limiters improve the fidelity and robustness of the RKDG method in

idealized settings. The bound-enforcing limiter improves robustness of the RKDG method in the adiabatic collapse application, while we find that slope limiting in characteristic fields is vulnerable to structures in the EoS — more specifically, in the phase transition from nuclei and nucleons to bulk nuclear matter. The success of these applications marks an important step toward applying RKDG methods to more realistic CCSN simulations with THORNADO in the future.

5.2 Introduction

Stars with zero-age main sequence (ZAMS) masses $M_{\mathrm{ZAMS}} \gtrsim 8M_\odot$ end their lives as spectacular explosions known as core-collapse supernovae (CCSNe). These explosions are at the heart of some of the most important questions in astrophysics. They are the primary catalysts of galactic chemical evolution, producing and dispersing many of the elements heavier than hydrogen and helium, and provide feedback into the interstellar medium. They may even be a source of the lighter first peak r-process elements (Martínez-Pinedo et al., 2014), though neutron star mergers are likely the primary production site for the r-process (Kasen et al., 2017). Their cores are the foundries for compact objects including those recently detected by Advanced LIGO and Virgo (Abbott et al., 2016, 2017a,b, 2020). Through their observables and the compact objects left behind, we may even begin to probe the nature of nuclear matter (Schneider et al., 2019).

Throughout their lives, these massive stars undergo successive cycles of nuclear fusion, forging heavier elements in their cores. At the end of a star's lifetime, fusion processes build up a degenerate iron core that is unable to undergo nuclear fusion itself. This iron core, supported thus far by electron degeneracy pressure, grows to the effective Chandrasekhar mass (Baron & Cooperstein, 1990) and, no longer able to balance gravity, subsequently collapses. During collapse, runaway electron capture processes accelerate the collapse and produce vast numbers of neutrinos, while photodissociation of iron group nuclei robs the core of more energy. Eventually the core reaches nuclear density and the nuclear strong force becomes repulsive, effectively stiffening the Equation of State (EoS) tremendously, and collapse is halted in the inner core. The collapse rebounds and produces a strong shock that is driven through the outer core. Ultimately, through a combination of neutrino cooling and dissociation of iron group nuclei, the shock runs out of energy and stalls before

escaping the core, becoming an accretion shock. Meanwhile, the inner core regains equilibrium in the form of a newborn proto-neutron star (PNS).

Providing a mechanism to revive the stalled shock and drive the explosion is among the forefront questions in the study of CCSNe. Of the proposed mechanisms, the most favored has been the delayed neutrino-driven mechanism (Bethe & Wilson, 1985). Neutrinos emitted from the surface of the cooling PNS, aided by hydrodynamic and magnetohydrodynamic instabilities, deposit energy below the stalled shock and reinvigorate the explosion. Of the other proposed mechanisms, the magneto-rotational mechanism – wherein a rapidly rotating PNS supplies energy to power the shock (Akiyama et al., 2003) – has potential, but likely doesn't account for most CCSNe. A key characteristic of magneto-rotationally driven SNe is the formation of collimated jets, which are not seen in the vast majority of supernova remnants (e.g., see Soderberg et al., 2010). Additionally, for this mechanism to be effective the stellar core must be very rapidly rotating, beyond the rotation rates commonly achieved through stellar evolution (Heger et al., 2005). Ultimately, any successful mechanism must not only revive the shock but also explain the observations of supernovae (e.g., light curves and spectra).

For several decades this was the state of the field. These mechanisms saw little success until relatively recently: spherically symmetric (spatially one-dimensional [1D]) simulations of CCSNe consistently failed to produce explosions. It wasn't until computing resources allowed for axisymmetric (spatially two-dimensional [2D]), and eventually full-physics three-dimensional (3D), simulations that successful explosions could be consistently produced without modified or parametrized physics. Ultimately, the reason for this is 1D fails to capture the fundamentally non-spherical nature of CCSNe and hydrodynamic instabilities are unable to develop. The CCSN explosion mechanism has been the subject of decades of work and still remains incompletely described (for in-depth reviews, see, e.g., Bethe, 1990; Mezzacappa, 2001, 2005; Janka et al., 2012, 2016; Burrows, 2013; Hix et al., 2014; Müller et al., 2016; Couch, 2017).

Hydrodynamics, along with gravity and neutrino transport, plays a key role in the dynamics of CCSNe. This starts with the progenitors, which in nature are multi-dimensional and likely

involve a complicated mixing of elements in the convectively burning shells (see, e.g., Arnett & Meakin (2011)). Further, it has been shown that asphericities in progenitors can mean the difference between a model that explodes, and a model that doesn't (Couch & Ott, 2013). However, regardless of the progenitor, after the core rebounds it is known that the shocked fluid develops instabilities.

Once the bounce-shock stalls and the neutrino hearing (or gain) region is established below the shock, at least two hydrodynamical instabilities may contribute to the evolution of the shock: neutrino-driven convection (Herant et al., 1992) and the Standing Accretion Shock Instability (SASI; Blondin et al. (2003)). Both of these instabilities create turbulence in the post-shock flow, and that turbulence contributes ram pressure that enlarges the extent of the gain region (Murphy et al., 2013), thus increasing the efficacy of neutrino heating, thus aiding the explosion (see Couch & Ott (2015), and references therein). Which effect is more dynamically important, however, may depend on the progenitor mass (Müller et al., 2012; Hanke et al., 2013; Summa et al., 2016; Vartanyan et al., 2019). Regardless of which effect is dominant, simulations should be able to satisfactorily quantify the turbulence, and in particular should be able to capture the turbulent energy cascade from the energy carrying scale through the inertial scale, down to the (numerical) dissipation scale. However, a consensus has not yet been reached as to what, in terms of angular resolution, is required to adequately capture the turbulent energy cascade. In particular, Radice et al. (2015), Abdikamalov et al. (2015), and Casanova et al. (2020) suggest that resolutions much lower than $1°$ may be necessary (due to the numerical dissipation of the scheme, which creates a "bottleneck" for energy transfer at a scale set by the scheme), but recently Melson et al. (2020) argued that $1°$ resolution is sufficient to obtain a clear distinction between the inertial and dissipation scales. Additionally, Endeve et al. (2012) showed that turbulence from the SASI can amplify magnetic fields, and more recently, Müller & Varma (2020) found that turbulently amplified magnetic fields can aid neutrino-driven explosions, even in slowly-rotating progenitors. See Radice et al. (2018) for a recent review of turbulence in CCSNe.

In addition to the hydrodynamic instabilities occurring in the shocked mantle, the PNS undergoes convection and potentially other instabilities due to entropy and electron fraction gradients (Bruenn

et al., 2004), which has an effect on the luminosity of heavy flavored neutrinos as well as the mean energies of all neutrino flavors (Buras et al., 2006). This may not directly affect the shock dynamics, but it does give rise to the recently discovered Lepton Number Emission Self-Sustained Asymmetry (LESA; Tamborra et al. (2014)), which may hold implications for the composition of the ejecta. For more detailed discussions on the role of hydrodynamic instabilities in CCSNe, we refer to the recent review by Müller (2020).

Insight into hydrodynamic phenomena can often be gained by treating the fluid as polytropic (in the CCSN context, see, e.g., Yahil, 1983; Blondin et al., 2003); i.e. the fluid pressure p is assumed to be proportional to a power law of the mass density ρ, which gives rise to the polytropic EoS, $p \propto \rho^{\Gamma}$, where $\Gamma = \left(\frac{\partial \ln p}{\partial \ln \rho}\right)$ is the adiabatic index.[1] However, relating the state variables by this expression neglects the nuclear interactions and compositions in stellar collapse; e.g. the polytropic EoS fails to capture the response in pressure due to the thermal or compositional changes that are typical in a stellar environment. For the conditions prevalent in stellar interiors, particularly in the high-density regimes of stellar collapse, a simple analytic form for the EoS likely does not exist. Instead, an EoS for this case is often created by minimizing a thermodynamic potential — e.g. the Helmholtz free energy — for a system of particles under stellar conditions (see, e.g., Swesty, 1996; Fryxell et al., 2000; Timmes & Swesty, 2000). Once the free energy is known, other relevant quantities, such as pressure, internal energy, and entropy, can easily be obtained.

The task of developing an equation of state for realistic CCSN simulations has remained a pertinent objective for several decades. Important contributions toward this effort include the Lattimer & Douglas Swesty (1991) (LS) and Shen et al. (1998) (STOS) EoSs. The LS EoS used a compressible liquid-drop model (see, e.g., Lattimer et al., 1985), while STOS used a relativistic mean field (RMF) model with the TM1 parameter set (see, e.g. Sugahara & Toki, 1994). However, due to the importance of including light nuclei in CCSN simulations, a notable drawback for both the LS and STOS EoS was their exclusion of all light nuclei other than alpha particles (Hempel et al., 2012; Steiner et al., 2013b). Further advances include the hadronic EoSs from G. Shen (Shen

---

[1]Contrary to a realistic model, the adiabatic index for a polytropic model remains constant through space and time.

et al., 2011a,b), which build upon the NL3 (Lalazissis et al., 1997) and FSUgold (Todd-Rutel & Piekarewicz, 2005) parameter sets. Additionally, unlike the LS and STOS EoSs, the statistical model of Hempel et al. (2012) (HS) (see also Steiner et al., 2013b) does not use the single-nucleus approximation for heavy nuclei, but includes a more realistic compositional distribution of nuclei.

Moreover, recent neutron star observations (see, e.g., Greif et al., 2020; Steiner et al., 2013a) and observations of other astronomical phenomena (see, e.g, Greif et al., 2020, and references therein), experiments in nuclear physics (see, e.g., Greif et al., 2020), and experiments in relativistic heavy-ion collisions (see, e.g., Oertel et al., 2017, and references therein), have led to the development of multiple EoSs for dense nuclear matter that are applicable to CCSN simulations (see, e.g., Steiner et al., 2013a,b). These equations of state provide thermodynamic quantities as functions of density, temperature, and electron fraction. The SHFo/SFHx EoSs from Steiner et al. (2013a,b) build upon the statistical model used in HS and constrain properties of nucleonic matter with an RMF model (see, e.g., Shen et al., 1998, 2011a,b). The most probable mass-radius relationship derived from neutron star (NS) observations was used to build the "optimal" SFHo EoS, while the "extreme" SFHx EoS is built around a minimized radius model for low-mass NSs (Steiner et al., 2010, 2013a). For our purposes, the importance of these equations of state lies in their ability to resolve various physical regimes in CCSNe, including the phase transition from nuclei and nucleons to bulk nuclear matter at high densities ($\rho \sim 10^{14}$ g cm^{-3}) (Steiner et al., 2013b), and the high-density rebound of the core, which determines the initial strength of the shock (Shen et al., 1998). We note that these EoSs do not include lower density/temperature regimes; i.e., they do not describe matter out of nuclear statistical equilibrium (NSE); but see, e.g., Bruenn et al. (2020) for treatment of non-NSE regions in CCSN models.

Clearly, multidimensional, multiphysics models of CCSNe require advanced simulation tools and massive computational resources, and to that end there are several production codes in existence; e.g., Aenus-Alcar (Just et al., 2015), Castro (Almgren et al., 2010), Chimera (Bruenn et al., 2020), CoCoNuT-Vertex (Müller et al., 2010), FLASH (Fryxell et al., 2000; Dubey et al., 2009; O'Connor & Couch, 2018), Fornax (Skinner et al., 2019), Prometheus-Vertex (Rampp & Janka, 2002), and

Zelmani (Ott et al., 2009; Roberts et al., 2016), and the codes of Sumiyoshi & Yamada (2012); Nagakura et al. (2014), and Kuroda et al. (2016). To solve the equations of hydrodynamics — with the aim of capturing shocks and resolving turbulent flows — these codes use variations of either the finite-difference or the finite-volume high-resolution shock capturing method, in either an Eulerian or semi-Lagrangian framework. In particular, the finite-volume method divides the computational domain into finite cells (or volumes), formulates the hydrodynamics equations in integral form, and solves for physical quantities (e.g., mass density) in terms of cell averages. The cell averages are updated by accounting for (1) fluxes through the surface enclosing each cell and (2) volume sources (e.g., due to gravity). The integral formulation leads naturally to good conservation properties, and allows for discontinuous solutions (e.g., shocks). In computing the surface fluxes, local polynomials are *reconstructed* using cell averages of the local cell and its neighbors. The local polynomials are then used to assign left and right states at each cell interface as inputs to a *Riemann solver*, which provides the numerical flux. To avoid non-physical oscillations around shocks, limiters are applied to the reconstructed polynomial to enforce some degree of monotonicity, which can degrade the formal order of accuracy of the hydrodynamics scheme. (We refer to the above citations for further details on the hydrodynamics algorithms implemented in the specific codes listed.)

As discussed above, turbulence is ubiquitous in the supernova environment and plays a role in the explosion mechanism. It is therefore desirable to maintain good spectral resolution to resolve as much of the turbulent spectrum as possible for a given spatial resolution, and this motivates the use of accurate Riemann solvers and high-order methods. On the other hand, due to their multiphysics nature, CCSN simulations with neutrino transport are computationally expensive, and must run efficiently on distributed memory architectures; e.g., using message passing interface (MPI). Furthermore, because of the high number of degrees of freedom involved in neutrino transport computations (a momentum space is attached to each spatial point), memory limitations require the number of spatial cells assigned to any given MPI process to not be large. For a code to scale well, the number of ghost cells should be limited relative to the number of compute cells to manage the communication overhead, since each MPI process will have a halo region comprised

of ghost cells populated with data from neighboring processes. While finite-difference and finite-volume methods can achieve high-order accuracy, the computational stencil width increases with increasing order of accuracy, thereby increasing the size of the halo region and the ratio of ghost cells to compute cells, thus impeding good scalability (e.g., Miller & Schnetter, 2017).

The discontinuous Galerkin (DG) method (e.g., Cockburn, 2001) is an alternative approach to solving the system of hydrodynamics equations (and many other systems). Similar to finite-volume methods, DG methods divide the computational domain into cells (or elements), and formulate the equations in integral form. However, contrary to finite-volume and finite-difference methods, in the DG method the solution is approximated by a local polynomial within each element, which implies that more local information is tracked in the solution process (i.e., not just the cell average). Because the full polynomial representation in each element is evolved, the reconstruction step needed in the finite-volume approach is not necessary. Meanwhile, Riemann solvers developed in the context of finite-volume methods can readily be used with DG methods to evaluate numerical fluxes on element interfaces. The DG method is a finite-element method, but does not demand continuity of the local polynomial approximation across element boundaries, and consequently, is well suited to capture shocks and other discontinuities. To prevent non-physical oscillations in the vicinity of a discontinuity, limiters are applied to the local polynomial to enforce monotonicity. More recently, so-called structure-preserving discretizations, which maintain fundamental physical properties of the system under consideration (e.g., positive mass density and pressure), have been developed within the DG framework (e.g., Zhang & Shu, 2011). Another advantage offered by the DG method is high-order spatial accuracy on a compact stencil. Only information from nearest neighbors is needed, independent of the order of accuracy. This makes the DG method well-suited for application on massively parallel architectures, since increasing the order of accuracy does not increase the communication overhead as much as other high-order methods (e.g., Miller & Schnetter, 2017). The desired combination of shock-capturing capabilities, high-order accuracy in smooth flows, and good scalability make DG methods an appealing choice. Additionally, DG methods are also amenable to hp-adaptivity (Remacle et al., 2003), wherein refinement of either the

spatial mesh (*h*-refinement) or the local degree of the polynomial approximation (*p*-refinement) can be used to improve the accuracy of the method near shocks while maintaining high-order accuracy in regions of smooth flow. DG methods are also well-suited for problems involving curvilinear coordinates (Teukolsky, 2016).

The DG method was introduced already in the 1970s by Reed & Hill (1973) to solve the steady state neutron transport equation, and the initial framework for solving time-dependent problems with explicit Runge–Kutta time integration (commonly referred to as RKDG methods) was established in a series of papers by Cockburn & Shu (Cockburn & Shu, 1989; Cockburn et al., 1989, 1990; Cockburn & Shu, 1991; Cockburn & Shu, 1998). Today, DG methods are widely used in science and engineering applications, and are rapidly gaining popularity in the computational astrophysics community (see, e.g., Radice & Rezzolla, 2011; Schaal et al., 2015; Teukolsky, 2016; Kidder et al., 2017; Fambri et al., 2018, and references therein), but have so far not been applied to multiphysics CCSN simulations.

The **t**oolkit for **h**igh-**or**der **n**eutrino **rad**iation hydr**o**dynamics[2] (THORNADO) is being developed with the goal of realizing multiphysics simulations of CCSNe and related problems with high-order methods. To this end, the hydrodynamics and neutrino transport algorithms in THORNADO are based on the DG method (see, e.g., Endeve et al., 2019; Chu et al., 2019; Laiu et al., 2020). It should be noted that, in addition to exhibiting favorable parallel scalability, DG methods are also an attractive choice for discretizing the neutrino transport equations because they recover the correct asymptotic behavior in the so-called diffusion limit (e.g., Larsen & Morel, 1989; Adams, 2001), which is characterized by frequent neutrino–matter interactions. Then, since the matter and neutrinos are strongly coupled in the CCSN environment, employing the DG method also for the hydrodynamics is most natural, as this enables treatment of the coupled physics in a unified mathematical framework. Currently, THORNADO is being developed as a collection of modules, focusing on single-node performance for updating structured data blocks using CPUs and/or GPUs, with the future aim of leveraging an external framework — e.g., AMReX[3] (Zhang et al., 2019) —

---

[2]https://github.com/endeve/thornado
[3]https://amrex-codes.github.io

to support mesh adaptivity.

This paper describes the DG algorithms for non-relativistic hydrodynamics in THORNADO. We adapt a three-covariant formalism that is sufficiently general to accommodate Cartesian, spherical-polar, and cylindrical spatial coordinates. Although we presented preliminary results obtained with similar algorithms for non-relativistic and relativistic hydrodynamics in the context of an ideal EoS in Endeve et al. (2019), this paper provides a more comprehensive description of the methods in THORNADO, and, more important, develops the algorithms further in order to accommodate a nuclear EoS. Introducing a nuclear matter EoS leads to more realistic models, but also complicates the numerical procedure. For instance, when solving the conservation equations for mass, momentum, and energy, the implementation of a nuclear EoS requires an additional conservation law for electrons, (see, e.g., Colella & Glaz, 1985; Zingale & Katz, 2015, for similar modifications). Moreover, on-the-fly numerical evaluation of a realistic EoS is computationally expensive (Swesty, 1996); thus, for computational expediency, EoSs are provided in tabulated form, and interpolations are used to access quantities away from table vertices, where a thermodynamically consistent interpolation scheme may be required (see, e.g., Swesty, 1996; Timmes & Swesty, 2000; Fryxell et al., 2000, for a discussion of such interpolation schemes). To limit the scope of this paper, we exclusively consider the SFHo EoS (Steiner et al., 2013a), which is provided in tabulated form by CompOSE[4]. In THORNADO, the interface to the tabulated EoS is through the WeakLib library[5], which provides auxiliary functionality needed for computations (e.g., input/output and interpolation). As such, the EoS is currently treated as a black box.

The Euler equations in curvilinear coordinates, extended to accommodate a nuclear EoS and self-gravity, are listed in Section 7.2. Then, in Section 5.4, we present the RKDG method in THORNADO. Sections 5.4.1 and 5.4.2 provide the spatial and temporal discretizations, respectively, which are based on the standard framework from Cockburn (2001). More specifically, we employ a nodal DG method (e.g., Hesthaven & Warburton, 2008) and adopt the spectral type nodal collocation approximation investigated by Bassi et al. (2013). Sections 5.4.3 and 5.4.4 discuss

---

[4]https://compose.obspm.fr
[5]https://github.com/starkiller-astro/weaklib

the slope limiter (to prevent non-physical oscillations) and the bound-enforcing limiter (to prevent non-physical states), respectively. The extension of these limiters to the case with a tabulated nuclear EoS is nontrivial. First, since slope limiting is most effective when applied to characteristic variables, we provide the characteristic decomposition of the flux Jacobian matrices for a nuclear EoS (Appendix 9C). Second, since the domain of validity of the nuclear EoS is more complex than the ideal case, we develop an enhanced version of the bound-enforcing limiter of Zhang & Shu (2010). Section 5.4.5 describes the Poisson solver for use in spherically symmetric problems with self-gravity, which uses the finite-element method. Section 5.4.6 provides details on the interpolation methods used to evaluate the tabulated EoS. We use basic trilinear interpolation, which is commonly employed in supernova simulation codes (e.g., Bruenn et al., 2020). In Section 5.5, to demonstrate the efficacy of the algorithms, we present numerical results from basic test problems (advection and Riemann problems) in idealized settings in one and two spatial dimensions. We also include a test of the Poisson solver. Then, in Section 5.6, we apply the DG method to the problem of adiabatic collapse, shock formation, and shock propagation in spherical symmetry, using a 15 $M_\odot$ progenitor. Here we focus on aspects of the limiters, resolution dependence, and total energy conservation. Our major goals in this paper are to (1) present the key algorithmic components of the hydrodynamics in THORNADO, (2) assess the implementation given the initial set of algorithmic choices, and (3) identify potential areas for improvement. This will clear the way for incorporating DG methods for neutrino transport and future neutrino radiation-hydrodynamics simulations with THORNADO.

5.3 Physical Model

5.3.1 Euler Equations

In this paper we adopt the non-relativistic Euler equations of gas dynamics in a coordinate basis (e.g., Rezzolla & Zanotti, 2013), supplemented with a nuclear equation of state (EoS), which are given by the mass conservation equation

$$\partial_t \rho + \frac{1}{\sqrt{\gamma}} \, \partial_i \left(\sqrt{\gamma} \, \rho \, v^i \right) = 0, \tag{5.1}$$

107

the momentum equation

$$\partial_t(\rho \, v_j) + \frac{1}{\sqrt{\gamma}} \, \partial_i \big(\sqrt{\gamma} \, \Pi^i{}_j \big) = \frac{1}{2} \, \Pi^{ik} \, \partial_j \gamma_{ik} - \rho \, \partial_j \Phi, \tag{5.2}$$

the energy equation

$$\partial_t E + \frac{1}{\sqrt{\gamma}} \, \partial_i \big(\sqrt{\gamma} \, [\, E + p \,] \, v^i \big) = -\rho \, v^i \, \partial_i \Phi, \tag{5.3}$$

and the electron conservation equation

$$\partial_t D_e + \frac{1}{\sqrt{\gamma}} \, \partial_i \big(\sqrt{\gamma} \, D_e \, v^i \big) = 0, \tag{5.4}$$

where ρ represents mass density, v^i the components of the fluid three-velocity, $\Pi^i{}_j = \rho \, v^i \, v_j + p \, \delta^i{}_j$ the stress tensor, p the fluid pressure, $D_e = \rho Y_e$, where Y_e is the electron fraction, $E = \epsilon \rho + \frac{1}{2} \rho v^2$ the total fluid energy density (internal plus kinetic), and ϵ is the specific internal energy. The Euler equations are closed with the EoS, where the pressure and specific internal energy are given functions of density, temperature T, and the electron fraction; e.g., $p = p(\rho, T, Y_e)$. Thus, Equation (5.4) is necessary for the inclusion of a nuclear EoS. (Unless stated otherwise, we use the Einstein summation convention where repeated latin indices run from 1 to 3.) Included on the right-hand sides of Equations (5.2) and (5.3), are gravitational sources from the Newtonian gravitational potential Φ, which is obtained from the Poisson equation

$$\frac{1}{\sqrt{\gamma}} \partial_i \big(\sqrt{\gamma} \, \gamma^{ij} \partial_j \Phi \big) = 4\pi \, G \, \rho, \tag{5.5}$$

where G is Newton's constant.

The use of curvilinear coordinates is enabled through the spatial metric tensor γ_{ik}, which gives the squared proper spatial interval

$$ds_x^2 = \gamma_{ik} \, dx^i \, dx^k. \tag{5.6}$$

The determinant of the spatial metric is denoted γ. The metric tensor is also used to raise and lower indices on vectors and tensors; e.g., $v_i = \gamma_{ik} v^k$. In this paper we only consider the commonly adopted Cartesian, cylindrical, and spherical-polar coordinate systems (see Table 5.1 for relevant quantities associated with each of these systems). Thus, the metric tensor is diagonal, and we assume

that it is time independent. Note that we also list the scale factors h_1, h_2, and h_3 in Table 5.1. By specifying the scale factors, components of the spatial metric are obtained from $\gamma_{11} = h_1 h_1$, $\gamma_{22} = h_2 h_2$, and $\gamma_{33} = h_3 h_3$, and the square root of the metric determinant is $\sqrt{\gamma} = h_1 h_2 h_3$.

For the discussion of the numerical method in Section 5.4, we rewrite Equations (5.1)–(5.4) in a more convenient way as a system of hyperbolic balance equations

$$\partial_t \mathbf{U} + \frac{1}{\sqrt{\gamma}} \, \partial_i \big(\sqrt{\gamma} \, \mathbf{F}^i(\mathbf{U}) \big) = \mathbf{S}(\mathbf{U}, \Phi), \tag{5.7}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho v_j \\ E \\ D_\mathrm{e} \end{bmatrix}, \quad \mathbf{F}^i(\mathbf{U}) = \begin{bmatrix} \rho \, v^i \\ \Pi^i{}_j \\ (E + p) \, v^i \\ D_\mathrm{e} \, v^i \end{bmatrix}, \quad \text{and} \quad \mathbf{S}(\mathbf{U}, \Phi) = \begin{bmatrix} 0 \\ \frac{1}{2} \Pi^{ik} \partial_j \gamma_{ik} - \rho \, \partial_j \Phi \\ -\rho \, v^i \, \partial_i \Phi \\ 0 \end{bmatrix} \tag{5.8}$$

are the vector of evolved quantities, the flux vectors, and the source vector, respectively. We split the source vector further as $\mathbf{S}(\mathbf{U}, \Phi) = \mathbf{S}^\gamma(\mathbf{U}) + \mathbf{S}^\Phi(\mathbf{U}, \Phi)$, where

$$\mathbf{S}^\gamma(\mathbf{U}) = \begin{bmatrix} 0 \\ \frac{1}{2} \Pi^{ik} \partial_j \gamma_{ik} \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{S}^\Phi(\mathbf{U}, \Phi) = -\rho \begin{bmatrix} 0 \\ \partial_j \Phi \\ v^i \, \partial_i \Phi \\ 0 \end{bmatrix}. \tag{5.9}$$

Table 5.1 Metric quantities for Cartesian, cylindrical, and spherical-polar coordinate systems.

Coordinates	x^1	x^2	x^3	h_1	h_2	h_3	γ_{11}	γ_{22}	γ_{33}	$\sqrt{\gamma}$	$\frac{1}{\gamma_{22}}\frac{\partial \gamma_{22}}{\partial x^1}$	$\frac{1}{\gamma_{33}}\frac{\partial \gamma_{33}}{\partial x^1}$	$\frac{1}{\gamma_{33}}\frac{\partial \gamma_{33}}{\partial x^2}$
Cartesian	x	y	z	1	1	1	1	1	1	1	0	0	0
Cylindrical	R	z	ϕ	1	1	R	1	1	R^2	R	0	$2/R$	0
Spherical	r	θ	ϕ	1	r	$r \sin\theta$	1	r^2	$r^2 \sin^2\theta$	$r^2 \sin\theta$	$2/r$	$2/r$	$2\cot\theta$

5.3.2 Equation of State

The EoS provides thermodynamic quantities such as pressure, internal energy, and entropy (dependent variables) as functions of the independent variables; e.g., density, temperature, and electron fraction. (Other choices for the independent variables — e.g., density, entropy, and electron fraction — are of course also possible, but in the nuclear astrophysics modeling community it is perhaps most common to use ρ, T, and Y_e.) These dependent variables, and in some cases their derivatives, are crucial for modeling hydrodynamics, nuclear reactions, and neutrino transport in core-collapse supernovae. Of particular importance for numerical methods for the hydrodynamics, is the relationship between the EoS and the well-posedness of the system given by Equation (5.7). Specifically, the system is said to be hyperbolic if the Jacobian matrices $\partial \mathbf{F}^i / \partial \mathbf{U}$ can be diagonalized with a set of real eigenvalues $\{\lambda_1^i, \ldots, \lambda_6^i\}$ and has a set of linearly independent right eigenvectors $\{\mathbf{r}_1^i, \ldots, \mathbf{r}_6^i\}$ such that (cf. LeVeque, 1992; Rezzolla & Zanotti, 2013)

$$\left(\partial \mathbf{F}^i / \partial \mathbf{U}\right) \mathbf{r}_j^i = \lambda_j^i \, \mathbf{r}_j^i, \quad \text{for} \quad j = 1, \ldots, 6. \tag{5.10}$$

(In Equation (5.10), repeated indices do not imply summation, but rather that it must hold for each of the three flux vectors.) For the system in Equation (5.7), the eigenvalues are given by $\{ v^i - c_s \sqrt{\gamma^{ii}}, v^i, v^i, v^i, v^i, v^i + c_s \sqrt{\gamma^{ii}} \}$, where c_s is the sound speed; $c_s^2 = \left(\partial p / \partial \rho\right)_{s,Y_e}$, where s is the entropy per baryon. A fundamental property of hyperbolic equations is that they are well-posed, which makes them suitable for numerical solution (see, e.g., Rezzolla & Zanotti, 2013, for a discussion). Thus, a necessary condition for our system to be suitable for numerical solution is $c_s^2 > 0$. When the independent variables are chosen to be ρ, T, and Y_e, the square of the sound speed can be written explicitly in terms of thermodynamic derivatives as

$$c_s^2 = \left(\frac{\partial p}{\partial \rho}\right)_{s,Y_e} = \left(\frac{\partial p}{\partial \rho}\right)_{T,Y_e} - \left(\frac{\partial s}{\partial T}\right)_{\rho,Y_e}^{-1} \left(\frac{\partial p}{\partial T}\right)_{\rho,Y_e} \left(\frac{\partial s}{\partial \rho}\right)_{T,Y_e}. \tag{5.11}$$

The sound speed, or a related quantity, is typically included with a tabulated EoS. In addition, advanced numerical methods make use of the eigenvectors in Equation (5.10), e.g., for the characteristic limiting described in Section 5.4.3. These eigenvectors in turn depend on additional thermodynamic derivatives, whose estimation from the EoS table is discussed in 5.4.6. For use

in computations, THORNADO has been developed to use the EoS infrastructure provided by the WeakLib library. (Specifically, WeakLib supplies trilinear interpolation, and derivatives computed by analytic differentiation of the trilinear interpolation formula.)

5.4 Numerical Method

5.4.1 Discontinuous Galerkin Method

In THORNADO we employ the Runge-Kutta discontinuous Galerkin (RKDG) method to solve the Euler equations given by Equation (5.7). (We refer to Cockburn (2001) for an excellent review on the RKDG method, and Shu (2016) for a summary of more recent developments.) To this end, the d-dimensional computational domain $D \subset \mathbb{R}^d$ is subdivided into the union $\mathcal{T}$ of non-overlapping elements K such that $D = \cup_{K \in \mathcal{T}} K$. We take each element to be a logically Cartesian box

$$K = \left\{ x : x^i \in K^i := (x_L^i, x_H^i), i = 1, \ldots, d \right\}, \tag{5.12}$$

where x_L^i and x_H^i are the low and high boundaries of the element in the ith dimension. We also define the surface elements $\tilde{K}^i = \times_{j \neq i}^d K^j$ (so that $K = \tilde{K}^i \times K^i$), the set $x = \{x^i, \tilde{x}^i\}$ to distinguish coordinates parallel and perpendicular to the ith dimension, and the element width $\Delta x^i = (x_H^i - x_L^i)$ and center $x_C^i = \frac{1}{2}(x_L^i + x_H^i)$. We also define $|K| = \prod_{i=1}^d \Delta x^i$ and $|\tilde{K}^i| = \prod_{j=1, j \neq i}^d \Delta x^j$. We let the volume of an element be denoted

$$V_K = \int_K dV_h, \quad \text{where} \quad dV_h = \sqrt{\gamma_h} \prod_{i=1}^d dx^i, \tag{5.13}$$

where γ_h is the determinant of the approximate spatial metric $(\gamma_h)_{ij}$. We will discuss the approximation to the spatial metric in more detail below.

On each element, we define the approximation space consisting of functions ψ_h

$$\mathbb{V}_h^k = \left\{ \psi_h : \psi_h|_K \in \mathbb{Q}^k(K), \forall K \in \mathcal{T} \right\}, \tag{5.14}$$

where $\mathbb{Q}^k$ is the tensor product space of one-dimensional polynomials of maximal degree k. In the DG method, the functions in $\mathbb{V}_h^k$ can be discontinuous across element interfaces. In THORNADO we use Lagrange polynomials,

$$\ell_p(\xi^i) = \prod_{\substack{q=1 \\ q \neq p}}^N \frac{\xi^i - \xi_q^i}{\xi_p^i - \xi_q^i}, \tag{5.15}$$

where $N = k + 1$ and the polynomials ℓ_p are defined on the unit reference interval $I^i = \{\xi^i : \xi^i \in (-\frac{1}{2}, \frac{1}{2})\}$ $(i = 1, \ldots, d)$. The physical coordinate x^i is related to the reference coordinate ξ^i by the transformation $x^i(\xi^i) = x_C^i + \Delta x^i \xi^i$. For the Lagrange polynomials, we define the set of interpolation points $S_N^i = \{\xi_1^i, \ldots, \xi_N^i\} \subseteq I^i$. Note that for $\xi_q^i \in S_N^i$, we have $\ell_p(\xi_q^i) = \delta_{pq}$, where δ_{pq} is the Kronecker delta. As an example, the multi-dimensional basis function $\phi_i(x(\xi)) \in \mathbb{V}_h^k$ takes the form

$$\phi_i(\xi) = \phi_{\{i_1,\ldots,i_d\}}(\xi^1, \ldots, \xi^d) = \ell_{i_1}(\xi^1) \times \ldots \times \ell_{i_d}(\xi^d), \tag{5.16}$$

where we have introduced the multi-index $i = \{i_1, \ldots, i_d\} \in \mathbb{N}^d$ (a d-tuple) to achieve a more compact notation. To further illustrate, in each element K we approximate the solution to Equation (5.7) by U_h, which is given by an expansion of functions in $\mathbb{V}_h^k$ of the form

$$U_h(x,t) = \sum_{i=1}^N U_i(t)\, \phi_i(x(\xi)) = \sum_{i_1=1}^N \ldots \sum_{i_d=1}^N U_{\{i_1,\ldots,i_d\}}(t)\, \ell_{i_1}(\xi^1) \times \ldots \times \ell_{i_d}(\xi^d), \tag{5.17}$$

where $N \in \mathbb{N}^d$ is the d-tuple $\{N, \ldots, N\}$. The DG method does not require that the approximate multidimensional solution is constructed from one-dimensional polynomials of the same degree k in each dimension, but we make this choice. In the multidimensional setting, we denote the set of interpolation points in element K by $S_N = \otimes_{i=1}^d S_N^i$. For $\xi_j \in S_N$, we have $\phi_i(\xi_j) = \delta_{ij} = \delta_{i_1 j_1} \times \ldots \times \delta_{i_d j_d}$, which follows from the Kronecker delta property of the Lagrange polynomials emphasized above. Therefore, for $\xi_j \in S_N$, a direct evaluation in Equation (5.17) shows that $U_h(x(\xi_j)) = U_j(t)$; i.e., the expansion coefficients in Equation (5.17) — the unknowns to be determined by the DG method — are simply the evolved quantities evaluated in the interpolation points on each element.

We are now ready to state the DG formulation, which forms the basis for the DG method implemented in THORNADO. The semi-discrete DG problem is to find $U_h \in \mathbb{V}_h^k$, which approximates U in Equation (5.7), such that

$$\langle \partial_t U_h, \psi_h \rangle_K = \mathcal{B}_h^{\mathsf{Flx}}(U_h, \psi_h)_K + \langle S(U_h, \Phi_h), \psi_h \rangle_K \equiv \mathcal{B}_h(U_h, \Phi_h, \psi_h)_K \tag{5.18}$$

holds for all test functions $\psi_h \in \mathbb{V}_h^k$ and all elements $K \in \mathcal{T}$. In Equation (5.18),

$$\langle \partial_t U_h, \psi_h \rangle_K = \int_K \partial_t U_h \psi_h \, dV_h, \qquad (5.19)$$

and we have defined the contributions from the fluxes as

$$\mathcal{B}_h^{\mathsf{Flx}}(U_h, \psi_h)_K = -\sum_{i=1}^d \int_{\tilde{K}^i} \left(\sqrt{\gamma_h} \, \widehat{F}^i(U_h) \, \psi_h|_{x_{\mathrm{H}}^i} - \sqrt{\gamma_h} \, \widehat{F}^i(U_h) \, \psi_h|_{x_{\mathrm{L}}^i} \right) d\tilde{x}^i + \sum_{i=1}^d \int_K F^i(U_h) \, \partial_i \psi_h \, dV_h, \qquad (5.20)$$

and the contributions from the sources as

$$\langle S(U_h, \Phi_h), \psi_h \rangle_K = \int_K S(U_h, \Phi_h) \, \psi_h \, dV_h. \qquad (5.21)$$

The approximation to the Newtonian gravitational potential, denoted Φ_h (not to be confused with the basis functions ϕ_i in Equation (5.17)), is obtained by solving Equation (5.5) using a finite element method. We discuss this in Section 5.4.5.

In Equation (5.20), the numerical flux $\widehat{F}^i(U_h)$ is introduced to define a unique flux in the ith surface of K. This numerical flux is computed from a numerical *flux function* (obtained, e.g., from solving an approximate Riemann problem)

$$\widehat{F}^i(U_h; x^i, \tilde{x}^i) = f^i\left(U_h(x^{i,-}, \tilde{x}^i), U_h(x^{i,+}, \tilde{x}^i) \right), \qquad (5.22)$$

where superscripts $-/+$ in the arguments of $U_h(x^{i,-/+}, \tilde{x}^i)$ indicate that the approximation is evaluated to the immediate left/right of the interface located at x^i. In THORNADO we have implemented the HLL (Harten et al., 1983a) and HLLC (Toro et al., 1994) flux functions, but in the numerical experiments in Sections 5.5 and 5.6, we use exclusively the HLL flux function given by

$$f^i\left(U_h^-, U_h^+\right) = \frac{\alpha^{i,+} F^i(U_h^-) + \alpha^{i,-} F^i(U_h^+) - \alpha^{i,-}\alpha^{i,+}\left(U_h^+ - U_h^-\right)}{\alpha^{i,-} + \alpha^{i,+}}, \qquad (5.23)$$

where $U_h^{\pm} = U_h(x^{i,\pm}, \tilde{x}^i)$, and where $\alpha^{i,-}$ and $\alpha^{i,+}$ are wave speed estimates for the fastest (in absolute value; $\alpha^{i,\pm} \geq 0$) left and right propagating waves, respectively. For these estimates we simply use (Davies, 1988)

$$\alpha^{i,-} = \max_{j \in \{1,\dots,6\}} \left(0, -\lambda_j^i(U_h^-), -\lambda_j^i(U_h^+) \right) \quad \text{and} \quad \alpha^{i,+} = \max_{j \in \{1,\dots,6\}} \left(0, +\lambda_j^i(U_h^-), +\lambda_j^i(U_h^+) \right), \quad (5.24)$$

where λ_j^i are the eigenvalues of the flux Jacobian introduced in Equation (5.10).

Motivated by results presented by Bassi et al. (2013), we employ a spectral-type collocation nodal DG method in THORNADO. To this end, we use Legendre–Gauss (LG) points to construct the interpolation points comprising $\boldsymbol{S}_N$. See the left panel of Figure 5.1 for the distribution of the interpolation points $\boldsymbol{S}_N$ in the two-dimensional case with $k = 2$ (black, filled circles). In the collocation nodal DG method, these interpolation points are also used as quadrature points to evaluate integrals in Equation (5.18). One of the benefits of this collocation method is computational efficiency since, even when using curvilinear coordinates, the mass matrix associated with the term in Equation (5.19) is diagonal and easily invertible. On the other hand, demanding exact evaluation of integrals — e.g., by using an extended quadrature set — results in mass matrices that are non-diagonal and vary from element to element because of the spatially dependent metric determinant in dV_h in Equation (5.19). The use of LG points, as opposed to Legendre–Gauss–Lobatto (LGL) points, provides better accuracy in evaluating the integrals. In the one-dimensional setting, the N-point LG quadrature evaluates polynomials of degree up to $2N - 1$ exactly, while the corresponding LGL quadrature evaluates polynomials of degree up to $2N - 3$ exactly. Let Q_N^i denote the one-dimensional N-point LG quadrature on the interval I^i with abscissas $\{\xi_q^i\}_{q=1}^N$ and weights $\{w_q^i\}_{q=1}^N$, normalized so that $\sum_{q=1}^N w_q^i = 1$. (Note that quadrature points and weights defined on the commonly used reference interval $[-1, 1]$ (e.g., Cockburn, 2001) must be scaled by a factor of $\frac{1}{2}$ before use on the reference interval $[-\frac{1}{2}, \frac{1}{2}]$ used in THORNADO.) Multidimensional integrals are evaluated by tensorization of one-dimensional quadratures. For volume integrals over the multidimensional reference element $\boldsymbol{I} = \times_{i=1}^d I^i$, we let $\boldsymbol{Q}_N = \otimes_{i=1}^d Q_N^i$ denote the tensorization of one-dimensional N-point LG quadrature rules with abscissas $\{\boldsymbol{\xi_q}\}_{\boldsymbol{q}=1}^N$ and weights $\{w_{\boldsymbol{q}}\}_{\boldsymbol{q}=1}^N$, where $\boldsymbol{q} = \{q_1, \ldots, q_d\} \in \mathbb{N}^d$, $\boldsymbol{\xi_q} = \{\xi_{q_1}^1, \ldots, \xi_{q_d}^d\}$, and $w_{\boldsymbol{q}} = w_{q_1} \times \ldots \times w_{q_d}$, so that the integral of a

polynomial $P(\boldsymbol{x}) \in \mathbb{V}_h^k$ in element $\boldsymbol{K}$ is evaluated as

$$\int_{\boldsymbol{K}} P(\boldsymbol{x}) \, d\boldsymbol{x} = |\boldsymbol{K}| \int_{\boldsymbol{I}} P(\boldsymbol{\xi}) \, d\boldsymbol{\xi} = |\boldsymbol{K}| \, \boldsymbol{Q}_N \big[P(\boldsymbol{\xi}) \big] = |\boldsymbol{K}| \sum_{q=1}^{N} w_{\boldsymbol{q}} \, P(\boldsymbol{\xi}_{\boldsymbol{q}})$$

$$= \Delta x^1 \times \ldots \times \Delta x^d \sum_{q_1=1}^{N} \ldots \sum_{q_d=1}^{N} w_{q_1} \times \ldots \times w_{q_d} \, P(\xi_{q_1}^1, \ldots, \xi_{q_d}^d). \tag{5.25}$$

Similarly, for surface integrals over the reference surface element $\tilde{\boldsymbol{I}}^i = \times_{j=1, j\neq i}^d I^j$, we let $\tilde{\boldsymbol{Q}}_N^i = \otimes_{j=1, j\neq i}^d Q_N^j$ denote the tensorization of one-dimensional N-point LG quadrature rules with abscissas $\{\tilde{\boldsymbol{\xi}}_{\tilde{q}_i}^i\}_{\tilde{q}_i=1}^N$ and weights $\{w_{\tilde{q}_i}\}_{\tilde{q}_i=1}^N$, where $\tilde{q}_i = \{q_j\}_{j=1, j\neq i}^d \in \mathbb{N}^{d-1}$, $\tilde{\boldsymbol{\xi}}_{\tilde{q}_i}^i = \{\xi_{q_j}^j\}_{j=1, j\neq i}^d$, and $w_{\tilde{q}_i} = \prod_{j=1, j\neq i}^d w_{q_j}$, so that for $P(x^i, \tilde{\boldsymbol{x}}^i) \in \mathbb{V}_h^k$, the integral over the surface element $\tilde{\boldsymbol{K}}^i$ is evaluated as

$$\int_{\tilde{\boldsymbol{K}}^i} P(x^i, \tilde{\boldsymbol{x}}^i) \, d\tilde{\boldsymbol{x}}^i = |\tilde{\boldsymbol{K}}^i| \int_{\tilde{\boldsymbol{I}}^i} P(x^i, \tilde{\boldsymbol{\xi}}^i) \, d\tilde{\boldsymbol{\xi}}^i = |\tilde{\boldsymbol{K}}^i| \, \tilde{\boldsymbol{Q}}_N^i \big[P(x^i, \tilde{\boldsymbol{\xi}}^i) \big] = |\tilde{\boldsymbol{K}}^i| \sum_{\tilde{q}_i=1}^{N} w_{\tilde{q}_i} \, P(x^i, \tilde{\boldsymbol{\xi}}_{\tilde{q}_i}^i)$$

$$\overset{(i=1)}{=} \Delta x^2 \times \ldots \times \Delta x^d \sum_{q_2=1}^{N} \ldots \sum_{q_d=1}^{N} w_{q_2} \times \ldots \times w_{q_d} \, P(x^1, \xi_{q_2}^2, \ldots, \xi_{q_d}^d), \tag{5.26}$$

where the specific case with $i = 1$ is given in the second line. The points used to evaluate volume integrals with the $\boldsymbol{Q}_N$ quadrature rule for the case with $d = k = 2$ are shown as black, filled circles in the right panel in Figure 5.1. (Note that these points are identical to the interpolation points displayed as black, filled circles in the left panel in Figure 5.1.) The quadrature points used to evaluate surface integrals with $\tilde{\boldsymbol{Q}}^1$ and $\tilde{\boldsymbol{Q}}^2$ are shown as the gray, open squares on the boundary of the element.

By inserting the expansion in Equation (5.17), letting $\psi_h = \phi_{\boldsymbol{p}}$, where $\phi_{\boldsymbol{p}}$ is one of the basis functions in the expansion in Equation (5.17), and using the quadrature rule in Equation (5.25), we can evaluate Equation (5.19) as

$$\langle \partial_t U_h, \, \phi_{\boldsymbol{p}} \rangle_{\boldsymbol{K}} := w_{\boldsymbol{p}} \, |\boldsymbol{K}| \sqrt{\gamma_{\boldsymbol{p}}} \, \partial_t U_{\boldsymbol{p}}, \tag{5.27}$$

where $w_{\boldsymbol{j}} \, |\boldsymbol{K}| \sqrt{\gamma_{\boldsymbol{j}}}$ are the elements of the diagonal mass matrix and $\gamma_{\boldsymbol{j}} = \gamma_h(\boldsymbol{x}_{\boldsymbol{j}})$. Similarly, using
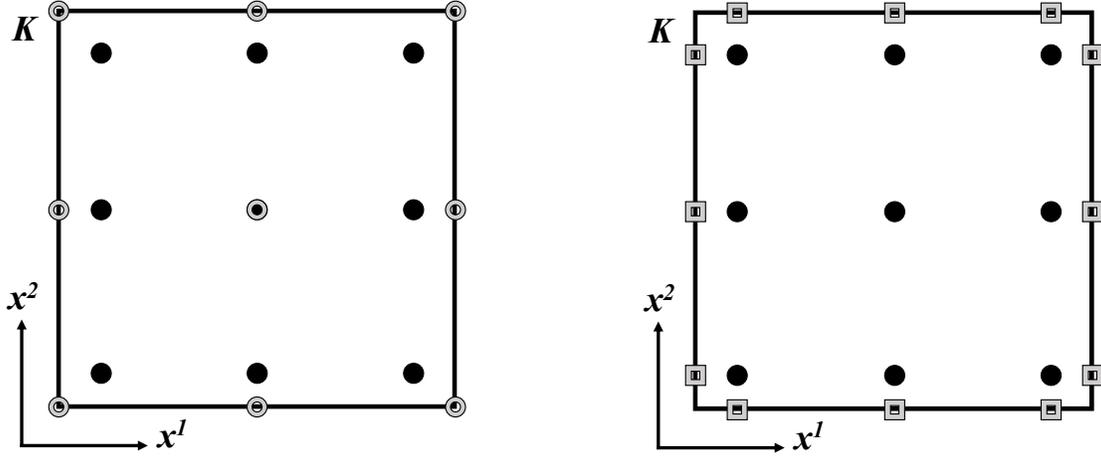
Figure 5.1 Reference elements with interpolation and quadrature points used in the DG method implemented in THORNADO for the two-dimensional case ($d = 2$) with polynomials of degree $k = 2$ ($N = 3$). In the left panel, interpolation points are shown for the hydrodynamics variables [$\boldsymbol{S}_N$ (based on LG quadrature points; black, filled circles)] and the geometry scale factors and the Newtonian gravitational potential [$\boldsymbol{\hat{S}}_N$ (based on LGL quadrature points; gray, open circles)]. In the right panel, quadrature points associated with volume integrals (black, filled circles) and surface integrals (gray, open squares) are shown. Note that in the collocation nodal DG method, the interpolation points in the left panel, $\boldsymbol{S}_N$, coincide with the quadrature points in the right panel. The quadrature points on the surface of the element are obtained as the projection of the quadrature points inside the element onto each surface.

the quadrature in Equation (5.26), the contributions from fluxes can be written as

$$
\mathcal{B}_h^{\mathsf{Flx}}\big(\boldsymbol{U}_h, \phi_{\boldsymbol{p}}\big)_K := -\sum_{i=1}^{d} w_{\tilde{\boldsymbol{p}}_i} \, |\tilde{\boldsymbol{K}}^i| \Big(\sqrt{\gamma_h(x_{\mathrm{H}}^i, \tilde{\boldsymbol{x}}_{\tilde{\boldsymbol{p}}_i}^i)} \, \widehat{\boldsymbol{F}}^i(x_{\mathrm{H}}^i, \tilde{\boldsymbol{x}}_{\tilde{\boldsymbol{p}}_i}^i) \, \ell_{p_i}(x_{\mathrm{H}}^{i;-}) - \sqrt{\gamma_h(x_{\mathrm{L}}^i, \tilde{\boldsymbol{x}}_{\tilde{\boldsymbol{p}}_i}^i)} \, \widehat{\boldsymbol{F}}^i(x_{\mathrm{L}}^i, \tilde{\boldsymbol{x}}_{\tilde{\boldsymbol{p}}_i}^i) \, \ell_{p_i}(x_{\mathrm{L}}^{i;+}) \Big)
$$

$$
+ \sum_{i=1}^{d} w_{\tilde{\boldsymbol{p}}_i} \, |\tilde{\boldsymbol{K}}^i| \sum_{q_i=1}^{N} w_{q_i} \sqrt{\gamma_h(x_{q_i}^i, \tilde{\boldsymbol{x}}_{\tilde{\boldsymbol{p}}_i}^i)} \, \boldsymbol{F}^i(x_{q_i}^i, \tilde{\boldsymbol{x}}_{\tilde{\boldsymbol{p}}_i}^i) \, \frac{\partial \ell_{p_i}}{\partial \xi^i}(\xi_{q_i}^i). \tag{5.28}
$$

Finally, the source term becomes

$$
\langle \, \boldsymbol{S}(\boldsymbol{U}_h, \Phi_h), \, \phi_{\boldsymbol{p}} \, \rangle_K := w_{\boldsymbol{j}} \, |K| \, \sqrt{\gamma_{\boldsymbol{p}}} \, \boldsymbol{S}_{\boldsymbol{p}}, \tag{5.29}
$$

where $\boldsymbol{S}_{\boldsymbol{p}}$ is the source vector in Equation (5.8), evaluated in $\boldsymbol{x}_{\boldsymbol{p}}$. Combining Equations (5.27), (5.28), and (5.29), we can now write the spectral-type collocation DG approximation to the semi-discrete DG problem in Equation (5.18) in terms of an evolution equation for the expansion

116

coefficient U_p in element K as

$$\partial_t U_p = - \sum_{i=1}^{d} \frac{1}{w_{p_i} \Delta x^i \sqrt{\gamma_p}} \left(\sqrt{\gamma_h(x_{\mathrm{H}}^i, \tilde{x}_{\tilde{p}_i}^i)} \, \widehat{F}^i(x_{\mathrm{H}}^i, \tilde{x}_{\tilde{p}_i}^i) \, \ell_{p_i}(x_{\mathrm{H}}^{i,-}) - \sqrt{\gamma_h(x_{\mathrm{L}}^i, \tilde{x}_{\tilde{p}_i}^i)} \, \widehat{F}^i(x_{\mathrm{L}}^i, \tilde{x}_{\tilde{p}_i}^i) \, \ell_{p_i}(x_{\mathrm{L}}^{i,+}) \right)$$

$$+ \sum_{i=1}^{d} \frac{1}{w_{p_i} \Delta x^i \sqrt{\gamma_p}} \sum_{q_i=1}^{N} w_{q_i} \sqrt{\gamma_h(x_{q_i}^i, \tilde{x}_{\tilde{p}_i}^i)} \, F^i(x_{q_i}^i, \tilde{x}_{\tilde{p}_i}^i) \, \frac{\partial \ell_{p_i}}{\partial \xi^i}(\xi_{q_i}^i) + S_p. \tag{5.30}$$

(For an example of Equation (5.30) in the simpler one-dimensional setting, see Endeve et al. (2019); their Equation (11).)

The cell averages in element K, defined as

$$U_K = \frac{1}{V_K} \int_K U_h \, dV_h := \frac{\sum_{p=1}^{N} w_p \sqrt{\gamma_p} \, U_p}{\sum_{p=1}^{N} w_p \sqrt{\gamma_p}}, \quad \text{where} \quad V_K = |K| \sum_{p=1}^{N} w_p \sqrt{\gamma_p}, \tag{5.31}$$

play an important role in the analysis and implementation of the DG method given by Equation (5.30). (Examples of the use of the cell averages are given in Sections 5.4.3 and 5.4.4, where we discuss limiting techniques.) From the definition of the cell average in Equation (5.31) and from Equation (5.30), the equation for the cell average can be written as

$$\partial_t U_K = - \frac{|K|}{V_K} \sum_{i=1}^{d} \tilde{Q}_N^i \left[\sqrt{\gamma_h(x_{\mathrm{H}}^i, \tilde{x}^i)} \widehat{F}^i(x_{\mathrm{H}}^i, \tilde{x}^i) - \sqrt{\gamma_h(x_{\mathrm{L}}^i, \tilde{x}^i)} \widehat{F}^i(x_{\mathrm{L}}^i, \tilde{x}^i) \right] / \Delta x^i + S_K, \tag{5.32}$$

where we used the quadrature rule in Equation (5.26) to represent the surface integrals, while the source term can be written in terms of the quadrature rule in Equation (5.25)

$$S_K = \frac{|K|}{V_K} Q_N \left[\sqrt{\gamma_h(x)} \, S(x) \right]. \tag{5.33}$$

To arrive at Equation (5.32), we used the property of the Lagrange polynomial in Equation (5.15) that $\sum_{p=1}^{N} \ell_p(\xi^i) = 1$ for any $\xi^i \in I^i$. Equation (5.32) exhibits the expected conservation form, with quadrature rules replacing integrals over the surface of K. In the absence of sources, the DG discretization in Equation (5.30) is conservative for mass, momentum, energy, and electron number. We also note that Equation (5.32) is familiar from the literature on finite-volume (FV) methods, which only evolve the cell averages. The DG and FV methods are in fact equivalent in the first-order case, when $k = 0$. However, for the extension to higher-order, FV methods reconstruct a local polynomial using cell averages in neighboring elements, while DG methods evolve all the

degrees-of-freedom in the local polynomial representation, so that the reconstruction step is not needed. Thus, one benefit of avoiding the reconstruction step becomes clear in the high-order case: while the FV stencil width increases with increasing spatial order of accuracy, the DG method only requires data from the local element and its nearest neighbors, independent of the order of accuracy.

We complete the specification of the basic DG method implemented in THORNADO by discussing the source terms due to the use of curvilinear coordinates and gravitational fields. In particular, we write [cf. Equation (5.9)]

$$S_p = S_p^\gamma + S_p^\Phi. \tag{5.34}$$

5.4.1.1 Geometric Source Terms

For the sources due to curvilinear coordinates, S_p^γ, the only nonzero components appear in the components of the momentum equation, which can be written in terms of the scale factors where, due to the diagonal metric, $\gamma_{ii} = h_i h_i$ and $\gamma^{ii} = 1/\gamma_{ii}$

$$\frac{1}{2} \Pi^{ik} \partial_j \gamma_{ik} = \frac{1}{2} \Pi^{11} \partial_j \gamma_{11} + \frac{1}{2} \Pi^{22} \partial_j \gamma_{22} + \frac{1}{2} \Pi^{33} \partial_j \gamma_{33} = \Pi^1{}_1 \frac{1}{h_1} \frac{\partial h_1}{\partial x^j} + \Pi^2{}_2 \frac{1}{h_2} \frac{\partial h_2}{\partial x^j} + \Pi^3{}_3 \frac{1}{h_3} \frac{\partial h_3}{\partial x^j}. \tag{5.35}$$

For the coordinate systems we consider here, the scale factors are independent of x^3, and only the first and second components of Equation (5.35) are nonzero (i.e., $j = 1, 2$; cf. Table 5.1). Note that $h_1 = 1$ for all the coordinate systems; therefore, spatial derivatives of h_1 vanish. For Cartesian coordinates, the scale factors are unity, and all the components of S_p^γ vanish. For cylindrical coordinates, only $h_3 = R$ contributes, while for spherical-polar coordinates both $h_2 = r$ and $h_3 = r \sin \theta$ contribute. In THORNADO, we approximate the scale factors by polynomials in each element. To this end, we define $\boldsymbol{h} = (h_1, h_2, h_3)^T$ and let the scale factors in $\boldsymbol{K}$ be given by the expansion

$$\boldsymbol{h}_h(\boldsymbol{x}) = \sum_{i=1}^{N} \boldsymbol{h_i}\, \hat{\phi}_i(\boldsymbol{x}) \in \mathbb{V}_h^k, \tag{5.36}$$

where $\hat{\phi}_i(\boldsymbol{x})$ are basis functions, similar to those defined in Equation (5.16). However, we demand that the scale factors are continuous across element interfaces. To achieve this we let $\hat{S}_N^i = \{\hat{\xi}_1, \ldots, \hat{\xi}_N^i\} \subseteq I^i$ denote the set of LGL points in the unit reference interval, since the LGL

points include the endpoints of I^i. For the scale factors (and, as discussed below, the Newtonian gravitational potential), we then let the interpolation points on K be given by $\hat{S}_N = \otimes_{i=1}^d \hat{S}_N^i$. The distribution of the interpolation points $\hat{S}_N$, used for the scale factors and the Newtonian gravitational potential, for the two-dimensional case with $k = 2$ are shown in the left panel of Figure 5.1 (gray, open circles). Hence, $\hat{\phi}_i(x)$ is defined as in Equation (5.16), but with the Lagrange polynomials in Equation (5.15) constructed with the LGL points $\hat{S}_N$, and the expansion coefficients h_i are given by the exact value of the scale factors in the LGL points. Scale factors in the LG points $x_i \in S_N$, which are needed, e.g., to compute the determinant of the spatial metric, are obtained from direct evaluation of Equation (5.36), $h_h(x_i)$, so that $\gamma_i = \gamma_h(x_i) := \gamma(h_h(x_i))$. Derivatives of the scale factors, needed for the source terms in Equation (5.35), are evaluated by analytic differentiation of Equation (5.36). Since in the present case the metric is time independent, the needed scale factors and their derivatives can be precomputed at program startup and stored for later use. Note that scale factors are polynomials and at most linear functions of the spherical-polar or cylindrical radius, so the representation is exact in the x^1-dimension if $N \geq 2$. However, for spherical-polar coordinates, h_3 is a trigonometric function in the x^2-dimension, and the representation in Equation (5.36) is only approximate.

Next we consider a special case where the geometric source terms, $\frac{1}{2}\Pi^{ik}\partial_j\gamma_{ik}$, and the divergence of the stress tensor, $\frac{1}{\sqrt{\gamma}}\partial_i(\sqrt{\gamma}\,\Pi^i_{\ j})$, appearing in the components of the momentum equation, Equation (5.2), must balance each other. Specifically, for a fluid associated with an isotropic and spatially homogeneous stress tensor, i.e., $\Pi^i_{\ k} = p_0\,\delta^i_{\ k}$ ($p_0 = $ constant), the divergence of the stress tensor must balance the geometry source exactly to prevent inducing spurious flows.

Considering Equation (5.32), with Equations (5.33) and (5.35), in spherical-polar coordinates and in the absence of gravity, assuming an isotropic and spatially homogeneous stress tensor, the equation for the first component of the momentum density (cf. Equation (5.8)), in the sense of the

cell-average, can be written as

$$\partial_t(\rho v_1)_{\boldsymbol{K}} = -p_0 \frac{|\tilde{\boldsymbol{K}}^1|}{V_{\boldsymbol{K}}} \left\{ \tilde{\boldsymbol{Q}}_N^1 \left[\sqrt{\gamma_h}(x_{\mathrm{H}}^1, \tilde{\boldsymbol{x}}^1) - \sqrt{\gamma_h}(x_{\mathrm{L}}^1, \tilde{\boldsymbol{x}}^1) \right] - \boldsymbol{Q}_N \left[2\sqrt{\gamma_h}(\boldsymbol{x})/x^1 \right] \right\}$$

$$= -p_0 \frac{|\tilde{\boldsymbol{K}}^1|}{V_{\boldsymbol{K}}} \tilde{\boldsymbol{Q}}_N^1 \left[(\sin\theta)_h \left((r_{\mathrm{H}}^2 - r_{\mathrm{L}}^2) - 2 Q_N^1[r] \right) \right], \tag{5.37}$$

where $(\sin\theta)_h$ is the polynomial approximation to $\sin\theta$. Because the stress tensor is isotropic and spatially homogeneous, the numerical flux in the first component of the momentum equation is simply $\widehat{F}_{(\rho v_1)}^i = p_0 \delta^i{}_1$. The right-hand side of Equation (5.37) vanishes because the LG quadrature, with $N \geq 1$, is exact for the radial integral; i.e., $2 Q_N^1[r] = (r_{\mathrm{H}}^2 - r_{\mathrm{L}}^2)$. Similarly, the second component of the momentum equation can be written as

$$\partial_t(\rho v_2)_{\boldsymbol{K}} = -p_0 \frac{|\tilde{\boldsymbol{K}}^2|}{V_{\boldsymbol{K}}} \tilde{\boldsymbol{Q}}_N^2 \left[r^2 \left((\sin\theta_{\mathrm{H}} - \sin\theta_{\mathrm{L}}) - Q_N^2 \left[\partial_{\xi^2}(\sin\theta)_h \right] \right) \right]. \tag{5.38}$$

Since $(\sin\theta)_h$ is approximated by a polynomial of degree $k = N - 1$, the N-point LG quadrature in the θ-direction is evaluated exactly, so that $Q_N^2 \left[\partial_{\xi^2}(\sin\theta)_h \right] = (\sin\theta_{\mathrm{H}} - \sin\theta_{\mathrm{L}})$, which implies that the right-hand side of Equation (5.38) vanishes. Note that these properties hold for polynomial approximations with $k \geq 1$. The first-order accurate scheme ($k = 0$) requires special treatment, and is not discussed here. (See, e.g., Mönchmeyer & Müller (1989) and Blondin & Lufkin (1993), for finite-volume schemes and associated challenges when using spherical-polar coordinates.)

In cylindrical coordinates, the source term in Equation (5.35) contributes only to the first component of the momentum equation. In this case, the equation for the cell-average can be written as

$$\partial_t(\rho v_1)_{\boldsymbol{K}} = -p_0 \frac{|\tilde{\boldsymbol{K}}^1|}{V_{\boldsymbol{K}}} \tilde{\boldsymbol{Q}}_N^1 \left[(R_{\mathrm{H}} - R_{\mathrm{L}}) - Q_N^1 \left[\partial_{\xi^1} R \right] \right]. \tag{5.39}$$

Again, since the quadrature in the R-direction is exact, $Q_N^1 \left[\partial_{\xi^1} R \right] = (R_{\mathrm{H}} - R_{\mathrm{L}})$, and the right-hand side of Equation (5.39) vanishes, as is desired under the conditions of an isotropic and spatially homogeneous stress tensor.

5.4.1.2 Gravitational Source Terms

For the gravitational source terms appearing in the momentum and energy equations, our approach is similar to that used for the geometric sources discussed above. The gravitational

potential in element $\boldsymbol{K}$ is approximated by the polynomial

$$\Phi_h(\boldsymbol{x}) = \sum_{i=1}^{N} \Phi_i \, \hat{\phi}_i(\boldsymbol{x}), \tag{5.40}$$

constrained to be continuous on the element interfaces, so that

$$\Phi_h(x_{\text{L/H}}^{i,+}, \tilde{\boldsymbol{x}}^i) = \Phi_h(x_{\text{L/H}}^{i,-}, \tilde{\boldsymbol{x}}^i). \tag{5.41}$$

(Continuity of the potential on the element interfaces is guaranteed by the finite-element method in Section 5.4.5.) We then compute derivatives of the gravitational potential by analytic differentiation of the expansion in Equation (5.40), and write the momentum and energy sources in the interpolation point $\boldsymbol{x}_{\boldsymbol{p}} \in \boldsymbol{S}_N$ as

$$\left(S_{\rho v_j}^{\Phi}\right)_{\boldsymbol{p}} = -\rho_{\boldsymbol{p}} \, (\partial_j \Phi_h)_{\boldsymbol{p}} \quad \text{and} \quad \left(S_E^{\Phi}\right)_{\boldsymbol{p}} = -\sum_{j=1}^{d} (\rho v^j)_{\boldsymbol{p}} \, (\partial_j \Phi_h)_{\boldsymbol{p}}, \tag{5.42}$$

where, $\rho_{\boldsymbol{p}}$, $(\rho v^j)_{\boldsymbol{p}}$, and $(\partial_j \Phi_h)_{\boldsymbol{p}}$ are, respectively, the mass density, momentum density, and the derivative of Equation (5.40), evaluated in $\boldsymbol{x}_{\boldsymbol{p}}$. We note that the source terms in Equation (5.42) are not well-balanced, i.e. designed specifically to capture steady states (e.g., hydrostatic equilibrium), which would require special treatment (see, e.g., Käppeli & Mishra, 2016; Li & Xing, 2018).

5.4.2 Time Integration

After application of the DG spatial discretization, Equation (5.18) can be viewed as a system of ordinary differential equations (ODEs), which can be written as

$$\frac{d}{dt} \langle \boldsymbol{U}_h, \psi_h \rangle_{\boldsymbol{K}} = \mathcal{B}_h \big(\boldsymbol{U}_h, \Phi_h, \psi_h \big)_{\boldsymbol{K}}. \tag{5.43}$$

This system of ODEs is evolved with the explicit strong stability-preserving Runge-Kutta (SSP-RK) methods of Shu & Osher (1988) (see also Gottlieb et al., 2001; Cockburn, 2001). Denoting the fluid fields and the gravitational potential at time t^n by $\boldsymbol{U}_h^n$ and Φ_h^n, respectively, the time stepping algorithm advancing the solution from t^n to $t^{n+1} = t^n + \Delta t^n$ with s stages is, $\forall \psi_h \in \mathbb{V}_h^k$ and $\forall \boldsymbol{K} \in \mathcal{T}$,

121

$$\langle \boldsymbol{U}_h^{(0)}, \psi_h \rangle_K := \Lambda^{\text{BE}} \left\{ \Lambda^{\text{TVD}} \left\{ \langle \boldsymbol{U}_h^n, \psi_h \rangle_K \right\} \right\}$$

$$\Phi_h^{(0)} \qquad := \Phi_h^n$$

for $i = 1, \ldots, s$ **do**

$$\langle \boldsymbol{U}_h^{(i)}, \psi_h \rangle_K := \Lambda^{\text{BE}} \left\{ \Lambda^{\text{TVD}} \left\{ \sum_{j=0}^{i-1} \alpha_{ij} \left(\langle \boldsymbol{U}_h^{(j)}, \psi_h \rangle_K + \frac{\beta_{ij}}{\alpha_{ij}} \Delta t^n \, \mathcal{B}_h^{(j)} \right) \right\} \right\},$$

$$\text{where } \mathcal{B}_h^{(j)} := \mathcal{B}_h \left(\boldsymbol{U}_h^{(j)}, \Phi_h^{(j)}, \psi_h \right), \text{ with } \Phi_h^{(j)} := \Phi \left(\boldsymbol{U}_h^{(j)} \right)$$

$$\Phi_h^{(i)} := \Phi \left(\boldsymbol{U}_h^{(i)} \right)$$

end

$$\langle \boldsymbol{U}_h^{n+1}, \psi_h \rangle_K := \langle \boldsymbol{U}_h^{(s)}, \psi_h \rangle_K$$

$$\Phi_h^{n+1} \qquad := \Phi_h \left(\boldsymbol{U}_h^{n+1} \right)$$

Algorithm 5.1 Algorithm for SSP-RK time integration.

Note that line 6 in Algorithm 5.1 invokes the Poisson solver for the gravitational potential. Details about the coefficients α_{ij} and β_{ij} can be found in Cockburn (2001). In order to for the evolution of the cell-average of the solution to be stable, the time step must satisfy the Courant–Friedrichs–Lewy (CFL) condition,

$$\Delta t \leq \frac{C_{\text{CFL}}}{d \, (2k + 1)} \times \min_{i \in \{1, \ldots, d\}} \left(\frac{\Delta x^i}{|\lambda^i|} \right), \tag{5.44}$$

where d is the number of spatial dimensions, k is the maximal degree of the one-dimensional polynomials comprising $\mathbb{V}_h^k$, $C_{\text{CFL}} \lesssim 1$ is the CFL number, and λ^i is the largest (in magnitude) eigenvalue of the flux Jacobian in Equation (5.10), corresponding to the fastest-moving wave in the ith spatial dimension.

In principle, one would also need an additional restriction on the time step to guarantee that the solution remains in the set of physically admissible states (see Section 5.4.4). However, we do not enforce such a condition because in practice we find the CFL condition given by Equation (5.44) to be sufficient.

The operators Λ^{TVD} and Λ^{BE} invoked in lines 1 and 4 in Algorithm 5.1 represent slope and bound-enforcing limiters, respectively, and play an important role in RKDG methods. In particular, the slope limiter is required in order for the SSP-RK method to guarantee stability when applied to non-linear problems (Cockburn, 2001).

5.4.3 Slope Limiting

To improve stability of the Runge-Kutta DG (RKDG) algorithm and prevent unphysical oscillations in the solutions around discontinuities, it is necessary to implement a limiting procedure for the polynomial $\mathbf{U}_h$. To this end, we use the basic minmod-type total variation diminishing (TVD) slope limiter (see, e.g., Cockburn & Shu, 1998) in conjunction with the troubled-cell indicator (TCI) proposed by Fu & Shu (2017). The TCI prevents excessive limiting by only flagging elements where limiting is needed. When using the basic TVD limiter one assumes that any spurious oscillations are evident in the part of the solution that is represented by piecewise linear functions, and under- and over-shoots of the higher-order solution at inter-cell boundaries are detected by comparing local slopes with slopes constructed using cell averages of the target cell and its neighbors. Our implementation follows closely the description in Schaal et al. (2015) for the case of an ideal EoS. Recall from Eq. (5.17) that in each cell the solution is expressed in the nodal form. It is convenient, however, for limiting purposes to express the solution in $\mathbf{K}$ using a modal representation

$$\mathbf{U}_h(\mathbf{x}, t) = \sum_{i=1}^{N} C_i(t)\, \tilde{\phi}_i(\mathbf{x}), \tag{5.45}$$

where the multidimensional modal basis functions $\tilde{\phi}_i(\mathbf{x}(\boldsymbol{\xi})) \in \mathbb{V}_h^k$ are constructed from one-dimensional Legendre polynomials $\{P_\ell(\xi^i)\}_{\ell=0}^{N-1}$ by tensorization; i.e.,

$$\tilde{\phi}_i(\boldsymbol{\xi}) = \tilde{\phi}_{\{i_1,\ldots,i_d\}}(\xi^1,\ldots,\xi^d) = P_{i_1-1}(\xi^1) \times \ldots \times P_{i_d-1}(\xi^d). \tag{5.46}$$

The Legendre polynomials are orthogonal on the unit interval I^i, and in THORNADO we use a normalization such that $P_0(\xi^i) = 1$ and $P_1(\xi^i) = \xi^i$ (i.e., the polynomials are orthogonal, but not orthonormal with respect to the standard L^2 inner product on I^i). Note that the case with $i = \{i_1, \ldots, i_d\} = \{1, \ldots, 1\} = \mathbf{1}$ corresponds to $\tilde{\phi}_{\mathbf{1}}(\mathbf{x}) = P_0(\xi^1) \times \ldots \times P_0(\xi^d) = 1$; therefore, the expansion coefficient C_1 is equal to the cell average when Cartesian coordinates are used ($\gamma_h = 1$); i.e.,

$$C_1 = \frac{1}{|\mathbf{K}|} \int_K \mathbf{U}_h \, d\mathbf{x}. \tag{5.47}$$

In our multi-index notation we define $|i| = \sum_{j=1}^{d} i_j$, so that the basis functions $\tilde{\phi}_i$ with i satisfying $|i| = d + 1$ are linear in one of the coordinates. For example, for the three-dimensional case ($d = 3$)

we have exactly three basis functions satisfying $|\boldsymbol{i}| = d + 1 = 4$

$$\tilde{\phi}_{\{2,1,1\}}(\boldsymbol{x}) = P_1(\xi^1) \times P_0(\xi^2) \times P_0(\xi^3) = P_1(\xi^1) = \xi^1, \tag{5.48}$$

$$\tilde{\phi}_{\{1,2,1\}}(\boldsymbol{x}) = P_0(\xi^1) \times P_1(\xi^2) \times P_0(\xi^3) = P_1(\xi^2) = \xi^2, \quad \text{and} \tag{5.49}$$

$$\tilde{\phi}_{\{1,1,2\}}(\boldsymbol{x}) = P_0(\xi^1) \times P_0(\xi^2) \times P_1(\xi^3) = P_1(\xi^3) = \xi^3, \tag{5.50}$$

which are linear in the reference coordinates ξ^1, ξ^2, and ξ^3, respectively. From orthogonality of the Legendre polynomials, we can identify the expansion coefficients satisfying $|\boldsymbol{i}| = 4$ in the modal representation in Equation (5.45) as the average derivative of $\boldsymbol{U}_h$ with respect to the reference coordinates ξ^1, ξ^2, and ξ^3, respectively; i.e.,

$$\boldsymbol{C}_{\{2,1,1\}} = \frac{1}{|\boldsymbol{K}|} \int_K (\partial_{\xi^1} \boldsymbol{U}_h) \, d\boldsymbol{x}, \quad \boldsymbol{C}_{\{1,2,1\}} = \frac{1}{|\boldsymbol{K}|} \int_K (\partial_{\xi^2} \boldsymbol{U}_h) \, d\boldsymbol{x}, \quad \text{and} \quad \boldsymbol{C}_{\{1,1,2\}} = \frac{1}{|\boldsymbol{K}|} \int_K (\partial_{\xi^3} \boldsymbol{U}_h) \, d\boldsymbol{x}. \tag{5.51}$$

These coefficients are here obtained by taking the derivative of Equation (5.45) with respect to ξ^1, ξ^2, and ξ^3, respectively, and integrating over the element.

We demand the representations of the solution in Equations (5.17) and (5.45) be equivalent in the least squares sense

$$\sum_{i=1}^{N} \int_K \left(\boldsymbol{U}_i(t) \, \phi_i(\boldsymbol{x}) - \boldsymbol{C}_i(t) \, \tilde{\phi}_i(\boldsymbol{x}) \right) \psi_h(\boldsymbol{x}) \, d\boldsymbol{x} = 0, \quad \forall \psi_h \in \mathbb{V}_h^k, \tag{5.52}$$

which provides a change of basis between Lagrange and Legendre polynomial representations, and relates the coefficients of nodal and modal representations by linear transformations. Setting $\psi_h = \phi_j$ in Equation (5.52) gives the nodal coefficients in terms of the modal coefficients

$$\boldsymbol{U}_j = \sum_{i=1}^{N} \tilde{\phi}_i(\boldsymbol{\xi}_j) \, \boldsymbol{C}_i, \tag{5.53}$$

while setting $\psi_h = \tilde{\phi}_j$ in Equation (5.52) gives the modal coefficients in terms of the nodal coefficients

$$\sum_{i=1}^{N} \int_I \tilde{\phi}_j(\boldsymbol{\xi}) \, \tilde{\phi}_i(\boldsymbol{\xi}) \, d\boldsymbol{\xi} \, \boldsymbol{C}_i = \sum_{i=1}^{N} \int_I \tilde{\phi}_j(\boldsymbol{\xi}) \, \phi_i(\boldsymbol{\xi}) \, d\boldsymbol{\xi} \, \boldsymbol{U}_i, \tag{5.54}$$

where the matrix on the left-hand side is diagonal and easily invertible. The matrix on the right-hand side is the same for all elements, and can be precomputed at program startup and stored with

minimal storage requirements. As illustrated in Equations (5.47) and (5.51), the representation in terms of Legendre polynomials $\{P_\ell\}_{\ell=0}^{N-1}$ is more convenient for limiting because the polynomial degree increases with increasing ℓ, and the identification of the expansion coefficients with average values and average derivatives is more straightforward. In the Lagrange basis, all the basis functions have the same polynomial degree.

We perform slope limiting by comparing the weights C_i — which for $|i| = d + 1$ and appropriate normalization of the Legendre polynomials are equal to the first derivatives of the solution in the cell — with the limited weights $\widetilde{C}_i$, computed from

$$\mathcal{M}\,\widetilde{C}_i = \text{minmod}\big(\,\mathcal{M}\,C_i,\ \beta_{\text{TVD}}\,\mathcal{M}\,(C_1^+ - C_1),\ \beta_{\text{TVD}}\,\mathcal{M}\,(C_1 - C_1^-)\,\big), \quad (\forall i \text{ satisfying } |i| = d + 1) \tag{5.55}$$

where the multivariate minmod function is defined as

$$\text{minmod}\big(a_1,\ a_2,\ a_3\big) = \begin{cases} s \times \min\{\,|a_1|, |a_2|, |a_3|\,\}, & \text{if } s = \text{sign}(a_1) = \text{sign}(a_2) = \text{sign}(a_3) \\ 0, & \text{otherwise.} \end{cases} \tag{5.56}$$

The minmod function returns the minimum argument if they all have the same sign, and zero otherwise. In three spatial dimensions we estimate limited slopes independently for all the coefficients in Equation (5.51), and limiting is applied to a component of U_h whenever the corresponding linear coefficient in the modal expansion in Equation (5.45) exceeds a given threshold value. Here we apply slope limiting when $|\widetilde{C}_i - C_i| > 10^{-6}\,C_i$, for any i satisfying $|i| = d + 1$. (C_i and $\widetilde{C}_i$ are arbitrary components of the vectors C_i and $\widetilde{C}_i$, respectively.) In Equation (5.55), the parameter β_{TVD} takes values in the closed interval $[1, 2]$, and determines how aggressively to apply limiting. The minimal β_{TVD} corresponds to a total variation diminishing scheme, which is more dissipative than a scheme with the maximal β_{TVD}, which is potentially more oscillatory. Increasing β_{TVD} puts more weight on the neighboring cell averages, making the minmod function more likely to set $\widetilde{C}_i = C_i$, which results in no limiting being applied. The superscripts $-/+$ on the C_1 coefficients in the minmod function in Equation (5.55) indicate that the coefficient belongs to the expansion in the previous/next element in the coordinate direction of the slope to be limited. Figure 5.2 illustrates

how the minmod limiter works in the one-dimensional case when applied to a scalar field $U(x)$. The transformation matrix $\mathcal{M}$ is included in Equation (5.55) to allow for limiting in characteristic fields (see discussion below). For *component-wise limiting*, $\mathcal{M}$ is set to the identity matrix. Thus, when slope limiting is applied, the local solution is truncated as

$$U_h(\boldsymbol{x}, t) := \widetilde{U}_h(\boldsymbol{x}, t) = \sum_{\substack{i=1 \\ |i| \leq d+1}}^{N} \widetilde{C}_i(t)\, \tilde{\phi}_i(\boldsymbol{x}), \tag{5.57}$$

where $\widetilde{C}_i = 0$ for all i with $|i| > d + 1$, and

$$\widetilde{C}_1 := U_K - \frac{1}{V_K} \sum_{\substack{i=1 \\ |i|=d+1}}^{N} \int_K \tilde{\phi}_i(\boldsymbol{x})\, dV_h\, \widetilde{C}_i. \tag{5.58}$$

Thus, the minmod limiter reduces the local polynomial degree to at most $k = 1$. If the arguments in the minmod function in Equation (5.55) have different signs, the minmod limiter further reduces the polynomial degree to $k = 0$. Because of this, we use the TCI as discussed below. Although not considered for THORNADO yet, we note that it is possible to generalize or improve the limiting strategy to maintain higher order of accuracy; see e.g., Biswas et al. (1994); Krivodonova (2007); Dumbser et al. (2014).

The readjustment of $\widetilde{C}_1$ in Equation (5.58), which occurs after computing the limited slopes in Equation (5.55), is necessary to preserve the cell average as defined in Equation (5.31), and is due to the use of curvilinear coordinates (see also related discussion by Radice & Rezzolla, 2011, their Section C1). Preservation of the cell average in the limiting procedure is needed, e.g., to conserve mass. Without the 'conservative correction' in Equation (5.58), the limiter preserves the cell average defined in Equation (5.47), which is undesirable in curvilinear coordinates. Note that the second term on the right-hand side of Equation (5.58) vanishes in Cartesian coordinates because of orthogonality of the Legendre polynomials. However, in curvilinear coordinates, this term does not vanish since the Legendre polynomials are not orthogonal with respect to the inner product weighted by $\sqrt{\gamma_h}$. In practice, we have found that the conservative correction is small, but necessary to maintain conservation to machine precision.
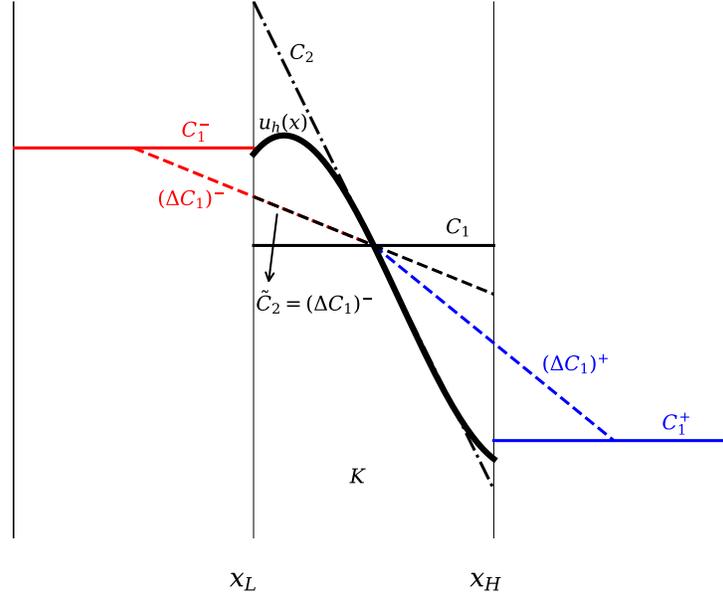
Figure 5.2 Illustration of how the minmod slope limiter works when applied to a one-dimensional, scalar field $U(x)$. The original, high-order polynomial $U_h(x) = \sum_{i=1}^{N} C_i \tilde{\phi}_i(x)$ is represented by the thick solid black curve, while its constant and linear contributions, $C_1 \tilde{\phi}_1(x)$ and $C_2 \tilde{\phi}_2(x)$, are represented by solid and dash-dot black lines, respectively. The slopes $(\Delta C_1)^+ = C_1^+ - C_1$ and $(\Delta C_1)^- = C_1 - C_1^-$ — the second and third argument in the minmod function in Equation (5.55), respectively — are represented by the blue and red dashed lines, respectively. In this example, all three slopes have the same sign. Then, since $(\Delta C_1)^- < (\Delta C_1)^+ < C_1$, $\widetilde{C}_2 := (\Delta C_1)^-$.

We note that, in order to improve the evolution of the electron fraction, $Y_e = D_e/\rho$, we also apply the minmod limiter directly to the electron fraction, and enforce limiting of both ρ_h and $D_{e,h}$ whenever oscillations in Y_e are detected by the minmod function.

In order to determine where slope limiting is necessary, we use the TCI of Fu & Shu (2017) to prevent excessive limiting. For example, it is well-known that the minmod limiter is overly diffusive around smooth extrema, where $\widetilde{C}_i = 0$, which kills off all the high-order accuracy. We note in passing that other TCIs have been proposed (see, e.g., Qiu & Shu, 2005), but we have chosen the one by Fu & Shu (2017) for its relative ease of implementation. This TCI is based on the function

$$I_K(G_h) = \frac{\sum_j |G_K - G_K^{(j)}|}{\max(\max_j |G_{K^{(j)}}^{(j)}|, |G_K|)}, \tag{5.59}$$

where $G_h \in \boldsymbol{G}_h \subseteq \boldsymbol{U}_h$ is in the subset of fields used to detect troubled cells. In Equation (5.59), the

127

sum in the numerator is taken over all the neighboring elements $K^{(j)}$ sharing a boundary with the target element K, while the max in the denominator is taken over neighboring elements $K^{(j)}$ and the target element K. The cell average of G_h in K is denoted G_K, and is here given by the right-hand side of Equation (5.47) — i.e., without the weighting factor $\sqrt{\gamma_h}$ used in the proper definition of the cell average in Equation (5.31). Computed in the same way, $G_K^{(j)}$ is the corresponding cell average computed by extrapolating the polynomial representation from the neighboring elements $K^{(j)}$ into the target K, and $G_{K^{(j)}}^{(j)}$ is the cell average native to the neighbor element $K^{(j)}$. An illustration of the troubled-cell indicator is given in Figure 5.3 for the one-dimensional case applied to a single field $G(x)$.

An element is flagged for limiting if, for any $G_h \in \boldsymbol{G}_h$, $I_K(G_h) > C_{\text{TCI}}(G)$, where $C_{\text{TCI}}(G)$ is a user-defined threshold, which can be set differently for each G. In the numerical results presented in Section 5.5, we use the mass density, fluid energy, and electron fraction as the variables to detect troubled cells; i.e., $\boldsymbol{G} = (\rho, E, \text{Y}_{\text{e}})^T$.

When solving a system of hyperbolic conservation laws, experience has shown that the slope limiting described above is more efficient when performed on the so-called 'characteristic variables', as opposed to the conserved variables $\mathbf{U}_h$ (see, e.g., Cockburn & Shu, 1998, for a description). Because the Euler equations form a system of hyperbolic partial differential equations (see, e.g., LeVeque, 1992), the flux Jacobian in Equation (5.10) can be decomposed as

$$\frac{\partial \mathbf{F}^i(\mathbf{U})}{\partial \mathbf{U}} = \mathcal{R}^i \, \Lambda^i \, (\mathcal{R}^i)^{-1} \quad (i = 1, \dots, d), \tag{5.60}$$

where the columns of the 6×6 matrix $\mathcal{R}^i$ contain the right eigenvectors of the flux Jacobian, the rows of $(\mathcal{R}^i)^{-1}$ contain the left eigenvectors, and Λ^i is a diagonal matrix containing the eigenvalues of the flux Jacobian. For hyperbolic systems, the eigenvalues are real and the eigenvectors form a complete set (see e.g., LeVeque, 1992). At this point, we introduce the characteristic variable $\mathbf{w} = \mathcal{R}^{-1}\mathbf{U}$. Recall in Equation (5.55) that we introduced the transformation matrix $\mathcal{M}$. If we let $\mathcal{M} = (\mathcal{R}^i)^{-1}$, limiting is performed on the characteristic variables. (For linear systems, the characteristic variables evolve independently, and limiting of one characteristic variable does not affect the others.) Once $\mathcal{M}\widetilde{\boldsymbol{C}_i}$ is estimated in the characteristic variables as in Equation (5.55), the
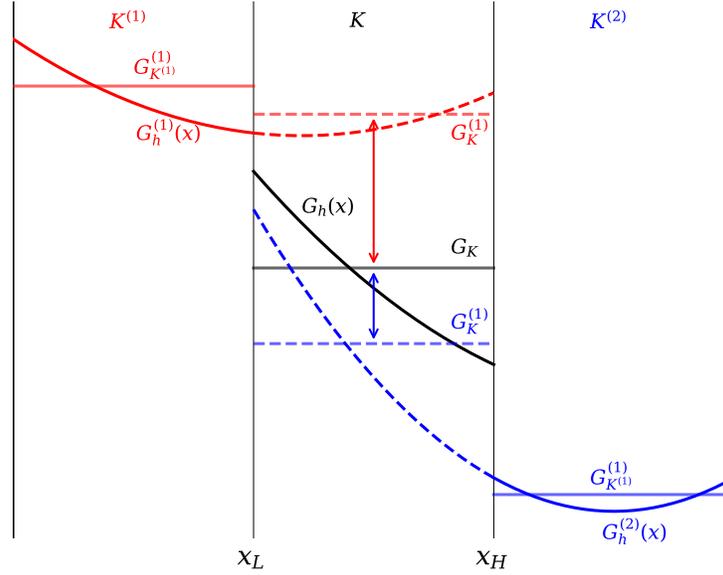
Figure 5.3 Illustration of how the troubled-cell indicator works in the one-dimensional case on a scalar field $G(x)$, to determine if limiting is needed in the target element K, where the polynomial representation is given by $G_h(x)$ (solid black curve), and the cell average is G_K (solid gray line). The polynomial representation in the left element, $K^{(1)}$, is given by $G_h^{(1)}(x)$ (solid red curve), with cell average $G_{K^{(1)}}^{(1)}$ (solid light red line). Similarly, the polynomial representation in the right element, $K^{(2)}$, is given by $G_h^{(2)}(x)$ (solid blue curve), with cell average $G_{K^{(2)}}^{(2)}$ (solid light blue line). The extrapolations of $G_h^{(1)}(x)$ and $G_h^{(2)}(x)$ into the target element are given by the dashed red and blue curves, respectively. Finally, the cell averages of the extrapolations of $G_h^{(1)}(x)$ and $G_h^{(2)}(x)$, computed over the target cell, are denoted $G_K^{(1)}$ (dashed light red line) and $G_K^{(2)}$ (dashed light blue line), respectively. The element is flagged for limiting if the difference in the cell averages, $|G_K - G_K^{(2)}|$ and/or $|G_K - G_K^{(2)}|$, becomes too large.

limited slopes in the conserved variables are obtained by left multiplication with $\mathcal{M}^{-1}$ (see e.g., Cockburn & Shu, 1998; Schaal et al., 2015), and the limiting process proceeds as in Equations (5.57) and (5.58). It should be noted that $\mathcal{R}^i$ and $(\mathcal{R}^i)^{-1}$ are computed using cell averages of the conserved and metric variables.

While this process of characteristic limiting has been done for an ideal EoS, and shown (e.g., Schaal et al., 2015) to give superior results when compared to component-wise limiting (especially for the high-order case; $k \geq 1$), the extension to the tabulated nuclear EoS case is nontrivial. The

reasons for this are (1) the increased complexity and dimensionality of the system due to the added electron conservation equation in Equation (5.4), and (2) the additional care that must be taken when computing the thermodynamic derivatives associated with the flux Jacobian. In the case of an ideal, or other simplified EoS, the necessary thermodynamic derivatives (such as derivatives of pressure) are analytically defined. For a nuclear EoS, the derivatives do not have analytic expressions and the necessary eigenvectors must be constructed generally. We provide the characteristic decomposition of the flux Jacobian for the Euler system with a nuclear EoS in Appendix 9C.

5.4.4 Bound-Enforcing Limiting

When solving the Euler equations of gas dynamics with an ideal EoS, the mass density ρ and pressure p (or, equivalently, internal energy density e) must remain positive. However, this property is not guaranteed by the basic DG method, which encourages the use of a more advanced procedure (Zhang & Shu, 2010). The internal energy density is given in terms of the conserved quantities as

$$e(\boldsymbol{U}) = E - \frac{m^2}{2\rho}, \tag{5.61}$$

where $m^2 = m_j m^j$, E is the fluid energy density, and $m_j = \rho v_j$ are the components of the momentum density. For the ideal EoS case, the set of physically admissible states is given by

$$\tilde{\mathcal{G}} = \left\{ \boldsymbol{U} = (\rho, \boldsymbol{m}, E)^{\mathrm{T}} \,|\, \rho > 0 \text{ and } e(\boldsymbol{U}) > 0 \right\}. \tag{5.62}$$

If mass density is positive, the internal energy density is a concave function of $\boldsymbol{U}$, and $\tilde{\mathcal{G}}$ is a convex set (Zhang & Shu, 2010). For many EoSs (including the ideal EoS), where the pressure only depends on the mass density and internal energy density, $\boldsymbol{U}$ must remain in $\tilde{\mathcal{G}}$ as defined in Equation (5.62), otherwise the initial value problem is ill-posed. To maintain $\boldsymbol{U} \in \tilde{\mathcal{G}}$, the combination of a suitable time step restriction, a strong stability-preserving time integrator, and a bound-enforcing limiter is used (e.g., Zhang & Shu, 2010). The time step restriction is derived as a sufficient condition to ensure that the updated *cell average* satisfies $\boldsymbol{U}_K^{n+1} \in \tilde{\mathcal{G}}$, and requires $\boldsymbol{U}_h^n \in \tilde{\mathcal{G}}$ *point-wise* within each element, while the limiter, which relies on $\boldsymbol{U}_K^{n+1} \in \tilde{\mathcal{G}}$ *and* the convexity of $\tilde{\mathcal{G}}$, is used to again enforce point-wise $\boldsymbol{U}_h^{n+1} \in \tilde{\mathcal{G}}$ within each element. (We do not attempt to derive a sufficient time step restriction for the present setting in this paper, and simply use the condition in Equation (5.44).)

We note here that for two arbitrary elements $U_a, U_b \in \tilde{\mathcal{G}}$, since the set $\tilde{\mathcal{G}}$ is convex, the convex combination $U_c := \vartheta\, U_a + (1 - \vartheta)\, U_b$, where $\vartheta \in [0, 1]$, is also in $\tilde{\mathcal{G}}$; i.e., $U_c \in \tilde{\mathcal{G}}$. Moreover, $e(U)$ in Equation (5.61) is concave since Jensen's inequality — $e(U_c) \geq \vartheta\, e(U_a) + (1 - \vartheta)\, e(U_b)$ — holds. The property of convex combinations is commonly used to design constraint-preserving numerical methods for systems where — for physical reasons — the dynamics is constrained to a convex set (see, e.g., Xing et al., 2010; Olbrant et al., 2012; Wu & Tang, 2015; Endeve et al., 2015; Chu et al., 2019, for examples beyond the non-relativistic Euler equations with an ideal EoS).

To maintain physically admissible states in the present setting with THORNADO, we draw inspiration from the limiting strategy proposed for an ideal EoS by Zhang & Shu (2010), which we have modified to work satisfactorily with a tabulated nuclear EoS. Specifically, thermodynamic quantities, including the specific internal energy $\epsilon = e/\rho$, are tabulated in terms of mass density, temperature, and electron fraction, which cover finite extents; i.e., $\rho \in [\rho_{\min}, \rho_{\max}]$, $T \in [T_{\min}, T_{\max}]$, and $Y_e \in [Y_{e,\min}, Y_{e,\max}]$. We use some of the table bounds to define the set of admissible states as

$$\mathcal{G} = \left\{ U = (\rho, \boldsymbol{m}, E, D_e)^{\mathrm{T}} \,\middle|\, (\rho, D_e)^{\mathrm{T}} \in \mathcal{G}_{\boldsymbol{u}} \text{ and } \epsilon(U) \geq \epsilon_{\min} \equiv \epsilon(\rho, T_{\min}, Y_e) \right\}, \qquad (5.63)$$

where we have defined the subset

$$\mathcal{G}_{\boldsymbol{u}} = \left\{ \boldsymbol{u} = (\rho, D_e)^{\mathrm{T}} \,\middle|\, \rho_{\min} \leq \rho \leq \rho_{\max}, \; D_e > 0, \text{ and } Y_{e,\min} \leq Y_e(\boldsymbol{u}) \leq Y_{e,\max} \right\}, \qquad (5.64)$$

and seek to maintain $U_h \in \mathcal{G}$.

First, we note that it is straightforward to show that the subset $\mathcal{G}_{\boldsymbol{u}}$ is convex. To do this, it is sufficient to show that a convex combination of two arbitrary elements of $\mathcal{G}_{\boldsymbol{u}}$ also belongs to $\mathcal{G}_{\boldsymbol{u}}$. To this end, let $\boldsymbol{u}_a \equiv (\rho_a, D_{e,a})^{\mathrm{T}} \in \mathcal{G}_{\boldsymbol{u}}, \boldsymbol{u}_b \equiv (\rho_b, D_{e,b}) \in \mathcal{G}_{\boldsymbol{u}}$, and define the convex combination $\boldsymbol{u}_c = \vartheta\, \boldsymbol{u}_a + (1 - \vartheta)\, \boldsymbol{u}_b$, where $\vartheta \in [0, 1]$. Then the first component of $\boldsymbol{u}_c$ is $\rho_c = \vartheta\, \rho_a + (1 - \vartheta)\, \rho_b$. Since, by assumption, $\rho_a, \rho_b \in [\rho_{\min}, \rho_{\max}]$ and $\vartheta \in [0, 1]$, it follows that $\rho_c \in [\rho_{\min}, \rho_{\max}]$. Similarly, the second component of $\boldsymbol{u}_c$ is $D_{e,c} = \vartheta\, D_{e,a} + (1 - \vartheta)\, D_{e,b}$. Then, since $D_{e,a}, D_{e,b} > 0$,

it follows that $D_{e,c} > 0$. Finally, we can write

$$Y_e(\boldsymbol{u}_c) = \frac{D_{e,c}}{\rho_c} = \frac{\vartheta\, D_{e,a} + (1-\vartheta)\, D_{e,b}}{\vartheta\, \rho_a + (1-\vartheta)\, \rho_b} = \alpha\, \frac{D_{e,a}}{\rho_a} + (1-\alpha)\, \frac{D_{e,b}}{\rho_b} = \alpha\, Y_e(\boldsymbol{u}_a) + (1-\alpha)\, Y_e(\boldsymbol{u}_b),$$
(5.65)

where

$$\alpha = \frac{\vartheta\, \rho_a}{\vartheta\, \rho_a + (1-\vartheta)\, \rho_b}.$$
(5.66)

Since $\rho_a, \rho_b \geq \rho_{\min} > 0$ and $\vartheta \in [0, 1]$, we have $\alpha \geq 0$. We also have $\alpha \leq 1$. Therefore, $\alpha \in [0, 1]$, which implies $Y_e(\boldsymbol{u}_c) \in [Y_{e,\min}, Y_{e,\max}]$ and $\boldsymbol{u}_c \in \mathcal{G}_{\boldsymbol{u}}$. Thus, the subset $\mathcal{G}_{\boldsymbol{u}}$ is convex.

While, strictly speaking, the Euler equations in Section 7.2 are valid for any mass density $\rho > 0$, we note that there are physical reasons for maintaining the mass density within the finite table bounds, which are $\rho_{\min} \approx 1.66 \times 10^3$ g cm^{-3} and $\rho_{\max} \approx 3.16 \times 10^{15}$ g cm^{-3} for the tables used in this paper. Indeed, in CCSN simulations, it is possible for the cell averaged mass density to evolve outside these limits, which would require extending the table bounds. However, when the mass density approaches the upper bound, a relativistic description should be adopted, and when the mass density approaches the lower bound, the nuclear EoS adopted here is invalid because the matter is not in nuclear statistical equilibrium. These bounds must, however, also be enforced to avoid algorithm failure. For the purpose of the bound-enforcing limiter, the finite bounds on the mass density in Equation (5.63) are included in case the bounds are violated for certain points within an element, e.g., in the vicinity of a shock, while the cell averaged mass density is still inside the table bounds. (The limiter developed here will not work if the cell averaged mass density exceeds the table bounds.) We have also equipped the set of admissible states $\mathcal{G}$ with the bounds $Y_e \in [Y_{e,\min}, Y_{e,\max}] \subseteq [0, 1]$, which are also required to avoid algorithm failure. (In this work, $Y_{e,\min} = 0.01$ and $Y_{e,\max} = 0.7$). We note, however, that for the test problems in Section 5.5 and the application in Section 5.6, we did not encounter a situation in which the mass density or the electron fraction exceeded their respective table bounds.

On the other hand, a complication that frequently arises in gravitational collapse simulations is that the specific internal energy falls below the minimum tabulated value (i.e., $\epsilon < \epsilon_{\min}$) — especially around core bounce and shock formation, which we discuss in further detail in Section 5.6. When

this happens, the EoS is not invertible for the temperature when given the state vector $(\rho, \epsilon, Y_e)^T$, and the algorithm fails since the temperature is needed to compute the pressure as well as other thermodynamic quantities. It is not feasible to merely generate tables with lower T_{min}, since — particularly for high mass densities — the specific internal energy does not tend to zero as $T \to 0$ due to the degeneracy (or zero temperature) contribution to the internal energy, as can be seen in Figure 5.4. In CCSN simulations, where the iron core is degenerate at the onset of collapse, the initial specific internal energy is already close to the minimum value. Then, around core bounce and shock formation, where steep gradients in the evolved fields form, conditions with $\boldsymbol{U}_h \notin \mathcal{G}$ can easily arise within certain elements, and a limiting strategy is needed. Fortunately, we have observed that $\boldsymbol{U}_K \in \mathcal{G}$ is always satisfied (although we do not seek to establish sufficient conditions to guarantee this here). This allows us to pursue the limiting strategy proposed by Zhang & Shu (2010), which we detail below.

There is, however, an additional complication that may cause the limiting strategy of Zhang & Shu (2010) to fail: the surface of specific internal energy at the minimum temperature T_{min} — that is $\epsilon_{min}(\rho, Y_e) \equiv \epsilon(\rho, T_{min}, Y_e)$ — is not globally convex in the sense that the second derivatives $(\partial^2 \epsilon_{min}/\partial \rho^2)_{Y_e}$ and $(\partial^2 \epsilon_{min}/\partial Y_e^2)_\rho$ are not strictly positive everywhere, which implies that the set $\mathcal{G}$ in Equation (5.63) is not strictly convex. We illustrate this in Figure 5.5, which shows $\epsilon(\rho, T_{min}, Y_e)$ as a function of ρ and Y_e for the SFHo EoS (Steiner et al., 2013b). Therefore, adopting the limiting procedure from the ideal EoS case to enforce $\boldsymbol{U}_h \in \mathcal{G}$ — even if $\boldsymbol{U}_K \in \mathcal{G}$ — can compromise the robustness of the limiter. The reason is that the amount of limiting applied to the polynomial $\boldsymbol{U}_h$ is determined by finding the intersection point of the boundary of $\mathcal{G}$ and the straight line connecting the cell average $\boldsymbol{U}_K$ and a non-physical point value $\boldsymbol{U}_q \notin \mathcal{G}$. If $\mathcal{G}$ is not convex, there may be multiple intersection points, which can cause the limiter to fail. However, the issue of globally non-convex $\mathcal{G}$ is avoided if the limiter is only activated in regions for which $\mathcal{G}$ is locally convex. That is, for the elements that require limiting, the cell average $\boldsymbol{U}_K$ and the DG solution $\boldsymbol{U}_h$, evaluated in the required quadrature points within each element K, are in a locally convex region and sufficiently close to each other in $\mathcal{G}$. The latter is typically the case in regions
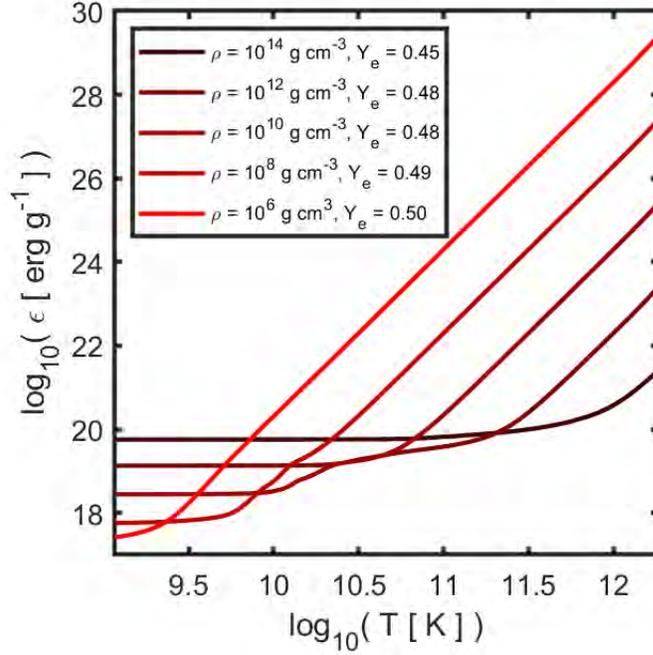
Figure 5.4 Relationship between specific internal energy and temperature from the SFHo EoS for select values of mass density and electron fraction. (Note that the electron fraction is only sampled from the narrow range encountered in the adiabatic simulations discussed in Section 5.6.) Due to degeneracy, the specific internal energy for all profiles demonstrates asymptotic behavior for low temperatures. Thus, the lower boundary on ϵ would not change if the table was reconstructed with a lower temperature limit.

of the flow characterized by small gradients, but may not be the case in the vicinity of a shock. Fortunately, as discussed further in Section 5.6, we do not encounter any situations in which the non-convexity of $\mathcal{G}$ causes the limiter to fail, but this needs to be further investigated in the context of multidimensional models with higher physical fidelity (i.e., models that include neutrino transport), which sample a larger part of the EoS than the simulations discussed in this paper.

The bound-enforcing limiter is completely local to each element, and can thus be discussed in terms of a single element K. As in (Zhang & Shu, 2010), we define a point set S^+, which includes the volumetric nodal points in an element K, as well as the points on the interface of K. For the two-dimensional case with $k = 2$, the point set is given by the union of all the points displayed in the right panel in Figure 5.1. Thus, S^+ comprises the points where U_h is evaluated to construct the update for each U_p in Equation (5.30). Using Equation (5.17), the solution is evaluated at all

the points $x_q \in S^+$, and limiting is applied if, for any point $x_q \in S^+$, $U_q \equiv U_h(x_q) \notin \mathcal{G}$. The step-by-step procedure for bound-enforcing limiting is described next, where it is assumed that the cell average satisfies $U_K \in \mathcal{G}$.

5.4.4.1 Step 1: Mass Density and Electron Density

The first step is to enforce $\rho_q \in [\rho_{\min}, \rho_{\max}]$ and $D_{e,q} \geq \delta_{D_e} > 0$ for all $x_q \in S^+$, where δ_{D_e} is arbitrarily small. (The bound $D_{e,q} > 0$ is needed in Step 2 below.) Following Zhang & Shu (2010), we use the linear scaling limiter from Liu & Osher (1996), and replace the polynomial $u_h(x) = \left(\rho_h(x), D_{e,h}(x)\right)^{\mathrm{T}}$ with the limited polynomial

$$u_h^{(1)}(x) := (1 - \vartheta_1)\, u_K + \vartheta_1\, u_h(x) \quad (\vartheta_1 \in [0, 1]), \tag{5.67}$$

where the limiter parameter $\vartheta_1 \in [0, 1]$ is found by a simple backtracing algorithm. Specifically, for any point $x_q \in S^+$ with $u_q = u_h(x_q) \notin \mathcal{G}_u$, we start with $\vartheta_{1,q} = 1$, which is recursively reduced (by 5%) until

$$u_q^{(1)} = (1 - \vartheta_{1,q})\, u_K + \vartheta_{1,q}\, u_q \in \mathcal{G}_u. \tag{5.68}$$

(In practice, to reduce the number of iterations, we set $\vartheta_{1,q} = 0$ whenever the backtracing algorithm has brought the value below 0.01.) We then set $\vartheta_1 := \min_q \vartheta_{1,q}$, where the minimum is taken over all the points within the element where u_h was found to violate the bounds associated with Step 1. The limiter in Equation (5.67) simply scales u_h as evaluated in the points within the element towards the cell average, and the value for ϑ_1 is determined in order to scale the solution in the points just enough to ensure that the bounds are satisfied for all $x_q \in S^+$. In the worst case scenario, $\vartheta_1 = 0$, and the DG solution is set equal to the cell average everywhere within the element. Note that this step is conservative and does not change the cell averages; i.e., $u_K^{(1)} = u_K$. Also note that if the bounds on the mass density and electron density are not violated, then $\vartheta_1 = 1$ and $u_h^{(1)}(x) = u_h(x)$.

5.4.4.2 Step 2: Electron Fraction

In the second step, we enforce $Y_{e,q} \equiv D_{e,h}(x_q)/\rho_h(x_q) \in [Y_{e,\min}, Y_{e,\max}]$ for all $x_q \in S^+$. To do this, we follow a procedure similar to the previous step, and replace $u_h^{(1)}(x)$ with the limited

polynomial

$$\boldsymbol{u}_h^{(2)}(\boldsymbol{x}) := (1 - \vartheta_2)\,\boldsymbol{u}_K + \vartheta_2\,\boldsymbol{u}_h^{(1)}(\boldsymbol{x}) \quad (\vartheta_2 \in [0,1]), \tag{5.69}$$

where

$$\vartheta_2 = \frac{\alpha\,\rho_K}{\alpha\,\rho_K + (1-\alpha)\,\rho_\alpha^{(1)}} \quad \text{and} \quad \alpha = \min\left\{1, \left|\frac{Y_{e,\min} - Y_{e,K}}{m_{Y_e} - Y_{e,K}}\right|, \left|\frac{Y_{e,\max} - Y_{e,K}}{M_{Y_e} - Y_{e,K}}\right|\right\}, \tag{5.70}$$

and where we have defined

$$M_{Y_e} = \max_{\boldsymbol{x} \in S^+} Y_e\big(\boldsymbol{u}_h^{(1)}(\boldsymbol{x})\big), \quad m_{Y_e} = \min_{\boldsymbol{x} \in S^+} Y_e\big(\boldsymbol{u}_h^{(1)}(\boldsymbol{x})\big), \quad \text{and} \quad Y_{e,K} = D_{e,K}/\rho_K. \tag{5.71}$$

and with the cell average for mass density and electron density computed according to the definition in Equation (5.31); i.e.

$$\rho_K = \frac{\sum_{p=1}^N w_p\,\sqrt{\gamma_p}\,\rho_p}{\sum_{p=1}^N w_p\,\sqrt{\gamma_p}} \quad \text{and} \quad D_{e,K} = \frac{\sum_{p=1}^N w_p\,\sqrt{\gamma_p}\,D_{e,p}}{\sum_{p=1}^N w_p\,\sqrt{\gamma_p}}, \tag{5.72}$$

respectively. In the expression for ϑ_2 in Equation (5.70), we simply set $\rho_\alpha^{(1)} = \max_{\boldsymbol{x} \in S^+} \rho_h^{(1)}(\boldsymbol{x})$, which is sufficient, but may not give the optimal value for ϑ_2 (i.e., this choice may not give the largest ϑ_2 while still maintaining $\boldsymbol{u}_h^{(2)} \in \mathcal{G}_{\boldsymbol{u}}$).

Step 2 is also conservative and does not change the cell averages; i.e., $\boldsymbol{u}_K^{(2)} = \boldsymbol{u}_K^{(1)} = \boldsymbol{u}_K$. Also, if the bounds on the electron fraction are not violated, $\vartheta_2 = \alpha = 1$ and $\boldsymbol{u}_h^{(2)}(\boldsymbol{x}) = \boldsymbol{u}_h^{(1)}(\boldsymbol{x})$. After the completion of Steps 1 and 2, we have ensured $\boldsymbol{u}_h^{(2)}(\boldsymbol{x_q}) \in \mathcal{G}_{\boldsymbol{u}}$ for all $\boldsymbol{x_q} \in S^+$.

5.4.4.3 Step 3: Specific Internal Energy

In the third, and final, step we enforce $\epsilon_q \geq \epsilon_{\min,q}$ for all $\boldsymbol{x_q} \in S^+$. To this end, we define $\boldsymbol{U}_h^{(2)} = \big(\rho_h^{(2)}, \boldsymbol{m}_h, E_h, D_{e,h}^{(2)}\big)^{\mathrm{T}}$, which is the full solution vector after steps 1 and 2. Using $\boldsymbol{U}_h^{(2)}$, the specific internal energy and electron fraction in each point $\boldsymbol{x_q} \in S^+$ are computed as

$$\epsilon_q^{(2)} = \epsilon\big(\boldsymbol{U}_h^{(2)}(\boldsymbol{x_q})\big) = \left(E_q - \frac{m_q^2}{2\,\rho_q^{(2)}}\right)/\rho_q^{(2)} \quad \text{and} \quad Y_{e,q}^{(2)} = Y_e\big(\boldsymbol{U}_h^{(2)}(\boldsymbol{x_q})\big) = D_{e,q}^{(2)}/\rho_q^{(2)}, \tag{5.73}$$

respectively. Then, if $\epsilon_q^{(2)} < \epsilon_{\min,q}^{(2)} \equiv \epsilon\big(\rho_q^{(2)}, T_{\min}, Y_{e,q}^{(2)}\big)$ for any $\boldsymbol{x_q} \in S^+$, we replace $\boldsymbol{U}_h^{(2)}$ with the limited polynomial

$$\boldsymbol{U}_h^{(3)}(\boldsymbol{x}) := (1 - \vartheta_3)\,\boldsymbol{U}_K + \vartheta_3\,\boldsymbol{U}_h^{(2)}(\boldsymbol{x}) \quad (\vartheta_3 \in [0,1]). \tag{5.74}$$

Here, the polynomial representation of the full solution is written as a convex combination of the cell average and the polynomial representation after Step 2. Since we assume $U_K \in \mathcal{G}$, setting $\vartheta_3 = 0$ will ensure $U_h^{(3)}(x) \in \mathcal{G}$. However, setting $\vartheta_3 = 0$, so that $U_h^{(3)} = U_K$, kills off all the high-order accuracy of the polynomial representation, which is undesirable. Instead, one would want to find the largest value for ϑ_3 to retain as much high-order accuracy as possible *and* enforce $U_h^{(3)}(x) \in \mathcal{G}$ for all $x_q \in S^+$. As discussed above, this is complicated by the fact that $\mathcal{G}$ is not strictly convex. It is further complicated by the fact that the surface $\epsilon_{\min}(\rho, Y_e)$ is only available at discrete points from the EoS table. Because of this, we will assume that $\mathcal{G}$ is locally convex and first obtain $\vartheta_{3,q}$ by solving

$$\epsilon\left(s(\vartheta_{3,q}) \right) = (1 - \vartheta_{3,q})\, \epsilon_{\min,K} + \vartheta_{3,q}\, \epsilon_{\min,q}^{(2)}, \tag{5.75}$$

for each x_q where $\epsilon_q^{(2)} < \epsilon_{\min,q}^{(2)}$. On the left-hand side of Equation (5.75) we have defined

$$s(\vartheta_{3,q}) = (1 - \vartheta_{3,q})\, U_K + \vartheta_{3,q}\, U_q^{(2)}, \tag{5.76}$$

while on the right-hand side of Equation (5.75) we have defined $\epsilon_{\min,K} = \epsilon(\rho_K, T_{\min}, Y_{e,K})$. Then we set

$$\vartheta_3 := \min_q \vartheta_{3,q}, \tag{5.77}$$

where the minimum is taken over all the points in S^+ where the specific internal energy fell below the minimum value.

We note that the limiter in Equation (5.74) is conservative in all the fields in the sense that the cell average is preserved; i.e.,

$$\frac{1}{V_K} \int_K U_h^{(3)}\, dV_h = (1 - \vartheta_3)\, U_K + \vartheta_3 \frac{1}{V_K} \int_K U_h^{(2)}\, dV_h = (1 - \vartheta_3)\, U_K + \vartheta_3\, U_K = U_K. \tag{5.78}$$

The motivation for solving Equation (5.75) is as follows (cf. Zhang & Shu, 2010, for the ideal EoS case): $s(\vartheta_{3,q})$ is the parametrized straight line connecting the cell average U_K and the point value $U_q^{(2)}$. Since $U_K \in \mathcal{G}$, we know that

$$\epsilon_K \equiv \left(E_K - \frac{m_K^2}{2\rho_K} \right)/\rho_K \geq \epsilon_{\min,K}. \tag{5.79}$$

137

On the other hand, if $\boldsymbol{U}_{\boldsymbol{q}}^{(2)} \notin \mathcal{G}$, there is at least one intersection point of the line $s(\vartheta_{3,\boldsymbol{q}})$ and the boundary of $\mathcal{G}$; i.e. the surface $\epsilon_{\min}(\rho, Y_e)$. (If $\mathcal{G}$ is convex, which we assume in this step, there is exactly one intersection point.) Since we do not know the exact shape of the surface, we approximate it by the line segment connecting the boundary points $\epsilon_{\min,\boldsymbol{K}}$ and $\epsilon_{\min,\boldsymbol{q}}^{(2)}$, and by the convexity assumption, this line lies above the surface $\epsilon_{\min}(\rho, Y_e)$. Thus, in Equation (5.75), the solution $\vartheta_{3,\boldsymbol{q}}$ provides the intersection point between the line connecting the points $\epsilon_{\boldsymbol{K}}$ and $\epsilon_{\boldsymbol{q}}^{(2)}$ and the line connecting the points $\epsilon_{\min,\boldsymbol{K}}, \epsilon_{\min,\boldsymbol{q}}^{(2)}$. See Figure 5.6 for an illustration.

Equation (5.75) is solved for $\vartheta_{3,\boldsymbol{q}}$ with a simple bisection algorithm, using the end points $\vartheta_{3,\boldsymbol{q}} = 0$ and $\vartheta_{3,\boldsymbol{q}} = 1$ as starting points. We note that, in practice, the solution to Equation (5.75) does not have to be accurate to many significant digits, and the bisection algorithm can be terminated after a few iterations. We also note that since $\epsilon_{\min}(\rho, Y_e)$ is not strictly convex, as is shown in Figure 5.5, Equation (5.75) can have multiple roots, and the bisection algorithm may result in a limited solution that is still outside $\mathcal{G}$. We have, however, not encountered a situation where this happens. On the contrary, in the numerical examples presented in Section 5.5, we find that the limiting procedure discussed in this section significantly improves the robustness of the DG algorithm. As can be seen by looking ahead to Figure 5.20 in Section 5.6, the bound-enforcing limiter is continuously activated, with $\vartheta_3 \in [0.4, 1]$, in a short time interval around core bounce in an adiabatic collapse simulation.

Finally, we have assumed that the cell average satisfies $\boldsymbol{U}_K \in \mathcal{G}$ when the limiter is applied. If this assumption does not hold, the bound-enforcing limiter will fail. By considering the equation for the cell average in Equation (5.32), in combination with forward Euler time stepping, it may be possible to derive a sufficient restriction on the time step such that $\boldsymbol{U}_K^{n+1} \in \mathcal{G}$, provided $\boldsymbol{U}_K^n \in \mathcal{G}$ and $\boldsymbol{U}_{\boldsymbol{q}}^n \in \mathcal{G}$ (possibly with additional points included in the set S^+). We do, however, not pursue this endeavor here. Instead, we use the time step restriction given in Equation (5.44), which may not be sufficient. In the absence of an explicit expression for a sufficient time step restriction (assuming one exists), one may design a time step control algorithm where the step size is recursively reduced, and the time step retaken, until a physically admissible cell average is obtained. On the other hand,

we have yet to encounter an application in which a solution with cell average $U_K \notin \mathcal{G}$ is passed to the bound-enforcing limiter.

5.4.5 Poisson Solver

In THORNADO, the approximate Newtonian gravitational potential, Φ_h, is obtained using the Poseidon code (Roberts et al., in preparation). Poseidon solves Equation (5.5) on a spherical-polar grid with a combination of an angular spectral expansion using spherical harmonics and a radial finite element solution method. Here, we discuss the case of spherical symmetry, and thus limit the angular expansion to the monopole harmonic function. Therefore we will focus only on the finite element method (Larson & Bengzon, 2013) used in the radial expansion. Because the Newtonian gravitational potential is expected to be continuous in space, we require the approximate solution, Φ_h, to be C^0 continuous across element interfaces. To enforce this continuity, Poseidon uses the continuous Galerkin (CG) finite element method instead of the DG method to solve the Poisson equation. However, we note that the DG method can also be used to solve elliptic equations (e.g., Rivière, 2008; Vincent et al., 2019).

The CG method expresses the approximate solution, Φ_h, to Equation (5.5) as a continuous expansion of functions of the form

$$\Phi_h(r,t) = \sum_{i=1}^{N_D} \Phi_i(t) v_i(r), \tag{5.80}$$

where N_D is the total number of interpolation nodes on the domain D, and $\Phi_i(t)$ are spatially constant expansion coefficients. As the method used to solve the Poisson equation is a purely spatial in nature, we will omit the time parameter, t, for the rest of this section. The basis functions $v_i(r)$ belong to the approximation space, V_h, defined by

$$V_h = \left\{ \psi_h : \psi_h|_{K^{(j)}} \in P^k(K^{(j)}), \ j = 1, \ldots, N_e \right\}, \tag{5.81}$$

where P^k is a space of one-dimensional piecewise polynomials of degree k, and $K^{(j)}$ are the radial elements of the same decomposition of the computational domain as expressed in Section 5.4.1. Given this choice of approximation space and domain decomposition, N_D is given by $N_D = N_e k + 1$, where N_e is the number of radial elements on the domain.

Continuity is achieved through the choice of interpolation points and approximation space polynomials. Within a specific element $K^{(j)}$, the interpolation points, $\hat{S}_j = \{\xi_{j,1}, \ldots, \xi_{j,m}, \ldots, \xi_{j,k+1}\} \subset I = [-\frac{1}{2}, \frac{1}{2}]$, are chosen to be the Legendre–Gauss–Lobatto (LGL) points. The physical coordinate r is related to the reference coordinate $\xi \in I$ by the transformation

$$r(\xi) = r_{c,j} + \Delta r_j \xi, \tag{5.82}$$

where $r_{c,j}$ is the physical coordinate for the center of element $K^{(j)}$ and is such that

$$r(\xi_{j,k+1}) = r(\xi_{j+1,1}). \tag{5.83}$$

The inverse relationship,

$$\xi(r) = \frac{(r - r_{c,j})}{\Delta r_j}, \tag{5.84}$$

allows us to express the chosen approximation space polynomials $v_i(r) \in V_h$ as

$$v_i(r) = \begin{cases} \ell_{j,m}(\xi(r)) & \text{for } r \in K^{(j)} \\ 0 & \text{else} \end{cases}, \tag{5.85}$$

where $\ell_{j,m}$ are the Lagrange polynomials in Equation (5.15) constructed with the LGL points, $\hat{S}_j$. Each approximation function $v_i(r)$ is associated with a node $\xi_{j,m}$ such that $v_i(r(\xi_{j,m})) = 1$ by the Kronecker delta property of the Lagrange polynomials. This choice of interpolation points and approximation functions enforces the C^0 continuity of the solution. See Figure 5.7 for an illustration of elements and associated basis functions in the CG method for the case with $k = 2$.

The CG method seeks to find $\Phi_h \in V_h$, which approximates Φ in Equation (5.5) such that

$$\langle \nabla^2 \Phi_h, \psi_h \rangle_D = \langle 4\pi G\rho, \psi_h \rangle_D \tag{5.86}$$

holds for all test functions $\psi_h \in V_h$. In Equation (5.86),

$$\langle \nabla^2 \Phi_h, \psi_h \rangle_D = 4\pi \int_{R_L}^{R_H} \nabla^2 \Phi_h \, \psi_h \, r^2 dr, \tag{5.87}$$

and

$$\langle 4\pi G\rho, \psi_h \rangle_D = 16\pi^2 G \int_{R_L}^{R_H} \rho \, \psi_h \, r^2 dr, \tag{5.88}$$

where R_L and R_H are the low and high radial boundary locations of the domain, respectively. Using integration by parts on Equation (5.87), Equation (5.86) becomes the weak form of Equation (5.5),

$$-\langle\, \partial_r\Phi_h, \partial_r\psi_h\,\rangle_D + \left(\partial_r\Phi_h\right)\psi_h\big|_{R_L}^{R_H} = \langle\, 4\pi G\rho, \psi_h\,\rangle_D. \tag{5.89}$$

For the gravitational collapse problem discussed in Section 5.6, we impose the Neumann boundary condition,

$$\partial_r\Phi_h(R_L) = 0, \tag{5.90}$$

on the inner boundary ($R_L = 0$) to preserve the symmetry of the solution, and the Dirichlet boundary condition,

$$\Phi_h(R_H) = -\frac{GM_{enc}}{R_H}, \tag{5.91}$$

on the outer boundary, where M_{enc} is the total enclosed mass given by

$$M_{enc} = 4\pi \int_{R_L}^{R_H} \rho_h(r)\, r^2 dr. \tag{5.92}$$

The Neumann condition in Equation (5.90) reduces Equation (5.89) to

$$-\langle\, \partial_r\Phi_h, \partial_r\psi_h\,\rangle_D + \left(\partial_r\Phi_h(R_H)\right)\psi_h(R_H) = \langle\, 4\pi G\rho, \psi_h\,\rangle_D. \tag{5.93}$$

Next, the expansion in Equation (5.80) and $\psi_h = v_j$ are substituted into Equation (5.93) to give

$$\sum_{i=1}^{N_D} \Phi_i\Big(-\langle\, \partial_r v_i, \partial_r v_j\,\rangle_D + \left(\partial_r v_i(R_H)\right)v_j(R_H) \Big) = \langle\, 4\pi G\rho, v_j\,\rangle_D, \quad j = 1, \cdots, N_D. \tag{5.94}$$

To enforce the Dirichlet condition, the expansion coefficient Φ_N is set to the boundary value given by Equation (5.91), and the dimensionality of the problem is reduced to $N_D - 1$, eliminating the $\left(\partial_r v_i(R_H)\right)v_j(R_H)$ term as $v_j(R_H) = 0, \forall\, j \neq N_D$. Equation (5.94) then becomes

$$\sum_{i=1}^{N_D-1} -\Phi_i\langle\, \partial_r v_i, \partial_r v_j\,\rangle_D = \langle\, 4\pi G\rho, v_j\,\rangle_D, \quad j = 1, \cdots, N_D - 1. \tag{5.95}$$

Defining the stiffness matrix as

$$S = \left\{s_{ij}\right\}_{i,j=1}^{N_D-1}, \quad s_{ij} = -\langle\, \partial_r v_i, \partial_r v_j\,\rangle_D, \tag{5.96}$$

141

the load vector as

$$L = \left\{ \langle 4\pi G\rho, v_j \rangle_D \right\}_{j=1}^{N_D - 1},$$ (5.97)

and the unknown coefficient vector as

$$C = \{\Phi_i\}_{i=1}^{N_D - 1},$$ (5.98)

the system in Equation (5.95) can then be written in matrix form as

$$SC = L.$$ (5.99)

The matrix S is a sparse symmetric band matrix, with bandwidth equal to k. When $k = 1$, the matrix S is tridiagonal. When $k > 1$, an overlapping block structure occurs within the diagonal band of S, see Figure 5.8. The sparsity of the matrix is given by

$$\text{Sparsity} = \frac{N_e k^2 - N_e + 1}{N_e^2 k^2 + 2N_e k + 1}.$$ (5.100)

To reduce memory overhead, S is stored in compressed column storage (CCS) format. The system is then solved using a CCS compatible Cholesky factorization. Once these coefficients are known the approximate solution can be reconstructed anywhere within the domain using Equation (5.80).

5.4.6 Table Interpolation

As in Bruenn (1985), Mezzacappa & Messer (1999), and Bruenn et al. (2020), we obtain a thermodynamic quantity F and its derivatives from the tabulated EoS through trilinear interpolation in the space spanned by $\left(\log_{10}(\rho), \log_{10}(T), Y_e \right)$. The software to compute these are provided by the WeakLib library, and, for completeness, we restate formulas here. To simplify the notation, let $X = \log_{10}(\rho)$, $Y = \log_{10}(T)$, and $Z = Y_e$. Then, $\bar{F} = \bar{F}(X, Y, Z)$, where $\bar{F}$ is related to the thermodynamic quantity by $F = 10^{\bar{F}} - F_{\text{offset}}$. That is, trilinear interpolations are performed on logged quantities, and the offset is used to ensure that $\bar{F}$ is well-defined when F is negative. Obtaining F first requires the eight points $(X_a, Y_b, Z_c) : a, b, c \in \{0, 1\}$ from the table that make up the corners of a "cube" of the points closest to (X, Y, Z). These points then satisfy

$$X_1 - X_0 = \frac{1}{N_\rho}, \quad Y_1 - Y_0 = \frac{1}{N_T}, \quad \text{and} \quad Z_1 - Z_0 = \frac{1}{N_{Y_e}},$$ (5.101)

where N_ρ and N_T are the number of the points per decade in ρ and T, respectively, and N_{Y_e} is the number of points per unit interval in Y_e. $\bar{F}$ is then given by the trilinear interpolation formula, e.g., found in Eq. (32) in Mezzacappa & Messer (1999), which, in multi-index notation, can be written compactly as

$$\bar{F}(X) = \sum_{i=0}^{1} w_i(X)\bar{F}_i, \tag{5.102}$$

where $X = (X, Y, Z)$. In this context, the weights $w_i(X)$ are given by

$$w_i(X) = B_{i_1}(X)B_{i_2}(Y)B_{i_3}(Z), \tag{5.103}$$

where $B_{i_1}(X)$ ($i_1 \in \{0, 1\}$) are linear Lagrange polynomials

$$B_0(X) = \frac{X_1 - X}{X_1 - X_0} \quad \text{and} \quad B_1(X) = \frac{X - X_0}{X_1 - X_0}, \tag{5.104}$$

and $B_{i_2}(Y)$ and $B_{i_3}(Z)$ are similarly defined by replacing X with Y or Z, respectively.

As in Mezzacappa & Messer (1999), derivatives with respect to ρ, T, and Y_e are calculated directly from this expression; i.e.

$$\left(\frac{\partial F}{\partial \rho}\right)_{T,Y_e} = \frac{(F + F_{\text{offset}})}{\rho}\left(\frac{\partial \bar{F}}{\partial X}\right)_{Y,Z} = \frac{(F + F_{\text{offset}})}{\rho}\sum_{i=0}^{1}\frac{\partial w_i}{\partial X}\bar{F}_i, \tag{5.105}$$

$$\left(\frac{\partial F}{\partial T}\right)_{\rho,Y_e} = \frac{(F + F_{\text{offset}})}{T}\left(\frac{\partial \bar{F}}{\partial Y}\right)_{X,Z} = \frac{(F + F_{\text{offset}})}{T}\sum_{i=0}^{1}\frac{\partial w_i}{\partial Y}\bar{F}_i, \tag{5.106}$$

$$\left(\frac{\partial F}{\partial Y_e}\right)_{\rho,T} = (F + F_{\text{offset}})\left(\frac{\partial \bar{F}}{\partial Z}\right)_{X,Y} = (F + F_{\text{offset}})\sum_{i=0}^{1}\frac{\partial w_i}{\partial Z}\bar{F}_i. \tag{5.107}$$

We note that this interpolation scheme does not, by construction, satisfy the Maxwell relations of thermodynamics. While this may impact the ability to resolve adiabatic flows (see Swesty (1996) and Timmes & Swesty (2000) for further discussion), we do not observe any clear evidence of this being a problem in our computations. In addition, while we believe that the low-order accuracy of the trilinear interpolation scheme may play a role in both the convergence rates observed with the high-order RKDG scheme in Section 5.5.1 and the issues with characteristic limiting around the phase transition observed in Section 5.6, additional investigations are required.

143

5.5 Numerical Results

In this section, we present results obtained with the DG method as implemented in THORNADO for various test problems relevant to CCSNe and other astrophysical phenomena. With the exception of few reference calculations obtained using an ideal EoS in Section 5.5.1.1, all the results were obtained using a tabulated version of the SFHo EoS of Steiner et al. (2013b), which covers the ranges $\rho \in [1.66 \times 10^3, 3.16 \times 10^{15}]$ g cm^{-3}, with $N_\rho = 25$, $T \in [1.16 \times 10^9, 1.84 \times 10^{12}]$ K, with $N_T = 50$, and $Y_e \in [0.01, 0.7]$, with $N_{Y_e} = 100$. (See, however, Endeve et al. (2019) for a documentation of results obtained with THORNADO using an ideal EoS.) In the first two subsections, we begin by presenting results from one-dimensional advection tests using Cartesian coordinates, and one- and two-dimensional Riemann problems using Cartesian, spherical-polar, and cylindrical coordinates (Sections 5.5.1 and 5.5.2, respectively). These tests serve as an initial gauge of the implementation of the DG algorithm in THORNADO with a nuclear EoS. Using Riemann problems with initial conditions adapted from their ideal EoS counterparts, we aim to investigate the performance of our implementation in curvilinear coordinates, as well as the slope limiter presented in Section 5.4.3 and the bound-enforcing limiter presented in Section 5.4.4. The Poisson solver is tested in Section 5.5.3. Then, in Section 5.6, our focus turns to the main application, adiabatic gravitational collapse in spherical symmetry, where we investigate the performance of THORNADO's DG algorithm by investigating various aspects of the solver with an eye towards future spherically symmetric — and eventually multidimensional — supernova simulations with neutrino transport. In all the tests, the CFL number in Equation (5.44) is set to $C_{\text{CFL}} = 0.5$.

5.5.1 Advection Tests

5.5.1.1 Rate of Convergence

The accuracy of the DG method can be manipulated by changing the number of nodes per cell $N = k + 1$ and/or the total number of cells K. The number of nodes per cell (or element) governs the expected order of accuracy of the method. (Nth order spatial accuracy is expected with N nodes.) This section covers the rate at which changing the number of degrees of freedom $n_{\text{DOF}} = (k + 1) \times K$ impacts the accuracy; i.e. the convergence rate. Inspired by Suresh & Huynh

144

([1997](#)), this test is performed over the 1D computational domain $D = [-100, \ 100]$ km, with smooth initial conditions, and periodic boundary conditions. The initial state for the tabulated EoS case is set with the primitive state vector $\mathbf{P}$ as

$$\mathbf{P} = \left(\rho, u, p, Y_e \right)^T = \left(\rho_0 \left(1 + 0.1 \ \sin^4(\pi x/L) \right), \ v_0, \ p_0, \ 0.3 \right)^T,$$

where $\rho_0 = 10^{12} \, \mathrm{g \, cm^{-3}}$ is the background density, $v_0 = 0.1 \, c$ is the velocity, $p_0 = 0.01 \, \rho_0 \, c^2$ the background pressure, and $L = 200$ km is the domain length. In this test, the mass density, a quartic sine wave, is advected for one period without any limiting, while the velocity, pressure, and electron fraction remain constant. The error in mass density between the initial and final states is then calculated in the L_1 error norm,

$$L_1 \equiv \sum_{j=1}^{n_{\mathrm{DOF}}} \left| \rho_{j,\mathrm{final}} - \rho_{j,\mathrm{initial}} \right|. \tag{5.108}$$

In Figure [5.9](#) we plot this quantity, scaled by both n_{DOF} and a background density ρ_0, versus n_{DOF} (crosses). (For reference, we also plot results obtained with an ideal EoS case with $\Gamma = 1.4$; open circles.) The solutions are obtained using $N = 2$ (black symbols) and $N = 3$ (red symbols) nodes with second and third-order time integration schemes, respectively. For each N, we use seven different values of K : 8, 16, 32, 64, 128, 256, and 512. For this smooth problem we always observe that for a fixed n_{DOF} the scheme with $N = 3$ is significantly more accurate than the scheme with $N = 2$. For the nuclear EoS case, the L_1 error for the second-order scheme ($N = 2$) crosses zero and generates a cusp at $n_{\mathrm{DOF}} = 512$. Otherwise, the results obtained with the second-order method agree well with the expected convergence rate for both the tabulated and ideal EoS cases. For $N = 3$, the ideal EoS case exhibits third-order accuracy throughout. However, for $n_{\mathrm{DOF}} > 96$, the results for $N = 3$ with the tabulated EoS appear to undergo a transition from third-order to second-order accuracy. We suspect that the trilinear interpolation method discussed in [5.4.6](#) may be the cause of the loss of accuracy for large n_{DOF}, but this requires further investigation.

5.5.1.2 Discontinuous Multi-Wave

This test from Suresh & Huynh ([1997](#)) involves the advection of a discontinuous initial state for mass density, which includes a Gaussian wave, a square wave, a triangular wave, and a semi-

ellipse (see light red lines in Figure 5.10). This test is performed over a periodic 1D domain $D = [-100, 100]$ km, with the initial state given as

$$\mathbf{P} = \left(\rho, u, p, Y_e\right)^T = \left(\rho(x, 0), v_0, p_0, 0.3\right)^T,$$

where v_0 and p_0 are given the same values as in the previous test, and $\rho(x, 0)$ is a piece-wise function defined as

$$
\begin{aligned}
\rho(x, 0) &= \rho_0 \left(1 + 0.1 \exp\left(-\log(2)(x/L + 0.7)^2/(0.0009)\right)\right) && \text{if } -80\,\text{km} \leq x \leq -60\,\text{km} \\
\rho(x, 0) &= \rho_0 (1 + 0.1) && \text{if } -40\,\text{km} \leq x \leq -20\,\text{km} \\
\rho(x, 0) &= \rho_0 (1 + 0.1 (1 - |10(x/L - 0.1)|)) && \text{if } \quad 0\,\text{km} \leq x \leq \quad 20\,\text{km} \\
\rho(x, 0) &= \rho_0 \left(1 + 0.1 \left(1 - 100(x/L - 0.5)^2\right)^{1/2}\right) && \text{if } \quad 40\,\text{km} \leq x \leq \quad 60\,\text{km} \\
\rho(x, 0) &= \rho_0 && \text{otherwise,}
\end{aligned}
$$

where $L = 100$ km.

We compare the performance of second- and third-order schemes in this test. Thus, a second- and third-order SSP-RK time integration scheme was used for $k = 1$ and $k = 2$, respectively. This test used the characteristic limiting procedure described in Section 5.4.3 with a TCI threshold $C_{\text{TCI}} = 1.0 \times 10^{-3}$ and a total variation diminishing parameter $\beta_{\text{TVD}} = 1.5$. Figure 5.10 shows the initial density profile (light red lines) along with four different cases of the mass density being evolved one (medium red lines) and ten (dark red lines) times across the periodic domain. Results obtained with the second-order method are displayed in the top panels, while results obtained with the third-order method are displayed in the bottom panels. Note that the results displayed in the top left and top right panels were obtained using the same total number of degrees of freedom as the results displayed in the bottom left and bottom right panels, respectively. Analytically, the evolved solution should match up exactly with the initial condition after each full domain crossing. However, the numerical solution is distorted by dissipation and dispersion. For fixed n_{DOF}, the third-order method appears to provide more accurate results. As the solution is evolved in the $k = 1$ case, accuracy is lost primarily around sharp edges, namely for the Gaussian and triangular waveforms. For the $k = 1$ case with 384 elements, the solution is not well-resolved around the base of each waveform, but some accuracy is gained around the maxima. Loss of accuracy around

146

sharp edges is also observed with the third-order method using 128 elements (bottom left panel). However, as is seen in the bottom right panel, the features of the solution are better captured with the third-order method using 256 elements. For the third-order method, we note that most of the distortion of the initial profile occurs in the first domain crossing, as the profiles after one and ten crossings are almost on top of each other. This is not so much the case for the second-order scheme, where the results after one and ten crossings are more easily distinguished. However, there is a trade-off between numerical accuracy and computational expense.

5.5.2 Riemann Problems

5.5.2.1 Sod Shock Tube: Cartesian Coordinates

This test is based on the classic Riemann problem from Sod (Sod, 1978). It involves an initially stationary fluid with a discontinuity separating two states – left and right – with high pressure and density on the left and low pressure and density on the right. This initial state evolves into a shock propagating into the low density region, followed by a contact discontinuity, and a rarefaction wave propagating back into the high density state. Shock tube problems such as this stress a method's ability to capture shocks and contact discontinuities without smearing or introducing unphysical oscillations. Given the importance of shocks in CCSNe, this serves as a critical first test for any method designed to model these explosions.

Here, the problem is modified to use physical units in a regime realizable in simulations of CCSNe. The computational domain is $D = [-5, 5]$ km with the discontinuity initially at $x = 0$ km, separating the left and right states

$$\mathbf{P}_L = (\rho, \, v, \, p, \, Y_e)_L^T = \left(10^{12} \text{ g cm}^{-3}, 0, 10^{32} \text{ erg cm}^{-3}, 0.4 \right)^T \tag{5.109}$$

$$\mathbf{P}_R = (\rho, \, v, \, p, \, Y_e)_R^T = \left(1.25 \times 10^{11} \text{ g cm}^{-3}, 0, 10^{31} \text{ erg cm}^{-3}, 0.3 \right)^T. \tag{5.110}$$

(Note that the initial Y_e profile is also discontinuous.)

The problem is evolved until $t = 0.021$ ms, using 100 uniform elements with $\beta_{\text{TVD}} = 1.75$, and no troubled-cell indicator ($C_{\text{TCI}} = 0$), so that limiting is applied everywhere. We use third-order spatial discretization ($k = 2$) and third-order temporal integration (SSP-RK3). A main

147

focus with this test is to compare results obtained with component-wise and characteristic limiting (discussed in Section 5.4.3). Figure 5.11 shows results for mass density (upper left), pressure (upper right), velocity (lower left), and electron fraction (lower right), using both characteristic (blue) and component-wise limiting (red), compared to a reference solution (black) computed using the first-order accurate spatial method ($k = 0$), third-order time integration, and 10000 elements. We note that both limiting schemes capture the general nature of the solution, including the rarefaction wave, which extends from about -3 to 0 km, the contact discontinuity, which is located at about 2 km, and the shock, located at about 4 km. The scheme based on characteristic limiting, however, is better at suppressing oscillations, and is less dissipative across the contact discontinuity. These observations are consistent with those made by Schaal et al. (2015) in the ideal EoS case.

5.5.2.2 Sod Shock Tube: Spherical-Polar and Cylindrical Coordinates

As a test of THORNADO's ability to work with non-Cartesian coordinate systems, we also solve a spherically symmetric version of the Sod shock tube problem in 1D spherical-polar and 2D cylindrical coordinates. For spherical-polar coordinates, the domain is $D = [0, 10]$ km, with the initial discontinuity placed at $r = 5$ km, while, for cylindrical coordinates, our domain is $D = [0, 10]$ km $\times [-10, 10]$ km, and the discontinuity is placed at $r = \sqrt{R^2 + z^2} = 5$ km. For the initial left and right states, we use those given in the 1D Cartesian Sod test in Equations (5.109)-(5.110), with the exception that the electron fraction is given a constant value of $Y_e = 0.4$ across the entire domain. We evolve both tests until $t = 0.025$ ms using 100 elements in the spherical case and 100×200 elements in the cylindrical case. Both tests use the third-order methods ($k = 2$ and SSP-RK3), characteristic limiting with $\beta_{\mathrm{TVD}} = 1.75$, and no troubled cell indicator ($C_{\mathrm{TCI}} = 0$). We note that for the 2D test with cylindrical coordinates, we used THORNADO's interface to AMReX to take advantage of AMReX's MPI infrastructure.

Results are shown in Figure 5.12. In the left panel of Figure 5.12, we show the 2D density distribution for the cylindrical test. In the right panel of Figure 5.12, we show the density, velocity, and pressure profiles of the spherical-polar test (solid lines), along with scatter plots of the corresponding quantities from the cylindrical test versus spherical-polar radius r for comparison.

We note that the characteristics of the solution profiles are similar to those obtained by others using an ideal EoS (e.g., Omang et al., 2006). There is also good agreement between the results obtained with spherical-polar and cylindrical coordinates. As in the Cartesian test, we note the clear resolution of the shock and contact discontinuity with no discernible oscillations. Furthermore, we note some spread in the scatter plots from the cylindrical solution, most notably in the velocity profile across the contact discontinuity. However, despite the truly multidimensional setup in the cylindrical case, there is decent preservation of the spherical symmetry inherit in the test.

5.5.2.3 Shock Tube Provoking the Bound-Enforcing Limiter

This test, performed in 1D with Cartesian coordinates, is similar to the Sod shock tube discussed in Section 5.5.2.1, but with initial conditions tuned to provoke the bound-enforcing limiter developed in Section 5.4.4. The goal is to demonstrate that the limiter keeps the solution within the set of admissible states (specifically that $\epsilon \geq \epsilon_{\min}$) while also conserving the total mass, energy, and electron number in time, given, respectively, by

$$\int_D \left\{ \rho_h(x,t),\ E_h(x,t),\ D_{e,h}(x,t) \right\} dx. \tag{5.111}$$

The computational domain is $D = [-5, 5]$ km, and a discontinuity is placed at $x = 0$ km, which separates the left and right states of the Riemann problem

$$\mathbf{P}_L = (\rho, v, p, Y_e)_L^T = \left(1.00 \times 10^{13}\ \mathrm{g\,cm^{-3}},\ 0\,\mathrm{m\,s^{-1}},\ 1.070 \times 10^{31}\ \mathrm{erg\,cm^{-3}},\ 0.04 \right)^T$$

$$\mathbf{P}_R = (\rho, v, p, Y_e)_R^T = \left(1.25 \times 10^{12}\ \mathrm{g\,cm^{-3}},\ 0\,\mathrm{m\,s^{-1}},\ 1.023 \times 10^{30}\ \mathrm{erg\,cm^{-3}},\ 0.10 \right)^T.$$

The numerical solution is evolved to $t = 0.2$ ms, using 256 elements with polynomial degree $k = 2$ and SSP-RK3 time integration. To fully test the bound-enforcing limiter, we run this test without the slope limiter discussed in Section 5.4.3. Moreover, it is possible to design an initial state for the Sod shock tube problem that does not place the solution close to or below the minimum table boundary. An example of this is seen in section 5.5.2.1, where the bound enforcing limiter is not required to keep the solution within the set of admissible states. However, we note that this particular test (using the initial condition described immediately above) fails without the bound enforcing limiter,

regardless of whether or not the slope limiter is implemented.[6] Thus, the bound enforcing limiter allows for a wider selection of initial states that would otherwise cause the algorithm to fail.

Numerical results from this test are displayed in Figure 5.13. In the left panel, we plot the specific internal energy versus position at the end of the simulation (solid black curve). We also plot the minimum internal energy $\epsilon_{\min}(\rho, Y_e)$ (dashed red curve). Around the shock, ϵ is very close to the minimum value, as can be seen in the inset in left panel of Figure 5.13. In fact, the specific internal energy remains very close to the minimum value throughout this test. The middle panel displays a space-time plot of the limiter parameter $\vartheta_3(x,t)$ in Equation (5.74), and shows the activation sites for the bound-enforcing limiter, where the average value for ϑ_3 when limiting is required is 0.9 and it ranges from $0.60 < \vartheta_3 < 0.99$. The bound-enforcing limiter is activated due to small oscillations slightly ahead of the shock, and produces a trace of the shock trajectory as seen in the middle panel in Figure 5.13. The slope of the prominent trace in $\vartheta_3(x,t)$ indicates a shock velocity of $v_{\text{shock}} \approx 1800$ km s^{-1}. Finally, the right panel in Figure 5.13 shows the relative change in the conserved quantities versus time. The change in these quantities are due to machine roundoff, indicating that the bound-enforcing limiter is sufficiently conservative for this test.

5.5.2.4 Shu-Osher Shock Tube

This test adopted from Shu & Osher (1988) involves a Mach=3 shock interacting with a lower density region with a sinusoidal perturbation. As the shock propagates and interacts with the density perturbations, the perturbations move upstream, forming high frequency oscillations just behind the shock. This problem tests the ability of a shock-capturing method to limit unphysical oscillations without destroying physical, small-scale features of the post-shock flow. We note that small-scale features resulting from hydrodynamical instabilities, such as turbulence and convection, are crucial to CCSN explosion dynamics (e.g., Murphy & Meakin, 2011; Murphy et al., 2013; Couch & Ott, 2015; Radice et al., 2016; Mabanta & Murphy, 2018; Couch et al., 2020) and many other astrophysical applications.

Here, the problem is modified to use physical units in a regime relevant to CCSNe. The

---

[6]Even when slope limiting is used, the bound enforcing limiter is required for this test, but we decide to deactivate the slope limiter to provoke the bound enforcing limiter even more.

computational domain is $D = [-5, 5]$ km, with a discontinuity initially located at $x = 1$ km separating the left and right states

$$\mathbf{P}_L = (\rho, v, p, Y_e)_L^T = \left(3.60632 \times 10^{12} \text{ g cm}^{-3}, 7.425 \times 10^4 \text{ km s}^{-1}, 1.333 \times 10^{32} \text{ erg cm}^{-3}, 0.5\right)^T$$

$$\mathbf{P}_R = (\rho, v, p, Y_e)_R^T = \left(\left[1 + 0.2 \times \sin(5 \text{ km}^{-1} x)\right] \times 10^{12} \text{ g cm}^{-3}, 0, 1.0 \times 10^{31} \text{ erg cm}^{-3}, 0.5\right)^T.$$

The fluid is evolved until $t = 0.0625$ ms, using 256 uniform elements and $\beta_{\text{TVD}} = 2.0$. We use third-order spatial discretization ($k = 2$) and third-order temporal integration (SSP-RK3). In this test we compare results obtained with characteristic and component-wise limiting, and, for each limiting method, we show results for various values of the TCI threshold.

In Figure 5.14, we show the density obtained using characteristic (top) and component-wise (bottom) limiting for various values of the troubled-cell indicator threshold C_{TCI}: 0.0 (full limiting, red), 0.03 (green), 0.3 (magenta), and 3.0 (blue); i.e., the same values that were used in Endeve et al. (2019) for the ideal EoS case. Larger values of C_{TCI} imply less slope limiting. These results are compared to a reference solution obtained using 2048 elements (black), with third-order spatial and temporal discretization, and $C_{\text{TCI}} = 0.0$. In both limiting schemes, full limiting washes out the density variations behind the shock, while increasing the TCI threshold allows for these features to be better captured. The results obtained with $C_{\text{TCI}} = 3.0$ are very close to the reference solution. However, for reasons discussed in Section 5.6.5, we do not recommend using such a high value for C_{TCI} in general, since some amount of limiting — even in smooth regions — seems to be required. For all values of the threshold (except perhaps the case with $C_{\text{TCI}} = 3.0$, which applies little limiting away from the shock), the characteristic limiting scheme better captures the shape and amplitude of the oscillations behind the shock (see insets in each panel, focusing on the higher frequency oscillations just behind the shock).

5.5.2.5 Two-Dimensional Riemann Problem

Here we consider a two-dimensional Riemann problem, adapted from Lax & Liu (1998), which involves a fluid with a different initial state in each quadrant given by

$$\mathbf{P}_{\mathrm{NW}} = \left(\rho,\, v^1,\, v^2,\, p,\, Y_{\mathrm{e}} \right)_{\mathrm{NW}}^{\mathrm{T}} = \left(10^{12}\ \mathrm{g\ cm^{-3}},\, 7.275 \times 10^4\ \mathrm{km\ s^{-1}},\, 0,\, 10^{32}\ \mathrm{erg\ cm^{-3}},\, 0.3 \right)^{\mathrm{T}},$$

$$\mathbf{P}_{\mathrm{NE}} = \left(\rho,\, v^1,\, v^2,\, p,\, Y_{\mathrm{e}} \right)_{\mathrm{NE}}^{\mathrm{T}} = \left(5.313 \times 10^{11}\ \mathrm{g\ cm^{-3}},\, 0,\, 0,\, 4.0 \times 10^{31}\ \mathrm{erg\ cm^{-3}},\, 0.3 \right)^{\mathrm{T}},$$

$$\mathbf{P}_{\mathrm{SE}} = \left(\rho,\, v^1,\, v^2,\, p,\, Y_{\mathrm{e}} \right)_{\mathrm{SE}}^{\mathrm{T}} = \left(10^{12}\ \mathrm{g\ cm^{-3}},\, 0,\, 7.275 \times 10^4\ \mathrm{km\ s^{-1}},\, 10^{32}\ \mathrm{erg\ cm^{-3}},\, 0.3 \right)^{\mathrm{T}},$$

$$\mathbf{P}_{\mathrm{SW}} = \left(\rho,\, v^1,\, v^2,\, p,\, Y_{\mathrm{e}} \right)_{\mathrm{SW}}^{\mathrm{T}} = \left(8.0 \times 10^{11}\ \mathrm{g\ cm^{-3}},\, 0,\, 0,\, 10^{32}\ \mathrm{erg\ cm^{-3}},\, 0.3 \right)^{\mathrm{T}},$$

on a domain $D = [0, 1.0]$ km $\times [0, 1.0]$ km. This test, which corresponds to "Configuration 12" in Lax & Liu (1998), involves two shocks moving into the northeastern quadrant and contact discontinuities (or slip lines) at the northern and eastern boundaries of the southwestern quadrant. It is adapted from the original works to use physical units in a regime relevant to CCSNe with a nuclear EoS. The initial configuration presented here is one of many possible configurations of 2D Riemann problems presented in Lax & Liu (1998). The fluid is evolved until $t = 0.0025$ ms using 400^2 uniform elements, $\beta_{\mathrm{TVD}} = 1.75$, and $C_{\mathrm{TCI}} = 0$ (i.e., limiting is applied everywhere). We use third-order spatial discretization ($k = 2$) and third-order temporal integration (SSP-RK3). To run this test, we used THORNADO's interface to AMReX in order to take advantage of AMReX's MPI parallelization.

Figure 5.5.2.5 shows the density (top panels) and pressure (bottom panels) at $t = 0.0025$ ms, from a run with component-wise limiting (left panels) and a run with characteristic limiting (right panels). Black lines on each plot show logarithmically spaced contours to highlight solution features. Overall, the morphology of the solutions obtained with THORNADO — using a nuclear EoS — agree well with the results displayed by Lax & Liu (1998). Moreover, the use of characteristic limiting presents a tremendous improvement over component-wise limiting, particularly as the higher dimensionality of the problem allows for more complex flow patterns and discontinuity geometries. Notably, the density and pressure contours in the component-wise limiting case reveal more oscillations and deformities. These oscillations are particularly prominent near the boundary

of the curved shock surface. There appears to be no oscillations present in the run performed with characteristic limiting. Similarly, the jet-like feature seen in the southwest quadrant of the density plots appear less resolved and are somewhat asymmetric in the component-wise limiting case.

5.5.3 Poisson Solver Test

The accuracy of the CG method used by Poseidon to solve Equation (5.5) is determined by the total number of degrees freedom used to solve the system. The number of degrees of freedom can be changed by either the p-method or the h-method. The p-method varies the degree k of the polynomials used in the approximation of the solution and requires $k + 1$ nodes per element. The h-method increases the number of elements N_e used to discretize the system. These two methods are used together in the hp-method where both the refinement of the mesh and the degree of the approximation polynomials can be varied. In the hp-method, the number of degrees of freedom is given by $n_{\text{DOF}} = (k + 1) \times N_e$. The accuracy of the hp-method increases with increasing n_{DOF}, and the error should decrease with increasing n_{DOF} as $1/n_{\text{DOF}}^{k+1}$.

We test the accuracy of Poseidon's Poisson solver using the density profile of a centrally condensed sphere of radius R. This test, from Stone & Norman (1992), was chosen because it has a non-polynomial analytic solution, thus allowing us to better explore the convergence properties of the solver. (Problems with polynomial solutions are solved exactly for sufficiently high k.) The density profile and analytic solution for the test are given by

$$
\rho(r) = \begin{cases} \dfrac{\rho_c}{1+\left(\frac{r}{r_c}\right)^2} & \text{if } r \leq R \\ 0 & \text{if } r > R \end{cases}
\tag{5.112}
$$

and

$$
\Phi(r) = \begin{cases} -4\pi G \rho_c r_c^2 \left[1 - \dfrac{\arctan\left(\frac{r}{r_c}\right)}{\frac{r}{r_c}} - \dfrac{1}{2}\left(\dfrac{1+\left(\frac{r}{r_c}\right)^2}{1+\left(\frac{R}{r_c}\right)^2}\right) \right] & \text{if } r \leq R \\ -4\pi G \rho_c \dfrac{r_c^3}{r} \left[\dfrac{R}{r_c} - \arctan\left(\dfrac{R}{r_c}\right) \right] & \text{if } r > R, \end{cases}
\tag{5.113}
$$

respectively, where ρ_c and r_c are the central density and core radius, respectively. For this test, we choose $\rho_c = 150\,\text{g cm}^{-3}$, $r_c = 0.2\,R_\odot$, and $R = R_\odot$, and perform the calculations over the 1D

153

computational domain $D = [0, 2 R_\odot]$. We compute the L_1 and L_∞ error norms as

$$L_1 \equiv \sum_{j=1}^{n_{\mathrm{DOF}}} |\Phi(r_j) - \Phi_h(r_j)|, \tag{5.114}$$

and

$$L_\infty \equiv \max_j \left(|\Phi(r_j) - \Phi_h(r_j)| \right) \quad \text{for} \quad j \in \{1, \ldots, n_{\mathrm{DOF}}\}. \tag{5.115}$$

In Figure 5.16, we plot the L_1 error norm (scaled by n_{DOF}; left panel) and the L_∞ error norm (right panel) versus n_{DOF}. The numerical solutions were obtained using $k = 1$ (black symbols) and $k = 2$ (red symbols). For each value of the polynomial degree k, seven values of N_e (8, 16, 32, 64, 128, 256, and 512) were used to create uniform grids. From these plots we see that for a specific value of N_e the higher order method always provides a more accurate solution. The rate of convergence observed for the third-order method is as expected (or better) in both error norms (cf. red, dashed reference lines). The second-order method converges at a rate somewhat slower than expected when the error is measured in the L_1 error norm, but the L_∞ error decreases roughly at the expected second-order rate (cf. black, dashed reference lines).

5.6 Adiabatic Collapse, Core-Bounce, and Shock Propagation

In this section we employ the DG method implemented in THORNADO to evolve a non-rotating progenitor through adiabatic collapse, core bounce, and post-bounce shock propagation. The initial conditions are provided by a 15 M$_\odot$ progenitor model from Woosley & Heger (2007). Overall, this section will cover the chronological evolution of the stellar collapse model in three stages: (1) adiabatic collapse of the core, (2) core rebound and the formation of the shock shortly after nuclear saturation, and (3) the propagation of the shock through the outer core thereafter. In total, the evolution covers about 800 ms of physical time, which is divided into about 300 ms for collapse, and almost 500 ms of post-bounce evolution, until the bounce shock reaches the outer boundary.

The following subsections will first discuss the physical conditions of the adiabatic collapse application that challenge any hydrodynamics method used for CCSN simulations. Then we focus on various features of the DG method in THORNADO, such as (1) the performance of the bound-enforcing limiter during bounce and shock formation, (2) the response of the numerical solution

154

to adjusting the troubled-cell indicator threshold parameter C_{TCI}, (3) resolution dependence in the inner core, (4) the challenge of maintaining energy conservation when applying limiters, and (5) difficulties associated with employing characteristic limiting in the vicinity of the phase transition. Of course, being spherically symmetric and without neutrino transport, this adiabatic model does not describe a realistic evolutionary trajectory for a CCSN progenitor. However, this test does subject the numerical method to some of the physical conditions encountered, and we deem it a necessary step towards more realistic models.

Using spherical-polar coordinates, the domain $D = [0, 8000]$ km is divided into $N = 512$ elements. In the interest of capturing important physical characteristics while maintaining computational efficiency, this application implements a geometrically progressing grid that uses a finer spatial resolution in the inner core, which becomes progressively coarser according to

$$\Delta r_i = z \times \Delta r_{i-1}, \quad i = 2, \ldots, N, \tag{5.116}$$

where $z > 1$ is the 'zoom factor'. This emphasizes the inner core, where most of the mass is concentrated after collapse, while deemphasizing the outer regions. To begin constructing the grid, the innermost cell width $\Delta r_1 = \Delta r_{\text{min}}$, the length $|D|$ of the spatial domain, and the number of elements N are defined. Then, the zoom factor is obtained by solving

$$\eta \times \left(z^N - 1\right) - (z - 1) = 0, \tag{5.117}$$

where $\eta = \Delta r_{\text{min}}/|D|$. The fiducial run in this section uses an inner cell width of $\Delta r_{\text{min}} = 0.5$ km. Then, with $|D| = 8000$ km and $N = 512$, this results in a zoom factor (in double precision) of $z = 1.009967685243838$, and an outer cell width of $\Delta r_N = 79.45$ km. Also, for the fiducial run, we use second-order spatial ($k = 1$) and temporal (SSP-RK2) discretization, combined with the component-wise limiting scheme discussed in Section 5.4.3, $\beta_{\text{TVD}} = 1.75$, and $C_{\text{TCI}} = 0.0$. For all the runs, we use reflecting boundary conditions at the inner boundary and Dirichlet conditions (provided by the initial condition) at the outer boundary. The gravitational potential is obtained with a second-order accurate CG method as discussed in section 5.4.5.

5.6.1 Stage 1: Collapse

Figure 5.17 illustrates the collapse phase prior to core bounce. We scale such that bounce occurs at $t - t_b = 0$ ms with $t_b = 302.9$ ms for this model, which is defined as the time when the central density, ρ_c, reaches its maximum. We plot the mass density (upper left panel), velocity (upper right panel), electron fraction (lower left panel), and entropy per baryon (lower right panel) versus radius for select times during collapse. We have chosen to display the collapse profiles at the times coinciding with each full decade in central density; i.e. $\rho_c = 10^{10,11,\dots,14}$ g cm^{-3}. The collapse dynamics is very similar to the self-similar solutions obtained by Yahil (1983), using a polytropic EoS. The central density increases with time and approaches nuclear densities (10^{14} g cm^{-3}) at $t - t_b = -1$ ms while the outer region rarefies as indicated by the steeper slope in density outside the innermost core. Meanwhile, the infall velocity increases linearly with radius in the inner core (consistent with homologous collapse), and approaches free-fall beyond the maximum infall velocity, where it eventually falls off roughly as $r^{-1/2}$. The maximum infall velocity reaches $11 - 12\%$ of the speed of light just before bounce. The electron fraction, Y_e, is a monotonically increasing function of radius and its inner profile shifts inward — in an approximately self-similar fashion — with the decreasing core radius during collapse. Because this test models adiabatic flows (i.e., no neutrino physics is included), the electron fraction remains constant in the core. Before core bounce and shock formation the entropy profile shifts inward due to the collapsing core. In fact, both the electron fraction and entropy profiles remain constant in the core throughout collapse, bounce, and shock propagation, which we quantify further in Section 5.6.6.

5.6.2 Stage 2: Core-Bounce

Figure 5.18 captures core-bounce and shock formation in the inner core ($r \in [0, 500]$ km). We plot the adiabatic index $\Gamma \equiv \left(\frac{\partial \ln p}{\partial \ln \rho}\right)$ (upper left), velocity (upper right), electron fraction (lower left), and entropy per baryon (lower right) versus radius. In each panel, blue curves illustrate the dynamics immediately before bounce (leading up to maximum ρ_c), while red curves illustrate the dynamics immediately after bounce (see color maps to the right of each panel). The bounce dynamics is in response to the stiffening of the EoS, which is illustrated by the evolution of the

adiabatic index during the transition to nuclear matter in the inner core. In the upper left panel, the adiabatic index is $\Gamma \approx 4/3$ at $t - t_b = -0.8$ ms. Once the core reaches nuclear densities and undergoes a phase transition to bulk nuclear matter, the EoS stiffens and the repulsive nuclear forces between the tightly packed nucleons results in a jump in Γ to around 2.5 at the inner boundary. After bounce, the inner core, $r \lesssim 10$ km is characterized by $\Gamma \approx 2.5$, while $\Gamma \lesssim 4/3$ at larger radii. Notice the sharp transition occurring around $r = 10$ km, which we refer to as the *phase transition*. The velocity profiles provide a clear demonstration of the genesis and evolution of the shock resulting immediately after bounce. When the EoS stiffens, collapse is halted, and a shockwave is formed in the region $r \in [10, 20]$ km. Once formed, the shock must push through the supersonically collapsing outer core. In this adiabatic simulation, without neutrinos, the shockwave propagates relatively unencumbered through the outer core, and eventually reaches the outer boundary. The constant value in electron fraction in the very inner core is preserved through bounce and shock formation, meanwhile the profile in the outer region (around 10 km) shifts as the shock travels through. There is no noticeable change in central entropy during bounce, but, as the shock forms, there is a large increase in the entropy across the shock, as expected.

5.6.3 Stage 3: Shock Propagation

Figure 5.19 shows the shock's trajectory through the outer core on its way towards the outer boundary. In this figure, we plot the mass density (upper left), velocity (upper right), electron fraction (lower left), and temperature (lower right) versus radius for select times after bounce. As can be seen by inspecting all panels, the inner core (inside about 50 km) settles into an approximate hydrostatic equilibrium once the bounce shock has cleared. Inside this region, the velocity is small (compared with the sound speed), and the mass density, electron fraction, and temperature profiles remain practically unchanged for hundreds of milliseconds. This suggests that the DG method is quite capable of capturing the adiabatic nature of the flow (this is further supported by the results shown in the left panel in Figure 5.23). In the velocity figure, the shock is seen to reduce in amplitude as it propagates towards the outer boundary. Early on, one can also observe secondary shocks, produced by the ring-down of the core as it settles into hydrostatic equilibrium, which later

catch up with the main shock. Rarefaction of the gas occurs in the outer core (beyond 100 km) as the shock pushes through the infalling matter, notably at $t - t_b = 362$ ms in the mass density profile. The thermal energy behind the shock is partially used to dissociate heavy nuclei and alpha particles in the supersonically infalling outer core, causing the shock to lose energy while leaving behind free nucleons in its wake. As the shock travels outward, the electron fraction profile in the outer core is advected with the flow; cf. the sharp gradient located around 100 km at $t - t_b = 2$, which has moved to about 1000 km when $t - t_b = 362$ ms. The temperature inherently rises across the shock, and a sharp rise in temperature that traces the path of the shock is seen in the lower right panel.

5.6.4 Bound-Enforcing Limiter

The microphysical conditions encountered in this test are constrained by the nuclear EoS. However, some extreme conditions encountered are difficult to resolve numerically, and thus may push the solutions beyond the boundaries of the admissible state set. For example, when the core bounces and launches the bounce shock, the discontinuity can generate oscillations in the numerical solution. These oscillations are to a certain degree suppressed by the slope limiting procedure described in Section 5.4.3, but the solution can still exceed the limits of the tabulated EoS. Thus, the bound-enforcing limiting procedure from Section 5.4.4 is required to ensure that the numerical solution remains physically valid, mostly at bounce and shock formation. When necessary, the bound-enforcing limiter acts to constrain the mass density, electron fraction, and specific internal energy. However, for the conditions encountered in the adiabatic collapse simulations discussed in this section, only violations of the bounds on the specific internal energy trigger limiting (cf. Step 3 in Section 5.4.4), namely during the early stages of shock formation. We note that, without the bound-enforcing limiter, the specific internal energy falls below the minimum possible value at certain locations, which then implies that a valid temperature — required, e.g., to compute the pressure — cannot be found, and the algorithm fails. Therefore, the bound-enforcing limiter is a critical component of the DG algorithm in THORNADO.

Figure 5.20 illustrates the action of the bound-enforcing limiter during bounce in the fiducial run

discussed in the previous subsections. The left panel is a space-time plot of the limiter parameter $\vartheta_3 \in [0, 1]$ (cf. Equation (5.74)), and shows the activation sites of the bound-enforcing limiter acting to constrain the specific internal energy ϵ. Values of $\vartheta_3 < 1$ imply some amount of limiting. The region displayed in the figure captures the brief moment around shock formation where ϵ drops below the minimum value, but is corrected by shifting the DG solution toward the cell average by an amount determined by ϑ_3. The darker regions indicate more aggressive limiting, and we find that ϑ_3 can become as small as 0.4 in this case. In the right panel in Figure 5.20, the specific internal energy is plotted versus radius for select times during the initial shock propagation (black lines). We also plot the minimum specific internal energy $\epsilon_{min}(\rho, Y_e)$, using the corresponding numerical solutions for ρ and Y_e (red lines). This figure captures ϵ being very close to, but above, ϵ_{min} — especially around the shock, which is located roughly $r = 20, 40,$ and 70 km for the times displayed.

Figure 5.21 shows activation sites of the bound-enforcing limiter in the ρY_e-plane (white dots). The majority of the activation sites are seen at higher mass densities, and correspond to the formation of the shock. These points appear to occupy a locally convex region of $\epsilon_{min}(\rho, Y_e)$. However, some points also appear at a low density and higher electron fraction. These points correspond to a moment toward the end of the simulation, specifically when the shock passes through the outer boundary. This portion of the EoS table may also be locally convex, thus the limiting scheme is expected to operate robustly in that region as well. Future work will involve an investigation of the EoS surface at minimum temperature to further challenge the robustness of our bound-enforcing limiter. This work, however, will need to be carried out in the context of neutrino radiation-hydrodynamics simulations of CCSNe, which access different and/or larger regions of the ρY_e-plane.

5.6.5 Troubled-Cell Indicator Threshold Dependence

In this section, we investigate the effect of varying the troubled-cell indicator threshold C_{TCI} on the adiabatic collapse simulations. The numerical results discussed in the previous subsections applied the slope limiter everywhere; i.e. the TCI threshold C_{TCI} was set to zero such that all

elements are flagged for (component-wise) limiting. As seen in Section 5.5.2.4, increasing the value of C_{TCI} prevents limiting at smooth extrema and preserves the accuracy of the solution. However, in contrast to the shock tube problem, the solutions for the adiabatic collapse problem exhibit nonzero slopes almost everywhere. This leads to more areas that may require limiting, and it becomes more difficult to find an optimal value for C_{TCI}. Moreover, various quantities vary by many orders of magnitude across the computational domain, and it is not clear which variables are optimal for detecting troubled cells. When using the mass density, the total fluid energy density, and the electron fraction as the variables to sense troubled cells, we find that if C_{TCI} is set too high, some areas that may require limiting are not flagged, and oscillations can start to develop.

In general, we have found that thermodynamic quantities such as the temperature and entropy per baryon demonstrate a higher sensitivity to C_{TCI} than the evolved quantities $\boldsymbol{U}$. Thus, this section will focus on the solution for the temperature and its sensitivity to C_{TCI}. Figure 5.22 shows the evolution of the troubled-cell indicator I_K (cf. Equation (5.59)) versus radius for adiabatic collapse simulations with various values of C_{TCI}: 0.01 (upper left panel), 0.03 (upper right panel), and 0.05 (lower left panel). The plotted quantities are derived from the maximum value across all fields in each element; i.e.,

$$I_K = \max_{G \in \boldsymbol{G}} I_K(G), \quad \text{where} \quad \boldsymbol{G} = (\rho, E, \mathrm{Y_e})^{\mathrm{T}}. \tag{5.118}$$

In each panel, the red curve represents the time-averaged value (from t_{b} to $t_{\mathrm{end}} - t_{\mathrm{b}} = 497.1$ ms), while the maximum and minimum values are given by the boundaries of the light gray-shaded region, and the positive standard deviations (i.e., average plus one σ) are given by the upper boundary of the dark gray-shaded region. We also plot C_{TCI} in each panel (dashed horizontal line). Recall that an element is flagged for limiting whenever $I_K > C_{\text{TCI}}$. In the lower right panel of Figure 5.22, we plot the temperature versus radius at the end of each simulation with a different value of C_{TCI}. For comparison, we also plot the temperature for the simulation from the previous subsections, with $C_{\text{TCI}} = 0$, which applies limiting everywhere.

As can be seen in the lower right panel in Figure 5.22, the temperature profiles from all four runs display the same general trend, and fall on top of each other outside $r = 50$ km. The simulations with

$C_{\text{TCI}} = 0$ and $C_{\text{TCI}} = 0.01$ (magenta and black lines, respectively) are practically indistinguishable everywhere. However, inside $r = 50$ km, the simulations with the larger values of C_{TCI} (0.03 and 0.05) exhibit some oscillations about the temperature profile from the fiducial run with $C_{\text{TCI}} = 0$, and the amplitude appears to increase with increasing C_{TCI}. The TCI maxima (upper boundary of the light gray-shaded region) are generally above the threshold in all cases, which implies that limiting has been applied at least once in most of the domain displayed. However, the average and the one sigma values serve as better indicators for where limiting occurs. Although the I_K values tend to be above the threshold inside the first 100 km in the $C_{\text{TCI}} = 0.01$ case, the solution is limited most frequently inside $r = 50$ km, which corresponds to the region where the temperature displays oscillatory behavior in the runs with larger values of C_{TCI}. As the threshold is increased, less of this region receives limiting. And in particular, for the 0.03 and 0.05 threshold cases, oscillations have developed in this region. Ideally, the limiting procedure should both preserve the original order of accuracy and prevent the development of spurious oscillations. However, for the adiabatic collapse simulations, inside $r = 50$ km, there seems to be a trade-off between these two features which leaves little flexibility for selecting a large value for C_{TCI}.

5.6.6 Resolution Dependence

In this section we investigate the effect of varying the spatial resolution in the adiabatic collapse simulation. To do this, we keep the number of elements fixed to $N = 512$, and vary the innermost cell width Δr_1 from 0.125 km to 1.0 km. Table 5.2 lists the inner- and outer-most cell widths along with the cell widths at $r = 10$ km and $r = 100$ km, and the corresponding zoom factors z, in Equation (5.116). Since we keep the number of elements fixed, the zoom factor increases with decreasing Δr_1, which also results in coarser resolution in the outer regions of the computational domain. We find that the general features of the solution — e.g., density and velocity profiles — are rather insensitive to the numerical resolution. Instead, we focus on the long term evolution (i.e., hundreds of milliseconds) of the central density, electron fraction, and entropy per baryon. After bounce, when the inner core settles into hydrostatic equilibrium, the central density should remain relatively constant with time. Similarly, since we do not include neutrinos and the evolution

161

is adiabatic, the central electron fraction and entropy per baryon should also remain constant throughout the simulation. Figure 5.23 shows results from varying the inner cell width. In the left panel, we plot the central density ρ_c versus time after bounce; i.e. the time when maximum central density is achieved. (To better visualize with a logarithmic abscissa, we have applied an arbitrary shift of 0.6 ms.) The right panel displays the evolution of the central entropy per baryon S_c (top) and electron fraction $Y_{e,c}$. During collapse, these quantities are plotted versus central density, while after bounce they are plotted versus time. There is some spread in the central density curves before bounce, but they all reach about the same maximum, $\rho_c \approx 4.2 \times 10^{14}$ g cm^{-3}, and, after the core stabilizes after bounce, ρ_c remains constant with time for all resolutions. For the coarsest resolution run ($\Delta r_1 = 1$ km), the central density settles down to about 3.425×10^{14} g cm^{-3}, while in the finer resolution models it settles down to about 3.475×10^{14} g cm^{-3}. Because the collapse is adiabatic and the profiles are constant with radius in the very inner core (cf. lower panels in Figure 5.17), S_c and $Y_{e,c}$ should remain constant throughout the evolution. All the simulations exhibit this behavior before $\rho_c \approx 10^{13}$ g cm^{-3}; i.e., before the phase transition into nuclear densities. (There is a slow increase in S_c, from 0.73 to 0.74, during collapse.) Just before core bounce, the profiles deviate somewhat from their constant values, and the lower resolutions exhibit larger deviations. For both central entropy and electron fraction, the profiles for the runs with $\Delta r_1 = 0.75$ km and $\Delta r_1 = 1.0$ km undergo notably larger changes than the higher resolution profiles. $Y_{e,c}$ remains nearly constant through bounce for the 0.125 km, 0.25 km, and 0.5 km simulations. For both $Y_{e,c}$ and S_c, the two lowest resolution profiles drop further down before maximum central density, and then exhibit a slight drift with time after bounce. However, both of these quantities remain relatively constant with time after bounce in the higher resolution cases. Thus, a threshold resolution seems to be required to accurately capture the physical behavior in the inner core. Considering the balance between computational cost and physical fidelity, an inner cell width of 0.5 km (as in the fiducial run) appears to be close to the optimal choice among the tested resolutions. For example, the central density for this run remains constant after bounce, as desired. It also maintains approximately constant central entropy and electron fraction through bounce. The central entropy deviates by no

more than 0.02 k_B, while the electron fraction changes by no more than about 10^{-6}.

Table 5.2 Inner, $r = 10$ km, $r = 100$ km, outer cell widths, and zoom factors for geometrically progressing grids with $N = 512$ elements.

Δr_1 [km]	$\Delta r_{10\,km}$ [km]	$\Delta r_{100\,km}$ [km]	Δr_N [km]	Zoom Factor
0.125	0.258	1.430	1.048×10^2	1.013260722382225
0.25	0.366	1.401	9.225×10^1	1.011634298318296
0.5	0.598	1.489	7.945×10^1	1.009967685243838
0.75	0.835	1.630	7.185×10^1	1.008968091682754
1.0	1.077	1.821	6.641×10^1	1.008244905346311

5.6.7 Energy Conservation

In this section we investigate total energy conservation with the DG method in THORNADO in the context of the adiabatic collapse simulations. Exact conservation of total energy is nontrivial to achieve in simulations of self-gravitating flows because of the adopted formulation of the fluid energy equation given by Equation (5.3), which is in non-conservative form due to the gravitational source term on the right-hand side. For simplicity, we limit the discussion to the present context of spherical-polar coordinates with spherical symmetry imposed. Then, by combining Equations (5.1), (5.3), and (5.5), it is possible to formulate a conservation law for the total energy

$$\partial_t \mathcal{E} + \frac{1}{r^2} \partial_r \left(r^2 \mathcal{F} \right) = 0, \tag{5.119}$$

where

$$\mathcal{E} = \rho \left(\epsilon + \frac{1}{2} v^2 + \frac{1}{2} \Phi \right) \quad \text{and} \quad \mathcal{F} = \left(E + p + \rho \, \Phi \right) v + \frac{1}{8\pi G} \left(\Phi \, \frac{\partial \dot{\Phi}}{\partial r} - \dot{\Phi} \, \frac{\partial \Phi}{\partial r} \right) \tag{5.120}$$

are the total energy density and total energy flux density, respectively, v is the radial component of the fluid three-velocity, and $\dot{\Phi} = \partial_t \Phi$. Because the corresponding RKDG discretization of Equations (5.1) and (5.3), and the CG discretization of Equation (5.5), do not combine exactly to form a discrete equivalent to Equation (5.119), the conservation of total energy is not expected to be exact in the adiabatic collapse simulations. Although we find that the combination of RKDG and CG discretizations exhibits surprisingly good energy conservation properties, we find evidence that

the application of the slope and bound-enforcing limiters, mainly around core bounce, compromise the conservation of total energy. As seen in Figure 5.13 for the Riemann problem invoking the bound-enforcing limiter, in the absence of gravity, the total fluid energy (i.e., internal plus kinetic) is by construction conserved to machine precision. The slope limiter is also conservative with respect to the total fluid energy. Conservation of total energy is more difficult to achieve for self-gravitating flows such as in the adiabatic collapse problem.

By integrating Equation (5.119) over the computational domain $D = [0, R]$, and from t_0 to t, the total energy in the system is given by

$$E_{\text{total}}(t) = E_{\text{total},0} - 4\pi R^2 \int_{t_0}^{t} \mathcal{F}(R, \tau)\, d\tau, \qquad (5.121)$$

where

$$E_{\text{total}} = \int_D \rho\, \epsilon\, dV + \frac{1}{2} \int_D \rho\, v^2\, dV + \frac{1}{2} \int_D \rho\, \Phi\, dV \equiv E_{\text{i}} + E_{\text{k}} + E_{\text{g}}, \qquad (5.122)$$

and $dV = 4\pi r^2 dr$. Figure 5.24 shows energy conservation results from adiabatic collapse simulations. In the left panel, we plot the kinetic, gravitational, internal, and total energy versus time for the fiducial run with $\Delta r_1 = 0.5$ km. Approaching core-bounce, the internal energy E_{i} and the gravitational energy E_{g} grow rapidly in concert (with opposite signs), before stabilizing after bounce with $E_{\text{i}} \approx 157$ B and $E_{\text{g}} \approx -158$ B, where 1 B = 10^{51} erg. The kinetic energy, E_{k}, peaks at approximately 10 B at bounce before decreasing again, and is down to 1 B when $t - t_{\text{b}} = 5.5$ ms. The kinetic energy continues to decrease, and reaches a minimum of about 0.55 B at $t - t_{\text{b}} \approx 40$ ms. Then, for $t - t_{\text{b}} \gtrsim 40$ ms, the kinetic energy starts to increase again, and is back up to 1 B for $t - t_{\text{b}} = 200$ ms.

The change in the total energy versus time, $E_{\text{total}} - E_{\text{total},0}$, is plotted in the middle panel of Figure 5.24 for the various spatial resolutions investigated in Section 5.6.6. As can be seen, the total energy remains relatively constant during collapse, makes an almost discontinuous jump around bounce, before remaining relatively constant again after bounce. (At $t - t_{\text{b}} \approx 375$ ms, the bounce shock reaches the outer boundary, and the total energy starts to decrease due to the energy flux through the boundary; cf. the second term on the right-hand side of Equation (5.121), which

has not been accounted for in the figure.) The magnitude of the jump in total energy at bounce decreases with increasing resolution in the core. Around $t = t_b$, the total energy in the fiducial run ($\Delta r_1 = 0.5$ km) increases by less than 0.5 B, as is seen from the middle curve (after bounce) in the middle panel in Figure 5.24. The difference $E_{\text{total}} - E_{\text{total},0}$ ought to remain zero throughout the simulation, but the extreme conditions during core-bounce — due to short time and length scales, and the necessity of applying limiters around the region of shock formation, which occurs at high energy densities — result in energy conservation violations. The change in the total energy in the fiducial run is less than 0.5% of the gravitational energy at bounce, and about 5% of the kinetic energy at bounce. Without accounting for the energy flowing through the outer boundary, the total energy changes by less than 1.5×10^{-3} B during collapse, until $t - t_b = -0.6$ ms, when the central density is about 1.5×10^{14} g cm^{-3}. Then, after bounce, from $t - t_b \approx 50$ ms to $t - t_b \approx 350$ ms, the total energy changes by less than 2.5×10^{-3} B, which is small compared to any of the individual components of the total energy.

We have found that the slope and bound-enforcing limiters contribute to the violation of total energy conservation at bounce. To investigate the impact of limiters on total energy conservation, we restarted the fiducial run, which employs slope and bound-enforcing limiters, at $t - t_b = -1$ ms, and ran one model with the slope limiter turned off, and one model with both the slope and bound-enforcing limiters turned off. The right panel in Figure 5.24 shows the total energy conservation versus time for these models. The largest violation of total energy conservation is observed in the fiducial run (red line). For the model where the slope limiter is turned off, but the bound-enforcing limiter is still active, the change in the total energy is noticeably reduced (black line). For example, the black line demonstrates no noticeable change in the total energy briefly before bounce, while the red line shows a minor increase starting at $t - t_b = -0.5$ ms. Thus, the slope limiter begins adding energy to the system shortly before bounce. Meanwhile, the bound-enforcing limiter remains inactive until about 0.2 ms before bounce. Once activated, the bound-enforcing limiter breaks total energy conservation, but to a lesser extent than when both limiters are active. The reason the limiters contribute to total energy violation is the gravitational potential energy, the third integral

on the right-hand side of Equation (5.122). While both limiters preserve the cell-averaged fluid energy, and thus leave the first two integrals on the right-hand side of Equation (5.122) unchanged, the cell-averaged gravitational potential energy density is defined as a higher moment of the mass density (Φ depends on position), which is not preserved by any of the limiters. It is interesting to note that the DG method manages to model core bounce and shock formation without the slope limiter activated. When both limiters are turned off, the run fails at bounce because ϵ may fall below the minimum value required by the EoS. Until then, the DG method maintains total energy conservation well. For example, we find $E_{\text{total}} - E_{\text{total}}(t_{\text{b}} - 1 \text{ ms}) = 8.6 \times 10^{-6}$ B at the time when the run crashes, which occurs when $\rho_{\text{c}} = 3.65 \times 10^{14}$ g cm^{-3}. In the future, we will investigate ways of improving the conservation of total energy while applying both limiters through bounce.

5.6.8 Characteristic Limiting

In contrast to the Riemann problems discussed in Section 5.5.2, the adiabatic collapse application does not currently benefit from characteristic limiting. As discussed earlier, toward the end of collapse, the core undergoes a phase transition from atomic nuclei and nucleons to bulk nuclear matter. However, the tabulated nuclear matter EoS appears to not be sufficiently smooth around this transition to enable robust construction of the characteristic fields, which depends on thermodynamic derivatives from the EoS (see Appendix 9D). Moreover, the interpolation scheme discussed in Section 5.4.6 is only C^0 continuous, which implies that derivatives are discontinuous across adjacent cubes in the table. As a result, the thermodynamic derivatives are not smooth around the phase transition, which appears to give rise to unphysical perturbations. These perturbations manifest as acoustic noise, in the characteristic and, eventually, the conserved fields, and this is clearly evident by considering the pressure. Figure 5.25 displays space-time plots of the logarithmic pressure gradient, $(\ln p/\ln r)$, shortly after bounce from two simulations — one employing component-wise limiting (left panel), and one employing characteristic limiting (right panel). In both panels, the dashed black line corresponds to the minimum in the adiabatic index (cf. upper left panel in Figure 5.18), which we refer to as the phase transition. The formation of the bounce shock and subsequent acoustic waves during the ring-down phase after bounce are

166

clearly seen in the lower part of both panels, which qualitatively agree. However, about 5 ms after bounce, acoustic waves are seen to continuously emanate from the core in the model with characteristic limiting. These waves are absent in the model with component-wise limiting. The accoustic waves in the model with characteristic limiting appear to emanate from the vicinity of the phase transition; i.e. originate around the vertical dashed black line (see also Figure 5.26). Another difference in the results obtained with the two limiters is the behavior of the maximum logarithmic pressure gradient. In the component-wise case, the peak in the pressure gradient remains fixed at approximately 30 km for the duration of the run after bounce. With characteristic limiting, this peak has a slow trajectory, starting at about $r = 15$ km and ending at $r = 30$ km.

Figure 5.26 shows a zoomed-in portion of the logarithmic pressure gradient for the characteristic limiting case displayed in the right panel in Figure 5.25. Spurious pressure waves appear to be generated around the phase transition (or slightly ahead of the dashed black line), which then propagate across the entire domain. Prominent examples of this are seen around $t - t_b = 0.5$ ms, 1.65 ms, and 2.45 ms, where pairs of left- and right-propagating waves emanate from the phase transition. The left-going waves propagate toward the inner boundary and are then reflected back out. These waves lead to the noisy pattern seen in the right panel in Figure 5.25.

As discussed in Section 5.4.3, characteristic limiting relies on transforming the set of conserved variables to the set of characteristic variables by applying the matrix of left eigenvectors from the eigendecomposition of the flux Jacobian. The construction of this matrix involves thermodynamic derivatives of the pressure and other quantities which, in the case of the tabulated EoS, do not have analytic expressions. Instead, these derivatives are obtained by differentiating the trilinear interpolation formula used to obtain quantities from the EoS table, and are not necessarily smooth — especially across the phase transition. The result is a discontinuity in every characteristic variable around the location of the phase transition. Because of this, it appears to no longer be beneficial to employ characteristic limiting, as it results in the waves seen in Figures 5.25 and 5.26, and destroys the accuracy gained from characteristic limiting observed in Section 5.5.2. Moreover, the expression for the sound speed given in Appendix 9C, obtained from the eigendecomposition

of the flux Jacobian, may transiently become imaginary due to variations in the derivatives. In this case we default to constructing the sound speed as provided by the EoS table. Future work will include improving the fidelity of THORNADO's interface with the EoS table, especially around the phase transition, in order to circumvent these problems.

5.7 Summary, Conclusions, and Outlook

5.7.1 Summary

We have extended the Runge–Kutta discontinuous Galerkin (RKDG) method for the Euler equations to accommodate an equation of state for dense nuclear matter, to solve problems in Cartesian, spherical-polar, and cylindrical coordinate systems in a three-covariant framework, and to simulate adiabatic, spherically-symmetric stellar collapse with self-gravity. More specifically, we have implemented a spectral-type nodal collocation DG approximation, which leads to simplifications in the semi-discrete equations — especially for problems that make use of curvilinear coordinates. In making these extensions to the RKDG method, we extended various limiters to maintain physically sound solutions:

- We have supplemented the RKDG method with a standard total variation diminishing slope limiter, combined with a troubled-cell indicator, to maintain time-integration stability and to reduce spurious oscillations around discontinuities. For our purposes, this involved a non-trivial adaptation of the limiter to nuclear equations of state, specifically when limiting the characteristic fields, and we have provided the necessary characteristic decomposition to achieve this in Appendix 9C.

- We have designed a bound-enforcing limiter to prevent the numerical solutions from becoming physically inadmissible; i.e. exceeding bounds imposed by the tabulated EoS. The tabulated EoS is supplied with strict boundaries in which the solution must be confined. However, critical thermodynamic quantities provided by the EoS are not necessarily globally convex, and this complicates the design of the bound-enforcing limiter, which currently operates under the assumption of a convex EoS.

168

We have developed THORNADO based on this extended RKDG method. THORNADO is written in modern Fortran, which is a general purpose programming language for high-performance scientific computing. Moreover, THORNADO is intended for multiphysics CCSN simulations with high-order methods, and to this end the RKDG method for hydrodynamics has been chosen, in part, for its ability to faithfully capture discontinuities and its ability to maintain high-order accuracy in smooth flows with a compact computational stencil. Distributed parallel computing capabilities with MPI are enabled through an interface with AMReX (Zhang et al., 2019). (The incorporation of AMReX's adaptive mesh refinement is deferred to future work.) We also mention that, in addition to distributed parallelism with MPI, THORNADO has been partially ported to utilize graphics processing units (GPUs) through the OpenACC[7] and OpenMP[8] standards, which will allow THORNADO to utilize heterogeneous architectures. Details on this progress will be reported in a future publication.

We have tested THORNADO against a suite of diverse and challenging problems incorporating a tabulated nuclear EoS in one and two spatial dimensions (see Endeve et al. (2019) for further tests in the ideal EoS case):

- To test the formal order of accuracy of the RKDG method with a nuclear EoS we performed an advection test with a smooth mass density profile using second- and third-order methods and various degrees of freedom to determine the rate of convergence. It was found that the third-order method is significantly more accurate than the second-order method, but the rate of convergence for the third-order method deteriorates to second-order at higher resolution, possibly due to the use of trilinear EoS interpolation. To further examine the efficacy of the high-order RKDG method, a discontinuous multi-shaped mass density profile was advected using characteristic limiting, and the initial condition was compared with the numerical solution after one and ten periods. We compared results obtained with second- and third-order methods using the same total number of degrees of freedom, by adjusting the number of cells. The third-order method was found to be superior to the second-order method in this

---

[7]https://www.openacc.org
[8]https://www.openmp.org

case as well.

- We conducted several well-known Riemann problem tests — adapted to the nuclear EoS case — in Cartesian, spherical-polar, and cylindrical coordinates, and one and two spatial dimensions, to examine THORNADO's ability to resolve discontinuities with high-order RKDG methods, without introducing spurious oscillations. It was demonstrated that results obtained with characteristic limiting are far superior to corresponding results obtained with component-wise limiting. Finally, a special version of the Sod shock tube test was constructed to examine the efficacy of the bound-enforcing limiter. In this case, it was demonstrated that the bound-enforcing limiter maintains physically admissible solutions, while at the same time preserving the inherent conservation properties of the RKDG method.

We have applied THORNADO to the problem of adiabatic stellar core collapse of a realistic non-rotating progenitor in spherical symmetry:

- We modeled the critical phases of collapse, through nuclear densities, the phase transition to bulk nuclear matter, core bounce, shock formation, and the propagation of the shock through the outer stellar layers.

- The complexity of this application necessitated additional investigations to probe the features of the RKDG method for hydrodynamics in THORNADO, such as the role of limiting and how it contributes to improved robustness of these simulations, the dependence of the solution on the troubled-cell indicator threshold and spatial resolution, and the conservation of energy through challenging stages in the simulation, such as stellar core bounce.

5.7.2 Conclusions and Outlook

- We successfully evolved a non-rotating, spherically symmetric, 15 $M_\odot$ progenitor with self-gravity through adiabatic collapse, bounce, and several hundred milliseconds of shock propagation past bounce, while maintaining adiabaticity (e.g. the entropy and electron fraction profiles remained constant in the core). The success of this application marks

170

an important step toward applying DG methods to more realistic CCSN simulations, and given the results obtained, we are in a position to develop THORNADO further towards more physically complete CCSN simulations; e.g., by incorporating neutrino transport.

- In the adiabatic collapse application, the bound-enforcing limiter is critical in allowing the solution to evolve through bounce. Without this limiter, the solution exceeds the limits of the EoS and the algorithm fails. The bound-enforcing limiter is required to maintain a physically valid solution for this application, but it, along with the slope limiter, interferes with the inherently good energy conservation properties of the RKDG scheme. Before and after bounce, the change in total energy is relatively low. However, when limiters are applied through bounce, an artificial jump in the total energy compromises the energy conservation. The change in total energy is less than 0.5 B for the fiducial run with inner cell width of 0.5 km, and decreases with increasing spatial resolution. While the change in total energy during bounce is relatively small, when compared to any of the individual energy components, future work focusing on reducing this unphysical change in total energy is warranted.

- For standard hydrodynamics tests with shocks, such as Riemann problems, we have shown that characteristic limiting is superior to component-wise limiting for resolving discontinuities while suppressing nonphysical, oscillatory features. However, characteristic limiting depends on derivatives of thermodynamic quantities, which are estimated from the tabulated EoS, and may not be sufficiently smooth. In particular, for the adiabatic collapse application, we observed anomalous behavior in the form of acoustic noise, which appears to originate around the phase transition. Thus, characteristic limiting currently does not provide the desired improvements for the adiabatic collapse application, or any problem that may involve a phase transition. The issue associated with these thermodynamic derivatives must be further investigated and resolved before our numerical method can be extended more generally to employ more sophisticated limiters, such as moment limiters (Krivodonova, 2007) or WENO-type limiters for DG (Zhu et al., 2020), which also rely on limiting of characteristic fields. This

may involve an improved EoS interpolation scheme that enforces thermodynamic consistency.

- As seen in the convergence tests, the RKDG method in THORNADO gained accuracy over lower-order schemes by implementing high-order discretization ($N = 3$) for a fixed number of degrees of freedom. However, third-order methods diminished to second-order accuracy for higher degrees of freedom, and the interpolation of the tabulated EoS may have been an agent in this loss of accuracy, but further investigation is required to confirm this. Moreover, for all the tests in Section 5.5, our method reliably captured physical discontinuities and oscillations with high-order instantiations of the RKDG scheme in THORNADO. However, for the adiabatic collapse application, we consistently employed a second-order accurate approach. The main reason: transient spurious oscillations (or perturbations) developed when we employed third-order discretization. Again, the interpolation of the EoS may be impacting the performance of the high-order scheme. We emphasize that the results for the gravitational collapse application obtained with second-order methods and component-wise limiting are satisfactory, and provides the basis for incorporating neutrino transport algorithms also based on DG methods. However, while the present paper represents a step towards our goal, further work is required to realize CCSN simulations with high-order DG methods.

- All results presented here were obtained with the HLL Riemann solver (Harten et al., 1983a). While we have also implemented the HLLC Riemann solver (Toro et al., 1994), which is designed to account for contact discontinuities and has been shown to give superior results (see, e.g., Cardall et al., 2014), we decided not to use this Riemann solver here. The known "odd-even" instability (Quirk, 1994), which develops with the HLLC Riemann solver in some multidimensional settings, is the main reason for our decision. Future work includes development of a hybrid solver, with the capability of applying the HLLC solver in regions of smooth flow while switching to the HLL solver in the vicinity of shocks by means of a shock detector.

- Because CCSNe are general relativistic in nature, we are extending the hydrodynamics in THORNADO to accommodate general relativity under the conformally-flat approximation (see, e.g., Wilson et al., 1996), some details of which are given in Dunham et al. (2020).
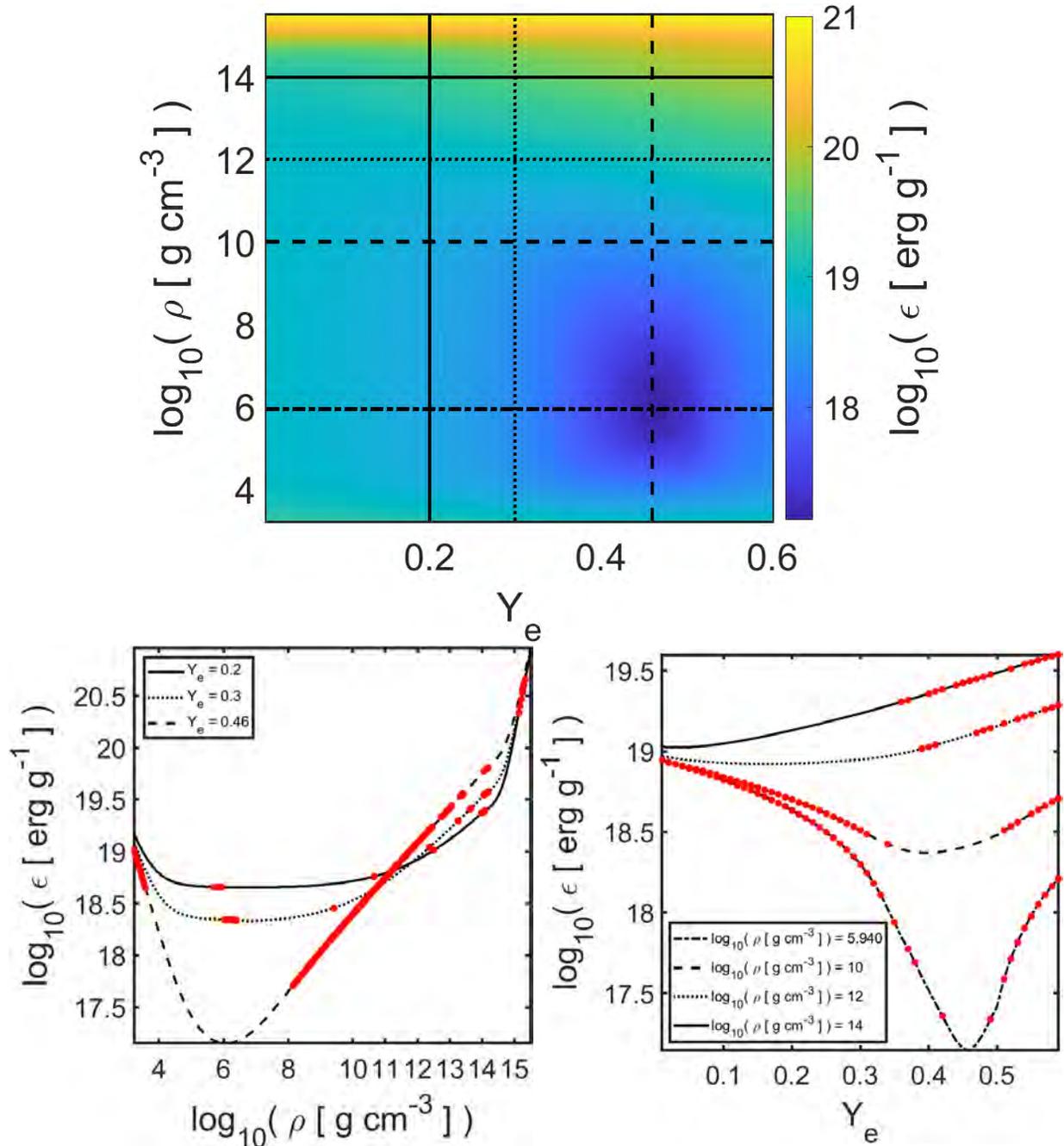
Figure 5.5 The minimum specific internal energy, $\epsilon_{min} = \epsilon(\rho, T_{min}, Y_e)$ for the SFHo EoS is displayed in the top panel as a function of ρ and Y_e. (Since the specific internal energy can be negative, we have added an offset of about 2.8×10^{17} erg g^{-1} to the actual value returned by the EoS). The vertical lines (constant Y_e) correspond to lines plotted in the lower left panel, while horizontal lines (constant ρ) correspond to lines plotted in the lower right panel. To further highlight the topology of the ϵ_{min} surface, we plot ϵ_{min} versus mass density for select values of Y_e in the lower left panel. Similarly, we plot ϵ_{min} versus electron fraction for select values of ρ in the lower right panel. These traces are selected to illustrate the non-convexity of $\mathcal{G}$. The red dots indicate non-convex regions, i.e. where the second derivative of the specific internal energy with respect to either ρ (left) or Y_e (right) is less than zero. From visual inspection, the constant Y_e profiles (lower left) appear to be convex; i.e., $(\partial^2 \epsilon_{min}/\partial \rho^2)_{Y_e} \geq 0$. However, the dashed line is not, and the dotted and solid lines are non-convex around $\rho = 10^{14}$ g cm^{-3}. Meanwhile the constant ρ profiles (lower right) are clearly not all convex since $(\partial^2 \epsilon_{min}/Y_e^2)_\rho$ can be negative (cf. dashed and dash-dot curves).
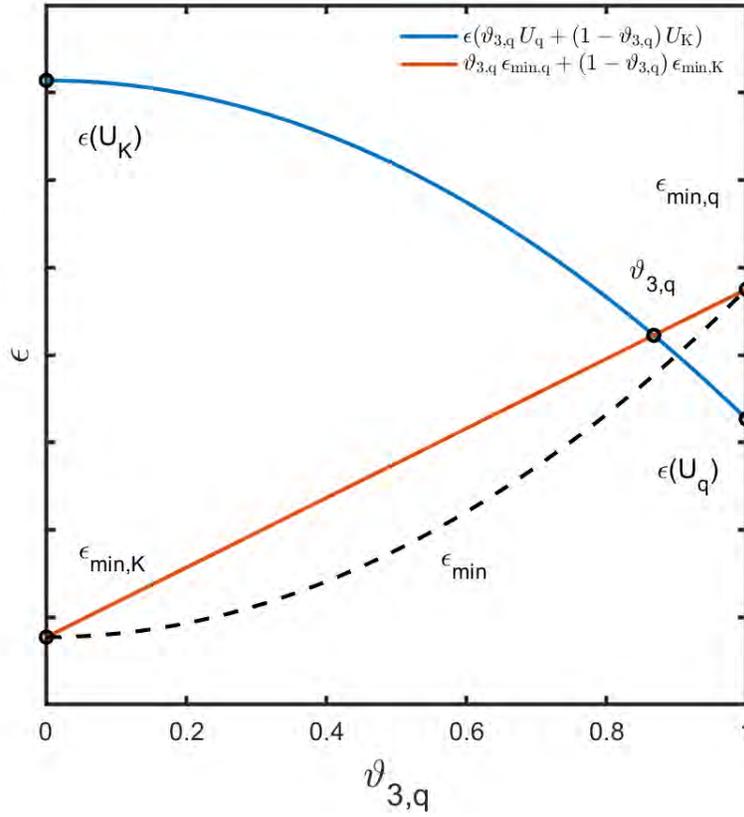
Figure 5.6 Illustration of the bisection problem used to find $\vartheta_{3,q}$ in Equation (5.75) to determine the extent of limiting needed to ensure that the specific internal energy does not fall below the table boundary $\epsilon_{\min}$ (dashed black curve). In the example depicted here, $\epsilon(U_q)$, the right endpoint of the blue curve, is below the table boundary, and limiting is needed. We find $\vartheta_{3,q}$ as the intersection point between the blue curve connecting $\epsilon(U_K)$ and $\epsilon(U_q)$, and the red curve connecting $\epsilon_{\min,K}$ and $\epsilon_{\min,q}$. In this case, $\vartheta_{3,q} \approx 0.87$.
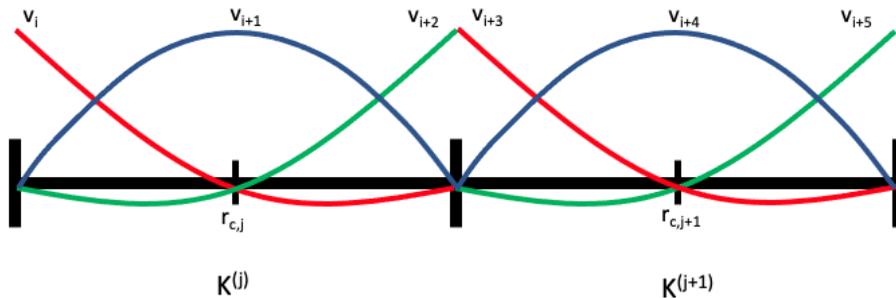
Figure 5.7 Illustration of the basis functions, v_i, used in the CG solution method of Poseidon for the case $k = 2$. Each function is associated with a specific element $K^{(j)}$ and a node $\xi_{j,m}$ within that element, such that $v_i\big(r(\xi_{j,m})\big) = 1$. Outside of the associated element, $v_i = 0$, and is therefore not depicted.
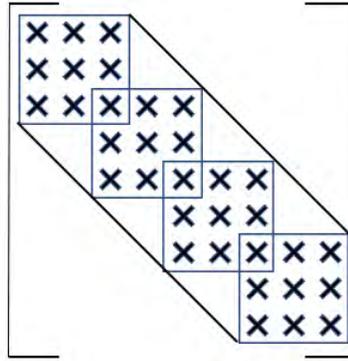
Figure 5.8 Nonzero structure of the matrix S used in Poseidon for the case of $k = 2$, and $N_e = 4$. The diagonal lines denote the band structure of the matrix. The squares denote the overlapping block structure within the band. The single overlapping element shared by the consecutive blocks comes from the shared interpolation node at element interfaces.
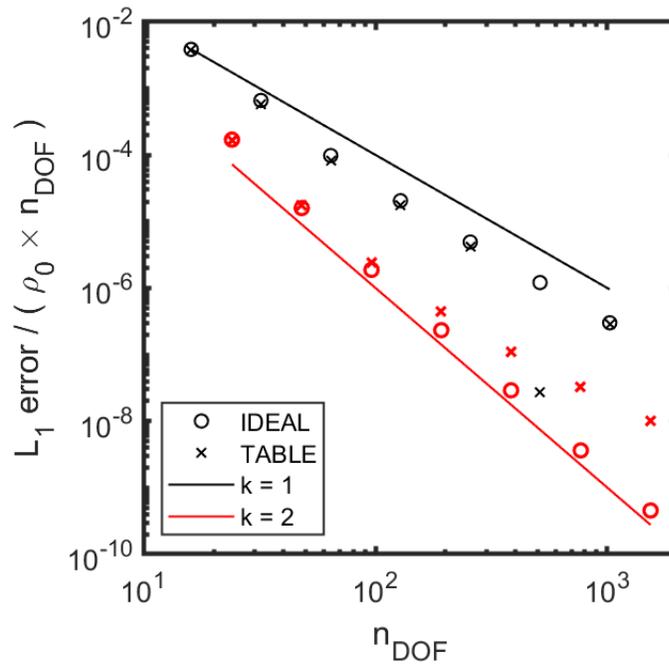
Figure 5.9 L_1 error between the initial and final states of an advected quartic sine wave, adopted from Suresh & Huynh (1997). The results are scaled by the number of degrees of freedom to obtain the average error per node. For the tabulated EoS results, the background density $\rho_0 = 10^{12}$ g cm^{-3} is also used to scale the error, but $\rho_0 = 1$ for the ideal case, which is run in dimensionless mode. The solid lines are proportional n_{DOF}^{k+1}, and serve as references for the convergence rates of solutions represented by polynomials of degree $k = 1$ and $k = 2$.
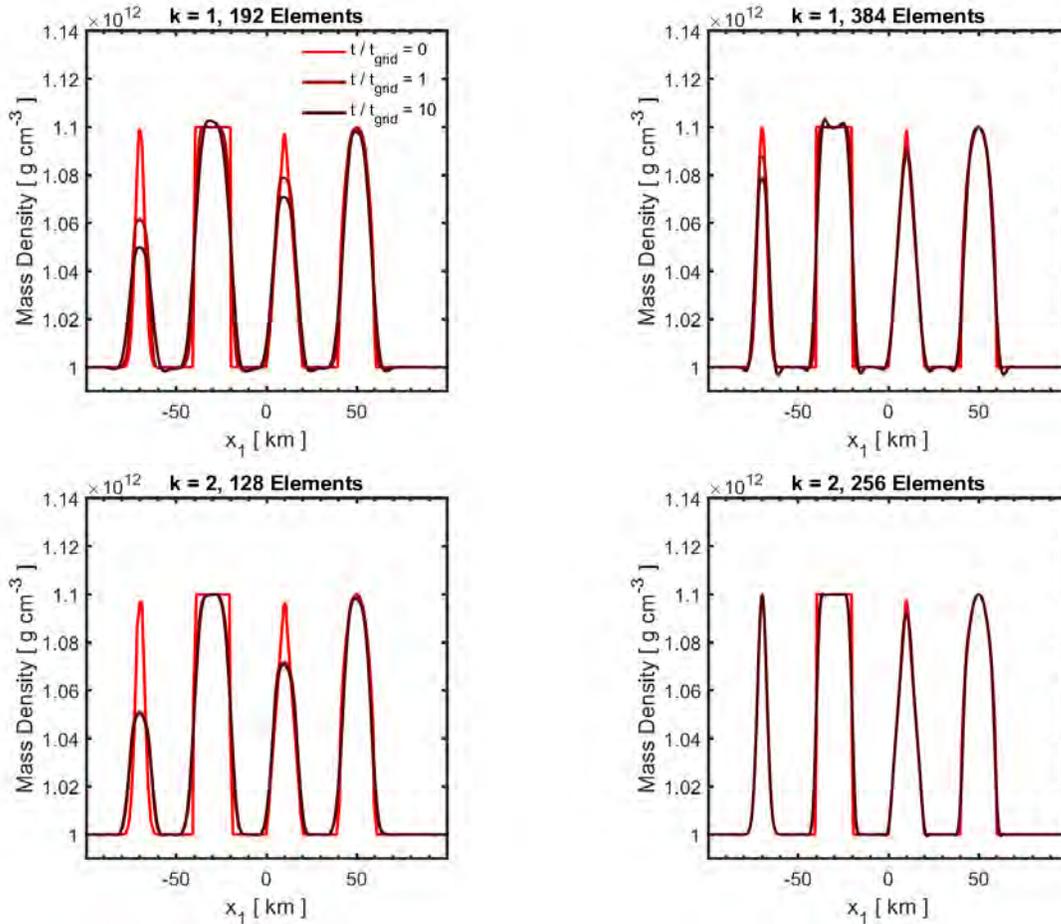
Figure 5.10 Mass density profiles for the discontinuous multi-wave advection test adopted from Suresh & Huynh (1997). In each panel we plot the initial condition (t/t_{grid} = 0; light red), the solution after one period (t/t_{grid} = 1; medium red), and after ten periods (t/t_{grid} = 10; dark red). (t_{grid} is the physical time required for one grid crossing.) In the top panels we plot results obtained with the second-order method (k = 1 and second-order SSP-RK time stepping) using 192 (left panel) and 384 (right panel) elements. In the bottom panels we plot results obtained with the third-order method (k = 2 and third-order SSP-RK time stepping) using 128 (left panel) and 256 (right panel) elements. Increasing the number of nodes and/or elements results in better resolution around sharp peaks.
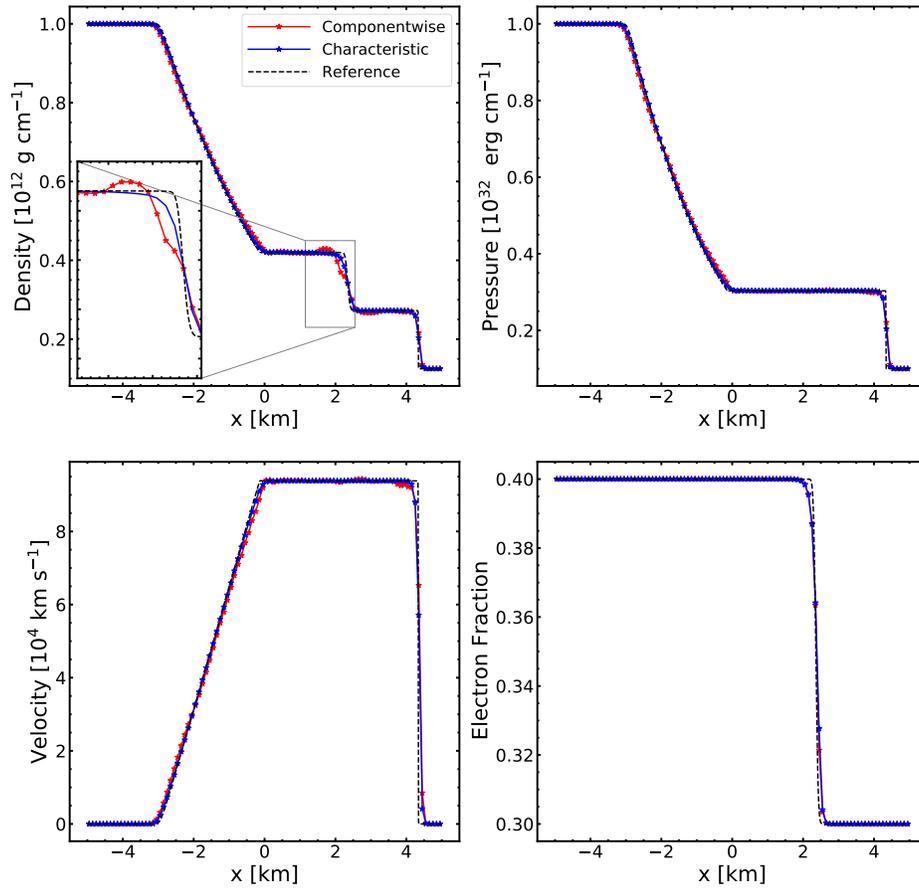
Figure 5.11 Numerical solution of the Sod shock tube at $t = 0.021$ ms using 100 elements and third-order accurate methods with characteristic (blue) and component-wise limiting (red) for density (upper left), pressure (upper right), velocity (lower left), and electron fraction (lower right), compared to a reference solution (black) using 10000 elements, obtained with first-order spatial discretization and third-order time integration.
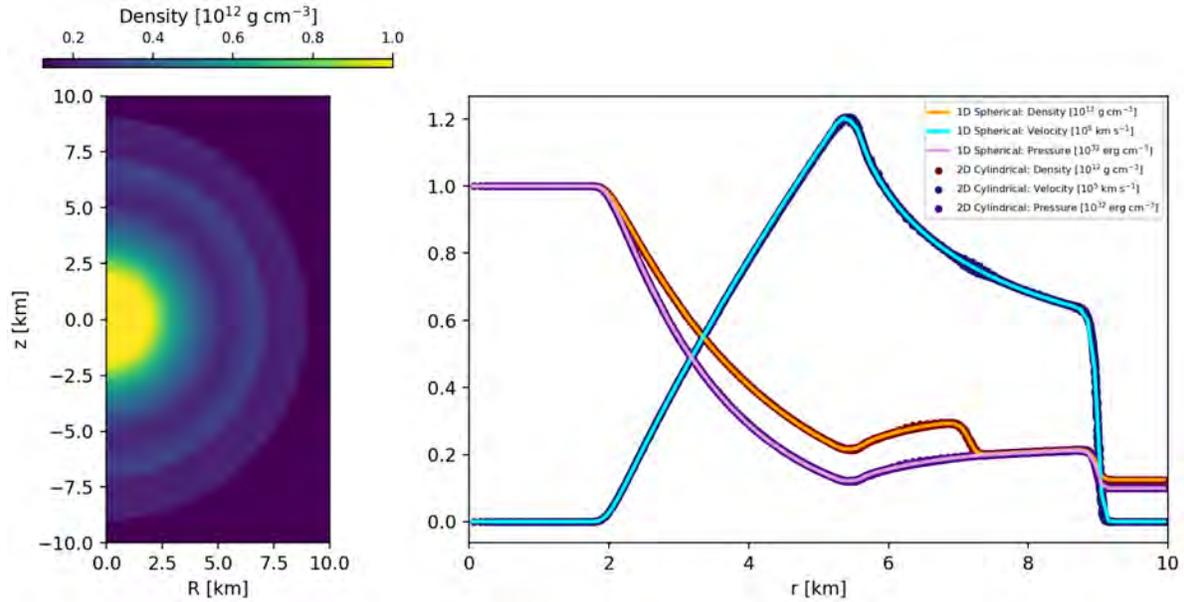
Figure 5.12 Two-dimensional density distribution (left panel), along with radial density, velocity, and pressure profiles (right panel) for the spherically symmetric Sod shock tube problem evolved to $t = 0.025$ ms using both 1D spherical-polar and 2D cylindrical coordinates; solid lines and scatter plots, respectively.
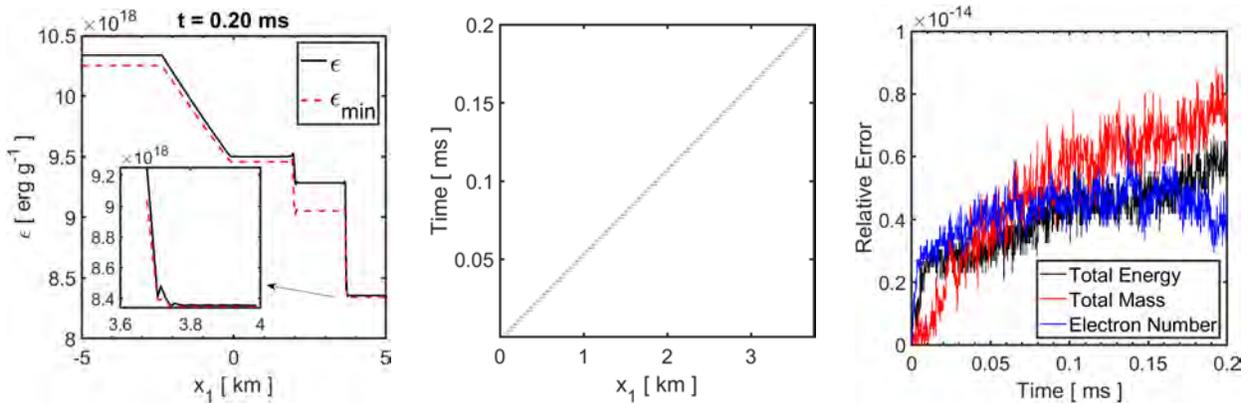
Figure 5.13 Numerical results for the shock tube provoking the bound-enforcing limiter. In the left panel we plot the specific internal energy (solid black line) and the minimum specific internal energy (dashed red line) versus position x. The middle panel shows the activation sites of the bound-enforcing limiter as indicated by a space-time plot of $\vartheta_3(x, t)$. The solution is closest to the boundary just ahead of the shock, as indicated by the inset in the left panel. In the right panel we plot the relative change in the conserved quantities; i.e., total fluid energy (black), mass (red), and electron number (blue).

179
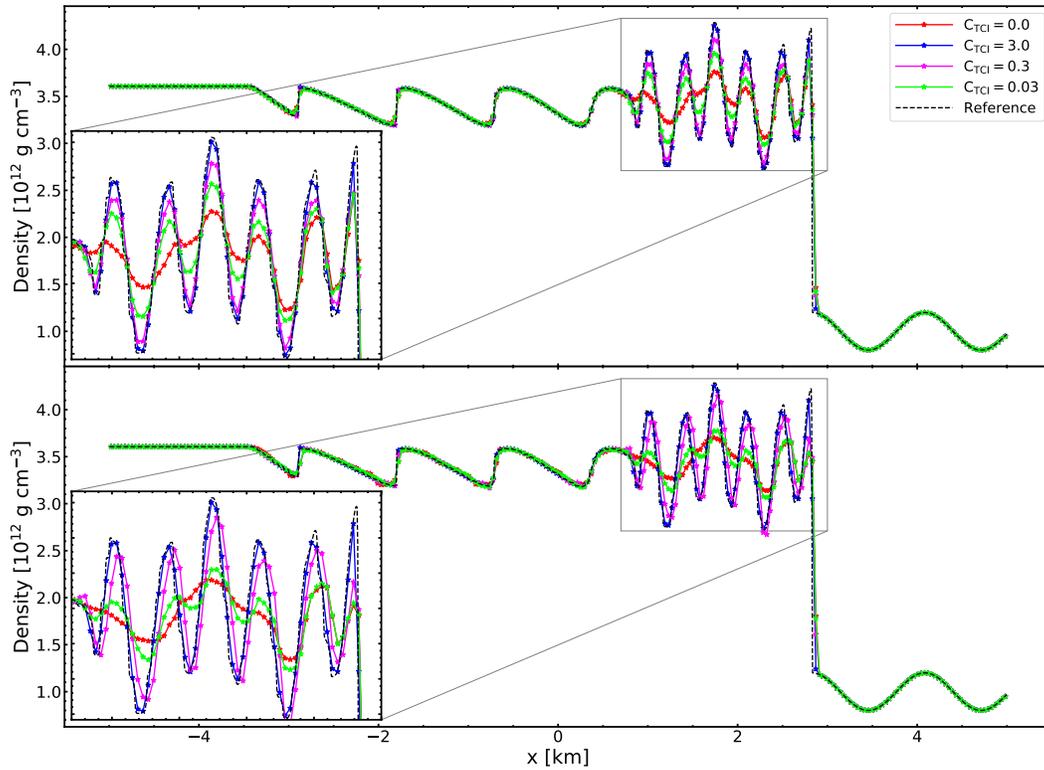
Figure 5.14 Numerical solution of the Shu-Osher shock tube with nuclear EoS at $t = 0.062$ ms, using 256 elements and third-order accurate methods with characteristic (top) and component-wise (bottom) limiting. In each panel, we plot the mass density versus position, obtained with various values of the troubled cell indicator threshold C_{TCI}: 0.0 (full limiting, red), 0.03 (green), 0.3 (magenta), and 3.0 (blue), compared to a reference solution (black) obtained using 2048 elements, third-order spatial discretization, and third-order time integration.
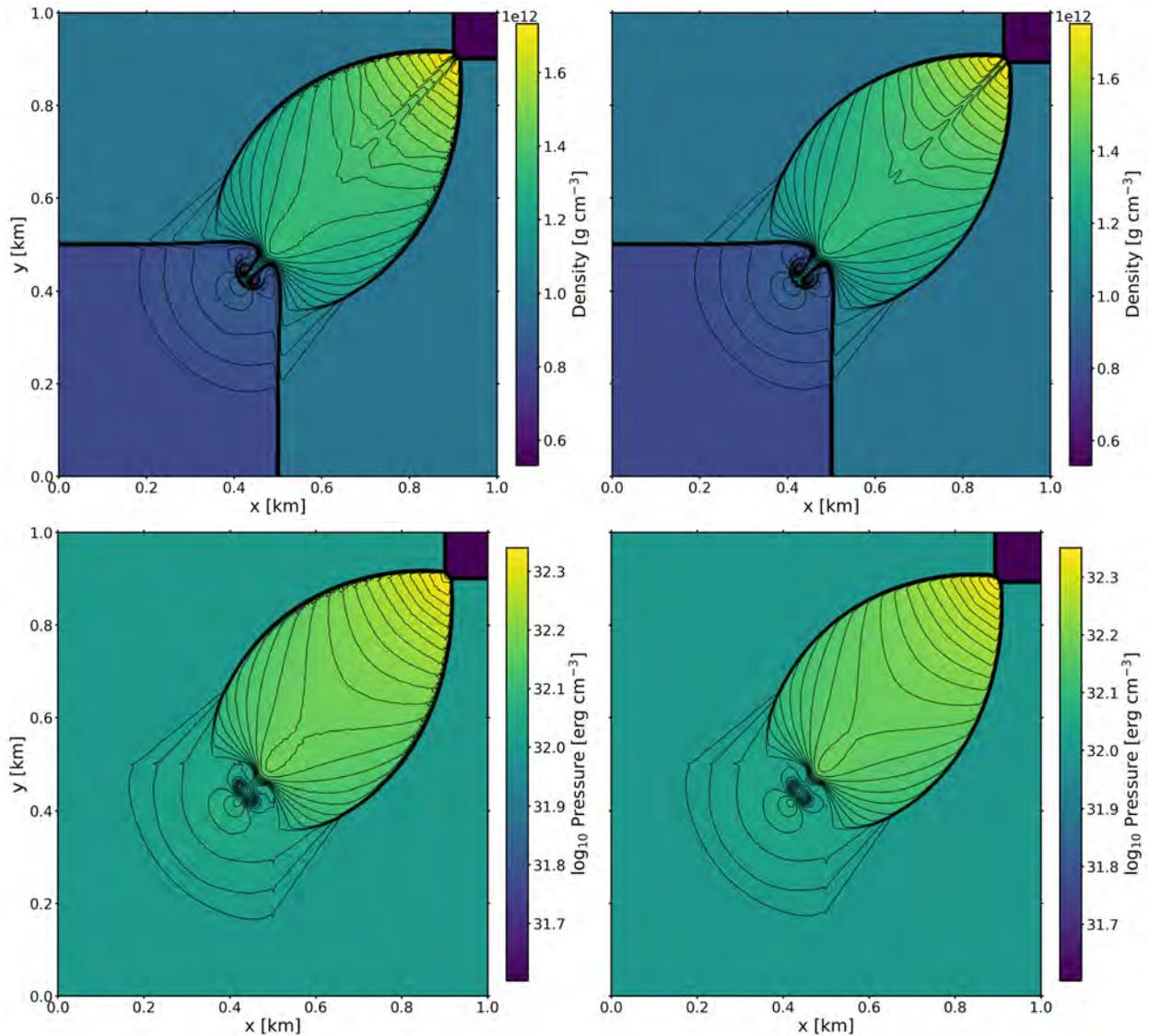
Figure 5.15 Numerical solution of a 2D Riemann problem (adopted from "Configuration 12" of Lax & Liu (1998)) with a nuclear EoS at $t = 0.0025$ ms using 400^2 elements and third order spatial and temporal discretization for density (top panels) and pressure (bottom panels). We compare results obtained with component-wise (left panels) and characteristic (right panels) limiting. Black lines on each plot show logarithmically spaced contours to highlight structures in the solutions.
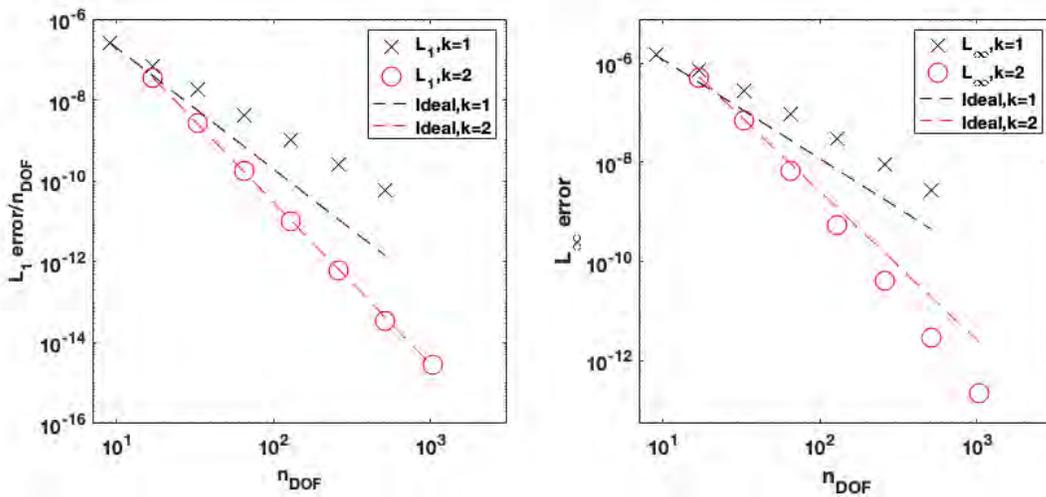
Figure 5.16 L_1 (left panel) and L_∞ (right panel) errors between the analytic and numerical solution calculated by the Poseidon solver for the case of a centrally condensed sphere. The L_1 error norms are scaled by the number of degrees of freedom to obtain an average error per node. The dashed lines are proportional to $1/n_{\mathrm{DOF}}^{k+1}$ and serve as references for the convergence rates of the numerical solutions.
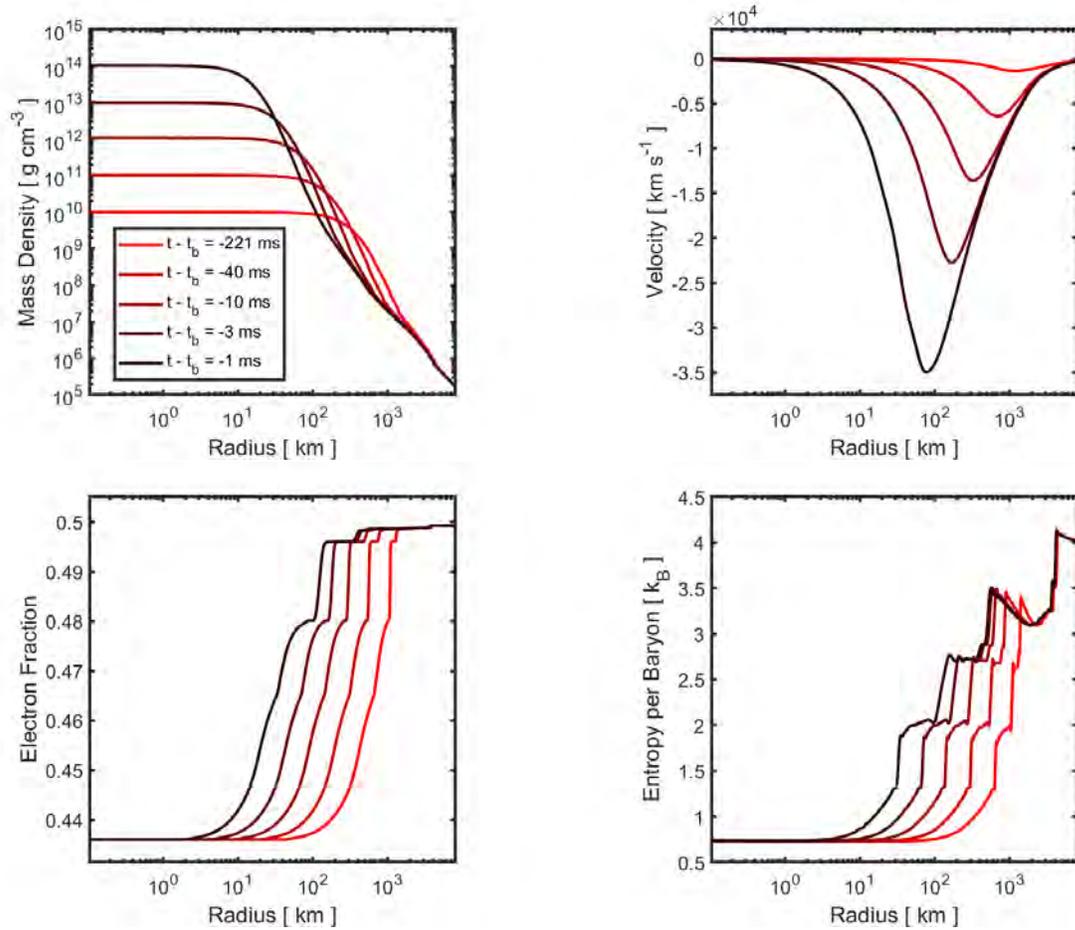
Figure 5.17 Numerical solutions for the adiabatic collapse of a 15 $M_\odot$ progenitor from Woosley & Heger (2007), obtained with THORNADO using 512 elements and a second-order DG scheme with component-wise limiting. Plotted versus radius are mass density (upper left), velocity (upper right), electron fraction (lower left), and entropy per baryon (lower right) during collapse. The time slices were chosen to depict the central mass density increasing by factors of 10.

183
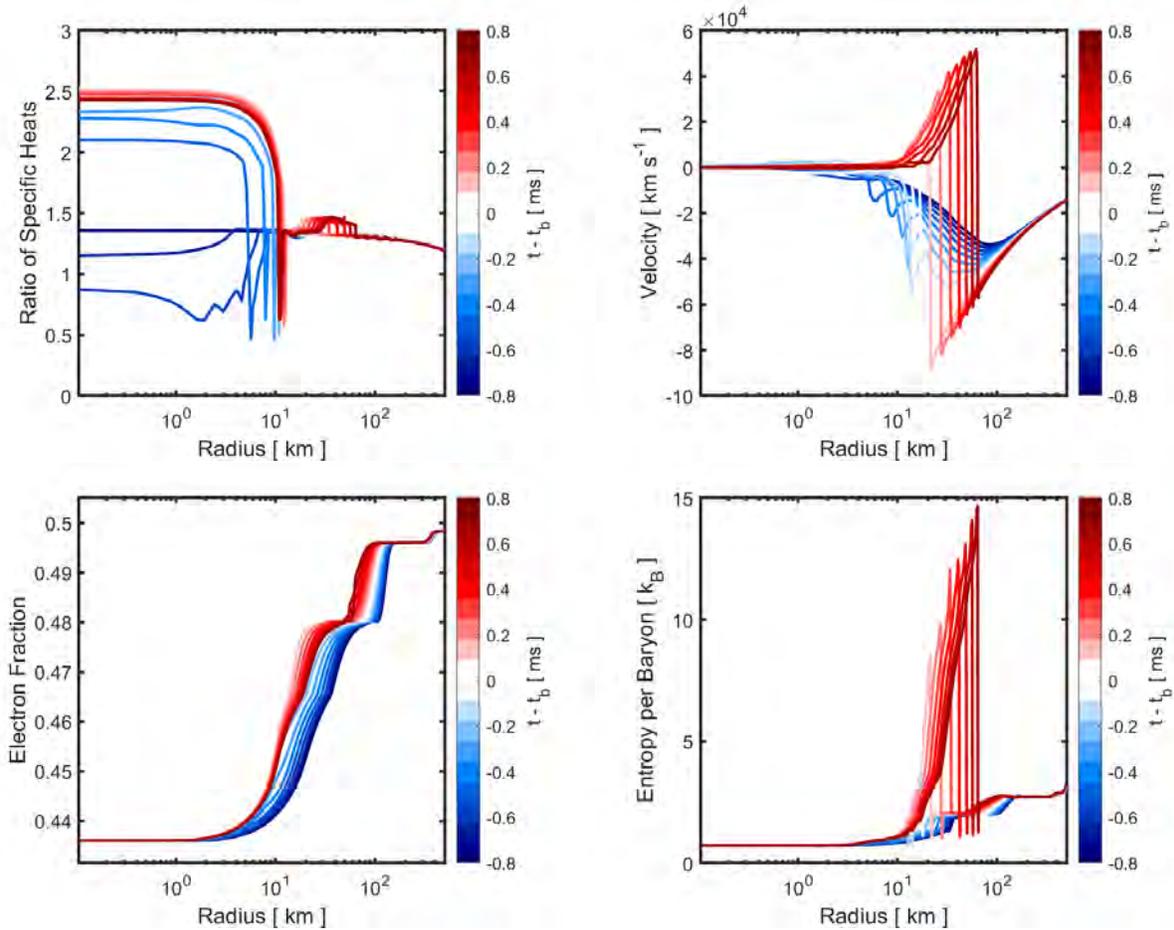
Figure 5.18 Numerical solutions of select quantities versus radius for adiabatic collapse evolved through bounce: adiabatic index $\Gamma \equiv \left(\frac{\partial \ln p}{\partial \ln \rho} \right)$ (upper left), velocity (upper right), electron fraction (lower left), and entropy per baryon (lower right). A finer time resolution is used here to exhibit the characteristics of bounce and shock formation, and the color map on the right of each panel is used to distinguish pre- and post-bounce profiles; blue and red, repsectively.
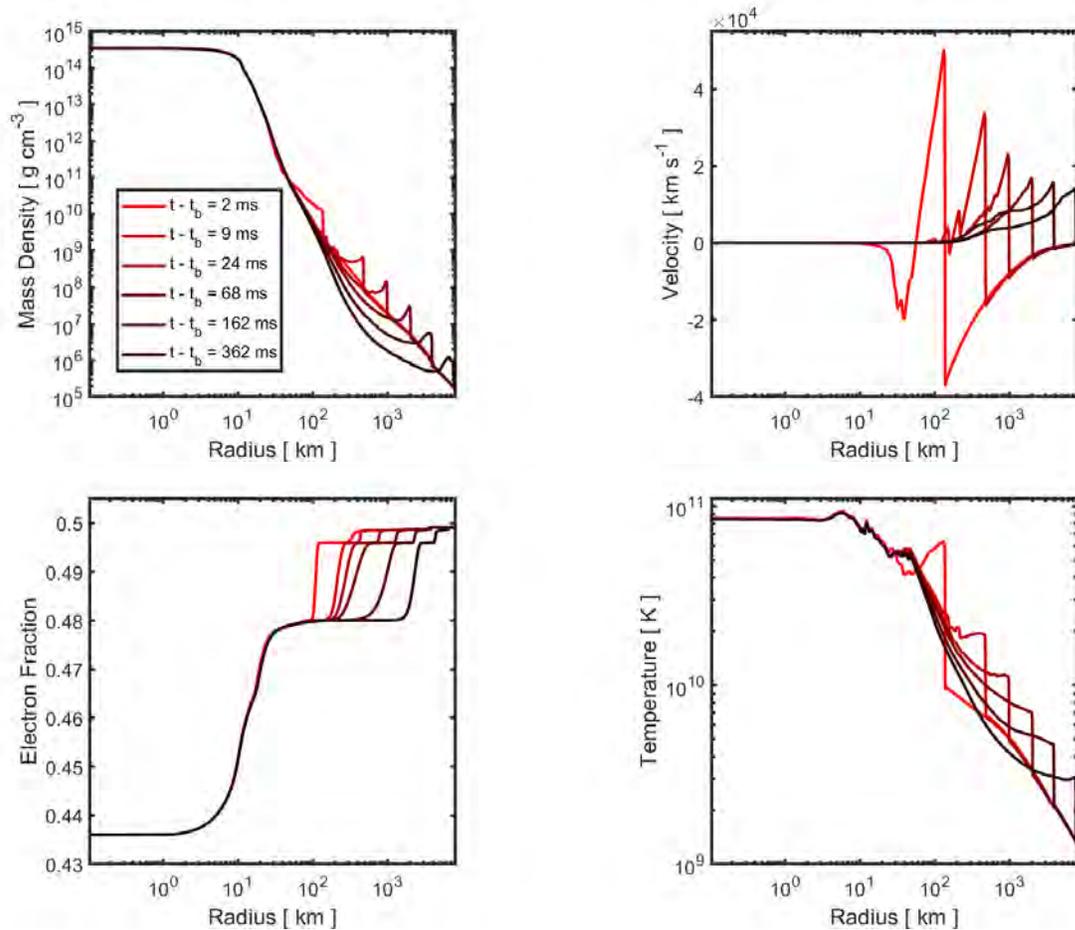
Figure 5.19 Numerical solutions for mass density (upper left), velocity (upper right), electron fraction (lower left), and temperature (lower right) versus radius at select times for the adiabatic collapse simulation evolved for several hundred milliseconds post bounce. This time domain partially captures the structure of the core as the shock propagates from its origin to the outer boundary.
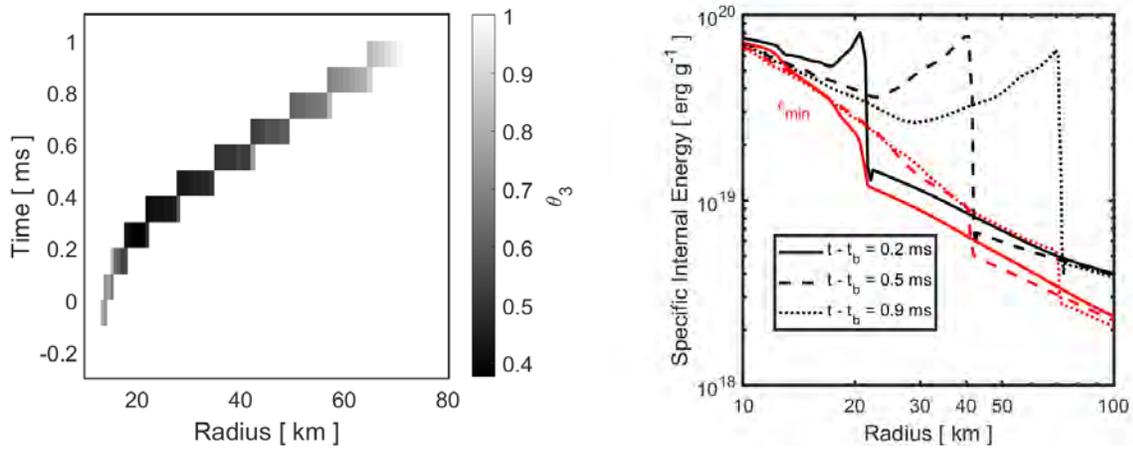
Figure 5.20 Activation of the bound-enforcing limiter in the fiducial adiabatic collapse simulation. The left panel shows the value of the limiter parameter ϑ_3 from Equation (5.74) in space and time. In the right panel we plot the solution for ϵ (black) and the minimum ϵ_{min} (red), described in Section 5.4.4. Each profile captures a moment in time briefly after bounce, when the bound-enforcing limiter is required to maintain $\epsilon > \epsilon_{min}$.
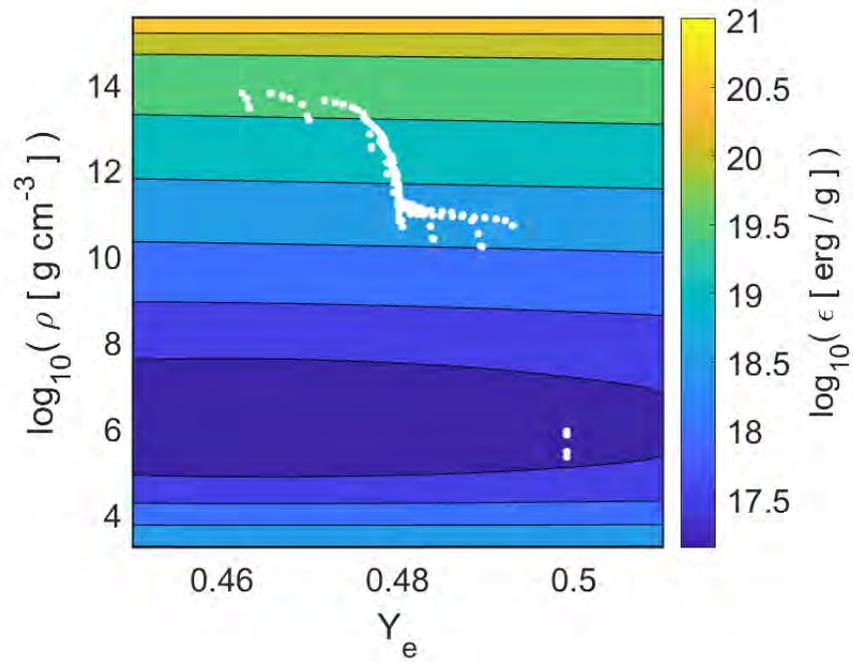
Figure 5.21 Activation sites (white dots) of the bound-enforcing limiter during the adiabatic collapse simulations in the ρY_e-plane, placed over a contour plot of the surface defined by $\epsilon_{min} = \epsilon(\rho, T_{min}, Y_e)$. The points in the mass density range $\log_{10}(\rho) \in [10, 14]$ show the limiter being activated during bounce and shock formation. The limiter is again briefly applied in the low density region, $\log_{10}(\rho) \in [5, 6]$. This corresponds to when the shock momentarily forces the solution below the lower EoS table boundary as the shock passes through the outer boundary of the spatial domain.
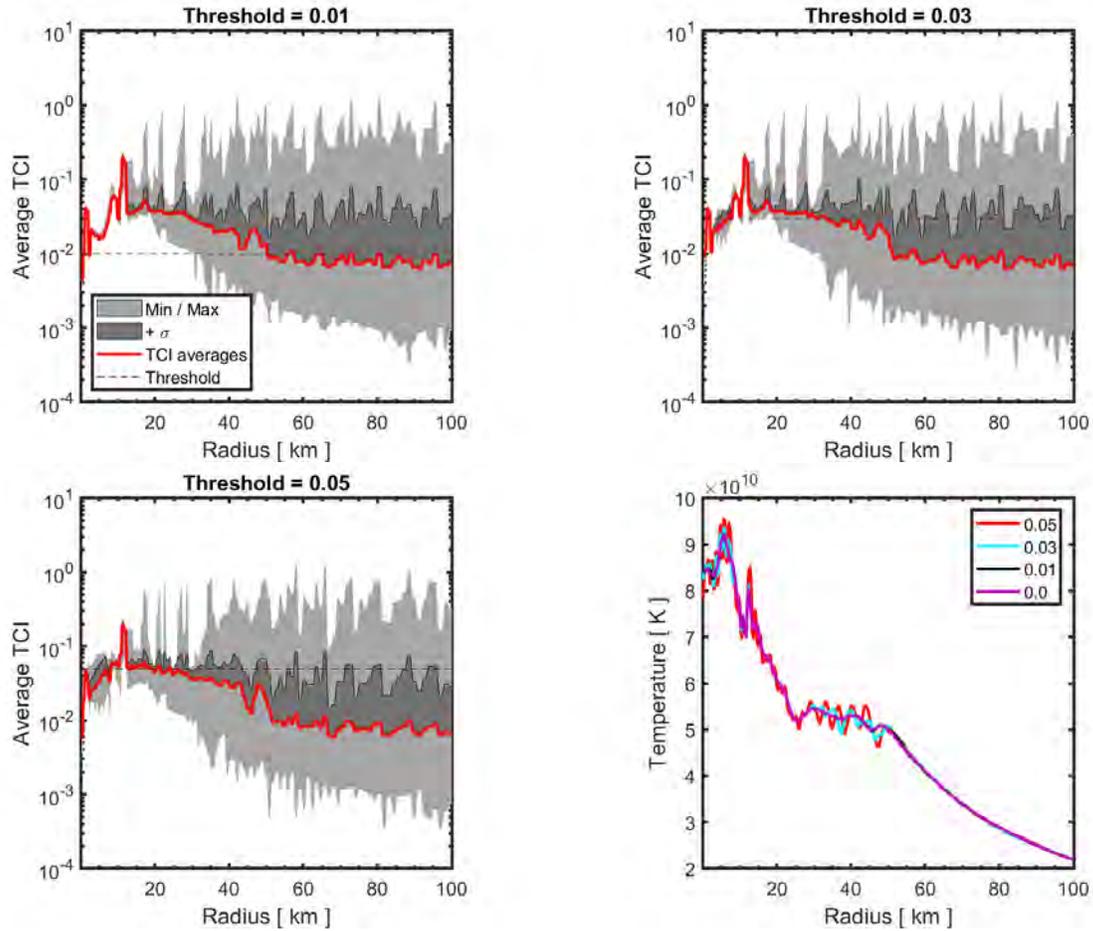
Figure 5.22 Troubled-cell indicator values for adiabatic collapse simulations with various thresholds C_{TCI} (upper panels and lower left panel). In each panel, the red curve represents the time-averaged troubled-cell indicator value (averaged from t_{b} to $t_{\text{end}} - t_{\text{b}} = 497.1$ ms of the simulation). The lighter gray-shaded regions represent the extreme TCI values in each element (taken over all post-bounce times). The darker shaded region represents the positive TCI standard deviation. In the lower right panel, the temperature at the end of each simulation with different C_{TCI} is plotted versus radius. A higher threshold results in less slope limiting, which allows for some oscillations to develop in the temperature profile.
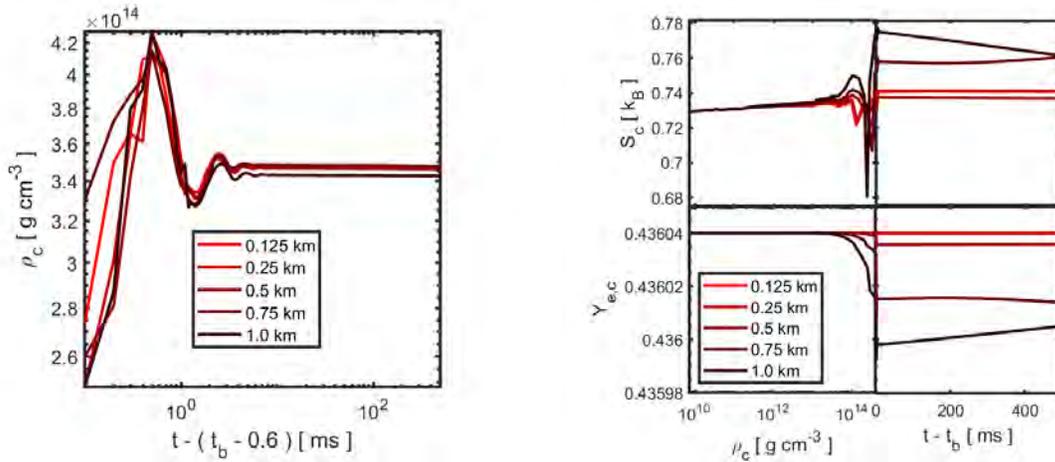
Figure 5.23 Results from adiabatic collapse simulations where the innermost cell width has been varied. The left panel shows the central density as a function of time for various Δr_1. The right panel shows the central entropy (top) and central electron fraction (bottom) versus central density (up to its maximum value). Beyond the maximum central density, the entropy and electron fraction are plotted versus time.
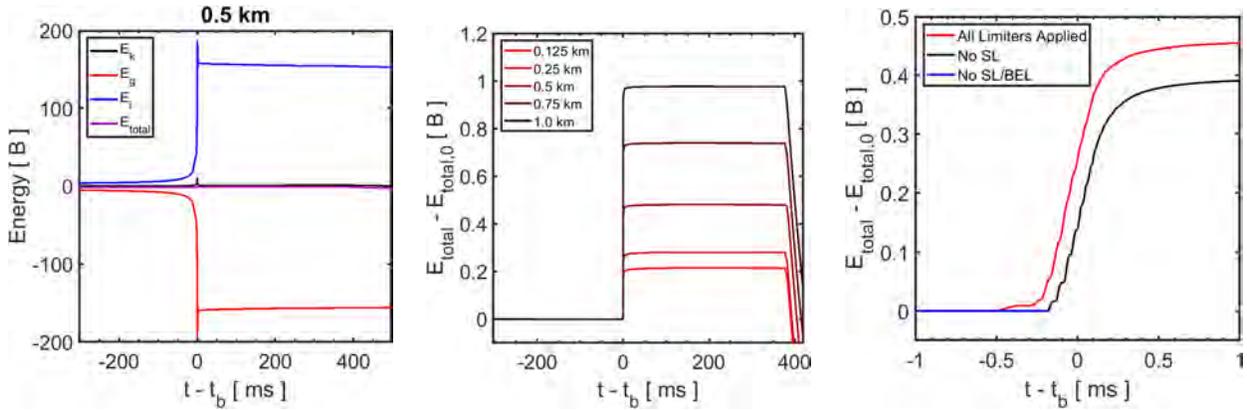
Figure 5.24 Energy conservation by the RKDG method in THORNADO for adiabatic collapse simulations. The left panel shows gravitational (red), kinetic (black), internal (blue), and total (magenta) energy versus time for the fiducial run with $\Delta r_1 = 0.5$ km. Due to the relative magnitude of E_i and E_g, the details in the kinetic and total energies are obscured. The middle panel shows the change in total energy versus time for all the resolutions considered in Section 5.6.6. The decrease in the total energies around $t - t_b \approx 375$ ms is due to the bounce shock reaching the outer boundary of the domain. The right panel shows the total energy versus time for models with various combinations of limiters enabled for the fiducial run. The red line represents the total energy when applying both the slope limiter and the bound-enforcing limiter. The black line shows this quantity when only applying the bound enforcing limiter. The blue line represents a model with both limiters off, which eventually crashed at bounce.

189
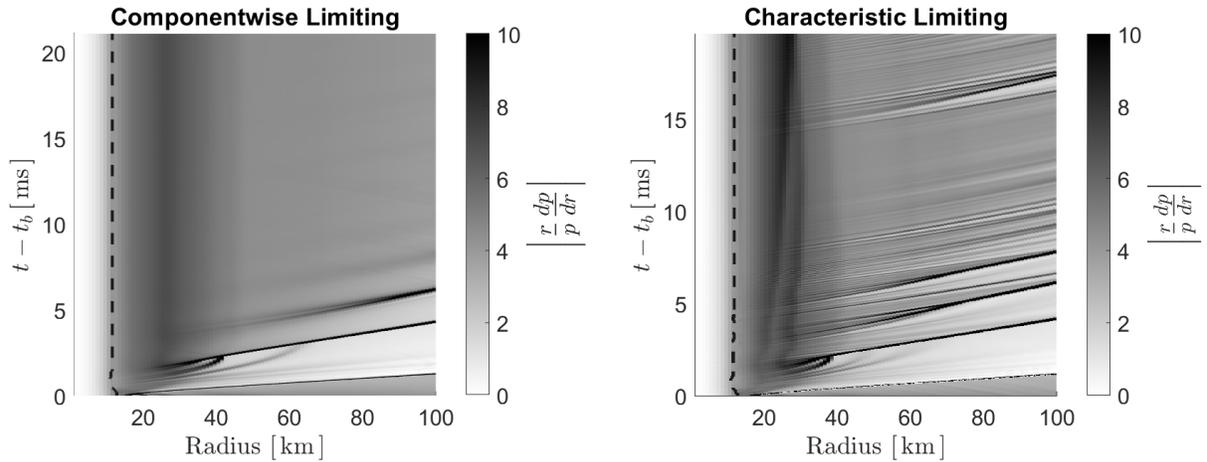
Figure 5.25 Space-time plots of the absolute value of the logarithmic pressure gradient for simulations employing component-wise (left) and characteristic (right) limiting. The time domain extends over a brief period after bounce $t \in [t_b, \ t_b + 20\,\mathrm{ms}]$. The vertical dashed line around 10 km, which traces the minimum of the adiabatic index Γ_{min}, represents the approximate position of the phase transition. Near the bottom of each figure, the black line extending from approximately 20 km to 100 km for a duration of 1 ms traces the bounce shock. The lines which form after this are traces of secondary or tertiary "ripples" that propagate outward from the inner core, and follow the shock shortly after bounce. With component-wise limiting, the pressure gradient is relatively smooth at about 10 ms after bounce and thereafter. With characteristic limiting, perturbations continue to develop in the solution after bounce, which is visible as noise in the pressure gradient. A high time resolution was used in this case to better capture details of the dynamics around the phase transition, such as the formation of acoustic waves and their reflections off the inner boundary.
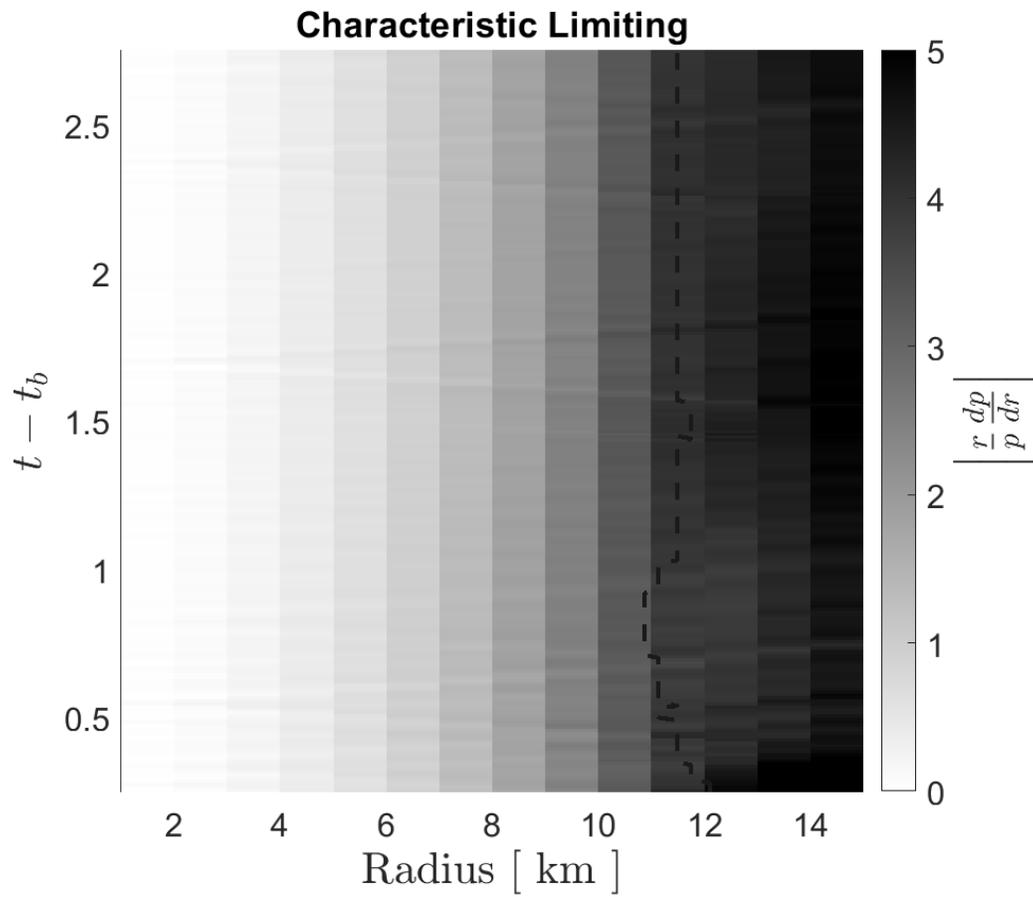
Figure 5.26 Zoomed-in view of the logarithmic pressure gradient (to better capture details of the dynamics around the phase transition) for the simulation employing characteristic limiting shown in the right panel in Figure 5.25.

CHAPTER 6

SINGULARITY-EOS : PERFORMANCE PORTABLE EQUATIONS OF STATE AND MIXED CELL CLOSURES

> Every day, once a day, give
>
> yourself a present.
>
> ───────────────────────
>
> Dale Cooper, *Twin Peaks*

This chapter is based on work recently submitted to the Journal of Open Source Software.

ABSTRACT

We present SINGULARITY-EOS , a new performance-portable library for equations of state and related capabilities. SINGULARITY-EOS provides a large set of analytic equations of state, such as the Gruneisen equation of state, and tabulated equation of state data under a unified interface. It also provides support capabilities around these equations of state, such as Python wrappers, solvers for finding pressure-temperature equilibrium between multiple equations of state, and a unique modifier framework, allowing the user to transform a base equation of state, for example by shifting or scaling the specific internal energy. All capabilities are performance portable, meaning they compile and run on both CPU and GPU for a wide variety of architectures.

6.1 Introduction

When expressed mathematically for continuous materials, the laws of conservation of mass, energy, and momentum form the Navier-Stokes equations of fluid dynamics. In the limit of zero molecular viscosity, they become the Euler equations. These laws have been used to describe phenomena as disparate as flow of air over an airplane wing, bacterial motion in fluids, and the cataclysmic deaths of stars. However, the fluid equations are not complete, and the system must be *closed* by a description of the material at a sub-continuum (e.g., molecular or atomic) scale. This closure is commonly called the *equation of state* (EOS).

Equations of state vary from the simple ideal gas law, to sophisticated descriptions multi-phase descriptions of the lattice structure of ice or wood, to models of quark-gluon plasma and nuclear pasta at ultra high densities. A common form to write an equation of state is as a pair of relations:

$$p = p(\rho, T, \vec{\lambda}) \text{ and } \varepsilon = \varepsilon(\rho, T, \vec{\lambda}), \qquad (6.1)$$

which relate the pressure p and specific internal energy ε to density ρ, temperature T, and potentially some unknown set of additional quantities $\vec{\lambda}$. However, other representations are possible, and in common parlance an EOS is the collection of knowledge needed to reconstruct some intrinsic thermodynamic quantities from others. For example, the speed of sound through a material or the specific heat capacity, which are thermodynamic derivatives of the pressure and the specific

internal energy, are both determined by the EOS.

In multi-material fluid dynamics simulations, one often will end up with a so-called *mixed cell*, where two materials exist within the same simulation zone. This can be an artifact of the numerical representation; for example a steel bar and the surrounding air may end up sharing a finite volume cell if the boundaries of the cell do not align exactly with the surface of the steel bar. Or it may represent physical reality; for example, air is a mixture of nitrogen and oxygen gases, as well as water vapor. Regardless of the nature of the mixed cell, one must somehow provide to the fluid code what the material properties of the cell are as a whole. This is called a *mixed cell closure*. One such closure is *pressure-temperature equilibrium* (PTE), where all materials in the cell are assumed to be at the same pressure and temperature.

6.2   State of the Field

Typically fluid dynamics codes each develop an EOS package individually to meet a given problem's needs. Databases of tabulated equations of state, such as the Sesame (Lyon & Johnson, 1992) and Stellar Collapse (O'Connor & Ott, 2010a) databases often come with tabulated data readers, for example, the EOSPAC library (Pimentel, 2021) and Stellar Collapse library (O'Connor & Ott, 2010b). However, these libraries typically do not include analytic equations of state or provide a unified API. They also don't provide extra equation-of-state capabilities, such as equilibrium solvers or production hardening. With a few exceptions, these libraries are also typically not GPU-capable.

We present SINGULARITY-EOS , which aims to be a "one stop shop" for EOS models for fluid and continuum dynamics codes. It provides a unified interface for both analytic and tabulated equations of state. It also provides useful surrounding capabilities, such as Python wrappers, `modifiers`, which allow the user to transform a given EOS, and solvers which can find the state in which multiple EOS's are in PTE. To support usability, the library is extensively documented and tested and supports builds through both `cmake` and `Spack` (Gamblin et al., 2015).

SINGULARITY-EOS leverages the "Kokkos" (Edwards et al., 2014; Trott et al., 2021, 2022)library for performance portability, meaning the code can run on both CPUs and GPUs, as well as other

accelerators. This fills an important need, as modern super computing capabilities increasingly rely on GPUs for performance. SINGULARITY-EOS is now used in the ongoing open-source PHOE-BUS (Chapter 7) [1] project which has a separate code paper in-prep.

6.3 Design Principles and Feature Highlights

Here we enumerate several design principles underlying SINGULARITY-EOS , and highlight a few feature of the library.

6.3.1 Flexibility in loop patterns

SINGULARITY-EOS provides both scalar and vector APIs, allowing the user to make EOS calls on both single points in thermodynamic space, and on collections of points. The vector calls may be more performant (as they may vectorize), however care is made to ensure both APIs operate at acceptable performance, to accommodate different code structures downstream.

6.3.2 Flexibility in memory layout

The vector calls in SINGULARITY-EOS use an *accessor* API and (with a few exceptions) accept any C++ object that has a "operator[]" function defined. This allows users to lay out their memory as they see fit and use SINGULARITY-EOS even on strided or sparsely allocated memory.

6.3.3 Expose APIs to aid performance

Many equations of state are most naturally represented as functions of density and temperature. However, fluid codes require pressure as a function of density and internal energy. Extracting this often requires computing a root find to invert the relation

$$\varepsilon = \varepsilon(\rho, T). \tag{6.2}$$

In these cases, we expose an initial guess for temperature, which helps the solution rapidly converge. Similarly, the performance of a sequence of EOS calls may depend on the ordering of the calls. For example, if both temperature and pressure are required from an equation of state that requires inversion, requesting pressure first will be less performant than requesting temperature

---

[1] https://github.com/lanl/phoebus

195

first, as the former requires two root finds, and the latter requires only one. To enable this, we expose a function `FillEos`, in which the user may request multiple quantities at once, and the code uses ordering knowledge to compute them as performantly as possible.

6.3.4 Performance-portable polymorphism

Accelerators provide new challenges to standard object-oriented programming. In particular, not all compiler stacks (such as Sycl (Reyes et al., 2020) or OpenMP Target Offload (Chandra et al., 2001)) support relocatable device code, which is required for standard C++ polymorphism. Even in programming models, such as CUDA (NVIDIA et al., 2020), which do support relocatable device code, polymorphism can be slower than naively expected, and the user-level API can be cumbersome, requiring operations such as `placement new`. To sidestep these issues, we use the C++ language feature `std::variant` to implement a polymorphism mechanism that works on device.

6.3.5 Modifiers

A given code may need to modify an EOS model to make it suitable for a given application. For example, the zero-point of the energy may need to be shifted, a porosity model may need to be added, or the unit system may need to be changed. We implement this with a system of `modifiers`, which can be applied on top of an EOS in a generic way. Modifiers may also be chained.

6.3.6 Fast log-lookups

To span the required orders of magnitude, tabulated equations of state are often tabulated on log-spaced grids. Logarithms and exponentials are, however, expensive operations and the performance of lookups can suffer. We instead use the not-quite-transcendental lookups described in Miller et al. (2022) to significantly enhance performance of log-like lookups.

6.3.7 Extensibility via modular parts and plugins

SINGULARITY-EOS is designed to be extensible. The `std::variant`-based polymorphism, combined with modifiers, as described above, already provides significant flexibility. However, downstream codes may wish to add functionality to the library. This may be implemented in several

ways. First, as SINGULARITY-EOS is open source, contributions from downstream developers are welcome. Second, a C++ code that depends on SINGULARITY-EOS may implement their own models and include them in a local variant object. SINGULARITY-EOS provides tooling to build variants up iteratively. Finally, SINGULARITY-EOS provides a flexible plugin infrastructure that allows downstream users to add capability to the core library locally by telling the build system to include a locally downloaded plugin. This final capability allows downstream users to share code with each other, even when committing that code to SINGULARITY-EOS proper is not possible due to, e.g., licensing issues.

CHAPTER 7

PHOEBUS: PERFORMANCE PORTABLE GRRMHD FOR RELATIVISTIC ASTROPHYSICS

> Burbidge, Burbidge, Fowler, Hoyle
>
> Took the stars and made them toil:
>
> Carbon, copper, gold, and lead
>
> Formed in stars, is what they said
>
> —————————————
>
> Ken Croswell

ABSTRACT

We introduce the open source code Phoebus (**ph**ifty **o**ne **e**rgs **b**lows **u**p a **s**tar) for astrophysical general-relativistic radiation magneto hydrodynamic simulations. Phoebus is designed for, but not limited to, high-energy astrophysical environments such as core-collapse supernovae, neutron star mergers, black-hole accretion disks, and similar phenomena. Phoebus is built on the parthenon (Grete et al., 2022) performance portable adaptive mesh refinement framework, is GPU capable, and capable of modeling a large dynamic range in space and time. We describe the physical model employed in Phoebus, the numerical methods used, and demonstrate a suite of test problems to demonstrate its abilities. We apply Phoebus to a problem of astrophysical interest – a relativistic black hole MHD accretion disk problem using a nuclear equation of state and neutrino transport.

7.1 Introduction

Compact objects such as neutron stars and black holes, through their formation channels or interactions with their environments, power some of the most energetic phenomena in the universe. Core-collapse supernovae (CCSNe), gamma-ray bursts, neutron star (NS) mergers, X-ray binaries, and quasars, to name a few, compose some of the most energetic phenomena observed. These events are linked to all of the post-Big Bang nucleosynthesis, the chemical and dynamical evolution of galaxies, and comprise many of the compact object formation channels. Furthermore, these phenomena probe matter at its most extreme, acting as grand laboratories for fundamental physics.

Our understanding of these phenomena relies on the union of theory and observation. For the former, computational methods are an essential tool necessary for modeling the underlying physics. However, these environments each have spatial and temporal scales that span many orders of magnitude on their own. Combined, these problems span such a range in spatio-temporal scale that was intractable in a single software, generally demanding instead specialized codes, each tuned for a specific problem of interest.

We present Phoebus (**ph**ifty **o**ne **e**rgs **b**lows **u**p a **s**tar), a new general-relativistic radiation magnetohydrodynamics (GRRMHD) code developed for modeling systems in relativistic astrophysics.

PHOEBUS includes all of the physics necessary to model these systems, including accurate radiation transport for both photon and neutrino fields, constrained-transport GRMHD, a wide variety of equations of state including those of dense nuclear matter, and the ability to model a wide dynamic range in space and time through adaptive mesh refinement and a GPU-resident development strategy.

An additional challenge, separate from numerically modeling the rich physics necessary, is the need to do so efficiently, across a *diverse range* of computing architectures – so called performance portability. Computing resources are becoming increasingly heterogeneous with compute nodes being comprised of both CPUs and GPUs and each GPU vendor supporting their own programming model and software stack. Hence, modern high performance simulation software must not only be able to leverage these architectures, but do so efficiently. To enable this, PHOEBUS is built upon PARTHENON [1], a performance portable, block-structured adaptive mesh refinement (AMR) library (Grete et al., 2022). PARTHENON, in turn, uses KOKKOS (Edwards et al., 2014; Trott et al., 2021, 2022), a hardware agnostic performance portability abstraction library, for on-node parallelism. This enables the user to, at compile time, select the target hardware, and KOKKOS specializes the relevant code to the target hardware. KOKKOS also exposes fine grained tuning of loop patterns to enable platform specific optimizations.

PHOEBUS adopts a fully free-and-open-source development model. Making scientific software open source constitutes good scientific practice as it enables transparency, full reproducability, and ultimately enables more science through serving the community. The code is publicly available[2] and developed on GitHub. We welcome, and hope for, bug reporting, issue tracking, feature or pull requests, and general feedback from the community. Continuous integration and unit testing are enabled with the Catch2[3] unit testing framework and all pull requests are reviewed before merging into the main codebase. PHOEBUS includes an expansive, and growing, suite of unit and regression tests that stress simple compilations and functionalities to large multiphysics problems.

---

[1]https://github.com/parthenon-hpc-lab/parthenon
[2]https://github.com/lanl/phoebus
[3]https://github.com/catchorg/Catch2

The software is licensed under the 3-clause Berkeley Software Distribution (BSD-3) clause which has relaxed rules for distribution.

In Section 7.2 we lay out the full system of equations that Phoebus is currently designed to solve. In Section 7.3 we describe the numerical methods used for each physics sector. In Section 7.4 we present a suite of tests designed to stress and verify the fidelity of Phoebus. Finally we offer concluding thoughts in Section 7.5 and discuss the future direction of Phoebus as well as its position as an open-source software.

7.2 Physical Model

In Phoebus we adopt the general relativistic Euler equations of magnetohydrodynamics, supplemented by an appropriate, but flexible, equation of state and set of radiative opacities. The relevant systems of equations and physical assumptions are given below. In all of the following, Greek indices run from 0 to 3 and Latin indices run from 1 to 3. We adopt the Einstein summation convention for repeated indices.

7.2.1 GRRMHD

For the fluid equations, we adopt the Valencia Formulation (Banyuls et al., 1997; Font et al., 2000) as summarized in Giacomazzo & Rezzolla (2007). We solve the conservation law

$$(\sqrt{\gamma}\mathbf{U})_{,t} + (\sqrt{-g}\mathbf{F}^i)_{,i} = \sqrt{-g}\mathbf{S} \tag{7.1}$$

for conserved vector

$$\mathbf{U} = \begin{pmatrix} D \\ S_j \\ \tau \\ B^k \end{pmatrix} = \begin{pmatrix} \rho W \\ (\rho h + b^2)W^2 v_j - \alpha b^0 b_j \\ (\rho h + b^2)W^2 - (p + b^2/2) - \alpha^2 (b^0)^2 - D \\ B^k \end{pmatrix}, \tag{7.2}$$

flux vector

$$\mathbf{F}^i = \begin{pmatrix} D\tilde{v}^i/\alpha \\ S_j \tilde{v}^i/\alpha + (p + b^2/2)\delta^i_j - b_j B^i/W \\ \tau \tilde{v}^i/\alpha + (p + b^2/2)v^i - \alpha b^0 B^i/W \\ B^k \tilde{v}^i/\alpha - B^i \tilde{v}^k/\alpha \end{pmatrix}, \tag{7.3}$$

201

and source vector

$$
\mathbf{S} = \begin{pmatrix} 0 \\ T^{\mu\nu}(g_{\nu j,\mu} - \Gamma^{\delta}_{\nu\mu}g_{\delta j}) + G_{\nu} \\ \alpha(T^{\mu 0}(\ln\alpha)_{,\mu} - T^{\mu\nu}\Gamma^{0}_{\nu\mu}) + G_{\nu} \\ 0^k \end{pmatrix},
\tag{7.4}
$$

where u^{μ} is the four-velocity of the fluid,

$$
v^i = \frac{u^i}{W} + \frac{\beta^i}{\alpha}
\tag{7.5}
$$

is the 3-velocity, with densitized 3-velocity

$$
\tilde{v}^i = \alpha v^i - \beta^i,
\tag{7.6}
$$

for Lorentz factor

$$
W = \alpha u^0,
\tag{7.7}
$$

lapse α, shift β^i, magnetic field four-vector b^{μ} defined by

$$
b^{\mu}u^n u - b^{\nu}u^m u = *F^{\mu\nu}
\tag{7.8}
$$

for the Hodge star of the Maxwell stress tensor $*F^{\mu\nu}$, baryon number density ρ, specific enthalpy h, Christoffel symbols $\Gamma^{\mu}_{\nu\sigma}$, four-metric $g_{\mu\nu}$, three-metric $\gamma_{\mu\nu}$, and stress-energy tensor

$$
T^{\mu\nu} = (\rho + u + P + b^2)u^{\mu}u^{\nu} + \left(P + \frac{1}{2}b^2\right)g^{\mu\nu} - b^{\mu}b^{\nu},
\tag{7.9}
$$

for pressure P. G_{ν} is the radiation 4-force including radiation-matter interactions. The magnetic field four-vector b^{μ} is related to the Eulerian observer magnetic field 3-vector B^i by

$$
b^0 = \frac{W}{\alpha}B^i v_i
\tag{7.10}
$$

$$
b^i = \frac{B^i + \alpha b^0 u^i}{W}
\tag{7.11}
$$

$$
b^2 = b^{\mu}b_{\mu} = \frac{B^2 + \alpha^2(b^0)^2}{W^2}.
\tag{7.12}
$$

We also track a *primitive* vector

$$\mathbf{P} = \begin{pmatrix} \rho \\ W v^i \\ \rho \epsilon \\ B^k \end{pmatrix}, \tag{7.13}$$

which is used for reconstructions and to compute fluxes.

In addition to the above equations, Phoebus supports the evolution of arbitrary passive scalars

$$(X \rho u^\mu)_{;\mu} = S \tag{7.14}$$

where X is some advected quantity that is neither intrinsic nor extrinsic, S, is some potentially non-zero source term, and the notation $\phi^\mu_{;\mu}$ denotes the covariant derivative. In Phoebus, we use this framework to model the lepton exchange between the matter and neutrino radiation fields, taking X to be the electron fraction Y_e and S to be $\sqrt{g} G_{ye}$, with G_{ye} capturing the rate of transfer.

7.2.2 Equation of State

The equation of state (EOS) provides the relationship between the independent and thermodynamic variables and, in general, encapsulates much of the required microphysics. These dependent variables, and on occasion their derivatives, are crucial for modeling astrophysical environments. Phoebus supports a wide range of equations of state of astrophysical interest, including tabulated dense matter and Helmholtz.

Software capable of modeling a range of astrophysical environments requires flexibility in its EOS. To this end, the EOS functionality of Phoebus is provided by an external library, singularity-eos [4] (Miller et al., in prep). singularity-eos provides downstream fluid codes with performance portable EOS access with a unified API across all EOS's. At present, singularity-eos implements more than ten EOS's including, of note, ideal gas, Helmholtz (Timmes & Swesty, 2000), and tabulated dense matter. Implementing the EOS microphysics with this framework allows us to switch, or add, EOS's without modifying Phoebus. singularity-eos provides one-to-one Python

---

[4] https://github.com/lanl/singularity-eos

bindings for testing and analysis. We support the ability to solve for adiabats of an arbitrary EOS
– a crucial capability as many initial condition setups require constant entropy.

In this work we use either an ideal equation of state or a tabulated nuclear matter EOS. For the
latter we use the "SFHo" EOS (Steiner et al., 2013a). SFHo is a relativistic mean field model built
upon Hempel et al. (2012) that, importantly, was constructed to reproduce observed neutron star
mass-radius relationships.

7.2.3 Gravity

PHOEBUS is a fully general relativistic code, and gravity is implemented via the curvature of a
metric tensor. We implement a generic metric infrastructure that supports a selection of tabulated,
analytically prescribed, and numerically computed metrics at compile time. The machinery is
highly flexible, allowing for simple compile-time switching of metric implementations. We provide
a method `GetCoordinateSystem`, which returns a `CoordinateSystem` object. This object has
reference semantics, but can be copied safely to device, similar to `Kokkos::View`s. Depending on
user selection at compile time, requesting, e.g., the spatial metric γ_{ij} from the `CoordinateSystem`
object may reference an evolved grid variable, an analytic formula, or tabulated data. Derivatives,
such as those needed for Christoffel symbols may be computed either analytically or numerically
via finite differences.

7.2.3.1 Monopole GR

For problems where gravitational waves aren't important and where the gravitational potential
is approximately spherically symmetric, PHOEBUS provides a *monopole* solver, which assumes a
spherically symmetric 3-metric with maximal slicing[5] and areal shift[6]

$$ds^2 = (-\alpha^2 + a^2(\beta^r)^2)dt^2 + 2a^2\beta^r\,dt\,dr + a^2dr^2 + r^2d\Omega^2. \tag{7.15}$$

with unknown metric function a, lapse α, and radial shift β^r.

It turns out that in spherical symmetry, under these gauge conditions, the Einstein constraint
equations are sufficient to specify the metric and extrinsic curvature components a and K_r^r. The

---

[5]That is, that the trace of the extrinsic curvature vanishes

[6]In other words, we choose a gauge in which spheres have surface area $4\pi r^2$.

Hamiltonian constraint provides an equation for a and the momentum constraint for K^r_r:

$$\partial_r a = \frac{a}{8r}\left\{4 + a^2\left[-4 + r^2\left(3(K^r_r)^2 + 32\pi\rho_{\text{ADM}}\right)\right]\right\} \tag{7.16}$$

$$\partial_r K^r_r = 8\pi a^2 j^r - \frac{3}{r}K^r_r. \tag{7.17}$$

Here

$$\rho = \tau + D \tag{7.18}$$

is the ADM mass and

$$j^i = S^i \tag{7.19}$$

is the ADM momentum. The ADM evolution equations can then be used to solve for the gauge variables, α and β^r:

$$\frac{1}{a^2}\partial_r^2\alpha = \alpha\left[\frac{3}{2}(K^r_r)^2 + 4\pi(\rho + S)\right] + \frac{a'}{a^3}\partial_r\alpha - \frac{2}{a^2 r}\partial_r\alpha \tag{7.20}$$

$$\beta^r = -\frac{1}{2}\alpha r K^r_r, \tag{7.21}$$

where the lapse α satisfies a second-order boundary-value problem, and the shift β is given algebraically. Here

$$S^i_j = (\rho_0 h + b^2)W^2 + 3(P + b^2/2) - P^i_\mu P^\nu_j b^\mu b_\nu \tag{7.22}$$

is the ADM stress tensor and P is the projection operator onto the hypersurface of constant coordinate time.

The boundary conditions are given by symmetry at the origin,

$$a(r = 0) = 1 \tag{7.23}$$

$$K^r_r(r = 0) = 0 \tag{7.24}$$

$$\partial_r\alpha(r = 0) = 0, \tag{7.25}$$

and the weak field limit at large radii:

$$\lim_{r\to\infty}\alpha = 1 - \frac{c}{r} \tag{7.26}$$

$$\Rightarrow \lim_{r\to\infty}\partial_r\alpha = \frac{c}{r^2} \tag{7.27}$$

$$\Rightarrow \lim_{r\to\infty}\alpha = 1 - r\partial_r\alpha. \tag{7.28}$$

We solve equations (7.16) and (7.17) by integration outward from the origin using a second-order Runge-Kutta method. If Equation (7.20) is discretized by second-order centered finite differences, it forms a matrix equation, where the matrix operator is tridiagonal. This operator may then be inverted via standard diagonal matrix inversion techniques.

To complete the monopole solver, time derivatives of the metric must be provided so that the time-components of the Christoffel symbols may be provided by the infrastructure. These equations are algebraically complex, and so are not included here. They are summarized in Appendix 9F.

7.2.4 Radiation

In problems of interest in relativistic astrophysics it is necessary to consider radiation fields and their impact on the matter field. These radiation fields may exchange four-momentum and, in the case of neutrino radiation, lepton number with the matter field. Here, we focus primarily on neutrino radiation.

The species-dependent neutrino distribution function $f_\nu(x^\alpha, p^\alpha)$, for 4-position and 4-momentum x^α and p^α, evolves according to the 6+1 Boltzmann equation

$$p^\alpha \left[\frac{\partial f_\nu}{\partial x^\alpha} - \Gamma^\beta_{\alpha\gamma} p^\gamma \frac{\partial f_\nu}{\partial p^\beta} \right] = \left[\frac{df_\nu}{d\tau} \right]_{\text{coll}} \tag{7.29}$$

where $\Gamma^\beta_{\alpha\gamma} p^\gamma$ are the Christoffel symbols and the right hand side is the collision term including neutrino-matter interactions. Full solution of the 6+1 Boltzmann equation in dynamical environments remains computationally intractable and simplifications must be made, as we discuss in detail in Section 7.3.3.

We include a suite of relevant neutrino-matter interactions. We list the absorption and emission processes in table 7.1. Those absorption and emission interactions involving electron type neutrinos and antineutrinos will exchange lepton number with the fluid, modifying the composition. We

206

Type	Processes	Current	Corrections/Approximations
Abs./Emis. on neutrons	$\nu_e + n \leftrightarrow e^- + p$ $\nu_\mu + n \leftrightarrow \mu^- + p$	Charged	Blocking/Stimulated Abs. Weak Magnetism Kinematic recoil
Abs./Emis. on protons	$\bar{\nu}_e + p \leftrightarrow e^+ + n$ $\bar{\nu}_\mu + p \leftrightarrow \mu^+ + n$	Charged	Blocking/Stimulated Abs. Weak Magnetism Kinematic recoil
Abs./Emis. on ions	$\nu_e A \leftrightarrow A' e^-$	Charged	Blocking/Stimulated Abs. Kinematic recoil
Electron capture on ions	$e^- + A \leftrightarrow A' + \nu_e$	Charged	Blocking/Stimulated Abs. Kinematic recoil
$e^+ - e^-$ Annihilation	$e^+ e^- \leftrightarrow \nu_i \bar{\nu}_i$	Charged + Neutral	single-ν Blocking Kinematic recoil
n_i-n_i Brehmsstrahlung	$n_i^1 + n_i^2 \rightarrow n_i^3 + n_i^4 + \nu_i \bar{\nu}_i$	Neutral	single-ν Blocking Kinematic recoil

Table 7.1 Emission and absorption processes used in PHOEBUS. The symbols previously used are defined as follows: n is a neutron, p a proton, e^- an electron, e^+ a proton, μ^- a muon, A an ion, μ^+ an antimuon, and n_i an arbitrary nucleon. ν_i is an arbitrary neutrino. ν_e is an electron neutrino, and $\bar{\nu}_e$ is an electron antineutrino. We describe the corrections and approximations used below, as tabulated in Skinner et al. (2018). Blocking and stimulated absorption are related to the Fermi-Dirac nature of neutrinos. Weak magnetism is related to the extended quark structure of nucleons. Single-ν blocking is an approximation of blocking that becomes exact for single neutrino processes. These interactions are summarized in Burrows et al. (2006).

include the elastic scattering processes listed below,

$$\nu_i + p \quad \leftrightarrow \quad \nu_i + p \tag{7.30}$$

$$\nu_i + n \quad \leftrightarrow \quad \nu_i + n \tag{7.31}$$

$$\nu_i + A \quad \leftrightarrow \quad \nu_i + A \tag{7.32}$$

$$\nu_i + \alpha \quad \leftrightarrow \quad \nu_i + \alpha \tag{7.33}$$

where n represent neutrons, p protons, ν_i neutrinos, A heavy ions, and α alpha particles. Emissivities and opacities are tabulated as presented in Burrows et al. (2006).

The above set of interactions, while sufficient for many applications, is not exhaustive. In particular, we neglect neutrino-electron inelastic scattering (Bruenn, 1985). Experience has repeatedly demonstrated that even small corrections can have a large impact on neutrino-matter interactions and the subsequent dynamics (e.g., Freedman, 1974; Arnett, 1977; Bethe & Wilson, 1985; Bruenn,

1985; Horowitz, 1997; Burrows & Sawyer, 1998; Reddy et al., 1998; Müller et al., 2012; Buras et al., 2003; Hix et al., 2003; Kotake et al., 2018; Bollig et al., 2017; Fischer et al., 2020; Betranhandy & O'Connor, 2020; Miller et al., 2020; Kuroda, 2021) Future work for production simulations will include more complete sets of neutrino-matter interactions.

Neutrinos exchange four-momentum and lepton number with the fluid. In a frame comoving with the fluid, the four-momentum source term is given as

$$G_{(a)} = \frac{1}{h} \int (\chi_{\epsilon,f} I_{\epsilon,f} - \eta_{\epsilon,f}) n_{(a)} d\epsilon d\Omega, \tag{7.34}$$

where $\chi_{\epsilon,f} = \alpha_{\epsilon,f} + \sigma^a_{\epsilon,f}$ is the flavor dependent extinction coefficient combining absorption $\alpha_{\epsilon,f}$ and scattering $\sigma^a_{\epsilon,f}$, $I_{\epsilon,f}$ is the radiation intensity, $\eta_{\epsilon,f} = j_{\epsilon,f} + \eta^s_{\epsilon,f}(I_{\epsilon,f})$ is the total emissivity combining fluid $j_{\epsilon,f}$ and scattering $\eta^s_{\epsilon,f}$ emission, and $n_{(a)} = p_{(a)}/\epsilon$. This is then mapped into the lab frame by a coordinate transformation

$$G^\mu = e^\mu_{(a)} G^{(a)} \tag{7.35}$$

where $e^\mu_{(a)}$ defines an orthonormal tetrad.

The lepton number exchange source term G_{ye} is given by

$$G_{ye} = \frac{m_p}{h} \text{sign}(f) \int \frac{\chi_{\epsilon,f} I_{\epsilon,f} - \eta_{\epsilon,f}}{\epsilon} d\epsilon d\Omega \tag{7.36}$$

where m_p is the proton mass and

$$\text{sign}(f) = \begin{cases} 1 & \text{for } f = \nu_e \\ -1 & \text{for } f = \bar{\nu}_e \\ 0 & \text{for } f = \nu_x \end{cases} \tag{7.37}$$

determines the sign of the lepton exchange.

7.3 Numerical Methods

Here we lay out the numerical methods used to solve the equations introduced above.

7.3.1 MHD

Magnetic field evolution is included in Phoebus using a constrained transport scheme described in Tóth (2000). This formulation of constrained transport uses cell centered magnetic fields. For further details on the formulations used in Phoebus, see Miller et al. (2019); Gammie et al. (2003). The details of magnetic field treatment will be the subject of future updates to Phoebus.

Phoebus currently supports the local Lax-Friedrichs (LLF) and Harten-Lax-van Leer (HLL) Riemann solvers (Harten et al., 1983b; Toro, 2009b). Additional Riemann solvers are planned to be supported in the future. Reconstruction methods currently supported in Phoebus include piecewise constant (denoted `constant`); piecewise linear (denoted `linear`) with a variety of limiter options, though the default is minmod (Van Leer, 1977; Roe, 1986; Kuzmin, 2006); the fifth-order monotonicity preserving scheme of Suresh & Huynh (1997) (denoted `mp5`); and a novel fifth-order weighted essentially non-oscillatory (Shu, 2009, WENO) implementation using the Z-type smoothness indicators from Borges et al. (2008) (denoted `weno5z`). We call this WENO scheme WENO5-Z-AOAH, and describe it in detail in Appendix 9E.

The recovery of the primitive variables from the conserved state vector is non-trivial, and must be computed numerically, as no analytic solution is available. We use the procedure described in Kastaun et al. (2021), which is guaranteed to always converge.

7.3.2 Atmosphere Treatment

Numerical modeling of accretion disk systems requires including vacuum originally outside of the disk – a feat infeasible for Eulerian hydrodynamics. Instead, artificial atmospheres must be imposed to ensure both physical validity and stability of the numerical scheme. The problem is further complicated by the use of a tabulated equation of state which has strict bounds on the range of grid variables. In general, an EOS with a set of n state variables Q has bounds

$$Q_i \in \{q_{\min}, q_{\max}\} \text{ for } i = 1, \ldots, n \tag{7.38}$$

For the Helmholtz EOS, take $Q = \{\rho, T\}$. Including electron fraction Y_e allows for tabulated nuclear matter EOS's commonly used for CCSN and merger simulations such as SFHo. These bounds must

be accounted for. We demand that density remain above some floor value, i.e., $\rho > \rho_{\text{flr}}$. There are several implemented forms for the floor density, including

$$\rho_{\text{flr}} = \begin{cases} \rho_0 \\ \rho_0 e^{-\alpha x^1} \\ \rho_0 (x^1)^{-\alpha} \\ \rho_0 r^{-\alpha} \end{cases}$$

where ρ_0 is some small, problem dependent constant and α is a positive exponent. x^1 is the radial coordinate which, depending on the coordinate system, may be transformed (e.g., $x^1 = \ln(r)$). We generally take $\alpha = 2.0$, but that is not required. In all cases, the density floor near the black hole is approximately ρ_0 and, besides the first constant case, decays with radius. This radial decay ensures that the floors do not interfere with winds from the disk. To ensure consistency with the tabulated EOS, we require that the floor does not extend below the minimum density in the table. The specific internal energy is set similarly to the above. For electron fraction we simply require that it stay within the bounds of the table. In general in PHOEBUS we use the second case, where relevant, unless otherwise noted.

7.3.3 Radiation

In contexts such as BNS mergers and CCSNe, neutrinos are responsible for exchanging four-momentum and lepton number with the fluid. Neutrino-matter interactions drive the chemical and dynamical evolution of these systems, with electron neutrino absorption (emission) driving the matter to be more proton (neutron) rich, and inversely for electron anti-neutrinos. An accurate treatment of the neutrino radiation field is necessary for following nucleosynthesis in problems of interest.

We implement in PHOEBUS several methods for evolving the radiation fields. First, we include a gray, two moment approach. Additionally, following closely to the methods outlined in BHLIGHT and NUBHLIGHT (Dolence et al., 2009; Ryan et al., 2015; Miller et al., 2019), we implement neutrino transport through Monte Carlo methods. There is also, for testing purposes, a simple "lightbulb"

approach where the neutrino luminosity is fixed at a constant value and a analytic form for the source terms is taken. We do not discuss that approach here.

For all of the above we consider three neutrino flavors: electron neutrinos, electron antineutrinos, and a characteristic heavy neutrino. With appropriate micophysics, the methods may be trivially extended to include, for example, muon neutrino evolution. Below we summarize the methods included in PHOEBUS and refer the reader to the aforementioned works for further details.

7.3.3.1   Monte Carlo

The probability distribution of emitted Monte Carlo packets is

$$\frac{1}{\sqrt{g}} \frac{dN_p}{d^3x\,dt\,dv\,d\Omega} = \frac{1}{\omega\sqrt{g}} \frac{dN}{d^3x\,dt\,dv\,d\Omega} = \frac{1}{w} \frac{j_{v,f}}{h\nu} \tag{7.39}$$

where N_p is the number of Monte Carlo packets with ω physical neutrinos per packet, N is the number of physical particles, $j_{v,f}$ is the fluid frame emissivity of neutrinos with frequency v, and flavor $f \in \{v_e, \bar{v}_e, v_x\}$[7], and h is Planck's constant. The number of emitted packets in timestep Δt is

$$N_{p,tot} = \Delta t \sum_f \int \sqrt{g} d^3x\,dv\,d\Omega \frac{1}{w} \frac{j_{v,f}}{h\nu} \tag{7.40}$$

and the number of packets of flavor f created in a computational cell i of volume Δ^3x is

$$N_{p,f,i} = \Delta t \Delta^3 x \int \sqrt{g} dv\,d\Omega \frac{1}{w} \frac{j_{v,f}}{h\nu}. \tag{7.41}$$

We control the total number of Monte Carlo packets created per timestep by setting the weights w as

$$w = \frac{C}{v} \tag{7.42}$$

where C is a constant. This ensures that packets of frequency v and weight $w(v)$ have energy

$$E_p = whv = hC, \tag{7.43}$$

such that packet energy is independent of frequency. The constant C is set by fixing the total number of Monte Carlo packets created to be N_{target} and setting C such that equation 7.40 is

---

[7]In practice the methods presented here may be straightforwardly extended to more neutrino species.

satisfied. N_{target} is set such that the total number of Monte Carlo packets is roughly constant in time. Thus, we have

$$C = \frac{\Delta t}{h N_{target}} \sum_f \int \sqrt{g} d^3 x d\nu d\Omega j_{\nu,f}. \quad (7.44)$$

Absorption of radiation is treated probabilistically in Monte Carlo fashion. A neutrino of flavor f that travels a distance $\Delta \lambda$ traverses an optical depth

$$\Delta \tau_{a,f}(\nu) = \nu \alpha_{\nu,f} \Delta \lambda \quad (7.45)$$

where $\alpha_{\nu,f}$ is the absorption extinction coefficient for radiation of frequency ν and flavor f. Absorption occurs if

$$\Delta \tau_{a,f}(\nu) > \ln(r_a) \quad (7.46)$$

where r_a is a random variable sampled uniformly from the interval [0, 1).

The implementation of Monte Carlo radiation leverages PARTHENON's `swarms` particle infrastructure. Monte Carlo scattering is not yet implemented in PHOEBUS. In the future, Monte Carlo scattering will be implemented following the methods in Miller et al. (2019).

7.3.3.2 Moments

We implement gray M1 moments scheme in PHOEBUS (Thorne, 1981; Shibata et al., 2011; Cardall et al., 2013; Foucart et al., 2015). We evolve three independent neutrino species: ν_e, $\overline{\nu}_e$, and ν_x, where the latter is the combination of (ν_μ, $\overline{\nu}_\mu$, ν_τ, $\overline{\nu}_\tau$). In gray approximation we consider energy-integrated moments and evolve first two moments. The energy density, flux, and radiation pressure in the inertial frame are defined as follows

$$E = \int \epsilon f_\nu(p^\mu, x^\mu) \delta(h\nu - \epsilon) d^3 p \,, \quad (7.47)$$

$$F^i = \int p^i f_\nu(p^\mu, x^\mu) \delta(h\nu - \epsilon) d^3 p \,, \quad (7.48)$$

$$P^{ij} = \int \frac{p^i p^j}{\epsilon} f_\nu(p^\mu, x^\mu) \delta(h\nu - \epsilon) d^3 p \,, \quad (7.49)$$

where ϵ is neutrino energy in the rest frame of medium. To obtain evolution equations, we decompose the stress-energy tensor for the radiation field as follows

$$T^{\mu\nu}_{\text{rad}} = n^\mu n^\nu E + n^\mu F^\nu + n^\nu F^\mu + P^{\mu\nu} \,, \quad (7.50)$$

212

Then the conservation equations in Valencia formalism are

$$\partial_t(\sqrt{\gamma}E) + \partial_i[\sqrt{\gamma}(\alpha F^i - \beta^i E)] = \alpha\sqrt{\gamma}(\alpha G^0 - F^i\partial_i\alpha + P^{ij}K_{ij}),$$

$$\partial_t(\sqrt{\gamma}F_i) + \partial_j[\sqrt{\gamma}(\alpha P_i^j - \beta^j F_i)] = \sqrt{\gamma}(\alpha\gamma_{i\nu}G^\nu + F^j\partial_i\beta_j + P^{jk}\frac{\alpha}{2}\partial_i\gamma_{jk} - E\partial_i\alpha),$$

where α is the lapse, β is the shift, γ is a three-metric, and K is the extrinsic curvature. The first terms on the right side of Equations (7.51) & (7.51) are the collisional source terms. We consider absorption, emission, and iso-energetic scattering from the background fluid. The source terms are very similar to those in Shibata et al. (2011) and Foucart et al. (2015):

$$G = \kappa_J(J^{\text{eq}} - J), \tag{7.51}$$

$$G_\nu = \kappa_J u_\nu(J^{\text{eq}} - J) - \kappa_H H_\nu, \tag{7.52}$$

where J and H_ν are the energy and flux in the fluid rest frame. κ_J is energy-averaged absorption and κ_H is the sum of energy-averaged absorption and scattering opacities. J^{eq} is evaluated from the equilibrium distribution function

$$J^{\text{eq}} = \int_0^\infty d\nu\nu^3 \int d\Omega \frac{1}{1 + exp[(\nu - \mu_\nu)/T_\nu]}, \tag{7.53}$$

where μ_ν and T_ν are the chemical potential and temperature of neutrinos that are in thermal equilibrium with matter, and ν is neutrino energy in the fluid frame.

To close the system of equations, we need to specify the closure relation, $P^{ij}(E, F^i)$. We use M1 closure which evaluates P^{ij} by interpolating between optically thin and optically thick regimes (Shibata et al., 2011)

$$P^{ij} = \frac{3\chi(f) - 1}{2}P_{\text{thin}}^{ij} + \frac{3(1 - \chi(f))}{2}P_{\text{thick}}^{ij}, \tag{7.54}$$

where $f = \sqrt{F^\alpha F_\alpha}/E$ is the flux factor and ranges from 0 to 1 and χ is an interpolant. We use maximum entropy closure for fermionic radiation (MEFD) which derives the closure relation by

maximizing the entropy for Fermi-Dirac distribution (Cernohorsky & Bludman, 1994). In the limit of maximum packing, the MEFD closure is (Smit et al., 2000)

$$\chi = \frac{1}{3}(1 - 2f + 4f^2).$$ (7.55)

Since P^{ij} is a function of f and f is a function of E and F^α, we use Newton-Raphson iteration algorithm to find the roots.

We solve Equations (7.51) & (7.51) by performing a backward Euler discretization. We fix J_{BB}, κ_J, κ_H and obtain a linear system of equations for (E, F_i) at time $(n + 1)$

$$\sqrt{\gamma}\frac{E^{n+1} - E^*}{\Delta t} = -\alpha\sqrt{\gamma}n^\nu G_\nu$$ (7.56)

$$\sqrt{\gamma}\frac{F_i^{n+1} - F_i^*}{\Delta t} = -\alpha\sqrt{\gamma}\gamma_i^\nu G_\nu$$ (7.57)

7.3.4   Gravity

7.3.4.1   Monopole GR

To solve the method in practice, matter quantities such as density are accumulated in a conservative way from a three-dimensional, potentially Cartesian AMR grid onto a single-dimensional radial grid, which is used by the monopole solver. The procedure in spherical coordinates is the following:

1. For each cell in a given meshblock, compute it's "integrand + measure". i.e., $\mathcal{M}(Q) = Qr^2\sin\theta\Delta\theta\Delta\phi$. (Fortunately, in the monopole approximation, this is the relevant part of the line element.)

2. Sum up $M(Q)$ in the θ and ϕ directions on the block, e.g., $SM(Q) = \sum_{j,i} M_{i,j}(Q)$. This creates a 1D radial grid aligned with the meshblock grid.

3. For each point on the radial grid that intersects the meshblock, interpolate $SM(Q)$ on to that point additively. In other words, add this meshblock's contribution to the total.

4. At the end, divide each point by 4π

In Cartesian coordinates, the procedure is similar, but more complex:

1. For each cell in a given meshblock, compute:

   - The radius r and angle θ of the cell

   - The width of the cell in the θ and ϕ directions by taking the vector $\{\Delta x, \Delta y, \Delta z\}$ and applying the Jacobian of the coordinate transformation to spherical coordinates to it

   - The measure, $\mathcal{M}(Q) = Qr^2 \sin\theta \Delta\theta \Delta\phi$

2. Reinterpret the cells in the meshblock as a 1D unstructured grid in radius. For each cell on the monopole GR grid, use this 1D unstructured grid to interpolate the measure on to it additively.

3. The reduction over all meshblocks onto this 1D grid is the integral over spherical shells of Q. Divide by 4π to get the average.

This solve is performed in a first-order operator-split way. To maximally expose concurrency on GPUs, the monopole solve is performed on CPU, concurrently with the fluid update. This implies a slight lag in the metric solution by one RK subcycle. We have found that this time lag has not significantly impacted the accuracy or stability of realistic simulations.

7.3.5 Tracer Particles

We include tracer particles in PHOEBUS. Tracer particles are a numerical representation of a Lagrangian fluid packet which is advected along with the fluid. Tracer particles allow for the post-processing of simulation data for, e.g., nucleosynthesis calculations. In the (3 + 1) split of general relativity, the equation of motion is

$$\frac{dx^i}{dt} = \frac{u^i}{u^0} = \alpha v^i - \beta^i, \tag{7.58}$$

where x^i are the tracer's spatial coordinates, α is the lapse, β^i is the shift vector, v^i is the fluid three-velocity, and u^μ is the fluid four-velocity. The implementation of tracer particles leverages

PARTHENON's `swarms` particle infrastructure. The fluid three-velocity, lapse, and shift are interpolated to the particle position before advecting. At present, tracer advection is coupled to the fluid via first order operator splitting and integrated in time with a second order Runge Kutta scheme. Initial sampling of tracer particles is, in general, problem dependent.

7.4 Numerical Tests

In this section we present results obtained with PHOEBUS with a comprehensive suite of test problems designed to stress its core functionalities. These tests serve the goal of verification and validation of PHOEBUS and allow for ease of reproducability.

7.4.1 Hydro

Here we present a suite of tests stressing the MHD solvers. Unless otherwise noted, all tests use an ideal gas EOS with an adiabatic index of 5/3.

7.4.1.1 Linear Waves

In this section we follow the propagation of various families of linear waves. Following the evolution stresses the ability of the code to converge in linear regimes (indeed, these linear waves are treatable analytically). While the ability of a astrophysical code to handle linear waves is not sufficient for scientific viability – as shocks are a fact of life – it is a necessary one. Indeed, while the presence of shocks will reduce the convergence of all schemes, accurate and high order solutions should be attainable in smooth regions. These tests stress the ability of PHOEBUS to converge to the correct solution in the linear regime. For all tests in this section, unless otherwise noted, we use a flat metric with coordinate boosts $v_x = v_y = 0.617213$ applied to the x and y directions. All tests are treated with two spatial dimensions. The tests presented here are adapted from `Athena` (Stone et al., 2008) and `Athena++` (Stone et al., 2020).

To measure the convergence of the tests presented here we use the L_1 norm scaled by the wave amplitude

$$L_1(q) = \frac{1}{kN^2} \sum_i \sum_j (q_{ij} - \hat{q}_{ij}) \tag{7.59}$$

for quantity q, number of grid points along a dimension N, wave amplitude k, and solution $\hat{q}$. The
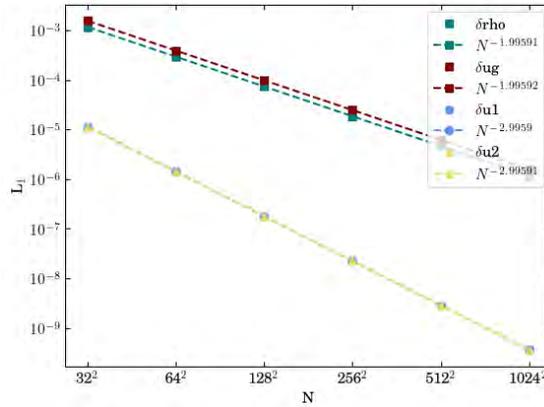
Figure 7.1 L_1 convergence for the pure sound wave test. Shown is convergence for density (teal), internal energy (red), v_x (light blue), and v_y (yellow).
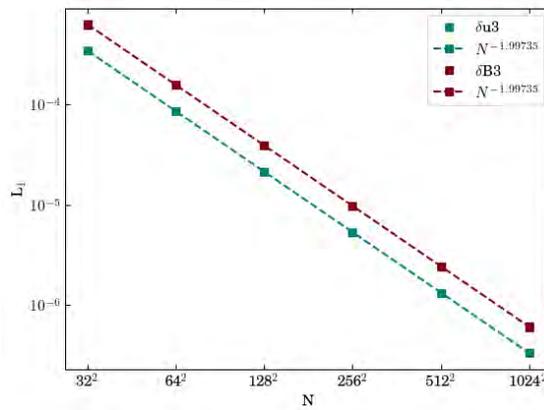
Figure 7.2 L_1 convergence for the pure Alfvén wave test. Shown is convergence for v_z (teal) and B_z (red).

tests are ran for one period such that the solution $\hat{q}$ is simply the initial condition. In Figures 7.1 – 7.4 we show L_1 convergence for relevant quantities for sound, Alfvén, fast, and slow magnetosonic waves, respectively. We consider resolutions $N^2 = 32^2$, 64^2, 128^2, 512^2, and 1024^2. For all cases, we observe roughly at least the expected second order convergence.

7.4.1.2 Riemann Problems

Here we present a modification of the classic shock tube Riemann problem of Sod (1978). The test involves an initially stationary fluid with two states separated by a discontinuity. The initial state develops a shock propagating into the low density region, followed by a contact discontinuity, and a rarefaction wave propagating into the high density region. This test stresses a code's ability
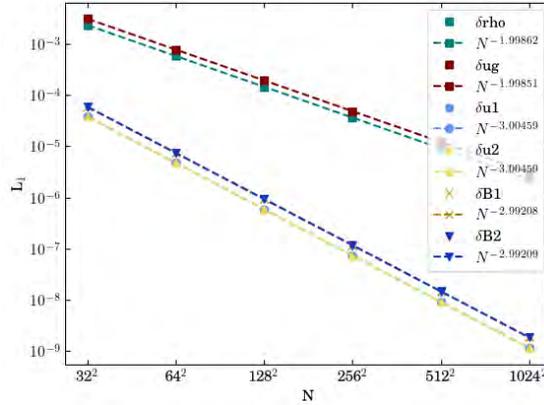
Figure 7.3 L_1 convergence for the pure fast magnetosonic wave test. Shown is convergence for density (teal), internal energy (red), v_x (light blue), v_y (yellow), B_x (gold x), and B_y (dark blue triangles).
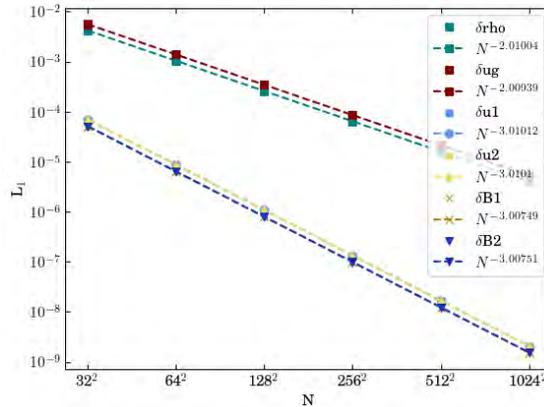
Figure 7.4 L_1 convergence for the pure slow magnetosonic wave test. Shown is convergence for density (teal), internal energy (red), x-velocity (light blue), and y-velocity (yellow) B_x (gold x), and B_y (dark blue triangles).

to capture various hydrodynamic waves without introducing unphysical oscillations or viscosity.

We modify the traditional shock tube problem by the use of a realistic nuclear EOS (SFHo). This allows us to stress the code in regimes of astrophysical interest while simultaneously stressing the implementation of the nuclear EOS. For this test, our computational domain is $D = [0, 300]$ km with an initial discontinuity located at $x = 150$ km. The initial conditions $\mathcal{S} = (\rho, p, Y_e)$ are
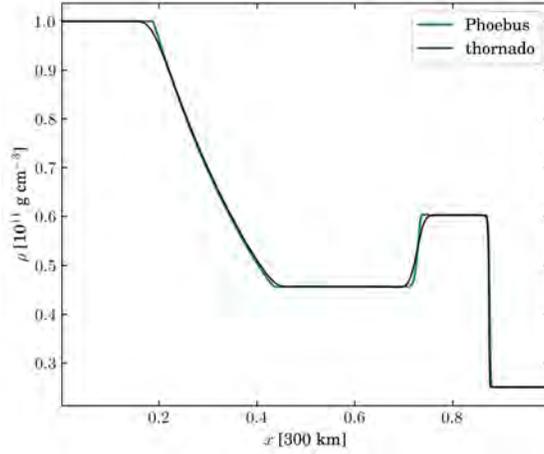
Figure 7.5 Numerical solution of the nuclear EOS shock tube at $t \approx 7.5$ms with Phoebus (teal) using 512 cells and piecewise linear reconstruction compared to a reference solution computed with thornado (black) using 10000 piecewise constant elements with 3rd order strong stability preserving Runge Kutta time integration.

given by

$$\mathcal{S} = \begin{cases} (10^{11}, 2.231 \times 10^{31}, 0.3) & \text{left} \\ (0.25 \times 10^{11}, 2.232 \times 10^{30}, 0.5) & \text{right} \end{cases} \tag{7.60}$$

for primitive density ρ (in g cm^{-3}), pressure p (in erg cm^{-3}) and electron fraction Y_e. The system is evolved until about $t = 7.5$ms using 512 computational cells, piecewise linear reconstruction, and an HLL approximate Riemann solver. As an analytic solution does not exist with the use of a non-trivial EOS, we compare to a reference solution computed using thornado, a discontinuous Galerkin based GRRMHD code, computed using 10000 piecewise constant (P0) elements, 3rd order strong stability preserving explicit Runge Kutta time integration, an HLLC approximate Riemann solver (Toro et al., 1994). The thornado reference solution was computed using the same SFHo EOS. Figure 7.5 shows the density profile obtained with Phoebus (teal) compared to the thornado reference solution (black). We see satisfactory agreement between the two codes.

7.4.1.3 Sedov–Taylor Blast Wave

Here we present the classic Sedov-Taylor blast wave (Sedov, 1946; Taylor, 1950). In this setup a large amount of energy is concentrated into a small volume, mocking an explosion and driving a spherical (or cylindrical in 2D) blast wave. This test stresses the scheme's ability to handle shocks

219
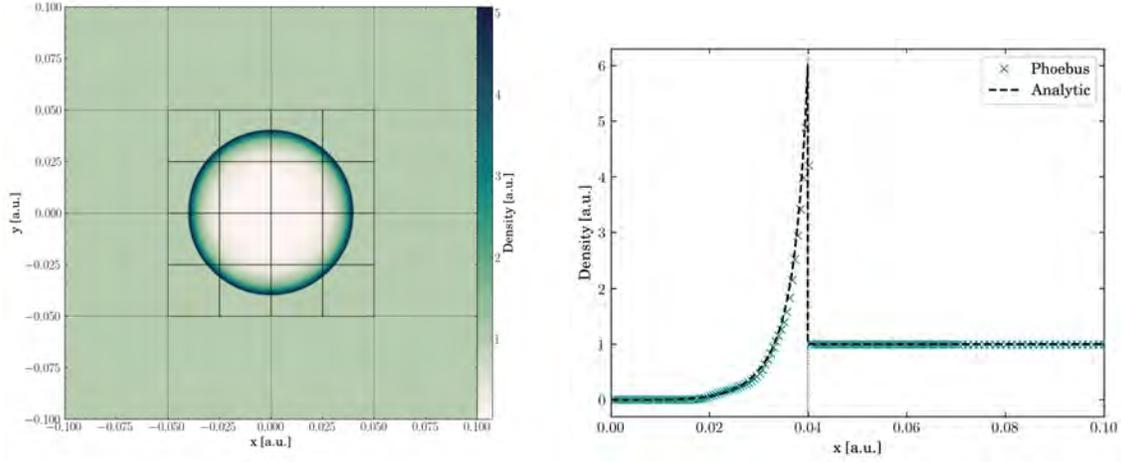
Figure 7.6 Left: 2D profile of density at $t = 0.5$ with AMR levels overlaid. Right: Density along the x coordinate at $y = 0$ for PHOEBUS (teal crosses) compared to the self similar solution (black dashed). The vertical dashed line denotes the analytic shock position.

and spherical geometries.

We perform the test in 2D Cartesian coordinates, implying a cylindrical blast wave. An amount of energy $E = 0.1$ is deposited into all cells with $r < r_{\text{init}}$ in an otherwise homogeneous medium. The medium has ambient density and pressure $\rho_{\text{ambient}} = 1.0$ and $p_{\text{ambient}} = 10^{-5}$. We take $r_{\text{init}} = 0.1$ to set the volume of deposition. The computational domain is $D = [-1.0, 1.0] \times [-1.0, 1.0]$.

We perform the test with $N_x \times N_y = 128 \times 128$ computational cells. As an additional test of the AMR capabilities of PHOEBUS, we allow for up to five levels of mesh refinement. We evolve the system until $t = 0.5$ Figure 7.6 shows the pressure profile from the blast wave (top). Overlaid on the profile are grid representative of the AMR refinement regions. We also show a 1D profile of pressure along the $x = y$ (bottom).

7.4.1.4 Blandford-McKee Blast Wave

Here we present the Blandford-McKee blast wave (Blandford & McKee, 1976) – a relativistic complement to the non-relativistic Sedov-Taylor blast wave of the previous section. This test involves an ultra-relativistic shock wave characterized by Lorentz factor W propagating into an ambient medium and stresses a scheme's treatment of relativity and ability to capture relativistic shocks. For this test we take $W = 8.5$ for the Lorentz factor of the shock and an ambient medium

220
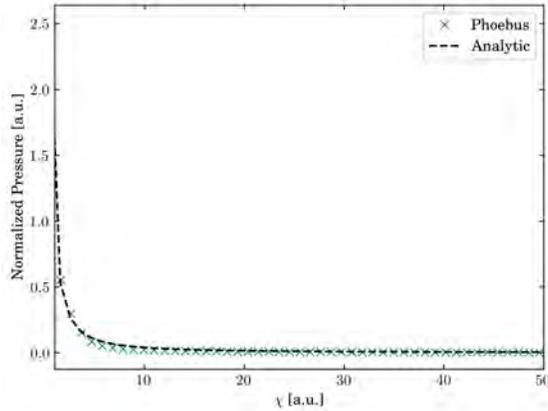
Figure 7.7 Normalized pressure profile as a function of the self similar radial variable χ for PHOEBUS (teal) and analytic (black, dashed) solutions. The shock front is located at $\chi = 1.0$ and $\chi > 1.0$ is the post-shock region.

with $\rho_0 = 10^{-2}$ and $p_0 = 10^{-4}$. Figure 7.7 shows the normalized post-shock pressure profile as a function of the similarity variable χ, where $\chi = 1.0$ is the shock position and $\chi > 1$ is the post-shock region.

7.4.2 Tracer Particles

Here we test the tracer particles infrastructure as described in Section 7.3.5. To stress the coupling to both the fluid and the spacetime, we model a 3D accretion disk in near hydrostatic equilibrium around a black hole. We adopt the torus configuration of Fishbone & Moncrief (1976). We initialize a torus of constant entropy and specific angular momentum with no initial magnetic field around a Kerr black hole. We assume an ideal gas equation of state for this test. The test is run in three spatial dimensions with $N_r \times N_\theta \times N_\phi = 128 \times 128 \times 128$ cells and 10^4 tracer particles. The system is evolved until $t = 2000 G M_{BH}/c^3$. Tracer particles are sampled uniformly in volume on the initial condition.

With no initial magnetic field we will not develop the magnetorotational instability (MRI) responsible for driving accretion on to the central compact object. Instead, the disk – and by extension, tracer particles – will orbit the black hole, until other hydrodynamic instabilities arise at later times. Figure 7.8 shows the trajectories of three select tracer particles throughout the evolution projected into the xy−plane. The innermost tracer covers several orbits through the evolution while
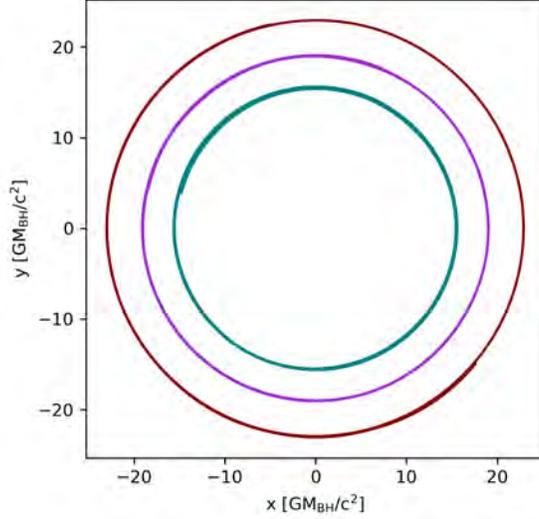
Figure 7.8 Paths of three select tracer particles evolved in the equilibrium disk.

the outermost covers slightly more than one. The tracer particles show the expected behavior, orbiting the central black hole (located here at $x = y = 0$). However, as the tracer particle integration is not symplectic, we do not expect, or observe, perfectly closed orbits. Future work includes the implementation of a symplectic integrator for tracer particle advection.

7.4.3 Transport

In this section we present a suite of tests stressing the radiation transport schemes. Unless otherwise noted, all tests use Monte Carlo transport.

7.4.3.1 Artificial Neutrino Cooling

We test the coupling of neutrinos to matter in a simplified context. We construct a homogeneous, isotropic gas cooled by only either electron neutrinos or antineutrinos using a simplified "tophat" emissivity

$$j_{\nu,f} = C y_f \left(Y_e \right) \chi \left(\nu_{\min}, \nu_{\max} \right), \tag{7.61}$$

where C is a constant ensuring correct units,

$$\chi \left(\nu_{\min}, \nu_{\max} \right) = \begin{cases} 1 & \text{for } \nu_{\min} \leq \nu \leq \nu_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{7.62}$$

222
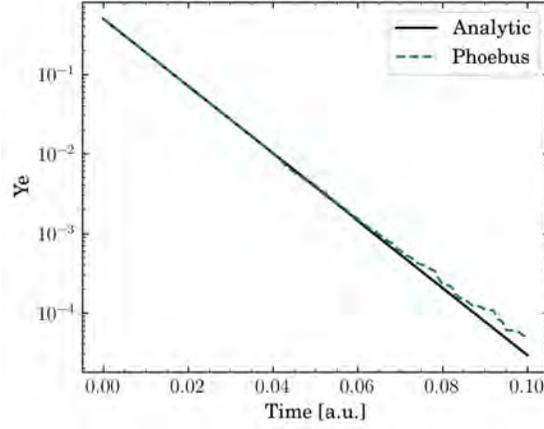
Figure 7.9 Electron fraction for a homogeneous isotropic gas cooling by electron neutrinos. The solid line is the analytic solution and the dashed line is the PHOEBUS solution.

and

$$
y_f\left(Y_e\right) = \begin{cases} 2Y_e & \text{for } \nu_e\text{emission} \\ 1 - 2Y_e & \text{for } \bar{\nu}_e\text{emission} \\ 0 & \text{otherwise.} \end{cases} \tag{7.63}
$$

The gas is at a uniform density of 10^6 g cm^{-1} with an internal energy density of 10^{20} erg cm^{-3} and electron fraction

$$
Y_e\left(t = 0\right) = \begin{cases} \frac{1}{2} & \text{for } \nu_e \\ 0 & \text{for } \bar{\nu}_e. \end{cases} \tag{7.64}
$$

In this simplified setting, the electron fraction evolution has an analytic solution (Miller et al., 2019)

$$
Y_e\left(t\right) = \begin{cases} -\frac{1}{2}e^{-2A_C t} & \text{for } \nu_e \\ \frac{1}{2}\left(1 - e^{-2A_C t}\right) & \text{for } \bar{\nu}_e \end{cases} \tag{7.65}
$$

where $A_C = \frac{m_p}{h\rho}C\ln(\frac{\nu_{\max}}{\nu_{\min}})$ for proton mass m_p, Planck's constant h, and density ρ.

This setup was run until $t = 0.1$ (in arbitrary code units) using 100 frequency bins and only 16 Monte Carlo packets. Figure 7.9 (7.10) shows the electron fraction as a function of time for a gas cooled by electron neutrinos (antineutrinos). Agreement with the analytic solution is very good, with very small deviations at late times due to Monte Carlo noise.

223
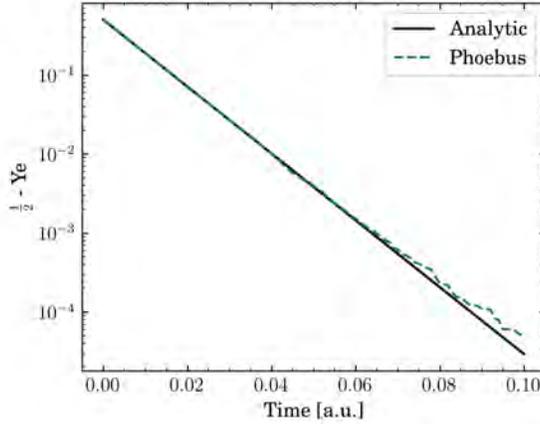
Figure 7.10 Electron fraction for a homogeneous isotropic gas cooling by electron antineutrinos. The solid line is the analytic solution and the dashed line is the PHOEBUS solution.

7.4.3.2 Neutrino-Driven Wind Setup

The tests of the previous section use artificial neutrino emissivities. While useful for comparison with a known analytic solution, they do not represent physically realistic or interesting settings. In this section we consider a setting of astrophysical interest and compare to the supernova code FORNAX (Skinner et al., 2018). FORNAX uses notably different methods from PHOEBUS, with PHOEBUS being fully general relativistic and FORNAX having an approximate treatment for gravity. FORNAX treats radiation using a multi-group moment based approach with the M1 closure (Shibata et al., 2011; Cardall et al., 2013) whereas PHOEBUS uses a Monte Carlo approach in addition to the other methods outlined in Section 7.3.

To facilitate comparison between codes and to test PHOEBUS in astrophysically motivated settings, we consider a homogeneous and isotropic gas at rest on a periodic domain in Minkowski space. PHOEBUS and FORNAX both use the same commonly adopted SFHo nuclear matter equation of state (Steiner et al., 2013b) and opacities. We consider the following initial state, motivated by conditions realized in neutrino-driven outflow

224

$$\rho_0 = 10^9 \text{ g cm}^{-3}$$

$$T = 2.5 \text{ MeV} \tag{7.66}$$

$$Y_e = 0.1.$$

The problem is evolved for 0.5 seconds assuming no initial radiation. Both codes are run with 200 frequency groups ranging from 1 to 300 MeV. PHOEBUS is run with a target number of 10^5 Monte Carlo packets. Both codes include three neutrino flavors: ν_e, $\bar{\nu}_e$, and ν_x, where ν_x is a representative "heavy" neutrino combining the $\mu - \tau$ neutrinos and antineutrinos.

We consider first the case of pure cooling by neutrinos, disabling absorption opacities. Electron fraction and temperature evolution for both PHOEBUS (blue dashed line) and FORNAX (red solid line) are shown in Figure 7.11. PHOEBUS displays the expected rapid cooling behavior and agrees very well with the FORNAX solution.

Next we consider the case of emission and absorption of neutrinos, allowing the radiation and gas to come to thermal equilibrium. We show electron fraction and temperature evolution for both PHOEBUS (blue dashed line) and FORNAX (red solid line) in Figure 7.12. As with the previous test, the electron fraction rapidly, but cooling is slowed due to absorption of neutrinos. Again we see good agreement between the codes with small Monte Carlo noise in the equilibrium electron fraction.

7.4.3.3 Two Dimensional Lepton Transport

Neutrinos, unlike photon radiation, can exchange energy, momentum, as well as lepton number with the matter field. This motivates an accurate treatment of neutrino transport, as the matter composition can influence the resulting nucleosynthesis, among other things. We test the ability for PHOEBUS to capture this lepton number exchange by considering a two-dimensional test problem. We let our domain be a periodic box with $(x, y) \in [-1, 1]^2$. The initial state is a gas with constant
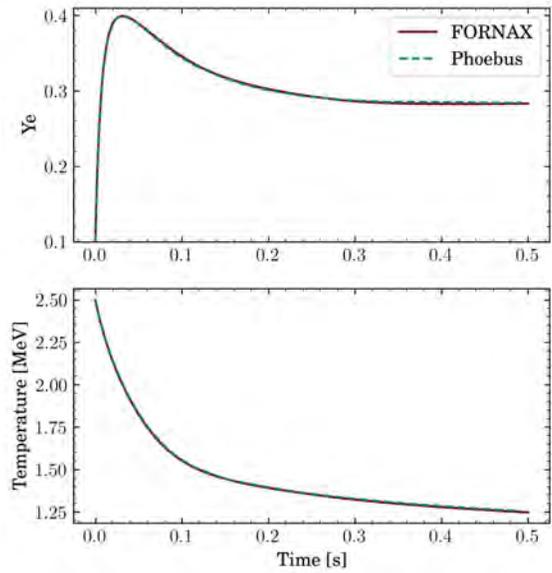
Figure 7.11 Electron fraction (top) and temperature (bottom) for the optically thin cooling comparison between PHOEBUS (dashed line) and FORNAX (solid line).
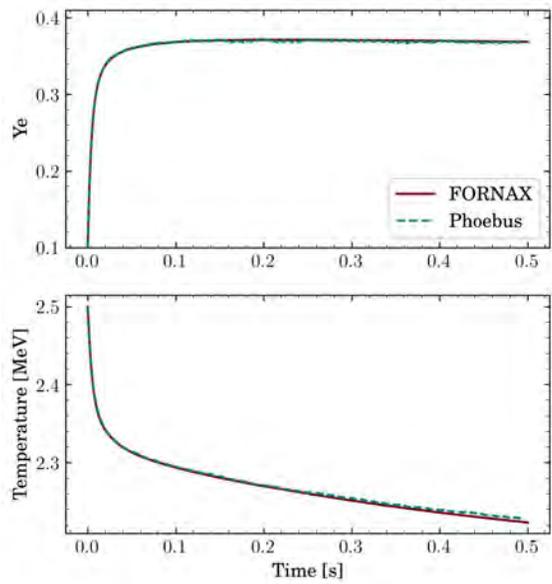
Figure 7.12 Electron fraction (top) and temperature (bottom) for the thermal equilibrium comparison between PHOEBUS (dashed line) and FORNAX (solid line).
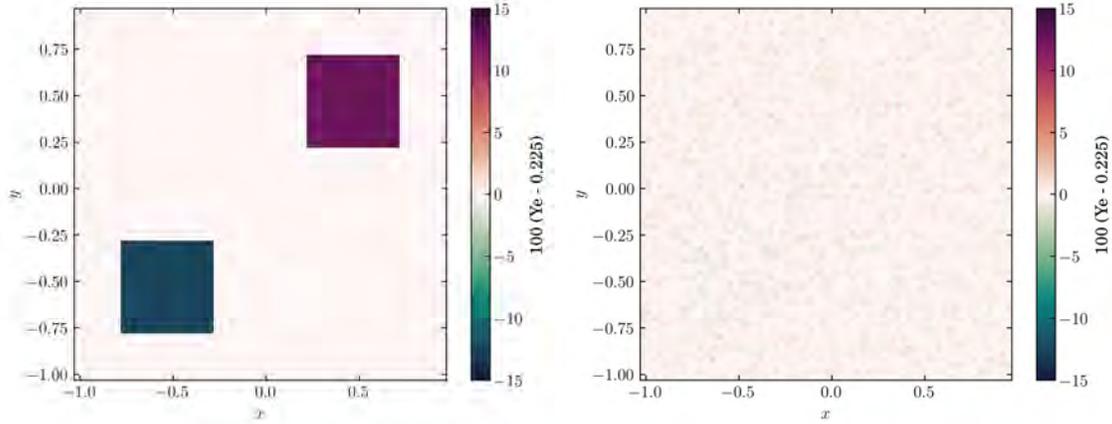
Figure 7.13 Left: Initial condition for the lepton equilibration problem. Right: The system after about 10 ms. The neutrino radiation field brings the electron fraction field into equilibrium.

density and temperature

$$\rho_0 = 10^{10} \text{ g cm}^{-3}$$
$$T = 2.5 \text{ MeV} \tag{7.67}$$

and electron fraction defined by

$$Y_e = \begin{cases} 0.100 & \text{for } (x, y) \in [-0.75, -0.25]^2 \\ 0.350 & \text{for } (x, y) \in [0.25, 0.75]^2 \\ 0.225 & \text{otherwise.} \end{cases} \tag{7.68}$$

This describes a region of stellar material with an electron fraction "hot spot" and "cold spot". We do not allow the gas to evolve due to pressure gradients, allowing instead only interaction with the radiation field in order to highlight the impact of lepton number transport. We use 2.5×10^5 Monte Carlo packets with 200 frequency bins distributed from roughly 100 keV to 300 MeV. We include three flavors of neutrinos.

Figure 7.13 shows the initial condition (left) and state at $t \approx 10$ms (right). We observe the expected behavior where the neutrinos equilibrate with the matter, with the final state being perturbed from the initial state.

7.4.4 Gravity

7.4.4.1 Homologous Collapse

Here we test the general relativistic gravity solver in PHOEBUS with the homologous collapse problem (Goldreich & Weber, 1980). For this test we use the spherically symmetric monopole gravity solver of Section 7.2.3.1. This problem serves to test the coupling between gravity and hydrodyamics in a setting relevant to CCSNe. It involves a homologously collapsing core ($\mathbf{u} \propto \mathbf{r}$) with mass M and size R. The system is described by the continuity equation, Euler's equation, and Poisson's equation:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = 0, \tag{7.69}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla}\left(\frac{|\boldsymbol{u}|^2}{2}\right) + (\boldsymbol{\nabla} \times \boldsymbol{u}) \times \boldsymbol{u} + \boldsymbol{\nabla} h + \boldsymbol{\nabla}\Phi = 0, \tag{7.70}$$

$$\nabla^2 \Phi - 4\pi G \rho = 0, \tag{7.71}$$

where $\boldsymbol{u}$ is the fluid velocity, Φ is the gravitational potential, h is the heat function $h = \int \frac{dp}{\rho} = 4\kappa\rho^{1/3}$. If one assumes vorticity-free flow and an analytic, $n = 3$, $\gamma = 4/3$ polytropic equation of state, one can find a semi-analytic solutions to the system of equations (Equations (7.69) - (7.71)). In this approximation, the fluid velocity, density, and the gravitational potential are given by:

$$\boldsymbol{u} = \dot{a}\boldsymbol{r}, \tag{7.72}$$

$$\rho = \left(\frac{\kappa}{\pi G}\right)^{3/2} a^{-3} f^3, \tag{7.73}$$

$$\Phi = \frac{\Psi}{c_s^{-2}} = \left(\frac{\gamma P_c}{\rho_c}\right)\Psi = \frac{4}{3}\left(\frac{\kappa^3}{\pi G}\right)^{1/2}\frac{\Psi}{a}, \tag{7.74}$$

where a is Jean's length and is found to be

$$a(t) = (6\lambda)^{1/3}\left(\frac{\kappa^3}{\pi G}\right)^{1/6}[t + t_0]^{2/3}, \tag{7.75}$$

where λ a constant determined by initial conditions. f is a normalization function for the density and is determined by the following differential equation:

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial f}{\partial r}\right) + f^3 = \lambda \tag{7.76}$$

228
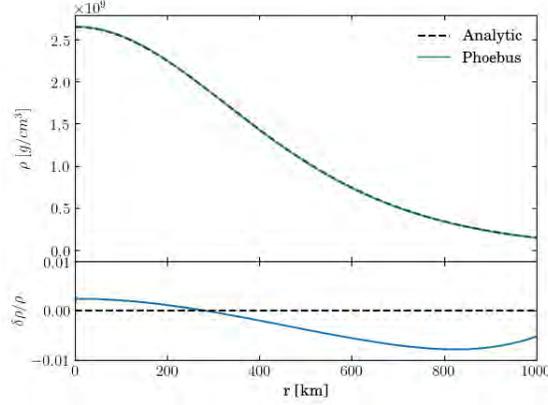
Figure 7.14 Density profile of the homologous collapsing star with mass $M = 1.4 M_\odot$ and size $R = 3000$km at $t = 0.12$s after the start of the collapse.

Finally, Ψ can be found using

$$\Psi = \frac{\lambda}{2} r^2 - 3f \tag{7.77}$$

We incorporate the homologous collapse test problem in PHOEBUS and compare results with the semi-analytic solutions of Equations (7.75) - (7.77). Figure 7.14 shows the comparison of the density profiles between the simulation and analytic solution. We simulate one-dimensional homologous collapse of a star with mass $M = 1.4 M_\odot$ and size $R = 3000$km. The number of zones in x direction is 10000. The analytic solution of Goldreich & Weber (1980) uses Newtonian approximation for gravity, while the simulation in PHOEBUS is solved in the monopole approximation for GR. This difference causes a different behavior in the time evolution between the simulation and analytic solution; for example, central density changes faster in the simulation. To compare the simulation and analytic results, we choose some time moment on the simulation, $t = 0.12$s and find corresponding Jeans length from the simulation. Then solve the analytic equations given this value of Jeans length. As a result, the density profile on the simulation matches well with the density profile obtained from analytic solutions; the variation is $\delta\rho/\rho_{\text{analytic}} \leq 1\%$.

7.5 Discussion and Conclusions

In this paper, we have introduced PHOEBUS, a new code for general relativistic radiation magnetohydrodynamic simulations of astrophysical phenomena, PHOEBUS.

PHOEBUS models general relativistic neutrino radiation magnetohydrodynamics. General rela-

tivistic hydrodynamics are incorporated with the Valencia formulation. MHD is incorporated using a cell centered constrained transport treatment and discretized with a finite volumes approach. Neutrino transport is incorporated using a Monte Carlo approach as well as a gray two moment scheme. Gravity is incorporated through analytic spacetimes and a novel monopole solver for core-collapse supernovae. The physics capabilities of PHOEBUS has been demonstrated through a suite of tests stressing individual physics implementations as well as the couplings between them.

PHOEBUS is developed on top of, and supported by, a large, open source ecosystem. PHOEBUS supports block based adaptive mesh refinement via the PARTHENON framework and achieves performance portability with the KOKKOS hardware agnostic library. Flexible, portable equations of state are supported through SINGULARITY-EOS. SPINER[8] enables performance portable storage and interpolation of tabular data, such as for equations of state and opacities (Miller et al., 2022). This open source ecosystem providing performance portability allows PHOEBUS developers to focus primarily on physics and numerics and ensures the longevity of the project.

There are a number of improvements planned for the near future that will greatly bolster the capabilities of PHOEBUS. Of note: we will adopt a proper face-centered-fields approach to MHD constrained transport. Moment based neutrino transport will be upgraded to allow for frequency dependent evolution. The finite volume discretization for hydrodynamics will be upgraded to be formally 4th order.

In the interest of open science, to provide a tool for the community, and allow for full reproducability, PHOEBUS and all parts of its ecosystem are open source. PHOEBUS is publicly available on GitHub. We welcome, and look forward to, contributions and engagement from the greater community.

---

[8]https://github.com/lanl/spiner

CHAPTER 8

SUMMARY

> Whatever happens next, I do
>
> not think it is to be feared.
>
> ———————————————
>
> The Prisoner, *Outer Wilds*

CHAPTER 9

APPENDICES

APPENDIX 9A

LIGHT CURVE COMPOSITIONAL DEPENDENCE

For our light curves, we modified the compositional profile in the FLASH part of the domain to be pure ^{4}He, as full composition is not currently tracked in the output. In this appendix, we provide comparisons of select light curves using thermal bombs with FLASH explosion energies for both the modified compositional profile and the original compositional profile. Figure 9A.1 shows light curves with the unaltered (orange) and modified (blue) compositional profiles for 9, 15.2, 25, and 30M$_\odot$ progenitors. For the cases considered here, the difference in luminosity on the plateau is bounded above by 0.1 dex, which has no meaningful affect on the iron core mass estimates and distributions of Section 2.3.3.
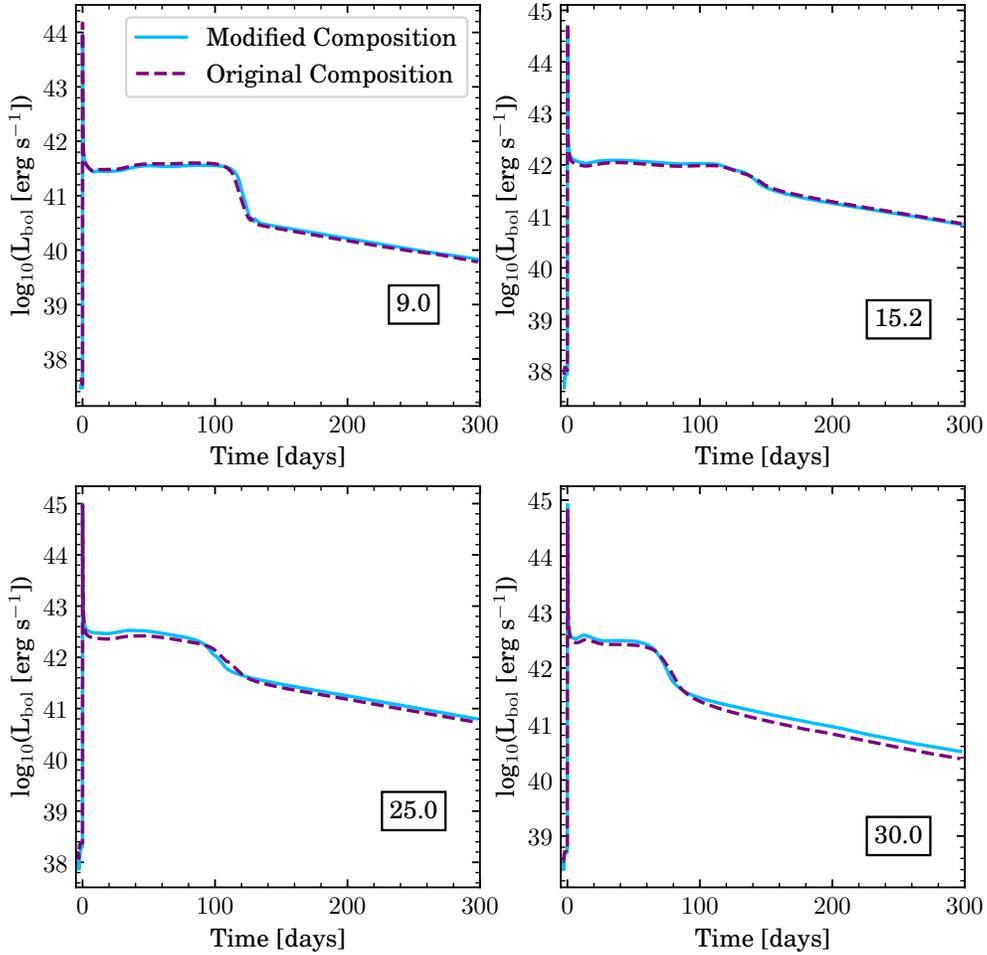
Figure 9A.1 Light curves using a thermal bomb driven explosion with STIR explosion energies using the modified compositional profile (blue) and unaltered profile (orange). We show light curves for 9, 15.2, 25, and 30M$_\odot$ progenitors.

χ^2 LIGHT CURVE FITTING

Here we show the effect of using the χ^2 metric to fit light curves, as opposed to the relative error metric discussed in Section 2.2.5. We define the chi-square (χ^2) metric for an observable quantity $f(t)$ as follows

$$\chi^2(f) = \sum_{t^*=t_1}^{t_N} \frac{\left(f_{t^*} - f_{t^*}^*\right)^2}{\sigma_f^2} \tag{9B.1}$$

where t^* are times coinciding with observations, f are synthetic observables, f^* are measured observational data, and σ_f is the uncertainty on the measurement f^* at a time t^*. Here we consider simultaneous fitting of luminosity and velocity data, i.e., minimizing the combined metric $\chi^2(v_{\mathrm{Fe}}) + \chi^2(L_{\mathrm{bol}})$. Figure 9B.1 shows the best fit model light curve for SN2017eaw using the chi-squared method (purple) and the relative error metric (blue). The light curve obtained with the chi-square method visibly fits the observations worse than the light curve obtained with the relative error approach, owing to the inverse square error weighting in the chi-square method. This weighting gives preference to the tail of the light curve where observational errors are reduced.
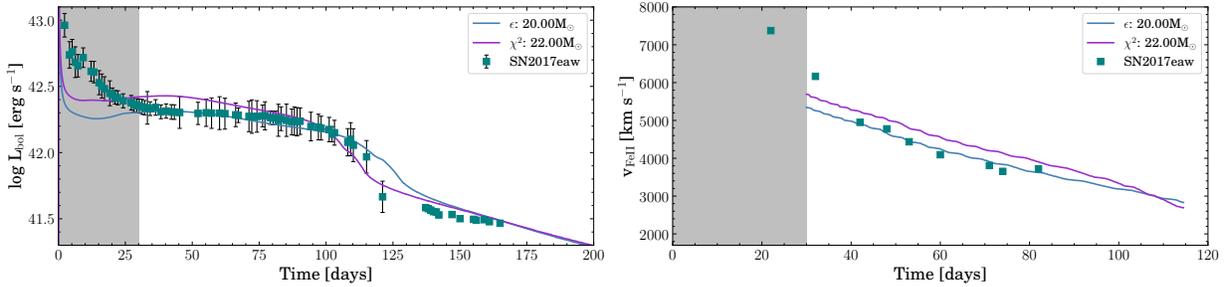
Figure 9B.1 Best fitting light curve for SN1017eaw obtained using a χ^2 metric (purple) and relative error metric (blue).

It is important to note that while chi-square minimization gave less satisfactory results for this study, this is likely sensitive to the details of the data being fit.

APPENDIX 9C

CHARACTERISTIC DECOMPOSITION

In this appendix, we provide the characteristic decomposition of the flux Jacobians, which are needed for slope limiting in characteristic fields. Recall that for the characteristic slope limiting described in 5.4.3, we require the eigendecomposition of the flux Jacobian

$$\frac{\partial \mathbf{F}^i(\mathbf{U})}{\partial \mathbf{U}} = \mathcal{R}^i \, \Lambda^i \, (\mathcal{R}^i)^{-1} \quad (i = 1, \dots, d). \tag{9C.1}$$

In the following, we will express the pressure from the EoS as $p = p(\tau, \epsilon, D_e)$; i.e., with independent variables $\tau = \rho^{-1}$, $\epsilon = e/\rho$, and $D_e = \rho \, Y_e$, instead of the usual function of ρ, T, and Y_e. This choice is arbitrary, but follows the approach outlined in Colella & Glaz (1985) for a general EoS without the addition of the conservation equation for electron number (cf. Equation (5.4)). The necessary transformations of thermodynamic derivatives between these two sets of independent variables are given in Appendix 9D. From the state and flux vectors given in Equation (5.7), we can calculate the following flux Jacobian matrices in each direction:

$$\frac{\partial \mathbf{F}^1(\mathbf{U})}{\partial \mathbf{U}} = \begin{bmatrix} 0 & \gamma^{11} & 0 & 0 & 0 & 0 \\ -v^1 v_1 - p_\tau \tau^2 - p_\epsilon \tau \left(\epsilon - \frac{v^i v_i}{2}\right) & v^1(2 - p_\epsilon \tau) & -p_\epsilon v^2 \tau & -p_\epsilon v^3 \tau & p_\epsilon \tau & p_{D_e} \\ -v^1 v_2 & \gamma^{11} v_2 & v^1 & 0 & 0 & 0 \\ -v^1 v_3 & \gamma^{11} v_3 & 0 & v^1 & 0 & 0 \\ v^1 \left(-H - p_\tau \tau^2 - p_\epsilon \tau \left(\epsilon - \frac{v^i v_i}{2}\right)\right) & \gamma^{11} H - p_\epsilon (v^1)^2 \tau & -p_\epsilon v^1 v^2 \tau & -p_\epsilon v^1 v^3 \tau & v^1(1 + p_\epsilon \tau) & v^1 p_{D_e} \\ -v^1 Y_e & \gamma^{11} Y_e & 0 & 0 & 0 & v^1 \end{bmatrix}, \tag{9C.2}$$

$$\frac{\partial \mathbf{F}^2(\mathbf{U})}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & \gamma^{22} & 0 & 0 & 0 \\ -v^2 v_1 & v^2 & \gamma^{22} v_1 & 0 & 0 & 0 \\ -v^2 v_2 - p_\tau \tau^2 - p_\epsilon \tau \left(\epsilon - \frac{v^i v_i}{2}\right) & -p_\epsilon v^1 \tau & v^2(2 - p_\epsilon \tau) & -p_\epsilon v^3 \tau & p_\epsilon \tau & p_{D_e} \\ -v^2 v_3 & 0 & \gamma^{22} v_3 & v^2 & 0 & 0 \\ v^2 \left(-H - p_\tau \tau^2 - p_\epsilon \tau \left(\epsilon - \frac{v^i v_i}{2}\right)\right) & -p_\epsilon v^2 v^1 \tau & \gamma^{22} H - p_\epsilon (v^2)^2 \tau & -p_\epsilon v^2 v^3 \tau & v^2(1 + p_\epsilon \tau) & v^2 p_{D_e} \\ -v^2 Y_e & 0 & \gamma^{22} Y_e & 0 & 0 & v^2 \end{bmatrix}, \tag{9C.3}$$

and

$$\frac{\partial \mathbf{F}^3(\mathbf{U})}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & 0 & \gamma^{33} & 0 & 0 \\ -v^3 v_1 & v^3 & 0 & \gamma^{33} v_1 & 0 & 0 \\ -v^3 v_2 & 0 & v^3 & \gamma^{33} v_2 & 0 & 0 \\ -v^3 v_3 - p_\tau \tau^2 - p_\epsilon \tau \left(\epsilon - \frac{v^i v_i}{2}\right) & -p_\epsilon v^1 \tau & -p_\epsilon v^2 \tau & v^3(2 - p_\epsilon \tau) & p_\epsilon \tau & p_{D_e} \\ v^3 \left(-H - p_\tau \tau^2 - p_\epsilon \tau \left(\epsilon - \frac{v^i v_i}{2}\right)\right) & -p_\epsilon v^3 v^1 \tau & -p_\epsilon v^3 v^2 \tau & \gamma^{33} H - p_\epsilon (v^3)^2 \tau & v^3(1 + p_\epsilon \tau) & v^3 p_{D_e} \\ -v^3 Y_e & 0 & 0 & \gamma^{33} Y_e & 0 & v^3 \end{bmatrix}, \tag{9C.4}$$

where we have defined the specific enthalpy of stagnation $H = \tau(E + p)$ and introduced the compact notation

$$p_\epsilon = \left(\frac{\partial p}{\partial \epsilon}\right)_{\tau, D_e}, \quad p_{D_e} = \left(\frac{\partial p}{\partial D_e}\right)_{\tau, \epsilon}, \quad p_\tau = \left(\frac{\partial p}{\partial \tau}\right)_{\epsilon, D_e} \tag{9C.5}$$

to express the necessary partial derivatives. The eigenvalues of the flux Jacobian are given by the diagonal matrix

$$\Lambda^i = \begin{bmatrix} v^i - c_s\sqrt{\gamma^{ii}} & 0 & 0 & 0 & 0 & 0 \\ 0 & v^i & 0 & 0 & 0 & 0 \\ 0 & 0 & v^i & 0 & 0 & 0 \\ 0 & 0 & 0 & v^i & 0 & 0 \\ 0 & 0 & 0 & 0 & v^i & 0 \\ 0 & 0 & 0 & 0 & 0 & v^i + c_s\sqrt{\gamma^{ii}} \end{bmatrix}, \tag{9C.6}$$

where $c_s = \sqrt{\Gamma p \tau}$, with

$$\Gamma = \left(\tau(pp_\epsilon - p_\tau) + p_{D_e}Y_e\tau^{-1}\right)p^{-1}, \tag{9C.7}$$

is the local sound speed. In the less general case where we ignore the electron contribution (i.e. $p_{D_e} = 0$), this reduces to the expression given by Colella & Glaz (1985). The right eigenvectors are then given by the column vectors of the following matrices

$$\mathcal{R}^1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ v_1 - c_s\sqrt{\gamma_{11}} & 0 & v_1 & v_1 & 0 & v_1 + c_s\sqrt{\gamma_{11}} \\ v_2 & 1 & 0 & 0 & 0 & v_2 \\ v_3 & 0 & 0 & 0 & 1 & v_3 \\ H - c_s\sqrt{\gamma_{11}}v^1 & v^2 & \beta_1 & 0 & v^3 & H + c_s\sqrt{\gamma_{11}}v^1 \\ Y_e & 0 & 0 & \frac{\tau \chi_1}{2p_{D_e}} & 0 & Y_e \end{bmatrix}, \tag{9C.8}$$

$$
\mathcal{R}^2 =
\begin{bmatrix}
1 & 0 & 1 & 1 & 0 & 1 \\
v_1 & 1 & 0 & 0 & 0 & v_1 \\
v_2 - c_\mathrm{s}\sqrt{\gamma_{22}} & 0 & v_2 & v_2 & 0 & v_2 + c_\mathrm{s}\sqrt{\gamma_{22}} \\
v_3 & 0 & 0 & 0 & 1 & v_3 \\
H - c_\mathrm{s}\sqrt{\gamma_{22}}v^2 & v^1 & \beta_2 & 0 & v^3 & H + c_\mathrm{s}\sqrt{\gamma_{22}}v^2 \\
Y_\mathrm{e} & 0 & 0 & \frac{\tau\chi_2}{2p_{D\mathrm{e}}} & 0 & Y_\mathrm{e}
\end{bmatrix},
\tag{9C.9}
$$

and

$$
\mathcal{R}^3 =
\begin{bmatrix}
1 & 0 & 1 & 1 & 0 & 1 \\
v_1 & 1 & 0 & 0 & 0 & v_1 \\
v_2 & 0 & 0 & 0 & 1 & v_2 \\
v_3 - c_\mathrm{s}\sqrt{\gamma_{33}} & 0 & v_3 & v_3 & 0 & v_3 + c_\mathrm{s}\sqrt{\gamma_{33}} \\
H - c_\mathrm{s}\sqrt{\gamma_{33}}v^3 & v^1 & \beta_3 & 0 & v^2 & H + c_\mathrm{s}\sqrt{\gamma_{33}}v^3 \\
Y_\mathrm{e} & 0 & 0 & \frac{\tau\chi_3}{2p_{D\mathrm{e}}} & 0 & Y_\mathrm{e}
\end{bmatrix},
\tag{9C.10}
$$

where the following definitions have been used:

$$
\begin{aligned}
\Delta_1 &= 2v^1 v_1 - v^i v_i, \\
\Delta_2 &= 2v^2 v_2 - v^i v_i, \\
\Delta_3 &= 2v^3 v_3 - v^i v_i, \\
\chi_i &= p_\epsilon(\Delta_i + 2\epsilon) + 2p_\tau\tau, \\
\beta_i &= \frac{1}{2}\left(\Delta_i + 2\epsilon + \frac{2p_\tau\tau}{p_\epsilon}\right).
\end{aligned}
$$

The left eigenvectors are given by the row vectors of the inverse matrix $\mathcal{L}^i = (\mathcal{R}^i)^{-1}$

$$(\mathcal{R}^1)^{-1} = \frac{1}{c_s^2} \begin{bmatrix} \frac{1}{4}\left(\omega + 2c_s\sqrt{\gamma_{11}}v^1\right) & -\frac{1}{2}\left(c_s\sqrt{\gamma^{11}} + \phi^1\right) & -\frac{1}{2}\phi^2 & -\frac{1}{2}\phi^3 & \frac{p_\epsilon\tau}{2} & \frac{p_{D_e}}{2} \\ -\frac{v_2\omega}{2} & \phi^1 v_2 & c_s^2 + \phi^2 v_2 & \phi^3 v_2 & -\phi_2 & -p_{D_e}v_2 \\ \frac{2\chi_1 c_s^2 + \alpha_1\omega\tau^{-1}}{2\chi_1} & -\frac{\phi^1\alpha_1}{\chi_1\tau} & -\frac{\phi^2\alpha_1}{\chi_1\tau} & -\frac{\phi^3\alpha_1}{\chi_1\tau} & \frac{p_\epsilon\alpha_1}{\chi_1} & \frac{p_{D_e}(\alpha_1 - 2c_s^2)}{\tau\chi_1} \\ -\frac{Y_e p_{D_e}\omega}{\chi_1\tau} & \frac{2Y_e p_{D_e}\phi^1}{\chi_1\tau} & \frac{2Y_e p_{D_e}\phi^2}{\chi_1\tau} & \frac{2Y_e p_{D_e}\phi^3}{\chi_1\tau} & -\frac{2Y_e p_{D_e}p_\epsilon}{\chi_1} & \frac{2p_{D_e}(c_s^2 - Y_e p_{D_e})}{\tau\chi_1} \\ -\frac{v_3\omega}{2} & \phi^1 v_3 & \phi^2 v_3 & c_s^2 + \phi^3 v_3 & -\phi_3 & -p_{D_e}v_3 \\ \frac{1}{4}\left(\omega - 2c_s\sqrt{\gamma_{11}}v^1\right) & \frac{1}{2}\left(c_s\sqrt{\gamma^{11}} - \phi^1\right) & -\frac{1}{2}\phi^2 & -\frac{1}{2}\phi^3 & \frac{p_\epsilon\tau}{2} & \frac{p_{D_e}}{2} \end{bmatrix}, \quad (9C.11)$$

$$(\mathcal{R}^2)^{-1} = \frac{1}{c_s^2} \begin{bmatrix} \frac{1}{4}\left(\omega + 2c_s\sqrt{\gamma_{22}}v^2\right) & -\frac{1}{2}\phi^1 & -\frac{1}{2}\left(c_s\sqrt{\gamma^{22}} + \phi^2\right) & -\frac{1}{2}\phi^3 & \frac{p_\epsilon\tau}{2} & \frac{p_{D_e}}{2} \\ -\frac{v_1\omega}{2} & c_s^2 + \phi^1 v_1 & \phi^2 v_1 & \phi^3 v_1 & -\phi_1 & -p_{D_e}v_1 \\ \frac{2\chi_2 c_s^2 + \alpha_2\omega\tau^{-1}}{2\chi_2} & -\frac{\phi^1\alpha_2}{\chi_2\tau} & -\frac{\phi^2\alpha_2}{\chi_2\tau} & -\frac{\phi^3\alpha_2}{\chi_2\tau} & \frac{p_\epsilon\alpha_2}{\chi_2} & \frac{p_{D_e}(\alpha_2 - 2c_s^2)}{\tau\chi_2} \\ -\frac{Y_e p_{D_e}\omega}{\chi_2\tau} & \frac{2Y_e p_{D_e}\phi^1}{\chi_2\tau} & \frac{2Y_e p_{D_e}\phi^2}{\chi_2\tau} & \frac{2Y_e p_{D_e}\phi^3}{\chi_2\tau} & -\frac{2Y_e p_{D_e}p_\epsilon}{\chi_2} & \frac{2p_{D_e}(c_s^2 - Y_e p_{D_e})}{\tau\chi_2} \\ -\frac{v_3\omega}{2} & \phi^1 v_3 & \phi^2 v_3 & c_s^2 + \phi^3 v_3 & -\phi_3 & -p_{D_e}v_3 \\ \frac{1}{4}\left(\omega - 2c_s\sqrt{\gamma_{22}}v^2\right) & -\frac{1}{2}\phi^1 & \frac{1}{2}\left(c_s\sqrt{\gamma^{22}} - \phi^2\right) & -\frac{1}{2}\phi^3 & \frac{p_\epsilon\tau}{2} & \frac{p_{D_e}}{2} \end{bmatrix}, \quad (9C.12)$$

and

$$(\mathcal{R}^3)^{-1} = \frac{1}{c_s^2} \begin{bmatrix} \frac{1}{4}\left(\omega + 2c_s\sqrt{\gamma_{33}}v^3\right) & -\frac{1}{2}\phi^1 & -\frac{1}{2}\phi^2 & -\frac{1}{2}\left(c_s\sqrt{\gamma^{33}} + \phi^3\right) & \frac{p_\epsilon\tau}{2} & \frac{p_{D_e}}{2} \\ -\frac{v_1\omega}{2} & c_s^2 + \phi^1 v_1 & \phi^2 v_1 & \phi^3 v_1 & -\phi_1 & -p_{D_e}v_1 \\ \frac{2\chi_3 c_s^2 + \alpha_3\omega\tau^{-1}}{2\chi_3} & -\frac{\phi^1\alpha_3}{\chi_3\tau} & -\frac{\phi^2\alpha_3}{\chi_3\tau} & -\frac{\phi^3\alpha_3}{\chi_3\tau} & \frac{p_\epsilon\alpha_3}{\chi_3} & \frac{p_{D_e}(\alpha_3 - 2c_s^2)}{\tau\chi_3} \\ -\frac{Y_e p_{D_e}\omega}{\chi_3\tau} & \frac{2Y_e p_{D_e}\phi^1}{\chi_3\tau} & \frac{2Y_e p_{D_e}\phi^2}{\chi_3\tau} & \frac{2Y_e p_{D_e}\phi^3}{\chi_3\tau} & -\frac{2Y_e p_{D_e}p_\epsilon}{\chi_3} & \frac{2p_{D_e}(c_s^2 - Y_e p_{D_e})}{\tau\chi_3} \\ -\frac{v_2\omega}{2} & \phi^1 v_2 & c_s^2 + \phi^2 v_2 & \phi^3 v_2 & -\phi_2 & -p_{D_e}v_2 \\ \frac{1}{4}\left(\omega - 2c_s\sqrt{\gamma_{33}}v^3\right) & -\frac{1}{2}\phi^1 & -\frac{1}{2}\phi^2 & \frac{1}{2}\left(c_s\sqrt{\gamma^{33}} - \phi^3\right) & \frac{p_\epsilon\tau}{2} & \frac{p_{D_e}}{2} \end{bmatrix}, \quad (9C.13)$$

where $\phi_i = p_\epsilon\,\tau\,v_i$, $\phi^i = p_\epsilon\,\tau\,v^i$, $\omega = \tau\left(p_\epsilon\left(v^i v_i - 2\epsilon\right) - 2\,p_\tau\,\tau\right)$, and $\alpha_i = 2Y_e p_{D_e} - \tau\chi_i$.

THERMODYNAMIC DERIVATIVES

In Appendix 9C, to compute the flux Jacobian matrices, we expressed the pressure $p = p(\tau, \epsilon, D_e)$ in terms of the independent variables $\tau = \rho^{-1}$, $\epsilon = e/\rho$, and $D_e = \rho Y_e$. On the other hand, the tabulated EoS constructs thermodynamic variables in terms of ρ, T, and Y_e. Thus, we need to express the thermodynamic derivatives of pressure necessary for the characteristic decomposition in terms of the independent variables from the EoS table. We start with the differential of pressure

$$dp = (\partial_\tau p)_{\epsilon, D_e} d\tau + (\partial_\epsilon p)_{\tau, D_e} d\epsilon + (\partial_{D_e} p)_{\epsilon, \tau} dD_e. \tag{9D.1}$$

Similarly, we may express the differentials of τ, ϵ, and D_e in terms of differentials of the table variables

$$d\tau = -\rho^{-2} d\rho,$$

$$d\epsilon = (\partial_\rho \epsilon)_{T, Y_e} d\rho + (\partial_T \epsilon)_{\rho, Y_e} dT + (\partial_{Y_e} \epsilon)_{T, \rho} dY_e,$$

$$dD_e = (\partial_\rho D_e)_{T, Y_e} d\rho + (\partial_T D_e)_{\epsilon, Y_e} dT + (\partial_{Y_e} D_e)_{T, \epsilon} dY_e \tag{9D.2}$$

$$= Y_e d\rho + \rho dY_e.$$

Inserting these differentials into Equation (9D.1), we find another expression for the pressure differential

$$dp = \left[-\rho^2 (\partial_\tau p)_{\epsilon, D_e} + (\partial_\rho \epsilon)_{T, Y_e} (\partial_\epsilon p)_{\tau, D_e} + Y_e (\partial_{D_e} p)_{\epsilon, \tau} \right] d\rho$$

$$+ (\partial_T \epsilon)_{\rho, Y_e} (\partial_\epsilon p)_{\tau, D_e} dT \tag{9D.3}$$

$$+ \left[(\partial_{Y_e} \epsilon)_{T, \rho} (\partial_\epsilon p)_{\tau, D_e} + \rho (\partial_{D_e} p)_{\epsilon, \tau} \right] dY_e.$$

On the other hand, we have the differential of pressure in terms of the table variables

$$dp = (\partial_\rho p)_{T, Y_e} d\rho + (\partial_T p)_{\rho, Y_e} dT + (\partial_{Y_e} p)_{T, \rho} dY_e. \tag{9D.4}$$

Comparing Equation (9D.3) and Equation (9D.4), we have the system of equations

$$\begin{bmatrix} -\rho^2 & (\partial_\rho \epsilon)_{T, Y_e} & Y_e \\ 0 & (\partial_T \epsilon)_{\rho, Y_e} & 0 \\ 0 & (\partial_{Y_e} \epsilon)_{T, \rho} & \rho \end{bmatrix} \begin{bmatrix} (\partial_\tau p)_{\epsilon, D_e} \\ (\partial_\epsilon p)_{\tau, D_e} \\ (\partial_{D_e} p)_{\epsilon, \tau} \end{bmatrix} = \begin{bmatrix} (\partial_\rho p)_{T, Y_e} \\ (\partial_T p)_{\rho, Y_e} \\ (\partial_{Y_e} p)_{T, \rho} \end{bmatrix}. \tag{9D.5}$$

Solving, with some simplifications, we find the derivatives of the pressure with respect to τ, ϵ, and $D_{\rm e}$ in terms of the table variables ρ, T, $Y_{\rm e}$

$$\left(\frac{\partial p}{\partial \epsilon}\right)_{\tau,D_{\rm e}} = \left(\frac{\partial \epsilon}{\partial T}\right)_{\rho,Y_{\rm e}}^{-1} \left(\frac{\partial p}{\partial T}\right)_{\rho,Y_{\rm e}}, \tag{9D.6}$$

$$\left(\frac{\partial p}{\partial D_{\rm e}}\right)_{\tau,\epsilon} = \tau \left[\left(\frac{\partial p}{\partial Y_{\rm e}}\right)_{\rho,T} - \left(\frac{\partial \epsilon}{\partial Y_{\rm e}}\right)_{\rho,T} \left(\frac{\partial p}{\partial \epsilon}\right)_{\tau,D_{\rm e}}\right], \tag{9D.7}$$

$$\left(\frac{\partial p}{\partial \tau}\right)_{\epsilon,D_{\rm e}} = \tau^{-2} \left[Y_{\rm e}\left(\frac{\partial p}{\partial D_{\rm e}}\right)_{\tau,\epsilon} + \left(\frac{\partial \epsilon}{\partial \rho}\right)_{Y_{\rm e},T} \left(\frac{\partial p}{\partial \epsilon}\right)_{\tau,D_{\rm e}} - \left(\frac{\partial p}{\partial \rho}\right)_{T,Y_{\rm e}}\right]. \tag{9D.8}$$

We use these relations to relate derivatives needed for the characteristic decomposition in Appendix 9C to derivatives obtained from table interpolations.

OUR NOVEL WENO5-Z-AOAH SCHEME

Consider a grid of cells of equal width Δx with centers at positions x_i for $i = 0, 1, \ldots, N$ for some $N \geq 5$. A function $f(x)$ is known at cell centers, with values $f_i = f(x_i)$. For some index j we wish to *reconstruct* the value of f at the cell face between centers x_j and x_{j+1}, i.e., at $x_{j+1/2}$. Ideally, if f is a smooth function, this reconstruction should be high-order, such that the truncation error is small. However, if f is *not* smooth, then the reconstruction should be robust and minimize the Gibbs oscillations that emerge from high-order representations of non-smooth functions (Wilbraham, 1848; Gibbs, 1898).

The WENO family of methods, first described by Shu (2009) seek to solve the above-described problem. For smooth problems, WENO constructs a high-order interpolant from the linear combination of several lower-order interpolants. For example, a fifth-order interpolant, $P^{(5)}(x_{j+1/2})$ evaluated at $x_{j+1/2}$ may be constructed from the linear combination of three third-order interpolants $P_k^{(3)}$, $k = 0, 1, 2$:

$$P^{(5)}(x_{j+1/2}) = \sum_{k=0}^{2} \gamma_k P_k^{(3)}(x_{j+1/2}), \tag{9E.1}$$

where $P_k^{(3)}$ are third-order Lagrange polynomials computed using stencils that are upwind of, downwind of, and centered around x_j respectively. In other words,

$$P_k^{(3)} = \sum_{l=0}^{2} \alpha_{kl} f_{j-2+k+l} \tag{9E.2}$$

for some coefficients α_{kl}. The lower-order stencils are combined via the *linear weights γ*.

The construction (9E.1) is ideal for smooth problems, but suffers the Gibbs phenomenon in when $f(x)$ is non-smooth. To resolve this issue, the linear weights γ are rescaled to become the *nonlinear weights w_i*. In the WENO-Z construction of Borges et al. (2008), the nonlinear weights are given by

$$w_k = \gamma_k \left[1 + \left(\frac{\tau_Z}{\beta_k + \varepsilon} \right)^p \right] \tag{9E.3}$$

for some small number ε and some power p, both free parameters, with normalization w such that $\sum_k w_k = 1$. Here

$$\tau_z = \beta_2 - \beta_0$$

is a *global smoothness indicator* and the β_ks are *local smoothness indicators* defined by[1]

$$\beta_k = \sum_{l=1}^{3} \Delta x^{2l-1} \int_{x_{j-1/2}}^{x_{j+1/2}} \left(\frac{d^l}{dx^l} P_k^{(3)}(x) \right)^2 dx. \qquad (9E.4)$$

Conceptually, the smoothness indicator β_k checks whether an interpolant $P_k^{(3)}$ suffers the Gibbs phenomenon, and if it does, the nonlinear weight w_k de-emphasizes that interpolant in favor of the others, suppressing the Gibbs phenomenon.

In a finite volumes context each face has two possible values, one reconstructed using the jth cell and one using the $j + 1$ cell. Both are required to pose a Riemann problem to pass into the Riemann solver. This implies that for each cell j, we must reconstruct values at both the $j + 1/2$ face and the $j - 1/2$ face. The value for the $j - 1/2$ face can be constructed by performing a mapping $x \rightarrow -x$ and then repeating the procedure described above.

In our experiments, we found that this approach still suffered Gibbs oscillations for strong shocks, more severely than a limited piecewise linear reconstruction. Therefore, inspired by the adaptive order WENO approaches introduced in Balsara et al. (2016), we introduce order ad-hoc order adaptivity to the WENO5-Z reconstruction described above. We do so by mixing in a linearized term into the reconstruction:

$$f(x_{j+1/2} = \sigma P^{(5)}(x_{j+1/2}) + (1 - \sigma) P_{\text{plm}}(x_{j+1/2}) \qquad (9E.5)$$

where σ is a weight, defined below and P_{plm} is an appropriately limited piecewise linear reconstruction of the face-centered value. We use a monotonized central limiter for the linear term.

The mixing term σ is constructed by leveraging the fact that, for a smooth problem, the reconstruction of the $j + 1/2$ face and the $j - 1/2$ face should be evaluations of the exact same polynomial. However, when the problem is non-smooth, and the smoothness indicators trigger, the

---

[1]Note we specify polynomial degree 3 explicitly here. Broadly the degree should be the order of the lower-order polynomials.

combination of the $x \rightarrow -x$ mapping and the nonlinearity in the weights means the polynomials are *not* the same.

To measure these differences, a measure of "nonlinearity" in the weights is constructed as a harmonic mean of the linear and nonlinear weights for either the $j + 1/2$ or $j - 1/2$ face.

$$\sigma_{j\pm1/2} = 3 \frac{w_0^{j\pm1/2} w_1^{j\pm1/2} w_2^{j\pm1/2}}{\gamma_2^{j\pm1/2} w_0^{j\pm1/2} w_1^{j\pm1/2} + \gamma_1^{j\pm1/2} w_0^{j\pm1/2} w_2^{j\pm1/2} + \gamma_2^{j\pm1/2} w_0^{j\pm1/2} w_1^{j\pm1/2}} \qquad (9E.6)$$

where here we temporarily introduce the $j \pm 1/2$ superscript to indicate these weights are for either the $j + 1/2$ face or the $j - 1/2$ face respectively. σ is then constructed as the harmonic mean of $\sigma_{j\pm1/2}$:

$$\sigma = 2 \frac{\sigma_{j+1/2} \sigma_{j-1/2}}{\sigma_{j+1/2} + \sigma_{j-1/2}}. \qquad (9E.7)$$

Thus, $0 < \sigma < 1$ and when $\sigma_{j+1/2} = \sigma_{j-1/2}$, our scheme reduces to the standard WENO5-Z method, when $\sigma_{j+1/2}$ and $\sigma_{j-1/2}$ differ significantly, σ will become small and our scheme reduces to a limited piecewise linear reconstruction.

We note that the recently developed adaptive order WENO approaches such as described in Balsara et al. (2016) formalize and generalize this idea. However, the approach described here has the advantage of being particularly simple compared to the more general treatment, and we have found it to be very effective. We call this method WENO5-Z-AOAH, for WENO5-Z-AO At Home.

TIME-DERIVATIVES OF THE MONOPOLE METRIC

To compute the Christoffel symbols, one needs derivatives of the metric in both space and time. The spatial derivatives are straightforward: with a, K, α, and β^r known, simply differentiate with respect to r. The time derivatives are more subtle. We use the Einstein evolution equations to derive them.

First we note that ADM evolution equation for the metric reduces to the following in spherical symmetry:

$$
\begin{aligned}
\partial_t g_{\mu\nu} &= \beta^\sigma \partial_\sigma g_{\mu\nu} + g_{\sigma\mu} \partial_\nu \beta^\sigma + g_{\sigma\nu} \partial_\mu \beta^\sigma - 2\alpha K_{\mu\nu} \\
&= \beta^r \partial_r g_{\mu\nu} + g_{r\mu} \partial_\nu \beta^r + g_{r\nu} \partial_\mu \beta^r - 2\alpha K_{\mu\nu}.
\end{aligned}
\tag{9F.1}
$$

The rr-component of equation (9F.1) yields an equation for the derivative of a:

$$
\begin{aligned}
\partial_t g_{rr} &= \beta^r \partial_r (a^2) + 2g_{rr} \partial_r \beta^r - 2\alpha K_{rr} \\
&= 2aa'\beta^r + 2a^2 \partial_r \beta^r - 2\alpha a^2 K^r_r \\
&= 2a(a'\beta^r + a(\beta^r)' - \alpha a K^r_r). \\
&= -2\alpha \partial_t \alpha + 2a(\beta^r)^2 \partial_t a - ra^2 \beta^r (\alpha \partial_t K^r_r + K^r_r \partial_t \alpha)
\end{aligned}
\tag{9F.2}
$$

But note that

$$
\partial_t g_{rr} = \partial_t a^2 = 2a \partial_t a,
$$

so

$$
\partial_t a = a'\beta^r + a(\beta^r)' - \alpha a K^r_r.
\tag{9F.3}
$$

The lapse proceeds similarly. We start with the fact that

$$
\begin{aligned}
g_{tt} &= -\alpha^2 + \beta^2 \\
&= -\alpha^2 + a^2 (\beta^r)^2 \\
\Rightarrow \partial_t g_{tt} &\quad -2\alpha \partial_t \alpha + 2a(\beta^r)^2 \partial_t a + 2a^2 \beta^r \partial_t \beta^r \\
&= -2\alpha \partial_t \alpha + 2a(\beta^r)^2 \partial_t a + 2a^2 \beta^r \left[-\frac{1}{2} r (\alpha \partial_t K^r_r + K^r_r \partial_t \alpha) \right]
\end{aligned}
$$

We then proceed to apply the metric evolution equation (9F.1) on g_{tt}:

$$
\begin{aligned}
\partial_t g_{tt} &= -\beta^r \partial_r g_{tt} + 2g_{rt}\partial_t \beta^r - 2\alpha K_{tt} \\
&= \beta^r \partial_r g_{tt} + 2g_{rt}\left[-\frac{1}{2}r(\alpha \partial_t K_r^r + K_r^r \partial_t \alpha)\right] - 2\alpha K_{tt} \\
&= \beta^r \partial_t g_{tt} - r a^2 \beta^r (\alpha \partial_t K_r^r + K_r^r \partial_t \alpha) - 2\alpha a^2 K_r^r (\beta^r)^2
\end{aligned}
\tag{9F.4}
$$

We now combine equations (9F.4) and (9F.4). The

$$
r a^2 \beta^r (\alpha \partial_t K_r^r + K_r^r \partial_t \alpha)
$$

term cancels and we find that

$$
-2\alpha \partial_t \alpha + 2a(\beta^r)^2 \partial_t a = \beta^r \partial_r g_{tt} - 2a^2 \alpha K_r^r (\beta^r)^2
$$

$$
\begin{aligned}
\Rightarrow \partial_t \alpha &= \frac{a}{\alpha}(\beta^r)^2 \dot{a} + a^2 K_r^r (\beta^r)^2 - \frac{\beta^r}{\alpha}\partial_r g_{tt} \\
&= \frac{a}{\alpha}(\beta^r)^2 \dot{a} + a^2 K_r^r (\beta^r)^2 - \frac{\beta^r}{\alpha}\left[-2\alpha \partial_r \alpha + 2\alpha(\beta^r)^2 \partial_r a + 2a^2 \beta^r \partial_r \beta^r\right] \\
&= \frac{a}{\alpha}(\beta^r)^2 \dot{a} + a^2 K_r^r (\beta^r)^2 + 2\beta^r \alpha' - 2(\beta^r)^3 a' - 2a^2 \frac{(\beta^r)^2}{\alpha}\partial_r \beta^r \\
&= \beta^r \left[\frac{a\beta^r \dot{a}}{\alpha} + a^2 K_r^r \beta^r + 2a'(1 - (\beta^r)^2) - 2\frac{a^2 \beta^r}{\alpha}\partial_r \beta^r\right]
\end{aligned}
\tag{9F.5}
$$

Finally, we reach the time derivative of β. To calculate it, we require the time derivative of K_r^r, which comes from the ADM evolution equation for the extrinsic curvature. We begin with the following relation:

$$
\begin{aligned}
K_j^i &= \gamma^{ik} K_{kj} \\
\Rightarrow (\partial_t - \mathcal{L}_\beta) K_j^i &= \gamma^{ik}(\partial_t - \mathcal{L}_\beta) K_{kj} + K_{kj}(\partial_t - \mathcal{L}_\beta)\gamma^{ik} \\
&= \gamma^{ik}(\partial_t - \mathcal{L}_\beta) K_{kj} - 2\alpha K_k^i K_j^k,
\end{aligned}
\tag{9F.6}
$$

We thus have

$$
\begin{aligned}
(\partial_t - \mathcal{L}_\beta) K_j^i &= -D^i D_j \alpha + \alpha \left[^{(3)}R_j^i + KK_j^i - 2K_k^i K_j^k\right] + 4\pi\alpha \left[\delta_j^i(S - \rho) - 2S_j^i\right] - 2\alpha K_k^i K_j^k \\
&= -D^i D_j \alpha + \alpha \left[^{(3)}R_j^i - 4K_k^i K_j^k\right] + 4\pi\alpha \left[\delta_j^i(S - \rho) - 2S_j^i\right] \\
\Rightarrow \partial_t K_j^i &= \beta^k \partial_k K_j^i - K_j^k \partial_k \beta^i + K_k^i \partial_j \beta^k - D^i D_j \alpha + \alpha \left[^{(3)}R_j^i - 4K_k^i K_j^k\right] + 4\pi\alpha \left[\delta_j^i(S - \rho) - 2S_j^i\right].
\end{aligned}
$$

When we specialize to $i = j = r$, this becomes

$$\partial_t K_r^r = \beta^r \partial_r K_r^r - K_r^r \partial_r \beta^r + K_r^r \partial_r \beta 2 - D^r D_r \alpha + \alpha \left[^{(3)} R_r^r - 4(K_r^r)^2\right] + 4\pi\alpha \left[S - \rho - 2S_r^r\right]$$

$$= \beta^r \partial_r K_r^r - D^r D_r \alpha + \alpha \left[^{(3)} R_r^r - 4(K_r^r)^2\right] - 4\pi\alpha(S - \rho - 2S_r^r). \qquad (9\text{F}.7)$$

Given metric ansatz (7.15), we have

$$^{(3)}R_{rr} = \frac{2}{abr} \{a'[(b + rb') - arb''] - 2ab'\}$$

$$= \frac{2}{ar} a'$$

$$\Rightarrow {}^{(3)} R_r^r = \frac{2}{a^3 r} a' \qquad (9\text{F}.8)$$

and since α is a scalar,

$$D^i D_j \alpha = D^i \partial_j \alpha$$

$$= \gamma^{ik} D_k \partial_j \alpha$$

$$= \gamma^{ik} (\partial_k \partial_j \alpha - {}^{(3)} \Gamma_{kj}^l \partial_l \alpha)$$

$$= \gamma^{ik} (\partial_k \partial_j \alpha - {}^{(3)} \Gamma_{kj}^r \partial_r \alpha) \text{ because only non-trivial derivatives are in } r$$

$$\Rightarrow D^r D_r \alpha = \gamma^{rk} (\partial_k \partial_r \alpha - {}^{(3)} \Gamma_{kr}^r \partial_r \alpha)$$

$$= \gamma^{rr} (\partial_r^2 \alpha - {}^{(3)} \Gamma_{rr}^r \alpha) \text{ because } \gamma \text{ is diagonal}$$

$$= \frac{1}{a^2} \left(\partial_r^2 \alpha - \frac{a'}{a} \partial_r \alpha\right)$$

$$= \frac{1}{a^2} \partial_r^2 \alpha - \frac{a'}{a^3} \partial_r \alpha \qquad (9\text{F}.9)$$

Which implies

$$\partial_t K_r^r = \beta^r \partial_r K_r^r - \frac{1}{a^2} \partial_r^2 \alpha + \frac{a'}{a^3} \partial_r \alpha + \alpha \left[\frac{2}{a^3 r} a' - 4(K_r^r)^2\right] + 4\pi\alpha \left[S - \rho - 2S_r^r\right] \qquad (9\text{F}.10)$$

which provides, along with equation (7.21) and the chain rule, a solution for the time-derivative of the shift.

We note that S_r^r can be computed simply in spherical symmetry from S via the following reasoning. In spherical symmetry, the spatial stress tensor must be diagonal. Moreover, the

diagonal components must be

$$S_\theta^\theta = S_\phi^\phi = P + \frac{1}{2}b^2.$$

Therefore,

$$
\begin{aligned}
S_r^r &= S - S_\theta^\theta - S_\phi^\phi \\
&= S - 2(P + \frac{1}{2}b^2) \\
&= \tau + D + P - (\rho + u) - P_\mu^i P_i^\nu b^\mu b_\nu
\end{aligned}
\tag{9F.11}
$$

We use this relation.

BIBLIOGRAPHY

Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2016, Phys. Rev. D, 94, 102001, doi: `10.1103/PhysRevD.94.102001`

Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2016, Physical Review Letters, 116, 061102, doi: `10.1103/PhysRevLett.116.061102`

—. 2017a, Physical Review Letters, 118, 221101, doi: `10.1103/PhysRevLett.118.221101`

—. 2017b, Physical Review Letters, 119, 161101, doi: `10.1103/PhysRevLett.119.161101`

Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2020, Phys. Rev. D, 101, 084002, doi: `10.1103/PhysRevD.101.084002`

Abbott, R., Abbott, T. D., Abraham, S., et al. 2020, ApJ, 896, L44, doi: `10.3847/2041-8213/ab960f`

Abdikamalov, E., Ott, C. D., Radice, D., et al. 2015, ApJ, 808, 70, doi: `10.1088/0004-637X/808/1/70`

Adams, M. L. 2001, Nuclear science and engineering, 137, 298

Adams, S. M., Kochanek, C. S., Beacom, J. F., Vagins, M. R., & Stanek, K. Z. 2013, ApJ, 778, 164, doi: `10.1088/0004-637X/778/2/164`

Adams, S. M., Kochanek, C. S., Gerke, J. R., Stanek, K. Z., & Dai, X. 2017, MNRAS, 468, 4968, doi: `10.1093/mnras/stx816`

Akiyama, S., Wheeler, J. C., Meier, D. L., & Lichtenstadt, I. 2003, ApJ, 584, 954, doi: `10.1086/344135`

Alcubierre, M. 2008, Introduction to 3+1 Numerical Relativity (Oxford University Press), doi: `10.1093/acprof:oso/9780199205677.001.0001`. `https://doi.org/10.1093/acprof:oso/9780199205677.001.0001`

Almgren, A. S., Beckner, V. E., Bell, J. B., et al. 2010, ApJ, 715, 1221, doi: `10.1088/0004-637X/715/2/1221`

Anderson, J. P., González-Gaitán, S., Hamuy, M., et al. 2014, ApJ, 786, 67, doi: `10.1088/0004-637X/786/1/67`

Anderson, T. W., & Darling, D. A. 1952, The Annals of Mathematical Statistics, 23, 193 , doi: `10.1214/aoms/1177729437`

Arnett, W. D. 1977, ApJ, 218, 815, doi: 10.1086/155738

—. 1980, ApJ, 237, 541, doi: 10.1086/157898

Arnett, W. D., Bahcall, J. N., Kirshner, R. P., & Woosley, S. E. 1989, ARA&A, 27, 629, doi: 10.1146/annurev.aa.27.090189.003213

Arnett, W. D., & Fu, A. 1989, ApJ, 340, 396, doi: 10.1086/167402

Arnett, W. D., & Meakin, C. 2011, ApJ, 733, 78, doi: 10.1088/0004-637X/733/2/78

Baade, W., & Zwicky, F. 1934, Proceedings of the National Academy of Science, 20, 254, doi: 10.1073/pnas.20.5.254

Balsara, D. S., Garain, S., & Shu, C.-W. 2016, Journal of Computational Physics, 326, 780, doi: https://doi.org/10.1016/j.jcp.2016.09.009

Banyuls, F., Font, J. A., Ibáñez, J. M., Martí, J. M., & Miralles, J. A. 1997, ApJ, 476, 221, doi: 10.1086/303604

Barbon, R., Ciatti, F., & Rosino, L. 1979, A&A, 72, 287

Barker, B. L., Harris, C. E., Warren, M. L., O'Connor, E. P., & Couch, S. M. 2022, ApJ, 934, 67, doi: 10.3847/1538-4357/ac77f3

Barker, B. L., O'Connor, E. P., & Couch, S. M. 2023, ApJ, 944, L2, doi: 10.3847/2041-8213/acb052

Baron, E., & Cooperstein, J. 1990, ApJ, 353, 597, doi: 10.1086/168649

Bassi, F., Franchina, N., Ghidoni, A., & Rebay, S. 2013, International Journal for Numerical Methods in Fluids, 71, 1322, doi: 10.1002/fld.3713

Baumgarte, T. W., & Shapiro, S. L. 2010, Numerical Relativity: Solving Einstein's Equations on the Computer

Bersten, M. C., Benvenuto, O., & Hamuy, M. 2011, ApJ, 729, 61, doi: 10.1088/0004-637X/729/1/61

Bersten, M. C., & Hamuy, M. 2009, ApJ, 701, 200, doi: 10.1088/0004-637X/701/1/200

Bethe, H. A. 1990, Reviews of Modern Physics, 62, 801, doi: 10.1103/RevModPhys.62.801

Bethe, H. A., & Wilson, J. R. 1985, ApJ, 295, 14, doi: 10.1086/163343

Betranhandy, A., & O'Connor, E. 2020, Phys. Rev. D, 102, 123015, doi: `10.1103/PhysRevD.102.123015`

Biswas, R., Devine, K. D., & Flaherty, J. E. 1994, Applied Numerical Mathematics, 14, 255 , doi: `https://doi.org/10.1016/0168-9274(94)90029-9`

Blandford, R. D., & McKee, C. F. 1976, Physics of Fluids, 19, 1130, doi: `10.1063/1.861619`

Blinnikov, S. I., & Bartunov, O. S. 1993, A&A, 273, 106

Blondin, J. M., & Lufkin, E. A. 1993, ApJS, 88, 589, doi: `10.1086/191834`

Blondin, J. M., Mezzacappa, A., & DeMarino, C. 2003, ApJ, 584, 971, doi: `10.1086/345812`

Boccioli, L., Mathews, G. J., & O'Connor, E. P. 2021, ApJ, 912, 29, doi: `10.3847/1538-4357/abe767`

Boccioli, L., Mathews, G. J., Suh, I.-S., & O'Connor, E. P. 2022, ApJ, 926, 147, doi: `10.3847/1538-4357/ac4603`

Bollig, R., Janka, H.-T., Lohs, A., et al. 2017, Physical Review Letters, 119, 242702, doi: `10.1103/PhysRevLett.119.242702`

Borges, R., Carmona, M., Costa, B., & Don, W. S. 2008, Journal of Computational Physics, 227, 3191, doi: `https://doi.org/10.1016/j.jcp.2007.11.038`

Branch, D., & Wheeler, J. C. 2017, Supernova Explosions, doi: `10.1007/978-3-662-55054-0`

Bruenn, S. W. 1985, ApJS, 58, 771, doi: `10.1086/191056`

Bruenn, S. W., Raley, E. A., & Mezzacappa, A. 2004, arXiv e-prints, astro. `https://arxiv.org/abs/astro-ph/0404099`

Bruenn, S. W., Lentz, E. J., Hix, W. R., et al. 2016, ApJ, 818, 123, doi: `10.3847/0004-637X/818/2/123`

Bruenn, S. W., Blondin, J. M., Hix, W. R., et al. 2020, ApJS, 248, 11, doi: `10.3847/1538-4365/ab7aff`

Buras, R., Janka, H.-T., Keil, M. T., Raffelt, G. G., & Rampp, M. 2003, ApJ, 587, 320, doi: `10.1086/368015`

Buras, R., Janka, H. T., Rampp, M., & Kifonidis, K. 2006, A&A, 457, 281, doi: `10.1051/0004-6361:20054654`

Burrows, A. 2013, Reviews of Modern Physics, 85, 245, doi: 10.1103/RevModPhys.85.245

Burrows, A., Hayes, J., & Fryxell, B. A. 1995, ApJ, 450, 830, doi: 10.1086/176188

Burrows, A., Radice, D., Vartanyan, D., et al. 2020, MNRAS, 491, 2715, doi: 10.1093/mnras/stz3223

Burrows, A., Reddy, S., & Thompson, T. A. 2006, Nuclear Physics A, 777, 356 , doi: https://doi.org/10.1016/j.nuclphysa.2004.06.012

Burrows, A., & Sawyer, R. F. 1998, Phys. Rev. C, 58, 554, doi: 10.1103/PhysRevC.58.554

Burrows, A., & Vartanyan, D. 2021, Nature, 589, 29, doi: 10.1038/s41586-020-03059-w

Cardall, C. Y., Budiardja, R. D., Endeve, E., & Mezzacappa, A. 2014, ApJS, 210, 17, doi: 10.1088/0067-0049/210/2/17

Cardall, C. Y., Endeve, E., & Mezzacappa, A. 2013, Phys. Rev. D, 87, 103004, doi: 10.1103/PhysRevD.87.103004

Casanova, J., Endeve, E., Lentz, E. J., et al. 2020, Phys. Scr, 95, 064005, doi: 10.1088/1402-4896/ab7dd1

Cernohorsky, J., & Bludman, S. A. 1994, ApJ, 433, 250, doi: 10.1086/174640

Chandra, R., Dagum, L., Kohr, D., et al. 2001, Parallel programming in OpenMP (Morgan kaufmann)

Chieffi, A., & Limongi, M. 2020, ApJ, 890, 43, doi: 10.3847/1538-4357/ab6739

Chu, R., Endeve, E., Hauck, C. D., & Mezzacappa, A. 2019, Journal of Computational Physics, 389, 62 , doi: https://doi.org/10.1016/j.jcp.2019.03.037

Clauset, A., Shalizi, C. R., & Newman, M. E. J. 2009, SIAM Review, 51, 661, doi: 10.1137/070710111

Cockburn, B. 2001, Journal of Computational and Applied Mathematics, 128, 187, doi: 10.1016/S0377-0427(00)00512-4

Cockburn, B., Hou, S., & Shu, C.-W. 1990, Mathematics of Computation, 54, 545, doi: 10.1090/S0025-5718-1990-1010597-0

Cockburn, B., Lin, S.-Y., & Shu, C.-W. 1989, Journal of Computational Physics, 84, 90, doi: 10.1016/0021-9991(89)90183-6

Cockburn, B., & Shu, C.-W. 1989, Math. Comp., 52, 411

Cockburn, B., & Shu, C.-W. 1991, ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique, 25, 337

Cockburn, B., & Shu, C.-W. 1998, Journal of Computational Physics, 141, 199, doi: 10.1006/jcph.1998.5892

Colella, P., & Glaz, H. M. 1985, Journal of Computational Physics, 59, 264, doi: 10.1016/0021-9991(85)90146-9

Colella, P., & Woodward, P. R. 1984, Journal of Computational Physics, 54, 174, doi: 10.1016/0021-9991(84)90143-8

Couch, S. M. 2017, Philosophical Transactions of the Royal Society of London Series A, 375, 20160271, doi: 10.1098/rsta.2016.0271

Couch, S. M., & Ott, C. D. 2013, ApJ, 778, L7, doi: 10.1088/2041-8205/778/1/L7

—. 2015, ApJ, 799, 5, doi: 10.1088/0004-637X/799/1/5

Couch, S. M., Warren, M. L., & O'Connor, E. P. 2020, ApJ, 890, 127, doi: 10.3847/1538-4357/ab609e

Curtis, S., Wolfe, N., Fröhlich, C., et al. 2021, ApJ, 921, 143, doi: 10.3847/1538-4357/ac0dc5

Davies, B., & Beasor, E. R. 2018, MNRAS, 474, 2116, doi: 10.1093/mnras/stx2734

—. 2020, MNRAS, 493, 468, doi: 10.1093/mnras/staa174

Davies, S. F. 1988, SIAM J. Sci. Stat. Comput., 9, 445

Dessart, L., & Audit, E. 2019, A&A, 629, A17, doi: 10.1051/0004-6361/201935794

Dessart, L., & Hillier, D. J. 2019, A&A, 625, A9, doi: 10.1051/0004-6361/201834732

Dib, S. 2014, MNRAS, 444, 1957, doi: 10.1093/mnras/stu1521

Dolence, J. C., Gammie, C. F., Mościbrodzka, M., & Leung, P. K. 2009, ApJS, 184, 387, doi: 10.1088/0067-0049/184/2/387

Dubey, A., Antypas, K., Ganapathy, M. K., et al. 2009, Parallel Computing, 35, 512 , doi: https://doi.org/10.1016/j.parco.2009.08.001

Dubey, A., Weide, K., O'Neal, J., et al. 2022, SoftwareX, 19, 101168, doi: 10.1016/j.softx.

2022.101168

Duffell, P. C. 2016, ApJ, 821, 76, doi: 10.3847/0004-637X/821/2/76

Dumbser, M., Zanotti, O., Loubère, R., & Diot, S. 2014, Journal of Computational Physics, 278, 47 , doi: https://doi.org/10.1016/j.jcp.2014.08.009

Dunham, S. J., Endeve, E., Mezzacappa, A., Buffaloe, J., & Holley-Bockelmann, K. 2020, Journal of Physics: Conference Series, 1623, 012012, doi: 10.1088/1742-6596/1623/1/012012

Ebinger, K., Curtis, S., Fröhlich, C., et al. 2019, ApJ, 870, 1, doi: 10.3847/1538-4357/aae7c9

Ebinger, K., Sinha, S., Fröhlich, C., et al. 2017, in 14th International Symposium on Nuclei in the Cosmos (NIC2016), ed. S. Kubono, T. Kajino, S. Nishimura, T. Isobe, S. Nagataki, T. Shima, & Y. Takeda, 020611, doi: 10.7566/JPSCP.14.020611

Edwards, H. C., Trott, C. R., & Sunderland, D. 2014, Journal of Parallel and Distributed Computing, 74, 3202 , doi: https://doi.org/10.1016/j.jpdc.2014.07.003

Efron, B. 1979, Ann. Statist., 7, 1, doi: 10.1214/aos/1176344552

Eldridge, J. J., & Xiao, L. 2019, MNRAS, 485, L58, doi: 10.1093/mnrasl/slz030

Endeve, E., Cardall, C. Y., Budiardja, R. D., et al. 2012, ApJ, 751, 26, doi: 10.1088/0004-637X/751/1/26

Endeve, E., Hauck, C. D., Xing, Y., & Mezzacappa, A. 2015, Journal of Computational Physics, 287, 151

Endeve, E., Buffaloe, J., Dunham, S. J., et al. 2019, Journal of Physics: Conference Series, 1225, 012014, doi: 10.1088/1742-6596/1225/1/012014

Ertl, T., Janka, H.-T., Woosley, S. E., Sukhbold, T., & Ugliano, M. 2016, ApJ, 818, 124, doi: 10.3847/0004-637X/818/2/124

Ertl, T., Woosley, S. E., Sukhbold, T., & Janka, H. T. 2020, ApJ, 890, 51, doi: 10.3847/1538-4357/ab6458

Falk, S. W., & Arnett, W. D. 1973, ApJ, 180, L65, doi: 10.1086/181154

—. 1977, ApJS, 33, 515, doi: 10.1086/190440

Fambri, F., Dumbser, M., Köppel, S., Rezzolla, L., & Zanotti, O. 2018, MNRAS, 477, 4543, doi: 10.1093/mnras/sty734

254

Faran, T., Poznanski, D., Filippenko, A. V., et al. 2014, MNRAS, 445, 554, doi: `10.1093/mnras/stu1760`

Farrell, E. J., Groh, J. H., Meynet, G., & Eldridge, J. J. 2020, MNRAS, 494, L53, doi: `10.1093/mnrasl/slaa035`

Ferguson, J. W., Alexander, D. R., Allard, F., et al. 2005, ApJ, 623, 585, doi: `10.1086/428642`

Fischer, T., Guo, G., Martínez-Pinedo, G., Liebendörder, M., & Mezzacappa, A. 2020, doi: `10.1103/PhysRevD.102.123001`

Fishbone, L. G., & Moncrief, V. 1976, ApJ, 207, 962, doi: `10.1086/154565`

Font, J. A., Miller, M., Suen, W.-M., & Tobias, M. 2000, Phys. Rev. D, 61, 044011, doi: `10.1103/PhysRevD.61.044011`

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, PASP, 125, 306, doi: `10.1086/670067`

Foucart, F., O'Connor, E., Roberts, L., et al. 2015, Phys. Rev. D, 91, 124021, doi: `10.1103/PhysRevD.91.124021`

Freedman, D. Z. 1974, Phys. Rev. D, 9, 1389, doi: `10.1103/PhysRevD.9.1389`

Fryxell, B., Olson, K., Ricker, P., et al. 2000, ApJS, 131, 273, doi: `10.1086/317361`

Fu, G., & Shu, C.-W. 2017, Journal of Computational Physics, 347, 305 , doi: `https://doi.org/10.1016/j.jcp.2017.06.046`

Gall, E. E. E., Polshaw, J., Kotak, R., et al. 2015, A&A, 582, A3, doi: `10.1051/0004-6361/201525868`

Gamblin, T., LeGendre, M., Collette, M. R., et al. 2015, in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '15 (New York, NY, USA: Association for Computing Machinery), doi: `10.1145/2807591.2807623`. `https://doi.org/10.1145/2807591.2807623`

Gammie, C. F., McKinney, J. C., & Tóth, G. 2003, ApJ, 589, 444, doi: `10.1086/374594`

Ghosh, S., Wolfe, N., & Fröhlich, C. 2022, ApJ, 929, 43, doi: `10.3847/1538-4357/ac4d20`

Giacomazzo, B., & Rezzolla, L. 2007, Classical and Quantum Gravity, 24, S235, doi: `10.1088/0264-9381/24/12/s16`

Gibbs, J. W. 1898, Nature, 59, 200, doi: `10.1038/059200b0`

Godunov, S. K. 1959, Mat. Sb., Nov. Ser., 47, 271

Goldberg, J. A., & Bildsten, L. 2020, ApJ, 895, L45, doi: 10.3847/2041-8213/ab9300

Goldberg, J. A., Bildsten, L., & Paxton, B. 2019, ApJ, 879, 3, doi: 10.3847/1538-4357/ab22b6

Goldreich, P., & Weber, S. V. 1980, ApJ, 238, 991, doi: 10.1086/158065

González-Gaitán, S., Tominaga, N., Molina, J., et al. 2015, MNRAS, 451, 2212, doi: 10.1093/mnras/stv1097

Gottlieb, E., Shu, C.-W., & Tadmor, E. 2001, SIAM Review, 43, 89

Greif, S. K., Hebeler, K., Lattimer, J. M., Pethick, C. J., & Schwenk, A. 2020, ApJ, 901, 155, doi: 10.3847/1538-4357/abaf55

Grete, P., Dolence, J. C., Miller, J. M., et al. 2022, arXiv e-prints, arXiv:2202.12309. https://arxiv.org/abs/2202.12309

Gutiérrez, C. P., Anderson, J. P., Hamuy, M., et al. 2017a, The Astrophysical Journal, 850, 89, doi: 10.3847/1538-4357/aa8f52

—. 2017b, The Astrophysical Journal, 850, 90, doi: 10.3847/1538-4357/aa8f42

Hamuy, M. 2005, in IAU Colloq. 192: Cosmic Explosions, On the 10th Anniversary of SN1993J, ed. J.-M. Marcaide & K. W. Weiler, Vol. 99, 535, doi: 10.1007/3-540-26633-X_71

Hanke, F., Müller, B., Wongwathanarat, A., Marek, A., & Janka, H.-T. 2013, ApJ, 770, 66, doi: 10.1088/0004-637X/770/1/66

Harten, A., Lax, P., & Leer, B. 1983a, SIAM Review, 25, 35, doi: 10.1137/1025002

Harten, A., Lax, P. D., & Leer, B. V. 1983b, SIAM Review, 25, 35

Heger, A., Fryer, C. L., Woosley, S. E., Langer, N., & Hartmann, D. H. 2003, ApJ, 591, 288, doi: 10.1086/375341

Heger, A., & Woosley, S. E. 2002, ApJ, 567, 532, doi: 10.1086/338487

Heger, A., Woosley, S. E., & Spruit, H. C. 2005, ApJ, 626, 350, doi: 10.1086/429868

Hempel, M., Fischer, T., Schaffner-Bielich, J., & Liebendörfer, M. 2012, ApJ, 748, 70, doi: 10.1088/0004-637X/748/1/70

Herant, M., Benz, W., & Colgate, S. 1992, ApJ, 395, 642, doi: 10.1086/171685

Herant, M., Colgate, S. A., Benz, W., & Fryer, C. 1997, Los Alamos Sci., 25, 164

Hesthaven, J. S., & Warburton, T. 2008, Nodal discontinuous Galerkin methods: Algorithms, analysis and applications (Springer)

Hix, W. R., Messer, O. E., Mezzacappa, A., et al. 2003, Phys. Rev. Lett., 91, 201102, doi: 10.1103/PhysRevLett.91.201102

Hix, W. R., Lentz, E. J., Endeve, E., et al. 2014, AIP Advances, 4, 041013, doi: 10.1063/1.4870009

Horowitz, C. J. 1997, Phys. Rev. D, 55, 4577, doi: 10.1103/PhysRevD.55.4577

Hummer, D. G., & Rybicki, G. B. 1968, ApJ, 153, L107, doi: 10.1086/180231

Iglesias, C. A., & Rogers, F. J. 1996, ApJ, 464, 943, doi: 10.1086/177381

Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, The Astrophysical Journal, 873, 111, doi: 10.3847/1538-4357/ab042c

Janka, H.-T., Hanke, F., Hüdepohl, L., et al. 2012, Progress of Theoretical and Experimental Physics, 2012, 01A309, doi: 10.1093/ptep/pts067

Janka, H.-T., Langanke, K., Marek, A., Martínez-Pinedo, G., & Müller, B. 2007, Phys. Rep., 442, 38, doi: 10.1016/j.physrep.2007.02.002

Janka, H.-T., Melson, T., & Summa, A. 2016, Annual Review of Nuclear and Particle Science, 66, 341, doi: 10.1146/annurev-nucl-102115-044747

Johnston, Z., Wasik, S., Titus, R., et al. 2022, ApJ, 939, 15, doi: 10.3847/1538-4357/ac9306

Just, O., Obergaulinger, M., & Janka, H. T. 2015, MNRAS, 453, 3386, doi: 10.1093/mnras/stv1892

Käppeli, R., & Mishra, S. 2016, A&A, 587, A94, doi: 10.1051/0004-6361/201527815

Kasen, D., Metzger, B., Barnes, J., Quataert, E., & Ramirez-Ruiz, E. 2017, Nature, 551, 80, doi: 10.1038/nature24453

Kasen, D., Thomas, R. C., & Nugent, P. 2006, ApJ, 651, 366, doi: 10.1086/506190

Kasen, D., & Woosley, S. E. 2009, ApJ, 703, 2205, doi: 10.1088/0004-637X/703/2/2205

Kastaun, W., Kalinani, J. V., & Ciolfi, R. 2021, Phys. Rev. D, 103, 023018, doi: 10.1103/PhysRevD.103.023018

Kharusi, S. A., BenZvi, S. Y., Bobowski, J. S., et al. 2021, New Journal of Physics, 23, 031201, doi: 10.1088/1367-2630/abde33

Kidder, L. E., Field, S. E., Foucart, F., et al. 2017, Journal of Computational Physics, 335, 84, doi: 10.1016/j.jcp.2016.12.059

Kilpatrick, C. D., & Foley, R. J. 2018, MNRAS, 481, 2536, doi: 10.1093/mnras/sty2435

Koplitz, B., Johnson, J., Williams, B. F., et al. 2021, ApJ, 916, 58, doi: 10.3847/1538-4357/abfb7b

Kotake, K. 2013, Comptes Rendus Physique, 14, 318, doi: 10.1016/j.crhy.2013.01.008

Kotake, K., Takiwaki, T., Fischer, T., Nakamura, K., & Martínez-Pinedo, G. 2018, ApJ, 853, 170, doi: 10.3847/1538-4357/aaa716

Kozyreva, A., Nakar, E., & Waldman, R. 2019, MNRAS, 483, 1211, doi: 10.1093/mnras/sty3185

Kozyreva, A., Nakar, E., Waldman, R., Blinnikov, S., & Baklanov, P. 2020, MNRAS, 494, 3927, doi: 10.1093/mnras/staa924

Krivodonova, L. 2007, Journal of Computational Physics, 226, 879, doi: 10.1016/j.jcp.2007.05.011

Kuroda, T. 2021, ApJ, 906, 128, doi: 10.3847/1538-4357/abce61

Kuroda, T., Takiwaki, T., & Kotake, K. 2016, ApJS, 222, 20, doi: 10.3847/0067-0049/222/2/20

Kuzmin, D. 2006, Journal of Computational Physics, 219, 513, doi: https://doi.org/10.1016/j.jcp.2006.03.034

Laiu, M. P., Harris, J. A., Chu, R., & Endeve, E. 2020, Journal of Physics: Conference Series, 1623, 012013, doi: 10.1088/1742-6596/1623/1/012013

Lalazissis, G. A., König, J., & Ring, P. 1997, Phys. Rev. C, 55, 540, doi: 10.1103/PhysRevC.55.540

Laplace, E., Justham, S., Renzo, M., et al. 2021, A&A, 656, A58, doi: 10.1051/0004-6361/202140506

Larsen, E. W., & Morel, J. E. 1989, Journal of Computational Physics, 83, 212

Larson, M. G., & Bengzon, F. 2013, The Finite Element Method: Theory, Implementation, and Applications (Springer Berlin Heidelberg)

Larsson, S., & Thomee, V. 2003, Partial Differential Equations with Numerical Methods, Texts in Applied Mathematics (Springer). https://books.google.com/books?id=mrmxylxQlPUC

Lattimer, J. M., & Douglas Swesty, F. 1991, Nuclear Physics A, 535, 331, doi: 10.1016/0375-9474(91)90452-C

Lattimer, J. M., Pethick, C. J., Ravenhall, D. G., & Lamb, D. Q. 1985, Nucl. Phys. A, 432, 646, doi: 10.1016/0375-9474(85)90006-5

Lax, P. D., & Liu, X.-D. 1998, SIAM Journal on Scientific Computing, 19, 319, doi: 10.1137/S1064827595291819

LeBlanc, J. M., & Wilson, J. R. 1970, ApJ, 161, 541, doi: 10.1086/150558

LeVeque, R. 1992, Numerical Methods for Conservation Laws, Lectures in Mathematics ETH Zürich, Department of Mathematics Research Institute of Mathematics (Springer). https://books.google.com/books?id=3WhqLPcMdPsC

—. 2002, Finite Volume Methods for Hyperbolic Problems, Cambridge Texts in Applied Mathematics (Cambridge University Press). https://books.google.com/books?id=O_ZjpMSZiwoC

—. 2007, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems, Other Titles in Applied Mathematics (Society for Industrial and Applied Mathematics). https://books.google.com/books?id=qsvmsXe8Ug4C

Li, G., & Xing, Y. 2018, Journal of Computational Physics, 352, 445, doi: 10.1016/j.jcp.2017.09.063

Li, W., Leaman, J., Chornock, R., et al. 2011, MNRAS, 412, 1441, doi: 10.1111/j.1365-2966.2011.18160.x

Lisakov, S. M., Dessart, L., Hillier, D. J., Waldman, R., & Livne, E. 2017, MNRAS, 466, 34, doi: 10.1093/mnras/stw3035

Lisakov, S. M., Dessart, L., Hillier, D. J., Waldman, R., & Livne, E. 2018, MNRAS, 473, 3863, doi: 10.1093/mnras/stx2521

Litvinova, I. Y., & Nadezhin, D. K. 1985, Soviet Astronomy Letters, 11, 145

Liu, X. D., & Osher, S. 1996, SIAM J. Numer. Anal., 33, 760

Lovegrove, E., & Woosley, S. E. 2013, ApJ, 769, 109, doi: 10.1088/0004-637X/769/2/109

Lyon, S. P., & Johnson, J. D. 1992, Sesame: The Los Alamos National Laboratory Equation of State Database, Tech. Rep. LA-UR-92-3407, Los Alamos National Laboratory

Mabanta, Q. A., & Murphy, J. W. 2018, ApJ, 856, 22, doi: 10.3847/1538-4357/aaaec7

Martinez, L., & Bersten, M. C. 2019, A&A, 629, A124, doi: 10.1051/0004-6361/201834818

Martinez, L., Bersten, M. C., Anderson, J. P., et al. 2020, A&A, 642, A143, doi: 10.1051/0004-6361/202038393

Martinez, L., Bersten, M. C., Anderson, J. P., et al. 2022, A&A, 660, A41, doi: 10.1051/0004-6361/202142076

Martínez-Pinedo, G., Fischer, T., & Huther, L. 2014, Journal of Physics G: Nuclear and Particle Physics, 41, 044008, doi: 10.1088/0954-3899/41/4/044008

Mattila, S., Smartt, S. J., Eldridge, J. J., et al. 2008, ApJ, 688, L91, doi: 10.1086/595587

McCorquodale, P., & Colella, P. 2011, Communications in Applied Mathematics and Computational Science, 6, 1

Melson, T., Kresse, D., & Janka, H.-T. 2020, ApJ, 891, 27, doi: 10.3847/1538-4357/ab72a7

Meskhi, M. M., Wolfe, N. E., Dai, Z., et al. 2021, arXiv e-prints, arXiv:2111.01815. https://arxiv.org/abs/2111.01815

Mezzacappa, A. 2001, Nucl. Phys. A, 688, 158, doi: 10.1016/S0375-9474(01)00690-X

—. 2005, Annual Review of Nuclear and Particle Science, 55, 467, doi: 10.1146/annurev.nucl.55.090704.151608

Mezzacappa, A. 2022, arXiv e-prints, arXiv:2205.13438. https://arxiv.org/abs/2205.13438

Mezzacappa, A. 2023, in The Predictive Power of Computational Astrophysics as a Discover Tool, ed. D. Bisikalo, D. Wiebe, & C. Boily, Vol. 362, 215–227, doi: 10.1017/S1743921322001831. https://ui.adsabs.harvard.edu/abs/2023IAUS..362..215M

Mezzacappa, A., & Bruenn, S. W. 1993, ApJ, 405, 669, doi: 10.1086/172395

Mezzacappa, A., Endeve, E., Messer, O. E. B., & Bruenn, S. W. 2020, Living Reviews in Computational Astrophysics, 6, 4, doi: 10.1007/s41115-020-00010-8

Mezzacappa, A., & Messer, O. 1999, Journal of Computational and Applied Mathematics, 109, 281 , doi: https://doi.org/10.1016/S0377-0427(99)00162-4

Mezzacappa, A., & Zanolin, M. 2024, arXiv e-prints, arXiv:2401.11635, doi: 10.48550/arXiv.2401.11635

Mihalas, D., Auer, L. H., & Mihalas, B. R. 1978, ApJ, 220, 1001, doi: 10.1086/155988

Mihalas, D., & Mihalas, B. W. 1984, Foundations of radiation hydrodynamics

Miller, J. M., Dolence, J. C., & Holladay, D. 2022, arXiv e-prints, arXiv:2206.08957, doi: 10.48550/arXiv.2206.08957

Miller, J. M., Holladay, D., Meyer, C. D., et al. 2022, Journal of Open Source Software, 7, 4367, doi: 10.21105/joss.04367

Miller, J. M., Ryan, B. R., & Dolence, J. C. 2019, ApJS, 241, 30, doi: 10.3847/1538-4365/ab09fc

Miller, J. M., & Schnetter, E. 2017, Classical and Quantum Gravity, 34, 015003, doi: 10.1088/1361-6382/34/1/015003

Miller, J. M., Sprouse, T. M., Fryer, C. L., et al. 2020, ApJ, 902, 66, doi: 10.3847/1538-4357/abb4e3

Mönchmeyer, R., & Müller, E. 1989, Astronomy & Astrophysics, 217, 351

Morozova, V., Piro, A. L., Renzo, M., & Ott, C. D. 2016, ApJ, 829, 109, doi: 10.3847/0004-637X/829/2/109

Morozova, V., Piro, A. L., Renzo, M., et al. 2015, ApJ, 814, 63, doi: 10.1088/0004-637X/814/1/63

Morozova, V., Piro, A. L., & Valenti, S. 2018, ApJ, 858, 15, doi: 10.3847/1538-4357/aab9a6

Müller, B. 2016, **??**jnlPASA, 33, e048, doi: 10.1017/pasa.2016.40

Müller, B. 2019, MNRAS, 487, 5304, doi: 10.1093/mnras/stz1594

Müller, B. 2020, Living Reviews in Computational Astrophysics, 6, 3, doi: 10.1007/s41115-020-0008-5

Müller, B., Heger, A., Liptai, D., & Cameron, J. B. 2016, MNRAS, 460, 742, doi: 10.1093/mnras/stw1083

Müller, B., Janka, H.-T., & Dimmelmeier, H. 2010, ApJS, 189, 104, doi: 10.1088/0067-0049/189/1/104

Müller, B., Janka, H.-T., & Marek, A. 2012, ApJ, 756, 84, doi: 10.1088/0004-637X/756/1/84

Müller, B., & Varma, V. 2020, arXiv e-prints, arXiv:2007.04775. https://arxiv.org/abs/

2007.04775

Murchikova, E. M., Abdikamalov, E., & Urbatsch, T. 2017, MNRAS, 469, 1725, doi: 10.1093/mnras/stx986

Murphy, J. W., Dolence, J. C., & Burrows, A. 2013, ApJ, 771, 52, doi: 10.1088/0004-637X/771/1/52

Murphy, J. W., Mabanta, Q., & Dolence, J. C. 2019, MNRAS, 489, 641, doi: 10.1093/mnras/stz2123

Murphy, J. W., & Meakin, C. 2011, ApJ, 742, 74, doi: 10.1088/0004-637X/742/2/74

Nagakura, H., Sumiyoshi, K., & Yamada, S. 2014, ApJS, 214, 16, doi: 10.1088/0067-0049/214/2/16

Nakar, E., & Sari, R. 2010, ApJ, 725, 904, doi: 10.1088/0004-637X/725/1/904

Neustadt, J. M. M., Kochanek, C. S., Stanek, K. Z., et al. 2021, MNRAS, 508, 516, doi: 10.1093/mnras/stab2605

Nugent, P., Sullivan, M., Ellis, R., et al. 2006, ApJ, 645, 841, doi: 10.1086/504413

NVIDIA, Vingelmann, P., & Fitzek, F. H. 2020, CUDA, release: 10.2.89. https://developer.nvidia.com/cuda-toolkit

O'Connor, E. 2015, ApJS, 219, 24, doi: 10.1088/0067-0049/219/2/24

O'Connor, E., & Ott, C. D. 2010a, Stellar Collapse: Microphysics. https://stellarcollapse.org/equationofstate

—. 2010b, Classical and Quantum Gravity, 27, 114103

O'Connor, E., & Ott, C. D. 2011, ApJ, 730, 70, doi: 10.1088/0004-637X/730/2/70

O'Connor, E. P., & Couch, S. M. 2018, ApJ, 854, 63, doi: 10.3847/1538-4357/aaa893

Oertel, M., Hempel, M., Klähn, T., & Typel, S. 2017, Rev. Mod. Phys., 89, 015007, doi: 10.1103/RevModPhys.89.015007

Olbrant, E., Hauck, C. D., & Frank, M. 2012, Journal of Computational Physics, 231, 5612

Omang, M., Børve, S., & Trulsen, J. 2006, Journal of Computational Physics, 213, 391, doi: 10.1016/j.jcp.2005.08.023

O'Neill, D., Kotak, R., Fraser, M., et al. 2021, A&A, 645, L7, doi: 10.1051/0004-6361/202039546

Ott, C. D., Schnetter, E., Burrows, A., et al. 2009, in Journal of Physics Conference Series, Vol. 180, Journal of Physics Conference Series, 012022, doi: 10.1088/1742-6596/180/1/012022

Paczynski, B. 1983, ApJ, 267, 315, doi: 10.1086/160870

Pajkos, M. A., Couch, S. M., Pan, K.-C., & O'Connor, E. P. 2019, ApJ, 878, 13, doi: 10.3847/1538-4357/ab1de2

Pajkos, M. A., Warren, M. L., Couch, S. M., O'Connor, E. P., & Pan, K.-C. 2021, ApJ, 914, 80, doi: 10.3847/1538-4357/abfb65

Paxton, B., Bildsten, L., Dotter, A., et al. 2011, ApJS, 192, 3, doi: 10.1088/0067-0049/192/1/3

Paxton, B., Cantiello, M., Arras, P., et al. 2013, ApJS, 208, 4, doi: 10.1088/0067-0049/208/1/4

Paxton, B., Marchant, P., Schwab, J., et al. 2015, ApJS, 220, 15, doi: 10.1088/0067-0049/220/1/15

Paxton, B., Schwab, J., Bauer, E. B., et al. 2018, ApJS, 234, 34, doi: 10.3847/1538-4365/aaa5a8

Paxton, B., Smolec, R., Schwab, J., et al. 2019, ApJS, 243, 10, doi: 10.3847/1538-4365/ab2241

Pejcha, O. 2020, The Explosion Mechanism of Core-Collapse Supernovae and Its Observational Signatures, 189–211

Pejcha, O., & Prieto, J. L. 2015, ApJ, 806, 225, doi: 10.1088/0004-637X/806/2/225

Pejcha, O., & Thompson, T. A. 2015, ApJ, 801, 90, doi: 10.1088/0004-637X/801/2/90

Perego, A., Hempel, M., Fröhlich, C., et al. 2015, ApJ, 806, 275, doi: 10.1088/0004-637X/806/2/275

Pimentel, D. A. 2021, EOSPAC User's Manual: V.6.5 (United States)

Popov, D. V. 1993, ApJ, 414, 712, doi: 10.1086/173117

Poznanski, D., Butler, N., Filippenko, A. V., et al. 2009, ApJ, 694, 1067, doi: 10.1088/0004-637X/694/2/1067

Pumo, M. L., Zampieri, L., Spiro, S., et al. 2017, MNRAS, 464, 3013, doi: 10.1093/mnras/stw2625

Qiu, J., & Shu, C.-W. 2005, SIAM J. Sci. Comput., 27, 995

Quirk, J. J. 1994, International Journal for Numerical Methods in Fluids, 18, 555, doi: 10.1002/fld.1650180603

Rabinak, I., & Waxman, E. 2011, ApJ, 728, 63, doi: 10.1088/0004-637X/728/1/63

Radice, D., Abdikamalov, E., Ott, C. D., et al. 2018, Journal of Physics G Nuclear Physics, 45, 053003, doi: 10.1088/1361-6471/aab872

Radice, D., Couch, S. M., & Ott, C. D. 2015, Computational Astrophysics and Cosmology, 2, 7, doi: 10.1186/s40668-015-0011-0

Radice, D., Ott, C. D., Abdikamalov, E., et al. 2016, ApJ, 820, 76, doi: 10.3847/0004-637X/820/1/76

Radice, D., & Rezzolla, L. 2011, Phys. Rev. D, 84, 024010, doi: 10.1103/PhysRevD.84.024010

Rampp, M., & Janka, H. T. 2002, A&A, 396, 361, doi: 10.1051/0004-6361:20021398

Reddy, S., Prakash, M., & Lattimer, J. M. 1998, Phys. Rev. D, 58, 013009, doi: 10.1103/PhysRevD.58.013009

Reed, W., & Hill, T. 1973, Proceedings of the Americal Nuclear Society

Remacle, J.-F., Flaherty, J. E., & Shephard, M. S. 2003, SIAM Review, 45, 53, doi: 10.1137/S00361445023830

Reyes, R., Brown, G., Burns, R., & Wong, M. 2020, in Proceedings of the International Workshop on OpenCL, IWOCL '20 (New York, NY, USA: Association for Computing Machinery), doi: 10.1145/3388333.3388649. https://doi.org/10.1145/3388333.3388649

Rezzolla, L., & Zanotti, O. 2013, Relativistic Hydrodynamics (Oxford University Press)

Richers, S., Nagakura, H., Ott, C. D., et al. 2017, ApJ, 847, 133, doi: 10.3847/1538-4357/aa8bb2

Ricks, W., & Dwarkadas, V. V. 2019, ApJ, 880, 59, doi: 10.3847/1538-4357/ab287c

Rivière, B. 2008, Discontinuous Galerkin methods for solving elliptic and parabolic equations theory and implementation, Frontiers in applied mathematics ; 35 (Philadelphia, Pa: Society for Industrial and Applied Mathematics SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104)

Roberts, L. F., Ott, C. D., Haas, R., et al. 2016, ApJ, 831, 98, doi: 10.3847/0004-637X/831/1/98

Roe, P. L. 1986, Annual Review of Fluid Mechanics, 18, 337, doi: 10.1146/annurev.fl.18.010186.002005

Rubin, A., & Gal-Yam, A. 2017, ApJ, 848, 8, doi: 10.3847/1538-4357/aa8465

Ryan, B. R., Dolence, J. C., & Gammie, C. F. 2015, ApJ, 807, 31, doi: 10.1088/0004-637X/807/1/31

Salpeter, E. E. 1955, ApJ, 121, 161, doi: 10.1086/145971

Sanders, N. E., Soderberg, A. M., Gezari, S., et al. 2015, ApJ, 799, 208, doi: 10.1088/0004-637X/799/2/208

Sandoval, M. A., Hix, W. R., Messer, O. E. B., Lentz, E. J., & Harris, J. A. 2021, ApJ, 921, 113, doi: 10.3847/1538-4357/ac1d49

Sapir, N., & Waxman, E. 2017, ApJ, 838, 130, doi: 10.3847/1538-4357/aa64df

Schaal, K., Bauer, A., Chandrashekar, P., et al. 2015, MNRAS, 453, 4278, doi: 10.1093/mnras/stv1859

Schneider, A. S., Roberts, L. F., Ott, C. D., & O'Connor, E. 2019, Phys. Rev. C, 100, 055802, doi: 10.1103/PhysRevC.100.055802

Scholberg, K. 2012, Annual Review of Nuclear and Particle Science, 62, 120726135758004, doi: 10.1146/annurev-nucl-102711-095006

Sedov, L. I. 1946, Journal of Applied Mathematics and Mechanics, 10, 241

Shen, G., Horowitz, C. J., & O'Connor, E. 2011a, Phys. Rev. C, 83, 065808, doi: 10.1103/PhysRevC.83.065808

Shen, G., Horowitz, C. J., & Teige, S. 2011b, Phys. Rev. C, 83, 035802, doi: 10.1103/PhysRevC.83.035802

Shen, H., Toki, H., Oyamatsu, K., & Sumiyoshi, K. 1998, Progress of Theoretical Physics, 100, 1013, doi: 10.1143/PTP.100.1013

Shibata, M., Kiuchi, K., Sekiguchi, Y., & Suwa, Y. 2011, Progress of Theoretical Physics, 125, 1255, doi: 10.1143/PTP.125.1255

Shu, C.-W. 2009, SIAM Review, 51, 82, doi: 10.1137/070679065

Shu, C.-W. 2016, Journal of Computational Physics, 316, 598

Shu, C.-W., & Osher, S. 1988, Journal of Computational Physics, 77, 439 , doi: https://doi.org/10.1016/0021-9991(88)90177-5

Shussman, T., Waldman, R., & Nakar, E. 2016. https://arxiv.org/abs/1610.05323

Skinner, M. A., Dolence, J. C., Burrows, A., Radice, D., & Vartanyan, D. 2018, ArXiv e-prints. https://arxiv.org/abs/1806.07390

—. 2019, ApJS, 241, 7, doi: 10.3847/1538-4365/ab007f

Smartt, S. J. 2009, ARA&A, 47, 63, doi: 10.1146/annurev-astro-082708-101737

—. 2015, ??jnlPASA, 32, e016, doi: 10.1017/pasa.2015.17

Smit, J. M., van den Horn, L. J., & Bludman, S. A. 2000, A&A, 356, 559

Sod, G. A. 1978, Journal of Computational Physics, 27, 1 , doi: https://doi.org/10.1016/0021-9991(78)90023-2

Soderberg, A. M., Chakraborti, S., Pignata, G., et al. 2010, Nature, 463, 513, doi: 10.1038/nature08714

Sotani, H., & Takiwaki, T. 2020, Phys. Rev. D, 102, 023028, doi: 10.1103/PhysRevD.102.023028

Soumagnac, M. T., Ganot, N., Irani, I., et al. 2020, ApJ, 902, 6, doi: 10.3847/1538-4357/abb247

Steiner, A. W., Hempel, M., & Fischer, T. 2013a, ApJ, 774, 17, doi: 10.1088/0004-637X/774/1/17

Steiner, A. W., Lattimer, J. M., & Brown, E. F. 2010, ApJ, 722, 33, doi: 10.1088/0004-637X/722/1/33

—. 2013b, ApJ, 765, L5, doi: 10.1088/2041-8205/765/1/L5

Stockinger, G., Janka, H. T., Kresse, D., et al. 2020, MNRAS, 496, 2039, doi: 10.1093/mnras/staa1691

Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., & Simon, J. B. 2008, ApJS, 178, 137, doi: 10.1086/588755

Stone, J. M., & Norman, M. L. 1992, ApJS, 80, 753, doi: 10.1086/191680

Stone, J. M., Tomida, K., White, C. J., & Felker, K. G. 2020, ApJS, 249, 4, doi: 10.3847/1538-4365/ab929b

Sugahara, Y., & Toki, H. 1994, Nucl. Phys. A, 579, 557, doi: 10.1016/0375-9474(94)90923-7

Sukhbold, T., Ertl, T., Woosley, S. E., Brown, J. M., & Janka, H.-T. 2016, ApJ, 821, 38, doi: 10.3847/0004-637X/821/1/38

Sukhbold, T., Woosley, S. E., & Heger, A. 2018, ApJ, 860, 93, doi: 10.3847/1538-4357/aac2da

Sumiyoshi, K., & Yamada, S. 2012, ApJS, 199, 17, doi: 10.1088/0067-0049/199/1/17

Summa, A., Hanke, F., Janka, H.-T., et al. 2016, ApJ, 825, 6, doi: 10.3847/0004-637X/825/1/6

Suresh, A., & Huynh, H. 1997, Journal of Computational Physics, 136, 83, doi: https://doi.org/10.1006/jcph.1997.5745

Suresh, A., & Huynh, H. T. 1997, Journal of Computational Physics, 136, 83, doi: 10.1006/jcph.1997.5745

Swartz, D. A., Sutherland, P. G., & Harkness, R. P. 1995, ApJ, 446, 766, doi: 10.1086/175834

Swesty, F. D. 1996, Journal of Computational Physics, 127, 118, doi: 10.1006/jcph.1996.0162

Szczepańczyk, M. J., Antelis, J. M., Benjamin, M., et al. 2021, Phys. Rev. D, 104, 102002, doi: 10.1103/PhysRevD.104.102002

Tamborra, I., Hanke, F., Janka, H.-T., et al. 2014, ApJ, 792, 96, doi: 10.1088/0004-637X/792/2/96

Taylor, G. 1950, Proceedings of the Royal Society of London Series A, 201, 159, doi: 10.1098/rspa.1950.0049

Teukolsky, S. A. 2016, Journal of Computational Physics, 312, 333, doi: 10.1016/j.jcp.2016.02.031

Thorne, K. S. 1981, MNRAS, 194, 439, doi: 10.1093/mnras/194.2.439

Timmes, F. X., & Swesty, F. D. 2000, ApJS, 126, 501, doi: 10.1086/313304

Todd-Rutel, B. G., & Piekarewicz, J. 2005, Phys. Rev. Lett., 95, 122501, doi: 10.1103/PhysRevLett.95.122501

Tolstov, A. G., Blinnikov, S. I., & Nadyozhin, D. K. 2013, MNRAS, 429, 3181, doi: 10.1093/mnras/sts577

Toro, E. 2009a, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction (Springer Berlin Heidelberg). https://books.google.com/books?id=SqEjX0um8o0C

—. 2009b, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction

Toro, E. F., Spruce, M., & Speares, W. 1994, Shock Waves, 4, 25, doi: 10.1007/BF01414629

Tóth, G. 2000, Journal of Computational Physics, 161, 605, doi: 10.1006/jcph.2000.6519

Trott, C., Berger-Vergiat, L., Poliakoff, D., et al. 2021, Computing in Science Engineering, 23, 10, doi: 10.1109/MCSE.2021.3098509

Trott, C. R., Lebrun-Grandié, D., Arndt, D., et al. 2022, IEEE Transactions on Parallel and Distributed Systems, 33, 805, doi: 10.1109/TPDS.2021.3097283

Trundle, C., Pastorello, A., Benetti, S., et al. 2009, A&A, 504, 945, doi: 10.1051/0004-6361/200911776

Ugliano, M., Janka, H.-T., Marek, A., & Arcones, A. 2012, ApJ, 757, 69, doi: 10.1088/0004-637X/757/1/69

Utrobin, V. P. 2007, in American Institute of Physics Conference Series, Vol. 937, Supernova 1987A: 20 Years After: Supernovae and Gamma-Ray Bursters, ed. S. Immler, K. Weiler, & R. McCray, 25–32, doi: 10.1063/1.3682879

Utrobin, V. P., & Chugai, N. N. 2008, A&A, 491, 507, doi: 10.1051/0004-6361:200810272

—. 2009, A&A, 506, 829, doi: 10.1051/0004-6361/200912273

Utrobin, V. P., Wongwathanarat, A., Janka, H. T., & Müller, E. 2015, A&A, 581, A40, doi: 10.1051/0004-6361/201425513

—. 2017, ApJ, 846, 37, doi: 10.3847/1538-4357/aa8594

Utrobin, V. P., Chugai, N. N., Andrews, J. E., et al. 2021, MNRAS, 505, 116, doi: 10.1093/mnras/stab1369

Valenti, S., Howell, D. A., Stritzinger, M. D., et al. 2016, MNRAS, 459, 3939, doi: 10.1093/mnras/stw870

Vallely, P. J., Kochanek, C. S., Stanek, K. Z., Fausnaugh, M., & Shappee, B. J. 2021, MNRAS, 500, 5639, doi: 10.1093/mnras/staa3675

Van Dyk, S. D., Li, W., & Filippenko, A. V. 2003, PASP, 115, 1289, doi: 10.1086/378308

Van Dyk, S. D., Davidge, T. J., Elias-Rosa, N., et al. 2012, AJ, 143, 19, doi: 10.1088/0004-6256/143/1/19

Van Dyk, S. D., Zheng, W., Maund, J. R., et al. 2019, The Astrophysical Journal, 875, 136, doi: 10.3847/1538-4357/ab1136

Van Leer, B. 1977, Journal of Computational Physics, 23, 263, doi: https://doi.org/10.1016/0021-9991(77)90094-8

Vartanyan, D., Burrows, A., & Radice, D. 2019, MNRAS, 489, 2227, doi: 10.1093/mnras/stz2307

Vincent, T., Pfeiffer, H. P., & Fischer, N. L. 2019, Physical review. D, 100

Warren, M. L., Couch, S. M., O'Connor, E. P., & Morozova, V. 2020, ApJ, 898, 139, doi: 10.3847/1538-4357/ab97b7

Weaver, T. A., Zimmerman, G. B., & Woosley, S. E. 1978, ApJ, 225, 1021, doi: 10.1086/156569

Weisz, D. R., Johnson, L. C., Foreman-Mackey, D., et al. 2015, ApJ, 806, 198, doi: 10.1088/0004-637X/806/2/198

Wilbraham, H. 1848, The Cambridge and Dublin Mathematical Journal, 3

Williams, B. F., Hillis, T. J., Blair, W. P., et al. 2019, ApJ, 881, 54, doi: 10.3847/1538-4357/ab2190

Wilson, J. R., Mathews, G. J., & Marronetti, P. 1996, Phys. Rev. D, 54, 1317, doi: 10.1103/PhysRevD.54.1317

Wongwathanarat, A., Janka, H.-T., & Müller, E. 2013, A&A, 552, A126, doi: 10.1051/0004-6361/201220636

Wongwathanarat, A., Müller, E., & Janka, H.-T. 2015, A&A, 577, A48, doi: 10.1051/0004-6361/201425025

Woosley, S. E., & Heger, A. 2007, Phys. Rep., 442, 269, doi: 10.1016/j.physrep.2007.02.009

Wu, K., & Tang, H. 2015, Journal of Computational Physics, 298, 539

Xing, Y., Zhang, X., & Shu, C.-W. 2010, Advances in Water Resources, 33, 1476

Yahil, A. 1983, ApJ, 265, 1047, doi: 10.1086/160746

Yasin, H., Schäfer, S., Arcones, A., & Schwenk, A. 2020, Phys. Rev. Lett., 124, 092701, doi: 10.1103/PhysRevLett.124.092701

Zhang, W., Almgren, A., Beckner, V., et al. 2019, Journal of Open Source Software, 4, 1370,

doi: 10.21105/joss.01370

Zhang, X., & Shu, C.-W. 2010, Journal of Computational Physics, 229, 8918 , doi: https://doi.org/10.1016/j.jcp.2010.08.016

Zhang, X., & Shu, C.-W. 2010, Journal of Computational Physics, 229, 3091

—. 2011, Proc. R. Soc. A, 467, 2752

Zhu, J., Qiu, J., & Shu, C.-W. 2020, Journal of Computational Physics, 404, 109105, doi: 10.1016/j.jcp.2019.109105

Zingale, M., & Katz, M. P. 2015, ApJS, 216, 31, doi: 10.1088/0067-0049/216/2/31