Dissertation
submitted to the
Combined Faculties for the Natural Sciences and for Mathematics
of the Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Put forward by
Aron Michel
born in: Schweinfurt
Oral examination: 09.01.2019

Modeling of Silicon Burning during late stages of Stellar Evolution

Referees: Prof. Dr. Friedrich Röpke
Prof. Dr. Cornelis Dullemond

Zusammenfassung

Aktuelle Studien mit parametrisierten Anfangsbedingungen haben gezeigt, dass Störungen im Strömungsmuster des Vorgängersterns eine wesentliche Zutat sein könnten, um zuverlässigere Explosionen in dreidimensionalen (3D) Simulationen von Kernkollaps-Supernovae zu ermöglichen. Diese Störungen entstehen auf natürliche Weise während den hydrostatischen Silizium und Sauerstoff Brennphasen in einem Stern. Der Einfluss von Störungen im Vorgängerstern auf die Dynamik der Kernkollaps-Supernova hängt ausschlaggebend von deren Stärke und dem räumlichen Ausmaß ab. Genaue Modellierung des Vorgängersterns ist daher notwendig, um ein realistische Anfangsmodell zu erhalten.

In dieser Arbeit präsentieren wir die ersten 3D Simulationen von Siliziumbrennen im vollen 4π Raumwinkel. Im Gegensatz zu bisherigen Arbeiten, verwenden wir die tatsächlichen Raten für Elektroneneinfänge auf ^{56}Fe. Mit unseren 3D Simulationen haben wir es geschafft, 154.41 s mit 3.9×10^6 CPU Stunden zu simulieren, was immer noch recht rechenaufwändig ist. Zukünftige Verbesserungen, wodurch die benötigten Computer-Ressourcen reduziert werden können, werden in dieser Arbeit diskutiert. Für die 3D Simulationen in dieser Arbeit haben wir einen standardmäßigen Poisson Gravitations-Löser auf einem kartesischen Gitter verwendet, welcher während dieser Arbeit entwickelt und implementiert wurde. Außerdem wurden unterschiedliche Methoden zur Lösung der Poisson-Gleichung implementiert, die auch auf krummlinigen Gittern funktionieren und auf eine große Anzahl Prozessoren skalieren.

Abstract

Recent studies with parametrized initial conditions have shown that perturbations in the flow pattern of the progenitor might be a crucial ingredient to enable more robust explosions in three-dimensional (3D) simulations of core-collapse supernovae. Those perturbations naturally develop during the hydrostatic silicon and oxygen burning phases in a star. However, the impact of perturbations in the progenitor on the dynamics of the core-collapse supernova largely depends on their magnitude and spatial scale. Accurate modeling of the progenitor is therefore necessary to obtain a realistic initial model.

In this work we present the first 3D simulations of silicon burning in full 4π solid angle. In contrast to previous work, we use the native electron capture rates on ^{56}Fe. With our 3D simulation we managed to cover 154.41 s with 3.9×10^6 core-hours, which is still rather CPU-intensive. Future improvements, which would reduce the required computational resources, are discussed in the thesis. For the 3D simulations in this thesis, we use a standard Poisson gravity solver on a Cartesian grid, which was developed and implemented during the course of the thesis. Additionally, we implemented different methods for solving Poisson's equation, which also work for curvilinear grids and scale to a large number of processors.

Contents

1 Introduction

Even though mankind was fascinated by the stars in the night sky since ancient times a thorough understanding of the processes in stars has only developed during the 20th century. In particular the work by S. Chandrasekhar, W. A. Fowler and M. Schwarzschild provided crucial insights into the structure and evolution of stars, as we know it today. However, even today there are many open questions regarding the details of nuclear processes in stars and their implications for stellar evolution.

In recent times it is possible to follow the complete evolution of a star on a computer from the formation of a star in molecular clouds through different nuclear burning processes until the formation of a white dwarf, neutron star or black hole. However, this is generally only possible under the assumption of spherical symmetry and requires several other approximations, which are discussed in Sec. 1.1.1. Another obstacle to overcome in the computational modeling of stars are the vastly different timescales on which astrophysical processes take place. The following timescale estimates are all based on Kippenhahn et al. (2012):

The *free-fall timescale* τ_{ff} is defined as the timescale on which a star would collapse due to its own gravity, if the pressure term in the hydrostatic equilibrium would suddenly "disappear". This timescale can be approximated via

$$\tau_{\mathrm{ff}} \approx \sqrt{\frac{R}{g}} = \sqrt{\frac{R^3}{GM}}, \tag{1.1}$$

where R is the radius and M is the total mass of the star and g is the gravitational acceleration given by Newton's law of gravitation $g = GM/R^2$ (see Sec. 2.3). In the case of the Sun this yields a value of $\tau_{\mathrm{ff}} \approx 27\,\mathrm{min}$ and even in the case of a red giant it is only roughly $\tau_{\mathrm{ff}} \approx 18\,\mathrm{d}$. In a similar fashion we can approximate the timescale on which a star would explode, if the gravity would suddenly disappear as

$$\tau_{\mathrm{expl}} \approx R\sqrt{\frac{\rho}{P}}, \tag{1.2}$$

where ρ and P are average values for the density and pressure over the entire star. If the star is near hydrostatic equilibrium those two timescales are about equal. In this case, we call this timescale the *hydrostatic timescale*

$$\tau_{\mathrm{hydr}} \approx \tau_{\mathrm{ff}} \approx \tau_{\mathrm{expl}}, \tag{1.3}$$

since it represents the timescale on which a star reacts to slight perturbations in the hydrostatic equilibrium.

The *Kelvin-Helmholtz timescale*

$$\tau_{\mathrm{KH}} = \frac{|E_{\mathrm{grav}}|}{L} \approx \frac{GM^2}{2RL} \tag{1.4}$$

is defined as the timescale for which a star could radiate with luminosity L using only its gravitational energy E_{grav}. In the case of the Sun the Kevin-Helmholtz timescale is $\tau_{\mathrm{KH}} = 1.6 \times 10^7\,\mathrm{yr}$. However, the decay of radioactive isotopes found in meteorites indicates that the solar system is at least $4.56 \times 10^9\,\mathrm{yr}$ old (e.g. Bouvier and Wadhwa, 2010). Clearly, this means that the Sun is not primarily powered by gravitational contraction. However, there are evolutionary phases in a star where E_{grav} is indeed the main energy source of a star.

Finally the *nuclear timescale*

$$\tau_{\mathrm{nuc}} = \frac{E_{\mathrm{nuc}}}{L} \tag{1.5}$$

of a star can be estimated by the energy release E_{nuc} through nuclear reactions. During the main sequence phase the most important nuclear reaction is the conversion of four ^{1}H into ^{4}He (see Sec. 3.1.1). The energy release of this reaction is $Q = 6.3 \times 10^{18}\,\mathrm{erg\,g^{-1}}$. Under the assumption that the Sun consists completely of hydrogen this yields a nuclear timescale of $\tau_{\mathrm{nuc}} = \frac{QM_\odot}{L_\odot} = 10^{11}\,\mathrm{yr}$. Comparing those three timescales clearly shows that

$$\tau_{\mathrm{nuc}} \gg \tau_{\mathrm{KH}} \gg \tau_{\mathrm{hydr}}. \tag{1.6}$$

Unfortunately this means that with the current computational resources it is clearly not possible to follow the whole lifetime of a star and resolving the hydrostatic timescale at the same time. Therefore, phenomena that occur on the hydrostatic timescale are typically neglected and treated very approximately in one-dimensional (1D) stellar evolution codes, whereas multidimensional hydrodynamic codes only cover short phases in the overall lifetime of a star.

However, during the very late burning stages of stellar evolution for massive stars the nuclear timescale and the hydrodynamic timescale are much closer. In the following, we give a rough estimate of the nuclear timescale during silicon burning. For this purpose we assume that silicon burning only consists of the simple fusion of two ^{28}Si nuclei into ^{56}Ni:

$$^{28}\mathrm{Si} + {}^{28}\mathrm{Si} \longrightarrow {}^{56}\mathrm{Ni}. \tag{1.7}$$

This is obviously a crude oversimplification of the nuclear reactions that are involved in silicon burning (see Sec. 3.1.4), but it should be sufficient for an order of magnitude estimate of the released nuclear energy. In particular, the reaction above completely neglects electron captures, which are extremely important for the neutronization during silicon burning and the resulting contraction of core (see Sec. 2.2.2). Under these assumptions the energy release for reaction (1.7) is $Q_{\mathrm{Si-burn}} = 1.88 \times 10^{17}\,\mathrm{erg\,g^{-1}}$. Furthermore, we assume that a total ^{28}Si mass of $M_{\mathrm{Si-burn}} = 1.4M_\odot$ is burned to ^{56}Ni, since the iron core becomes unstable once it reaches the Chandrasekhar mass limit $M_{\mathrm{ch}} \approx 1.4M_\odot$ (see Sec. 3.2). Finally, we take the peak luminosity of the convective silicon burning region during silicon burning from the 1D stellar evolution code *MESA*. This model from the stellar evolution code is also used as starting point for our simulations of silicon burning in Sec. 5.3 and 5.4. This yields a peak luminosity of $L_{\mathrm{Si-burn}} \approx 3.408 \times 10^{12} L_{\odot}$. With those values we obtain a nuclear timescale of $\tau_{\mathrm{nuc,\,Si-burn}} \approx 4 \times 10^{4}\,\mathrm{s} = 0.46\,\mathrm{d}$ during silicon burning. 1D stellar evolution calculations with a more realistic nuclear network also confirm that silicon burning typically only lasts several days. This value is much closer to the hydrostatic timescale τ_{hydr}, which means that we can cover substantial parts of this burning phase with multidimensional hydrodynamic simulations.

1.1 Stellar modeling

The following sections focus mainly on modeling techniques and recent results in stellar modeling. A more rigorous discussion of the astrophysical processes in stars with a focus on nuclear burning stages is given in Chapter 3.

1.1.1 1D stellar evolution

The basic equations of stellar structure and evolution under the assumption of a non-rotating, spherically symmetric star in hydrostatic equilibrium are given in Sec. 3.1. Consequently, processes that are inherently multidimensional can only be treated approximately in this framework. In particular convection and rotation are complicated multidimensional phenomena that are subject of ongoing research (see Sec. 1.1.2). Since a full treatment of those phenomena is not possible in a 1D framework, parametrized prescriptions are used instead. In the case of convection the so-called *mixing-length theory* by Böhm-Vitense (1958) is commonly used in 1D stellar evolution calculations. The following summary of mixing-length theory is based on Kippenhahn et al. (2012).

The total energy flux $l/4\pi r^2$ at a given point in a star is given by as the sum of the radiative flux F_{rad} and the convective flux F_{con}:

$$l/4\pi r^2 = F_{\mathrm{rad}} + F_{\mathrm{con}} = \frac{4acG}{3}\frac{T^4 m}{\kappa P r^2}\nabla_{\mathrm{rad}}, \tag{1.8}$$

where l is the luminosity as defined in Sec. 3.1, $a = 7.57 \times 10^{-15}\,\mathrm{erg\,cm^{-3}\,K^{-4}}$ is the radiation density constant, c is the speed of light, G is the gravitational constant (see Sec. 2.3.1), T is the temperature, κ is the Rosseland mean of the opacity, P is the pressure, m is the enclosed mass within radius r and ∇_{rad} is the gradient that would be necessary to transport the whole flux by radiation. In a similar fashion the actual radiative flux can be expressed as

$$F_{\mathrm{rad}} = \frac{4acG}{3}\frac{T^4 m}{\kappa P r^2}\nabla, \tag{1.9}$$

where $\nabla = \partial \ln T / \partial \ln P$ is now the actual temperature gradient of the stratification. To obtain an expression for the convective flux we consider a convective element (a blob) with an excess temperature DT over its surroundings. Furthermore, we assume that this element moves radially with velocity v and remains in complete pressure balance, i.e. $DP = 0$. Under these conditions the convective flux can be expressed as

$$F_{\mathrm{con}} = \rho v c_P DT, \tag{1.10}$$

where c_P is the specific heat and ρ is the mass density. One of the main assumptions of mixing-length theory is that the convective element travels the *mixing-length* l_{m} before it is absorbed into the surroundings. This means that elements passing through a sphere of constant radius r have on average traveled a distance of $l_{\mathrm{m}}/2$. Hence we can approximate the excess temperature as

$$\frac{DT}{T} = \frac{1}{T}\frac{\partial(DT)}{\partial r}\frac{l_{\mathrm{m}}}{2} = \left(\nabla - \nabla_{\mathrm{e}}\right)\frac{l_{\mathrm{m}}}{2}\frac{1}{H_P}, \tag{1.11}$$

where $H_P = -dr/d\ln P$ is the pressure scale height and the subscript e indicates that the respective quantity is computed for the convective element (details of this notation are introduced in Sec. 3.3). With similar considerations the velocity of the element due to buoyancy forces $k_r = -g\frac{D\rho}{\rho}$ can be estimated as

$$v^2 = g\delta\left(\nabla - \nabla_{\mathrm{e}}\right)\frac{l_{\mathrm{m}}^2}{8H_P}, \tag{1.12}$$

where $\delta = -\left(\partial\ln\rho/\partial\ln T\right)$ is a partial derivative of the equation of state as defined later in Sec. 3.3. Plugging Eqs. (1.11) and (1.12) into Eq. (1.10) then leads to the following expression for the convective flux

$$F_{\mathrm{con}} = \rho c_P T\sqrt{g\delta}\frac{l_{\mathrm{m}}^2}{4\sqrt{2}}H_P^{-3/2}\left(\nabla - \nabla_{\mathrm{e}}\right)^{3/2}. \tag{1.13}$$

Finally we also need to consider temperature changes of the convective element due to adiabatic expansion or compression and due to radiative change with the surroundings. With certain assumptions about the geometry the following expression can be derived

$$\frac{\nabla_{\mathrm{e}} - \nabla_{\mathrm{ad}}}{\nabla - \nabla_{\mathrm{e}}} = \frac{6acT^3}{\kappa\rho^2 c_P l_{\mathrm{m}} v}. \tag{1.14}$$

This means that with (1.8), (1.9), (1.12), (1.13), (1.14) we have now five equations for the five quantities F_{rad}, F_{con}, v, ∇_{e} and ∇ as a function of the mixing-length l_{m}. With further substitutions this system of equations can now be further reduced to a single cubic equation that gives a unique real solution for the temperature gradient ∇ in presence of convection (Kippenhahn et al., 2012, sec. 7.2).

The mixing-length theory as described above is based on several assumptions. The most obvious one is the mixing-length l_{m} itself as a free parameter of the model. The mixing-length is often expressed in units of the pressure scale height H_P as the *mixing-length parameter* α_{MLT}. This parameter generally is assumed to be constant within a star and in addition to that it does not vary with stellar mass, composition or evolutionary stage. Through comparison of stellar models with observed stars this parameter is inferred to be of order 1.

Another crucial assumption of this model is that all convective elements have the same size and shape. However, the theory of turbulence, numerical simulations, laboratory experiments and astrophysical observations clearly show that this is not the case. Therefore, convective energy is actually not transported in laminar flows of blobs of identical size and energy content, but rather by turbulent elements ("eddies") of all sizes. This turbulent energy spectrum is taken into account by the *full spectrum turbulence* theory by Canuto and Mazzitelli (1991), which is an extension of the mixing-length theory. This model gives significantly better results than the original mixing-length theory, however, it is still dependent on a mixing-length parameter.

1.1.2 Stellar evolution modeling in 2D and 3D

Clearly the 1D description from the previous section has many deficits. More accurate results can be obtained from hydrodynamic simulations of the star in 2D and 3D. The governing equations for this approach are discussed in Sec. 2.1. In the following, we give a general overview of recent results from stellar modeling in 2D and 3D. More specific details about hydrodynamic simulations of silicon burning are given in Sec. 1.2. The following is a short overview of those hydrodynamic codes that are most relevant for this work. A more detailed overview of hydrodynamic codes used for stellar modeling can be found in the review article by Kupka and Muthsam (2017).

Many astrophysical hydrodynamic codes are based on the piecewise-parabolic method (PPM) by Colella and Woodward (1984), which uses piecewise quadratic instead of piecewise linear functions in the reconstruction of the approximate Riemann solver (see Sec. 4.1.1). For instance this method is used in the *PROMPI* code (Meakin and Arnett, 2007b), which is a parallelized version of *PROMETHEUS* (Fryxell et al., 1989). Since this code was mainly used for simulations of oxygen burning and to some degree silicon burning, we discuss the applications of this code in Sec. 1.2.1. In Meakin and Arnett (2007b) they also adopt a simple model of turbulent entrainment for convective boundary mixing in stars, which was originally used in a geophysical context. This is further investigated in Cristini et al. (2017) in the case of convective boundary mixing during carbon burning.

Another astrophysical hydrodynamics code that uses the PPM scheme from Colella and Woodward (1984) is the low Mach number code *MAESTRO* (Almgren et al., 2007; Nonaka et al., 2010), which is built on the adaptive mesh refinement (AMR) framework AMReX. In contrast to other astrophysical codes it does not solve the compressible Euler equations. The hydrodynamic equations are instead reformulated to filter out sound waves but preserve the correct large-scale fluid motions and hydrostatic balance. This limits the code to strictly low Mach number applications such as core convection in massive stars (Gilet et al., 2013) or convection in helium shells on sub-Chandrasekhar white dwarfs (Zingale et al., 2013; Jacobs et al., 2016). Another code using the AMReX framework is the radiation-hydrodynamics code *CASTRO* (Almgren et al., 2010). Both *MAESTRO* and *CASTRO* are open source.

Piecewise-parabolic Boltzmann (PPB) is another method based on the PPM scheme. It is primarily used in the hydrodynamics code *PPMstar*, which is described in Woodward et al. (2015). This description of the code includes a detailed discussion of the PPB scheme. *PPMstar* was mainly used for 3D simulations of He-shell flash convection (Woodward et al., 2015) and oxygen-burning shell convection (Jones et al., 2017) in 4π geometry. In both cases they are mainly interested in the properties of mass entrainment at convective boundaries. For this purpose, they use a simplified description of microphysics. Based on those results, they developed a mixing description for 1D stellar evolution codes.

A different approach of solving the Euler equations is used in the *ANTARES* code by Muthsam et al. (2010). It employs high-order WENO schemes (see e.g. Liu et al., 1994) and uses the low Mach number method by Kwatra et al. (2009). In Grimm-Strele et al. (2014) they extended their WENO treatment to curvilinear grids. So far this code was mainly used for multidimensional modeling of Cepheid variables (Mundprecht et al., 2013, 2015) and surface convection in white dwarfs (Kupka et al., 2018).

The development of the implicit hydrodynamics code *MUSIC* was described in Viallet et al. (2011, 2013a, 2016). Similar to our code *SLH* the spatial discretization is performed independently of the time discretization (*method of lines*). The resulting system of ordinary differential equations for the spatial discretization is then solved using a combination of the Newton-Raphson method with Jacobian-free Krylov-subspace methods. Benchmarking results and comparisons with other hydrodynamic codes are shown in Goffrey et al. (2017). Recent applications of this code include the simulation of envelopes in red giants (Viallet et al., 2013b), lithium depletion in solar-like stars (Baraffe et al., 2017) and convective penetration in pre-main sequence stars (Pratt et al., 2017).

1.1.3 The hydrodynamics code SLH

In this section, we give a short overview of the most important features of the *Seven-League-Hydro* (henceforth *SLH*) code that is used as basis for this work. This code was originally developed during the PhD thesis of Miczek (2013) under the name *Low Mach number Hydro Code* (*LHC*). Due to the fact that there is a particle collider with the same name it was later renamed to *SLH*.

It solves the compressible Euler equations in one, two or three spatial dimensions with a finite-volume discretization, which is described in more detail in Sec. 4.1.1. In addition to that, it features both *explicit* and *implicit time integration*. The main advantage of implicit time stepping is that the time step can be extended beyond the CFL time step, which is required for explicit time stepping (see Sec. 4.1.2). This is particularly useful in the case of Low Mach number flows ($\lesssim 10^{-3}$), since the explicit CFL criterion scales with the sound speed c_s, which is much larger than the actual fluid velocity u. For the purpose of modeling low Mach number flows a *preconditioned Roe solver* that greatly reduces the dissipation in the low Mach number regime has been developed by Miczek et al. (2015). A more detailed analysis of the properties of this method is given in Barsukow et al. (2017). Alternative low Mach number schemes that are also implemented in *SLH* are presented in Miczek (2013) and Edelmann (2014). However, for the present work on silicon burning those low Mach number methods have not been used. Nuclear reaction networks have been incorporated into *SLH* during the PhD thesis of Edelmann (2014) along with the *Helmholtz equation of state* from Timmes and Swesty (2000). Aside from that, the code is implemented in a modular approach, which allows flexible combinations of reconstruction schemes, flux functions, nuclear networks and other components.

The non-linear system of equations resulting from the implicit time stepping is solved using the *Newton-Raphson* method, which reduces the solution of the non-linear system to an iterative solution of linear systems. Those linear systems of equations are then solved using *preconditioning* and *iterative solvers for sparse linear systems of equations* (e.g. BiCGSTAB, GMRES), which are described in more detail in Sec. 4.3.5.

One particular feature of the code is the transformation of the hydrodynamic equations to *general coordinates*. This means that different grid geometries can be defined through appropriate coordinate transformations as described in Sec. 4.1.3. With this approach the geometry of the grid can be easily adjusted to different problem geometries. However, as we shall see in Sec. 5.1.2 this approach is slightly problematic in the case of discontinuous coordinate transformations.

The code is parallelized using a hybrid approach with MPI and OpenMP. It has been shown in Edelmann and Röpke (2016) that this parallelization scales efficiently up to more than 300 000 cores on JUQUEEN. In addition to that, *SLH* is part of the High-Q Club of the Jülich Supercomputing Centre (JSC), which proves that it is capable of using the entire 28-rack BlueGene/Q system at JSC, i.e. 458 752 cores.

Until this thesis one of the major shortcomings of the code was the simplified, time independent treatment of gravity due to the fact that a gravity solver was not yet implemented. As such, the implementation of a gravity solver that also works on general curvilinear grids and scales to a large number of cores was one of the main objectives of this thesis. Since we already have iterative solvers for sparse linear systems in the code we decided to implement the mixed method Poisson solver, which is described in Sec. 4.3.4. In addition to that, a standard Poisson solver on Cartesian grids (Sec. 4.3.2) and a monopole gravity solver (Sec. 4.3.1) have also been implemented during this thesis. Benchmarking results for those gravity solvers are shown in Sec. 5.1.

The code has previously been used for simulations of convective mixing in massive Population III stars as part of Edelmann (2014). In addition to that, it has been used for simulations of Classical Novae by Bolaños Rosales (2016) and Lach (2016). Simulations of shear instabilities in two and three dimensions have been carried out by Edelmann et al. (2017) and Horst (2016), respectively. The latter is still an ongoing project. Another recent application of the code is the simulation of convective boundary mixing by Botto Poala (2017) using the *box in a star* approach.

1.2 Simulations of silicon and oxygen burning in 2D and 3D

As we discuss in Sec. 3.2.2 perturbations arising during convective silicon and oxygen burning are a possible candidate to enable shock revival in three-dimensional models of core-collapse supernovae. However, due to a lack of sophisticated three-dimensional models of the silicon burning phase the impact of such perturbations on shock revival remains inconclusive. In the following section, we give an overview on previous attempts to model silicon burning. In addition to that, we discuss the deficiencies of each of those modeling attempts.

1.2.1 Previous work

First attempts of modeling oxygen burning in massive stars prior to core-collapse have been carried out by Arnett (1994) and Bazán and Arnett (1994) with the hydrodynamics code *PROMETHEUS* in two spatial dimensions. They found that significant mixing beyond convective boundaries determined by mixing-length theory leads to entrainment of ^{12}C into the convective region. This additional fuel enhances nuclear burning and causes hot spots and inhomogeneities in the electron fraction Y_e, which may influence the core-collapse and the explosive nucleosynthesis. Follow-up studies by Bazán and Arnett (1998) confirm that convection is more dynamic than predicted by one-dimensional diffusion-like approximations. Similar results were found in two-dimensional simulations of silicon burning with a 123 isotope nuclear network (Bazán and Arnett, 1997). Asida and Arnett (2000) covered a longer period of oxygen burning than Bazán and Arnett (1994, 1998) with the two-dimensional hydrodynamics code *VULCAN* and confirmed the earlier results.

With *PROMPI*, a parallelized version of *PROMETHEUS*, first 3D simulations of oxygen shell burning with more realistic physics (i.e. compressibility, nuclear network, realistic EoS, appropriate boundaries, etc.) were performed (Meakin and Arnett, 2007a,b). Their main conclusions are that 2D simulations are significantly different from 3D simulations, in terms of both flow morphology and velocity amplitude. Moreover, they find that convective regions are better predicted with a dynamic boundary condition based on the bulk Richardson number than purely local criteria like the Schwarzschild or Ledoux criterion (see Sec. 3.3). Arnett and Meakin (2011) performed 2D simulations with multiple, simultaneously active burning shells of carbon, oxygen, neon and silicon. This approach allows additional interactions between different burning shells. Substantial deviations from spherical symmetry are clearly visible in this simulation. This development of large scale asymmetries during silicon burning and oxygen burning is particularly relevant for shock revival in the modeling of Type II supernovae (see Sec. 3.2.2).

Arnett and Meakin (2011) also list a number of requirements that need to be fulfilled in future 3D simulations of silicon burning to study these phenomena in more detail:

1. Simulation of the full core in 4π steradians to get the lowest order fluid modes ($l = 1$), rotation and low-order MHD modes

2. Multiple shells (C, Ne, O, and Si) to get shell interactions

3. A realistic equation of state that captures the effect of electron-positron pairs and relativistic partial degeneracy

4. A nuclear network that incorporates realistic burning of C, Ne, O and Si with electron captures in a dynamic environment
5. Resolve turbulent motions
6. Compressible fluid dynamics

In addition to that, accurate modeling of these phenomena also requires a gravity solver that takes deviations of the density distribution from spherical symmetry into account. Consequently, a simple monopole gravity solver, which assumes a spherically symmetric density distribution, is not suitable for the modeling of the perturbations in Type II supernova progenitors.

A more recent attempt of modeling silicon burning in 3D until the onset of core-collapse was made by Couch et al. (2015) for a $15M_\odot$ star. For this purpose, they followed the evolution of a pre-main sequence star until the onset of core-collapse in the open source 1D stellar evolution code *MESA* (Paxton et al., 2011, 2013, 2015). Afterwards, they followed the last 160 s in a 3D hydrodynamics code. However, as discussed by Müller et al. (2016) they use several approximations that might influence the structure and magnitude of the perturbations. For instance, they use a monopole gravity solver, simulate only one octant of the whole star and artificially enhance the electron capture rate on ^{56}Fe by a factor of 50. The implications of these approximations for the modeling of core-collapse supernovae in 3D are further discussed in Sec. 3.2.2.

1.2.2 Goals of this work

With the simulations presented in this thesis we want to alleviate some of the above artificial constraints that were imposed in previous simulations of silicon burning. For this purpose we simulate the whole core in three spatial dimensions up to a radius of 3.2×10^8 cm. This covers the full silicon burning shell and also large parts of the oxygen burning shell, however, we were still unable to include neon and carbon burning shells in our 3D modeling. We employ the Helmholtz equation of state from Timmes and Swesty (2000), which was also used in the simulations of Arnett and Meakin (2011) and Couch et al. (2015). In addition to that, we adopted the effective rate α-chain nuclear reaction network (see Sec. 4.2.2) that is used in the stellar evolution code *MESA* into *SLH* during the course of this thesis. We decided to use this nuclear network to maintain consistency between the composition in the stellar evolution model and our hydrodynamic model, since the conversion of the composition between different nuclear networks is not necessarily straightforward. As shown in Sec. 5.4.1 the resolution in our 2D and 3D simulations is sufficient to obtain a $k^{-5/3}$ turbulent energy spectrum according to Kolmogorov (1941). Since we are using a compressible hydrodynamic scheme we fulfill most of the points listed by Arnett and Meakin (2011). In addition to that, we apply our newly implemented standard Poisson solver for a more accurate treatment of self-gravity. At the same time, it serves as a first test of the performance of the gravity solver in a realistic physical application.

However, there is still room for improvements with regard to the nuclear network and the simulated domain. A QSE-reduced nuclear network based on Hix et al. (2007) might be more suited for simulations of silicon burning, since it would give more accurate yields than the effective rate nuclear network that we are currently using. However, it would require substantial effort to implement such a network into *SLH*. In addition to that, we would also have to construct new stellar evolution models with the same nuclear network. Consequently, this a more long-term project, which requires major changes in the implementation of the nuclear network.

Extending the simulated domain to cover also neon and carbon burning is mainly an issue of performance and computational resources. However, since this is our first attempt to model silicon burning in 3D, we expect that there is still a lot of room for improvements in this regard.

1.3 Outline

Chapter 2 introduces the physical ingredients that are necessary to describe the phenomena that are discussed in this thesis. In particular Sec. 2.1 covers the theoretical foundations of our hydrodynamic simulations. The physics of nuclear reactions are discussed in Sec. 2.2, whereas the relevant aspects of Newtonian gravity are introduced in Sec. 2.3. Chapter 3 gives a quick overview of astrophysical concepts during the evolution of stars with a particular focus on massive stars. The hydrostatic nuclear burning stages in massive stars are discussed in Sec. 3.1, whereas Sec. 3.2 covers core-collapse supernovae towards the end of stellar evolution in massive stars. Sec. 3.3 introduces the concept of dynamical instabilities in stars. In Chapter 4 we formulate numerical methods based on the physical concepts that were introduced in Chapter 2. The discretization of the hydrodynamic equations on curvilinear grids is discussed in Sec. 4.1, whereas nuclear reaction networks are covered in Sec. 4.2. Our implementation of the gravity solvers is described in Sec. 4.3, in particular the newly developed mixed method Poisson solver is discussed in Sec. 4.3.4. The results of this thesis are presented in Chapter 5. Sec. 5.1 covers the benchmarking for the different gravity solvers that were implemented during the course of this thesis. The initial setup for the simulations of silicon burning is introduced in Sec. 5.2. The results for the 2D and 3D simulations of silicon burning are presented in Sec. 5.3 and Sec. 5.4, respectively. In Chapter 6 we summarize the most important results of this thesis and give an outlook for future work.

2 Physical background

2.1 Hydrodynamics

For the simulations in this work we describe the star as a continuous fluid. This assumption holds if the following requirements are fulfilled (see e.g. Müller, 1998; Thompson, 2006). The macroscopic length scale L of interest has to be much larger than the collisional mean free path l of the particles. This allows us to introduce the concept of a *fluid element* that is large in comparison to l, but small in comparison to L. In addition to that, mean physical quantities e.g. the fluid density ρ or the fluid velocity u can be defined due to the large number of particles in the fluid element. Furthermore, only "short range" forces are allowed between particles, since otherwise collective effects have to be considered. This is for example the case for gravity, which is treated as a macroscopic external force in the hydrodynamic equations.

These conditions are usually fulfilled in astrophysical environments due to the large length scales of astrophysical objects. Consequently, we can describe the star with a set of hydrodynamic balance equations for mass, momentum and energy. In astrophysical environments the viscosity ν is similar to liquids and gases on earth. Thus, comparing the vast length and velocity scales of astrophysical objects to their viscosity leads to very large *Reynolds numbers*.

$$\mathrm{Re} = \frac{Lu}{\nu}. \tag{2.1}$$

According to Thompson (2006) the Reynolds number in the Sun can be as large as 10^{18}. Therefore, we will only consider the inviscid Euler equations instead of the viscous Navier-Stokes equations in the following. Here it is important to note that the Euler equations form a hyperbolic set of differential equations, whereas the Navier-Stokes equations also have a parabolic part due to the viscous terms. However, truncation errors of the numerical scheme and the finite grid resolution lead to an intrinsic numerical viscosity, which depends on the numerical method and the grid resolution.

2.1.1 Euler equations

The Euler equations consist of hydrodynamical balance equations for mass, momentum and energy. In general a hydrodynamical balance equation for an extensive quantity A in a fluid volume V can be expressed in the form (see hydrodynamics textbooks e.g. Landau and Lifschitz, 1991; Batchelor, 2000)

$$\frac{dA}{dt} = \frac{d_\mathrm{s}A}{dt} + \frac{d_\mathrm{f}A}{dt}, \tag{2.2}$$

where $d_{\mathrm{s}}A$ is the change of A due to creation or destruction inside the volume V and $d_{\mathrm{f}}A$ represents the flux of A over the boundary ∂V of the volume. Eq. (2.2) is the integral form of a balance equation. The differential form of the balance equation can be derived using the divergence theorem. This leads to

$$\frac{\partial}{\partial t}a(\boldsymbol{x},t) + \boldsymbol{\nabla}\cdot\boldsymbol{j}_a(\boldsymbol{x},t) = s_a(\boldsymbol{x},t), \tag{2.3}$$

where $a(\boldsymbol{x},t)$ is the density of A, $\boldsymbol{j}_a(\boldsymbol{x},t)$ is the flux density of A over the boundary ∂V and $s_a(\boldsymbol{x},t)$ is the source density of A inside V.

Using this framework we can formulate the Euler equations for mass, momentum and energy conservation in the following way:

$$\begin{aligned} \partial_t\rho + \boldsymbol{\nabla}(\rho\boldsymbol{u}) &= 0, \\ \partial_t(\rho\boldsymbol{u}) + \boldsymbol{\nabla}(\rho\boldsymbol{u}\otimes\boldsymbol{u}) &= -\boldsymbol{\nabla}P + \rho\boldsymbol{f}, \\ \partial_t(\rho e_{\mathrm{tot}}) + \boldsymbol{\nabla}(\rho e_{\mathrm{tot}}\boldsymbol{u}) &= -\boldsymbol{\nabla}(P\boldsymbol{u}) + \rho\boldsymbol{u}\boldsymbol{f}, \end{aligned} \tag{2.4}$$

where ρ is again the mass density of the fluid, $\boldsymbol{u} = (u,v,w)$ is the fluid velocity, P is the pressure, $\boldsymbol{f}$ an external force (e.g. gravity $\boldsymbol{g}$, Sec. 2.3) and $e_{\mathrm{tot}} = 0.5\ \boldsymbol{u}^2 + e_{\mathrm{i}}$ is the specific total energy (e_{i} is the specific internal energy). Eq. (2.4) constitutes a system of five partial differential equations for the six quantities ρ, $\boldsymbol{u}$ (three components), p and e_{tot}. Therefore, an additional equation is required to close the system. The missing equation is provided by the equation of state (EoS) that relates ρ, P and e_{tot} as a function of the chemical composition (see Sec. 2.2).

2.1.2 Source terms for nuclear reactions

Since we want to model nuclear burning processes in stellar environments, we have to extend Eq. (2.4) to account for energy release or consumption due to nuclear reactions. Therefore, we have to consider additional source terms in the energy conservation, which leads to (see e.g. Müller, 1998):

$$\partial_t(\rho e_{\mathrm{tot}}) + \boldsymbol{\nabla}(\rho e_{\mathrm{tot}}\boldsymbol{u}) = -\boldsymbol{\nabla}(P\boldsymbol{u}) + \rho\boldsymbol{u}\boldsymbol{f} + \rho Q, \tag{2.5}$$

where Q depends on the reaction rates r_i of the chemical species ($i = 1,\ldots,N$). In addition to that, we have to introduce balance equations for the N chemical species

$$\partial_t(\rho X_i) + \boldsymbol{\nabla}(\rho\boldsymbol{u}X_i) = r_i \qquad i = 1,\cdots,N, \tag{2.6}$$

where X_i is the mass fraction of the chemical species i. The mass fractions X_i are defined in the following way:

$$X_i = \frac{n_i A_i}{\sum_j n_j A_j}, \tag{2.7}$$

where n_i is the number density and A_i the nucleon number of species i. Note that the sum of all equations in (2.6) yields the mass conservation of Eq. (2.4). This leads to the following constraint on the reaction rates:

$$\sum_{i=1}^{N} r_i = 0. \tag{2.8}$$

The reaction rates r_i are determined by a network of nuclear reactions (cf. Sec. 2.2). As such, we have now extended the Euler equations (2.4) with one additional equation for each of the species, resulting in a set of $5 + N$ partial differential equations. Even for a low number of species a full coupling of the nuclear reactions to the hydrodynamics is usually not possible. The dense blocks in the sparse Jacobian become larger and therefore it becomes more inefficient to invert this matrix. Consequently, this approach of using *active scalars* is limited to very small nuclear networks. For only five nuclear species, the block size of the sparse matrix already doubles in comparison to a purely hydrodynamic Jacobian, which makes it significantly more expensive in terms of both storage and computation time. Therefore, in practical applications the nuclear species are often treated as *passive scalars*. This means that the Euler equations are first solved under the assumption that the composition does not change. Afterwards, the chemical species are updated according to the obtained mass flux and the composition changes due to nuclear reactions. This alternating sequence of evaluating the Euler equations, the species advection and the nuclear reactions separately is a typical operator splitting approach as described in Press et al. (2007). This splitting introduces a numerical error that scales with the time-step. It has been shown in Edelmann (2014) that the splitting error is usually negligible. More details of the operator splitting are discussed in Sec. 4.2.1.

2.2 Nuclear reactions

Nuclear reactions are crucial for understanding the burning processes in the interior of stars. The role of thermonuclear burning in stars will be discussed in more detail in Sec. 3.1. In this section, we are going to focus on the equations that describe the nuclear burning processes and their implications. The following discussion is based on Müller (1998); Iliadis (2015).

Nuclear reactions are classified according to the number of input nuclei. The most common reactions in astrophysical environments either involve one or two species. One-body reactions include photo-disintegrations and weak reactions such as β-decays or electron captures. The latter ones will be discussed in more detail in Sec. 2.2.2. Two-body reactions include processes involving protons, neutrons or α-particles. In some cases heavier nuclei such as ^{12}C or ^{16}O may also collide with each other, however these processes are more unlikely due to the higher Coulomb barrier. As shown in Iliadis (2015) the *Gamow factor* for a two-body reaction yields:

$$\hat{T} \approx \exp\left(-\frac{2\pi}{\hbar}\sqrt{\frac{m}{2E}} Z_1 Z_2 e^2\right), \tag{2.9}$$

where $\hat{T}$ is the transmission coefficient through the Coulomb barrier, m is the reduced mass of the participating nuclei, E is the center-of-mass energy, Z_1 and Z_2 are the proton numbers of the participating nuclei and e is the elementary charge. We can see that the transmission coefficient rapidly decreases with increasing proton numbers due to the product of Z_1 and Z_2 in the exponential function.

The cross sections for reactions that involve three input species are much lower, however, the triple α reaction is particular important during helium burning, as we will see in Sec. 3.1.2. Reactions that involve four or more input species are negligible in astrophysical environments due to their low cross sections.

In order to formulate differential equations for the nuclear reactions we introduce the specific abundance Y_i (see e.g. Müller, 1998):

$$Y_i = \frac{X_i}{A_i} = \frac{n_i}{\sum_j n_j A_j}. \tag{2.10}$$

This is the number of nuclei of species i in relation to the total number of nucleons in the system. The change in Y_i is then given as

$$\begin{aligned} \dot{Y_i} = \sum_j c_i(j)\lambda_j Y_j + \sum_{j,k} c_i(j,k)\rho N_\mathrm{A} < j,k > Y_j Y_k \\ + \sum_{j,k,l} c_i(j,k,l)(\rho N_\mathrm{A})^2 < j,k,l > Y_j Y_k Y_l. \end{aligned} \tag{2.11}$$

λ_j is the decay rate of species j into species i and N_A is the Avogadro constant. In a similar fashion $< j,k >$ and $< j,k,l >$ denote the thermally averaged products of cross sections and relative velocity in the center of mass system for the two-body and three-body reactions, respectively. The coefficients c_i are introduced to prevent double counting of reactions with identical particles:

$$c_i(j) = \pm N_i, \qquad c_i(j,k) = \pm\frac{N_i}{N_j!N_k!}, \qquad c_i(j,k,l) = \pm\frac{N_i}{N_j!N_k!N_l!}, \tag{2.12}$$

where the N_i denote absolute numbers for the corresponding species i. The sign of the coefficients depends on whether the species i is produced (+) or destroyed (-). Eq. (2.11) is a *stiff* set of differential equations due to the difference in reaction timescales. Therefore, specific numerical methods are required to solve this system efficiently. Some of them are discussed in Sec. 4.2.1.

2.2.1 Equilibria of nuclear reactions

At sufficiently high densities and temperatures ($T \gtrsim 3 \times 10^9\,\mathrm{K}$) the strong and electromagnetic interactions are in equilibrium with their reverse reactions (Hix and Meyer, 2006). However, the reactions that involve weak nuclear processes are usually not in equilibrium. This equilibrium of the strong and electromagnetic reactions is known as *nuclear statistical equilibrium* (NSE). Under these conditions the composition can be expressed as function of the thermodynamic variables temperature T, density ρ and the electron fraction Y_e, which is given by the weak reactions (Sec. 2.2.2).

The chemical potential of a nucleus ${}^A\mathrm{Z}$ with Z protons and $N = A - Z$ neutrons can be expressed as a function of the chemical potential for the free protons (μ_p) and neutrons (μ_n).

$$\mu_{Z,A} = Z\mu_\mathrm{p} + N\mu_\mathrm{n} \tag{2.13}$$

With the Maxwell Boltzmann distribution for the chemical potential this can be used to derive an expression for the abundance $Y(^A\mathrm{Z})$ of every nuclear species in terms of the abundances of free protons (Y_p) and neutrons (Y_n) (Hix and Meyer, 2006):

$$Y(^A\mathrm{Z}) = \frac{G\left(^A\mathrm{Z}\right)}{2^A}\left(\frac{\rho N_\mathrm{A}}{\theta}\right)^{A-1} A^{\frac{3}{2}} \exp\left(\frac{B(^A\mathrm{Z})}{k_\mathrm{B}T}\right) Y_\mathrm{n}^N Y_\mathrm{p}^Z \equiv C(^A\mathrm{Z}) Y_\mathrm{n}^N Y_\mathrm{p}^Z, \tag{2.14}$$

where $G\left(^A\mathrm{Z}\right)$ and $B(^A\mathrm{Z})$ are the partition function and the binding energy of nucleus $^A\mathrm{Z}$, respectively, k_B is Boltzmann's constant and $\theta = (m_u k_B T/2\pi\hbar^2)^{3/2}$.

The nuclear composition can then be computed using Eq. (2.14) under the assumption of nucleon number conservation ($\sum A_i Y_i = 1$) for a given temperature T, density ρ and electron fraction Y_e. The latter one is determined by the weak interaction history (Sec. 2.2.2). The computational cost of computing the nuclear abundances for all species can be reduced significantly with this method.

At low NSE temperatures the NSE composition heavily favors the most bound nuclei for which $Z/A \sim Y_\mathrm{e}$. With increasing temperature the equilibrium shifts towards ^{4}He at intermediate temperatures and towards free neutrons and protons at high temperatures. The transition temperatures between those three regions shift towards higher temperatures with increasing density.

The strict requirements for the above NSE treatment are often only partially fulfilled in astrophysical environments and therefore not all the reactions are in equilibrium. During hydrostatic silicon burning the temperatures are typically high enough for NSE treatment, however, the equilibrium composition is often not fully established. Under these conditions one can often identify sub-groups of nuclei that are in local equilibrium with each other.

The *quasi-statistical equilibria* (QSE) between those sub-groups can be used to obtain the abundances for all nuclear species within the group from the abundance of a single group member. Let us consider a QSE-reduced α-network as an example. We introduce the silicon QSE group (^{28}Si, ^{32}S, ^{36}Ar, ^{40}Ca, ^{44}Ti) and the iron peak QSE group (^{48}Cr, ^{52}Fe, ^{56}Ni, ^{60}Zn). The abundances for individual species in those groups can be calculated via (Hix and Meyer, 2006):

$$\begin{aligned} Y_\mathrm{QSE,Si}\left(^A\mathrm{Z}\right) &= \frac{C\left(^A\mathrm{Z}\right)}{C\left(^{28}\mathrm{Si}\right)} Y\left(^{28}\mathrm{Si}\right) Y_\alpha^{\frac{A-28}{4}} \\ Y_\mathrm{QSE,Ni}\left(^A\mathrm{Z}\right) &= \frac{C\left(^A\mathrm{Z}\right)}{C\left(^{56}\mathrm{Ni}\right)} Y\left(^{56}\mathrm{Ni}\right) Y_\alpha^{\frac{A-56}{4}}, \end{aligned} \tag{2.15}$$

where $C\left(^A\mathrm{Z}\right)$ is obtained from the NSE relation (2.14). As such, we have reduced the original 14-isotope α-network consisting of ^{4}He, ^{12}C, ^{16}O, ^{20}Ne, ^{24}Mg, ^{28}Si, ^{32}S, ^{36}Ar, ^{40}Ca, ^{44}Ti, ^{48}Cr, ^{52}Fe, ^{56}Ni and ^{60}Zn to a 7-isotope reduced QSE network containing the species ^{4}He, ^{12}C, ^{16}O, ^{20}Ne, ^{24}Mg, ^{28}Si and ^{56}Ni. This network only considers nuclei with $N = Z$. A more general QSE-reduced network for silicon burning that also includes weak reactions is discussed in Hix et al. (2007).

2.2.2 Weak interactions and neutrino losses

Weak interactions are important in late stages of stellar evolution (e.g. silicon burning) as they lead to changes in the neutron excess parameter η (see e.g. Iliadis, 2015)

$$\eta \equiv \sum_i (N_i - Z_i)\, Y_i = \sum_i \frac{N_i - Z_i}{A_i} X_i, \tag{2.16}$$

where N_i, Z_i, A_i, Y_i and X_i denote the number of neutrons, protons, nucleons, specific abundance and mass fraction of species i, respectively. We can see that η goes to 1, if the plasma is composed entirely of neutrons ($Z = 0$) and to -1, if it is composed entirely of protons ($N = 0$). Moreover, η goes to 0, if only symmetric nuclei with $N = Z$ are present such as ^{4}He, ^{12}C or ^{16}O. A related quantity is the electron fraction Y_e, defined as

$$Y_\mathrm{e} = \sum_i \frac{Z_i}{A_i} X_i. \tag{2.17}$$

It is related to η via

$$\eta = 1 - 2Y_\mathrm{e}. \tag{2.18}$$

The following weak interaction processes are important in astrophysical environments:

$$\begin{align}
{}^{A}_{Z}\mathrm{X} &\longrightarrow {}_{Z+1}^{A}\mathrm{Y} + \mathrm{e}^- + \bar{\nu} && \beta^-\text{- decay} \tag{2.19}\\
{}^{A}_{Z}\mathrm{X} &\longrightarrow {}_{Z-1}^{A}\mathrm{Y} + \mathrm{e}^+ + \nu && \beta^+\text{- decay} \tag{2.20}\\
{}^{A}_{Z}\mathrm{X} + \mathrm{e}^- &\longrightarrow {}_{Z-1}^{A}\mathrm{Y} + \nu && \text{electron capture} \tag{2.21}
\end{align}$$

Electron capture reactions are particularly important in Type II supernova progenitors (Sec. 3.2) as it reduces the number of electrons and thus the electron degeneracy pressure, which stabilizes the core against collapse. This decrease in pressure is very important for the dynamics of core collapse.

Neutrinos emitted in weak interactions typically do not interact with nuclei or electrons, as the cross sections for neutrino scattering are very small at the prevailing densities. Therefore, neutrinos can escape from the reaction site and transport away a portion of the thermonuclear energy release. This decrease in energy due to neutrino losses further accelerates the collapse of the core in Type II supernovae.

Weak reactions also influence the nucleosynthesis in late stages of stellar evolution and during explosive burning (e.g. in thermonuclear supernovae). The NSE composition shifts towards species that are favored by Y_e. For example ^{56}Ni is favored over $^{54}\mathrm{Fe} + 2\,^{1}\mathrm{H}$, if $Y_e = 0.5$, even though ^{54}Fe is more tightly bound than ^{56}Ni. However, for lower Y_e the NSE composition would shift towards ^{54}Fe.

The released energy in Eq. (2.19) to Eq. (2.21) is almost entirely transferred to the emitted leptons (i.e. $\mathrm{e}^-, \mathrm{e}^+, \nu, \bar{\nu}$). In β-decays two leptons are emitted. Therefore, the total energy release of the reaction is continuously distributed between those two leptons (i.e. the neutrino energy can range from zero to the total reaction energy). However, in the case of electron captures only one lepton is produced. Consequently, the resulting neutrino carries the whole reaction energy.

Detailed evaluations of weak reaction rates for more than 100 nuclei in the mass range $A = 45 - 65$ for densities $\rho Y_e = 10^7 - 10^{10}\,\mathrm{mol\,cm^{-3}}$ and temperatures $T = 10^9 - 10^{10}\,\mathrm{K}$ have been carried out by Langanke and Martinez-Pinedo (2000), using shell model calculations. They consider several hundred states in both the parent and daughter nucleus.

2.3 Gravity

Gravity is another crucial ingredient for a consistent modeling of the physics in stars. In hydrostatic equilibrium it balances out the radiation pressure due to nuclear fusion in the center of the star and holds it together. During the late stages of silicon burning and core collapse it is responsible for the contraction of the core. Therefore, accurate modeling of gravity is particularly important during these phases of stellar evolution. For our purpose of modeling silicon burning Newtonian gravity is accurate enough. However, under more extreme conditions a fully relativistic treatment of gravity may be necessary. As such, we only cover some aspects of Newtonian gravity in the following sections and neglect contributions of general relativity.

2.3.1 Poisson's equation

The gravitational field $\boldsymbol{g}$ for a mass distribution $\rho(\boldsymbol{r})$ is given by:

$$\boldsymbol{g}(\boldsymbol{r}) = -\int G\rho(\boldsymbol{r'})\frac{\boldsymbol{r}-\boldsymbol{r'}}{|\boldsymbol{r}-\boldsymbol{r'}|^3}d^3\boldsymbol{r'}, \tag{2.22}$$

where $G = (6.674\,08 \pm 0.000\,31) \times 10^{-11}\,\mathrm{m^3\,kg^{-1}\,s^{-2}}$ is the gravitational constant according to CODATA 2014 and ρ is the mass density. As we can see the gravitational field at location $\boldsymbol{r}$ depends on the contributions of all the infinitesimal mass points located at $\boldsymbol{r'}$ inside the integration volume (*superposition principle*). This implies that a change of the density at location $\boldsymbol{r'}$ immediately influences the gravitational field $\boldsymbol{g}$ at location $\boldsymbol{r}$.

Since gravity is a conservative force, the gravitational field can be expressed as gradient of a potential $\boldsymbol{g} = -\boldsymbol{\nabla}\Phi$ with

$$\Phi(\boldsymbol{r}) = -\int G\frac{\rho(\boldsymbol{r'})}{|\boldsymbol{r}-\boldsymbol{r'}|}d^3\boldsymbol{r'}. \tag{2.23}$$

The relation

$$\boldsymbol{\nabla}^2\left(\frac{1}{|\boldsymbol{r}|}\right) = -4\pi\delta_\mathrm{D}(\boldsymbol{r}) \tag{2.24}$$

then leads to *Poisson's equation* of gravity (δ_D is the Dirac delta function):

$$\boldsymbol{\nabla}^2\Phi = 4\pi G\rho. \tag{2.25}$$

This is an elliptic partial differential equation of second order. It can be transformed into a set of four partial differential equations of first order:

$$\begin{aligned} -\boldsymbol{\nabla}\Phi &= \boldsymbol{g}, \\ \boldsymbol{\nabla}\boldsymbol{g} &= -4\pi G\rho. \end{aligned} \tag{2.26}$$

This alternative formulation of Poisson's equation will be used in Sec. 4.3.4 for the *mixed method Poisson solver*. The last equation in (2.26) is analogous to Gauss's law for the electrical field in electrostatics, which is part of Maxwell's equations.

2.3.2 Monopole gravity

A simple solution for the gravitational potential $\Phi(\boldsymbol{r})$ can be obtained by assuming spherical symmetry. Eq. (2.23) can be expanded into spherical harmonics similar to the electrostatic potential (see e.g. Jackson, 1999) using

$$\frac{1}{|\boldsymbol{r}-\boldsymbol{r'}|} = 4\pi\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\frac{1}{2l+1}\frac{r_<^l}{r_>^{l+1}}Y_{lm}^*(\theta',\phi')Y_{lm}(\theta,\phi), \tag{2.27}$$

where $r_< = \min(|\boldsymbol{r}|,|\boldsymbol{r'}|)$, $r_> = \max(|\boldsymbol{r}|,|\boldsymbol{r'}|)$ and $Y_{lm}(\theta,\phi)$ are the spherical harmonics. This leads to the following expression for the gravitational potential in terms of multipole moments:

$$\Phi(\boldsymbol{r}) = -G\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\frac{4\pi}{2l+1}Y_{lm}(\theta,\phi)\int\frac{\rho(\boldsymbol{r'})r_<^l}{r_>^{l+1}}Y_{lm}^*(\theta',\phi')d^3\boldsymbol{r'}. \tag{2.28}$$

For the gravitational monopole moment ($l = m = 0$) this reduces to

$$\Phi(r) = -4\pi G\int_0^{\infty}\frac{\rho(r')}{r_>}r'^2dr', \tag{2.29}$$

since $Y_{00} = Y_{00}^* = 1/\sqrt{4\pi}$ and the integration over the angular part becomes trivial leading to another factor 4π. With the definition of $r_>$ we can now divide the integral in Eq. (2.29) into inner and outer parts:

$$\Phi(r) = -\frac{4\pi G}{r}\int_0^{r}\rho(r')r'^2dr' - 4\pi G\int_r^{\infty}\rho(r')r'dr'. \tag{2.30}$$

The angular dependencies of the gravitational potential have been eliminated in this expression. Therefore, this equation can be integrated numerically by using radial bins (see Sec. 4.3.1). This treatment only gives good results if the deviations of the density distribution from spherical symmetry are not too large, which is not necessarily fulfilled in all astrophysical applications. The above treatment could be improved by considering also higher-order multipole moments in Eq. (2.28).

2.3.3 Hydrostatic equilibrium

As we will see in Chapter 3 stars spend most of their lifetime in hydrostatic burning phases. Dynamic phases (e.g. stellar explosions, core contraction due to gravity) during stellar evolution are often very short lived. In *hydrostatic equilibrium* the forces due to gravity and the pressure gradient exactly compensate each other. In a formal way this can be achieved by setting the velocities $\boldsymbol{u}$ in the Euler equations (2.4) to zero, which yields:

$$\begin{aligned} \partial_t \rho &= 0, \\ \boldsymbol{\nabla} P &= \rho \boldsymbol{g}, \\ \partial_t (\rho e_{\mathrm{tot}}) &= 0, \end{aligned} \tag{2.31}$$

where $\boldsymbol{g}$ is the external force due to gravity. Under the assumption of spherical symmetry $\boldsymbol{\nabla} P = \rho \boldsymbol{g}$ reduces to

$$\frac{\partial P}{\partial r} = -\frac{Gm}{r^2}\rho, \tag{2.32}$$

where m is the enclosed mass at radius r. This is one of the basic equations of stellar evolution (see Sec. 3.1). Therefore, one-dimensional initial models from stellar evolution codes typically fulfill this condition. However, maintaining hydrostatic equilibrium during the mapping from the one-dimensional stellar evolution code to the multidimensional hydrodynamics code is not trivial (see Sec. 5.2.1). If the hydrostatic equilibrium is not fulfilled in the initial hydrodynamic model this might result in rapid contraction or expansion of the model.

3 Stellar Evolution

In this chapter, we give a quick overview over the most relevant aspects of stellar evolution for massive stars $M \gtrsim 10 M_\odot$. Sec. 3.1 covers hydrostatic nuclear burning processes in stars with a particular focus on silicon burning, whereas Sec. 3.2 focuses on core contraction and shock revival during core-collapse supernovae (Type II, Ib, Ic) and highlights some recent developments and challenges in the computational modeling of these phenomena. The growth of dynamical instabilities in stars is discussed in Sec. 3.3.

3.1 Hydrostatic nuclear burning in massive stars

In hydrostatic equilibrium the evolution of a spherically symmetric star is governed by the following set of differential equations (see e.g. Kippenhahn et al., 2012):

$$\frac{\partial r}{\partial m} = \frac{1}{4\pi r^2 \rho}, \tag{3.1}$$

$$\frac{\partial P}{\partial m} = -\frac{Gm}{4\pi r^4}, \tag{3.2}$$

$$\frac{\partial l}{\partial m} = \epsilon_\mathrm{n} - \epsilon_\nu - c_P \frac{\partial T}{\partial t} + \frac{\delta}{\rho}\frac{\partial P}{\partial t}, \tag{3.3}$$

$$\frac{\partial T}{\partial m} = -\frac{GmT}{4\pi r^4 P}\nabla, \tag{3.4}$$

$$\frac{\partial X_i}{\partial t} = \frac{m_i}{\rho}\left(\sum_j r_{ji} - \sum_k r_{ik}\right), \quad i = 1, \ldots, N. \tag{3.5}$$

Eq. (3.2) describes the hydrostatic equilibrium in mass coordinates. With Eq. (3.1) this is equivalent to Eq. (2.32) derived in Sec. 2.3.3. ϵ_n in Eq. (3.3) accounts for energy production due to nuclear reactions, whereas ϵ_ν accounts for energy losses due to neutrinos. $\delta = -(\partial \ln \rho / \partial \ln T)$ is a partial derivative of the equation of state, which is more rigorously defined in Sec. 3.3, and $\nabla = d \ln T / d \ln P$ is the actual temperature gradient in the star. If the energy in the star is mainly transported by radiation (and conduction), the temperature gradient

$$\nabla = \nabla_\mathrm{rad} = \frac{3}{16\pi a c G}\frac{\kappa l P}{m T^4} \tag{3.6}$$

is given by Eq. (1.8). In the case of energy transport by convection ∇ has to be replaced with a value obtained from a proper theory of convection, e.g. *mixing-length theory* (see Sec. 1.1.1). The luminosity l is defined as the net energy per second passing outward through a sphere at radius r and c_P is the specific heat. The Eqs.(3.2) to (3.4) describe the structure for a star at a given time and for a given composition $X_i(m)$. Changes of the composition with time due to nuclear reactions are taken into account by Eq. (3.5), where r_{ji} is the reaction rate for the reaction that transforms species j into species i. For realistic material functions (equation of state, opacities, nuclear reaction rates) it is not possible to solve this system of equations analytically, however, there are different ways to solve this problem numerically (Kippenhahn et al., 2012).

The following overview on nuclear burning in stars is mainly based on Iliadis (2015). A more detailed and general description can be found in Kippenhahn et al. (2012).

The evolution of stars is mainly determined by their initial mass at the *zero age main sequence* (ZAMS). This is due to the fact that the temperature and pressure in the core increase with the total mass of the star. Since nuclear reaction rates are very sensitive to temperature, this also means that the nuclear energy generation rate and the luminosity of the star increase with the stellar mass. However, the main sequence (i.e. hydrogen burning; see Sec. 3.1.1) lifetime of a massive star is much shorter, because the nuclear fuel is consumed at a faster rate. For instance, a $1M_\odot$ star spends around 10 Gyr on the main sequence, whereas a $15M_\odot$ star only spends 12 Myr on the main sequence. The initial mass of the star also determines which nuclear burning stages can be ignited in the core of the star. This leads to a wide range of evolutionary tracks with different outcomes depending on the initial mass of the star. For the purpose of this work we only distinguish between low mass stars ($M < 10M_\odot$) and massive stars ($M \gtrsim 10M_\odot$).

Due to the Gamow factor (2.9) thermonuclear fusion in stars always starts with light nuclei such as hydrogen and helium. Therefore, hydrogen burning (see Sec. 3.1.1) is always the first nuclear burning stage in a star. After the hydrogen fuel in the core has been exhausted, nuclear burning in the core is no longer possible and the core starts to contract under its own gravity. At the same time the envelope of the star expands due to the virial theorem and the star becomes a red giant. However, hydrogen fusion may still continue in a shell around the core where hydrogen is still present. This shell burning further increases the mass of the core and accelerates the contraction. This contraction leads to an increase in the core temperature and may ignite further burning stages, if the required temperature threshold for the next burning stage is reached.

Low mass stars ($M < 10M_\odot$) are unable to ignite all the nuclear burning stages that are available in a star. They usually either fail to ignite non-degenerate helium or carbon burning, because their core temperatures are not high enough. Consequently, those stars contract further and the electron gas in their core becomes degenerate at sufficiently high densities. Degeneracy is a consequence of the Pauli exclusion principle, which only allows up to two electrons in the same quantum state. This implies that further compression of the core is no longer possible, because all the lower energy levels are already occupied. In addition to that the pressure of a degenerate electron gas is not dependent on the temperature. Consequently, nuclear burning in a degenerate star does not lead to an expansion of the star. Eventually, the temperature in the degenerate core may become high enough to ignite the next burning stage. In a non-degenerate gas, the energy release from the nuclear reactions would lead to an expansion, which would decrease the temperature and thus the nuclear energy generation rate. However, this expansion does not occur in a degenerate gas and therefore the energy generation rate does not decrease. This results in a *thermonuclear runaway* and only stops after sufficient energy is released to lift the degeneracy. Eventually, the outer envelope of a low mass star is removed by stellar winds and the nuclear burning in the degenerate core stops. The degenerate core cools slowly and settles down into a helium, carbon-oxygen or oxygen-neon white dwarf. Since those stars do not reach silicon burning, they are not relevant for this thesis. Further details on the evolution of such stars are given in Kippenhahn et al. (2012).

Massive stars with $M \gtrsim 10 M_\odot$ can ignite all nuclear burning stages up to silicon burning (see Sec. 3.1.3 and 3.1.4) and develop an onion shell structure with iron group elements in the center and successively lighter elements in the outer regions, where the conditions for the later burning stages are not fulfilled anymore. It should be noted that the evolution of the star rapidly accelerates during the advanced burning stages (i.e. neon, oxygen and silicon burning) due to neutrino losses. For instance, silicon burning only lasts for a couple days, whereas hydrogen burning typically lasts several Myr in a massive star. Since isotopes in the iron region are the isotopes with the highest binding energy per nucleon, exothermic nuclear reactions are no longer possible after silicon burning and the iron core starts to contract and becomes degenerate. When the mass of the core exceeds the Chandrasekhar mass limit ($\approx 1.4 M_\odot$), the degenerate electron gas can no longer support itself against gravitational collapse, which eventually leads to a core-collapse supernova (Sec. 3.2). The remnants of such explosions are either neutron stars or black holes, depending on the initial mass of the star and the mass loss during the stellar evolution.

In the following subsections, we discuss the nuclear burning processes for each of the burning stages in a massive star based on Kippenhahn et al. (2012). A detailed discussion of the nuclear processes and network calculations for each burning stage can be found in Iliadis (2015).

3.1.1 Hydrogen burning

For the purpose of motivating our study and providing the necessary context, we here give a short overview of hydrogen burning and helium burning based on Kippenhahn et al. (2012). This discussion mainly serves as an example to introduce the concepts of nuclear burning in stars and as a comparison for the more advanced burning stages.

During *hydrogen burning* four hydrogen atoms (^{1}H) are fused into ^{4}He. For this fusion two protons need to be transformed into neutrons via β^+-decays, which leads to the emission of two neutrinos (see Sec. 2.2.2). The cross section for a direct interaction of four protons is far too low to explain the luminosity of stars during hydrogen burning. Therefore, this fusion can only be achieved by a sequence of several two-body interactions. During hydrogen burning there are two main channels by which this reaction is mediated, the *proton-proton chain* (pp chain) and the *CNO cycle*.

The pp chain starts with the fusion of two hydrogen atoms into deuterium (^{2}He). In this reaction, one of the hydrogen atoms has to experience a β^+-decay during the closest encounter. Therefore, this reaction has a very small cross section. Afterwards, the deuterium from this reaction can interact with another proton to form ^{3}He:

$$\begin{aligned} {}^1\mathrm{H} + {}^1\mathrm{H} &\longrightarrow {}^2\mathrm{H} + \mathrm{e}^+ + \nu, \\ {}^2\mathrm{H} + {}^1\mathrm{H} &\longrightarrow {}^3\mathrm{He} + \gamma. \end{aligned} \tag{3.7}$$

From this point onwards there are three possible channels to produce ^{4}He. The first one (pp1) involves two ^{3}He nuclei, so the reactions in Eq. (3.7) have to be completed twice:

$$\text{pp1:} \quad {}^3\mathrm{He} + {}^3\mathrm{He} \longrightarrow {}^4\mathrm{He} + 2\,{}^1\mathrm{H}. \tag{3.8}$$

The other two channels require that ^{4}He is already present, which may either be produced as part of this burning process or during the nucleosynthesis in the early Universe. The reaction of ^{3}He with ^{4}He produces ^{7}Be, which can either react with an electron (pp2) or a proton (pp3).

$$\begin{aligned} &{}^3\mathrm{He} + {}^4\mathrm{He} \longrightarrow {}^7\mathrm{Be} + \gamma \\ &\text{pp2:} \begin{cases} {}^7\mathrm{Be} + \mathrm{e}^- & \longrightarrow {}^7\mathrm{Li} + \nu \\ {}^7\mathrm{Li} + {}^1\mathrm{H} & \longrightarrow {}^4\mathrm{He} + {}^4\mathrm{He} \end{cases}, \\ &\text{pp3:} \begin{cases} {}^7\mathrm{Be} + {}^1\mathrm{H} & \longrightarrow {}^8\mathrm{B} + \gamma \\ {}^8\mathrm{B} & \longrightarrow {}^8\mathrm{Be} + \mathrm{e}^+ + \nu \\ {}^8\mathrm{Be} & \longrightarrow {}^4\mathrm{He} + {}^4\mathrm{He} \end{cases}. \end{aligned} \tag{3.9}$$

Assuming a composition of $X(^1\mathrm{H}) = X(^4\mathrm{He}) = 0.5$ the pp1 channel is favored at low temperatures ($T < 1.8 \times 10^7\,\mathrm{K}$) (Iliadis, 2015). With increasing temperature the dominant channel shifts first towards the pp2 channel (between $1.8 \times 10^7\,\mathrm{K}$ and $2.5 \times 10^7\,\mathrm{K}$) and later to the pp3 channel ($\mathrm{T}{>}2.5 \times 10^7\,\mathrm{K}$), if ^{4}He is already present.

The CNO cycle is a catalytic cycle of reactions that involves some isotopes of C, N and O to fuse four hydrogen atoms into one ^{4}He atom. A detailed discussion of the nuclear reactions during this cycle is given in Kippenhahn et al. (2012) and Iliadis (2015). The bottleneck of this reaction cycle is the proton capture on ^{15}N and therefore the energy generation rate of the overall CNO cycle depends mainly on the reaction rate of this particular reaction. The temperature sensitivity of the CNO cycle is much higher than for the pp chain. Therefore, the pp chain dominates at temperatures below $1.5 \times 10^7\,\mathrm{K}$, whereas the CNO cycle dominates at higher temperatures.

3.1.2 Helium burning

In this subsection, we give a short overview of helium burning based on the description in Kippenhahn et al. (2012). The required temperatures ($T \gtrsim 1 \times 10^8\,\mathrm{K}$) for *helium burning* are already much higher than for hydrogen burning due to the higher Coulomb barrier in the Gamow factor. The key reaction during helium burning is the formation of ^{12}C from three ^{4}He nuclei through the so-called *triple α reaction*. Since 3-body encounters are rather unlikely this reaction is performed in two sub steps:

$$\begin{aligned} {}^4\mathrm{He} + {}^4\mathrm{He} &\rightleftharpoons {}^8\mathrm{Be}, \\ {}^8\mathrm{Be} + {}^4\mathrm{He} &\longrightarrow {}^{12}\mathrm{C} + \gamma. \end{aligned} \tag{3.10}$$

In the first reaction two α particles temporarily form a ^{8}Be nucleus, which is unstable and decays back into two α particles after roughly 10^{-16} s (Kippenhahn et al., 2012). A small concentration of ^{8}Be builds up until a reaction rate equilibrium (compare Sec. 2.2.1) between ^{4}He and ^{8}Be is established. A second α capture on ^{8}Be then leads to ^{12}C. During helium burning further α captures on ^{12}C and the resulting heavier isotopes can occur. However, substantial amounts of ^{12}C are still left after helium burning. This leads to the conclusion that α captures on ^{12}C and heavier isotopes during helium burning are rather slow. Unfortunately the reaction rate for α captures on ^{12}C is still rather uncertain (Kunz et al., 2002; An et al., 2016) and the relative abundances of ^{12}C and ^{16}O at the end of helium burning are very sensitive to this rate (Iliadis, 2015). This ratio between ^{12}C and ^{16}O at the end of helium burning also influences the nucleosynthesis in later burning stages and the evolution of massive stars.

3.1.3 Advanced burning stages

The following overview of the advanced burning stages from carbon burning to oxygen burning is mainly based on Iliadis (2015). *Carbon burning* takes place at temperatures $T = 0.5 - 1.0\,\mathrm{GK}$. This burning stage is initiated by the fusion of two ^{12}C nuclei into a compound ^{24}Mg nucleus, since this is the reaction with the lowest Coulomb barrier in an environment of ^{12}C and ^{16}O nuclei. The resulting ^{24}Mg nucleus is highly excited and decays mainly by the emission of light particles leading to reactions such as:

$$\begin{aligned} {}^{12}\mathrm{C} + {}^{12}\mathrm{C} &\longrightarrow {}^{23}\mathrm{Na} + \mathrm{p}, \\ {}^{12}\mathrm{C} + {}^{12}\mathrm{C} &\longrightarrow {}^{20}\mathrm{Ne} + \alpha, \\ {}^{12}\mathrm{C} + {}^{12}\mathrm{C} &\longrightarrow {}^{23}\mathrm{Mg} + \mathrm{n}. \end{aligned} \tag{3.11}$$

The light particles resulting from these reactions are immediately consumed in secondary reactions with ^{12}C or ^{16}O. Since so many different reactions are involved, the situation during carbon burning is already very complicated. For a detailed analysis of the nucleosynthesis during carbon burning it is necessary to apply an appropriate nuclear network. According to network calculations in Iliadis (2015) the most abundant species at the end of carbon burning are ^{16}O, ^{20}Ne, ^{24}Mg and ^{23}Na.

One would assume that the next nuclear burning stage is oxygen burning with the fusion of two ^{16}O nuclei. However, at temperatures above $T = 1\,\mathrm{GK}$ photodisintegration reactions become important. ^{20}Ne has a relatively small α particle separation energy of 4.73 MeV and is therefore the first nucleus that is subject to photodisintegration. The photodisintegration of ^{20}Ne is endothermic, but the α particles enable secondary reactions with ^{20}Ne, ^{23}Na or ^{24}Mg. Some of these α-particle-induced reactions release protons and neutrons, which can also contribute in further reactions. This network of reactions resulting from the photodisintegration of ^{20}Ne is called *neon burning*. Neon core burning takes place at temperatures $T = 1.2 - 1.8\,\mathrm{GK}$. The most important reactions during neon burning are

$$
\begin{aligned}
{}^{20}\mathrm{Ne} + \gamma &\longrightarrow {}^{16}\mathrm{O} + \alpha, \\
{}^{20}\mathrm{Ne} + \alpha &\longrightarrow {}^{24}\mathrm{Mg} + \gamma, \\
{}^{24}\mathrm{Mg} + \alpha &\longrightarrow {}^{28}\mathrm{Si} + \gamma.
\end{aligned}
\tag{3.12}
$$

The most abundant nuclei at the end of neon burning are ^{16}O, ^{24}Mg, and ^{28}Si.

Oxygen burning is similar to carbon burning, since it also involves the fusion of two heavy nuclei (^{16}O + ^{16}O). However, the resulting ^{32}S compound nucleus is even more excited than the ^{24}Mg compound nucleus. Consequently, there are more exit channels than during carbon burning. The most common primary reactions are the following:

$$
\begin{aligned}
{}^{16}\mathrm{O} + {}^{16}\mathrm{O} &\longrightarrow {}^{31}\mathrm{P} + \mathrm{p}, \\
{}^{16}\mathrm{O} + {}^{16}\mathrm{O} &\longrightarrow {}^{30}\mathrm{Si} + 2\,\mathrm{p}, \\
{}^{16}\mathrm{O} + {}^{16}\mathrm{O} &\longrightarrow {}^{28}\mathrm{Si} + \alpha, \\
{}^{16}\mathrm{O} + {}^{16}\mathrm{O} &\longrightarrow {}^{24}\mathrm{Mg} + 2\,\alpha, \\
{}^{16}\mathrm{O} + {}^{16}\mathrm{O} &\longrightarrow {}^{30}\mathrm{P} + {}^{2}\mathrm{H}, \\
{}^{16}\mathrm{O} + {}^{16}\mathrm{O} &\longrightarrow {}^{31}\mathrm{S} + \mathrm{n}.
\end{aligned}
\tag{3.13}
$$

Afterwards, the liberated light particles are once again consumed in secondary reactions, which results in a big network of nuclear reactions. The temperatures during oxygen burning are $T = 1.5 - 2.7\,\mathrm{GK}$. Iliadis (2015) showed that photodisintegrations of ^{16}O, ^{24}Mg and ^{28}Si are not relevant at these temperatures. During this burning stage the neutron excess increases significantly due to β^+-decays (e.g. ${}^{31}\mathrm{S}(\mathrm{e}^+\nu){}^{31}\mathrm{P}$ and ${}^{30}\mathrm{P}(\mathrm{e}^+\nu){}^{30}\mathrm{Si}$) and electron captures (e.g. ${}^{33}\mathrm{S}(\mathrm{e}^-,\nu){}^{33}\mathrm{P}$, ${}^{35}\mathrm{Cl}(\mathrm{e}^-,\nu){}^{35}\mathrm{S}$ and ${}^{37}\mathrm{Ar}(\mathrm{e}^-,\nu){}^{37}\mathrm{Cl}$). It is noteworthy that the first quasi-equilibrium cluster (see Sec. 2.2.1) centered around ^{28}Si starts to form towards the end of oxygen burning. The network calculations in Iliadis (2015) show that the most abundant species at the end of oxygen burning are ^{28}Si, ^{32}S, ^{38}Ar, ^{34}S, ^{36}Ar and ^{40}Ca.

3.1.4 Silicon burning

The discussion of silicon burning in this subsection is based on Iliadis (2015). *Silicon burning* is once again a photodisintegration rearrangement process comparable to neon burning, but on a much larger scale. Fusion reactions of two ^{28}Si nuclei are not possible due to the Coulomb barrier. Since ^{32}S is more susceptible to photodisintegrations this nucleus is destroyed first providing α particles and protons for secondary reactions. With increasing temperature the photodisintegration of ^{28}Si also becomes relevant. The resulting nucleosynthesis is very complex and it requires a large nuclear network to capture all the relevant processes. The composition of the resulting iron peak nuclei is very sensitive to the initial neutron excess. After the iron peak nuclei have formed, the neutron excess increases further due to electron captures on e.g. ^{53}Mn, ^{54}Fe, ^{55}Fe, ^{55}Co and ^{56}Co. Typical temperatures for silicon core burning are $T = 2.8 - 4.1\,\mathrm{GK}$.

During silicon burning two quasi-equilibrium clusters form in the regions $A = 25 - 44$ (silicon QSE group) and $A = 46 - 64$ (iron peak QSE group). At the start of silicon burning (after 0.01 s in the network calculations of Iliadis (2015)) only the first QSE group centered around silicon is present. As time goes on (at around 1.0 s) a second QSE cluster forms in the region $A = 50 - 67$ and later on (at around 100 s) those two clusters are connected. Towards the end of silicon burning full NSE of all strong and electromagnetic interactions is obtained (see Sec. 2.2.1).

A close look at the decay constants λ_α and λ_γ (in units of s^{-1}) of the nuclei in the α-chain centered around ^{28}Si reveals further details on the driving mechanism behind silicon burning. The decay constant for the α separation of ^{28}Si ($\lambda_\gamma = 0.14\,\mathrm{s}^{-1}$) is considerably smaller than the decay constants for other α separations in the chain. Meanwhile the α capture on ^{24}Mg back to ^{28}Si is much more likely than another photodisintegration into ^{20}Ne. Therefore, the equilibrium between ^{28}Si and ^{24}Mg is quickly established. Similarly, the photodisintegration of ^{32}S into ^{28}Si is more likely than α captures to ^{36}Ar. Hence the equilibrium between ^{28}Si and ^{32}S is also quickly established. However, nuclides lighter than ^{24}Mg are not in equilibrium with ^{28}Si, since photodisintegrations of those isotopes are more likely than α captures. Due to the slow rate of ^{28}Si decomposition it is reasonable to assume that the abundances of those light α nuclides are small in comparison to the ^{28}Si abundance. This leads to the conclusion that those isotopes are in steady state, i.e. the flow into those nuclei is balanced by the flow out. Under these assumptions it is possible to obtain an analytical expression for the effective rate of ^{28}Si consumption, which is comparable to the rate of ^{28}Si consumption obtained from nuclear network calculations (Iliadis, 2015). While the above considerations are helpful to get a fundamental understanding of the physics behind silicon burning, one should be aware that these simplifications do not capture the full picture of this complex burning process.

In stellar evolution calculations and hydrodynamic simulations of silicon burning it is not feasible to include a full nuclear network, which describes all aspects of the nucleosynthesis accurately. However, a rough estimate of the nuclear energy release and the neutronization during silicon can be obtained with a smaller nuclear reaction network. In this thesis, we use a 21-isotope α-chain reaction network, which uses an effective rate treatment. This reaction network is described in Sec. 4.2.2.

3.2 Core-collapse supernovae

In this section, we give a short overview of the relevant processes during core-collapse supernovae based on Iliadis (2015). More details on the theory and state-of-the-art modeling of core-collapse supernovae can be found in Janka et al. (2007, 2016) and Müller (2016).

Core-collapse supernovae are explosions towards the end of a massive star's lifetime. They are powered by the release of gravitational energy during the collapse of the core into a proto-neutron star. After silicon burning in the core has subsided the core still grows due to silicon shell burning until it reaches the Chandrasekhar mass ($M_{\mathrm{ch}} \approx 1.4 M_\odot$). At this point the degeneracy pressure of the electrons is no longer enough to support the core against gravitational collapse. Since thermonuclear burning is no longer possible, the core collapses under its own gravity. Typical temperatures and densities at the onset of core collapse are $T \approx 10\,\mathrm{GK}$ and $\rho \approx 10^{10}\,\mathrm{g\,cm^{-3}}$.

3.2.1 Core collapse and neutrino-driven shock revival

The following discussion of the core collapse and neutrino-driven shock revival is based on Iliadis (2015) and Müller (2016).

The collapse of the core accelerates due to electron captures at high densities, which further decreases the electron fraction Y_e and thus the stabilizing degeneracy pressure of the electrons. At the same time the electron captures also lead to a massive burst of electron neutrinos. In addition to that, the high temperatures in the collapsing core lead to thermal disintegrations of iron peak nuclei into lighter less stable nuclei. Due to those two effects the core quickly collapses from several thousand kilometers to a proto-neutron star with around 10 km radius.

After $t \approx 0.1\,\mathrm{s}$ the densities in the core are so high ($\rho \approx 10^{12}\,\mathrm{g\,cm^{-3}}$) that neutrinos are trapped inside the *neutrino sphere* R_ν, since their diffusion time becomes longer than the collapse time scale (Bethe, 1990). Meanwhile, neutrinos outside the neutrino sphere escape almost freely with an average energy that depends on R_ν. The collapse of the core proceeds further until it reaches nuclear densities ($\rho \approx 10^{14}\,\mathrm{g\,cm^{-3}}$) after $t \approx 0.11\,\mathrm{s}$. Since it cannot be compressed further, the core bounces and drives a shock wave into the surrounding matter. The shock propagates outwards and dissociates iron peak nuclei into free nucleons, which leads to substantial energy losses ($\approx 9\,\mathrm{MeV}$ per nucleon). As the shock approaches the neutrino sphere, electron captures on free protons also contribute to the energy loss of the shock while giving rise to another burst of electron neutrinos. Consequently, the shock stalls after $t \approx 0.2\,\mathrm{s}$ at a radius around 100 km to 200 km.

In the *delayed neutrino-driven mechanism* (e.g. Bethe and Wilson, 1985; Wilson, 1985) this stalled accretion shock is revived by neutrinos that emerge from the neutrino sphere of the proto-neutron star. The region between the neutrino sphere and the stalled shock is divided into two parts by the *gain radius*. The first region between the neutrino sphere and the gain radius is dominated by neutrino cooling through neutrino emission due to $\mathrm{p}\,(\mathrm{e}^-, \nu_\mathrm{e})\,\mathrm{n}$ and $\mathrm{n}\,(\mathrm{e}^+, \bar{\nu}_\mathrm{e})\,\mathrm{p}$. In the second region between the gain radius and the shock there is mainly neutrino heating due to neutrino absorption by the reverse processes $\mathrm{n}\,(\nu_\mathrm{e}, \mathrm{e}^-)\,\mathrm{p}$ and $\mathrm{p}\,(\bar{\nu}_\mathrm{e}, \mathrm{e}^+)\,\mathrm{n}$. This energy deposition in the *gain region* may revive the shock, which then leads to a supernova explosion. However, this mechanism depends crucially on the product of the neutrino luminosity and the neutrino interaction cross section, which both depend on the average neutrino energy.

The following overview of recent advances in the multidimensional modeling of core-collapse supernovae is based on Müller (2016). It has been shown in 1D simulations of the post-bounce phase that neutrino-driven explosions cannot be achieved in spherical symmetry (see e.g. Rampp and Janka, 2000, 2002; Liebendörfer et al., 2001, 2004, 2005). Successful neutrino-driven explosions have been obtained in sophisticated 2D models that also take hydrodynamic instabilities into account. Particular important for shock revival in 2D are convective instabilities in the gain region (e.g. Herant et al., 1994;

Burrows et al., 1995; Janka and Müller, 1996) and the so-called *standing accretion shock instability* (SASI; e.g. Blondin et al., 2003; Foglizzo et al., 2007; Laming, 2007; Guilet and Foglizzo, 2012). This instability is characterized by sloshing motions, which saturate and later develop into large-scale spiral motions. The development of these instabilities relies crucially on multidimensional phenomena, which may already be seeded during the silicon burning phase in the progenitor. The main problem with the recent generation of 2D models (Janka et al., 2012; Nakamura et al., 2015; O'Connor and Couch, 2015) is that the explosion energy grows slowly after shock revival. As such, they are unable to reproduce typical explosion energies of observed supernovae.

It is possible that this *energy problem* can be resolved by fully consistent 3D simulations of the core collapse. After initial problems with shock revival in 3D, it is now possible to produce successful explosions in 3D (Melson et al., 2015; Lentz et al., 2015). However, the fact remains that 3D simulations are clearly less likely to explode than their 2D counterparts; they either barely fail to explode or explode later than in 2D. It was suggested by Hanke et al. (2012) that this might be due to the inverted turbulent energy cascade[1] in 2D, which leads to larger structures in 2D and might enhance shock revival. This suggests that there are still missing physics that could lead to more robust explosions in 3D. Some of these recent developments will be discussed in the next subsection.

3.2.2 Possible solutions for more robust explosions in 3D

Different approaches to obtain more robust explosions in 3D simulations of core-collapse supernovae are discussed in Müller (2016).

Nakamura et al. (2014) and Janka et al. (2016) suggested that rapid rotation might significantly reduce the critical luminosity that is required for shock revival. Janka et al. (2016) show that the pre-shock infall velocities may be reduced due to centrifugal forces. This also implies that the radius of the stalled shock increases and the required critical luminosity decreases in the presence of rapid rotation. However, stellar evolution calculations indicate that such a large amount of rotation is rather untypical for progenitors of Type II supernovae. For slowly rotating progenitors the reduction of the critical luminosity due to rotation is most likely negligible.

A different approach to achieve more robust explosions in 3D consists in increasing the neutrino luminosities and the mean energy for electron neutrinos. The neutrino luminosity mainly depends on the neutrino opacities, but other properties of the model, e.g. general relativity or the equation of state may also have an impact. Any changes in the neutrino luminosities often come with counterbalancing side effects (*Mazurek's law*). However, it has been shown that this can be circumvented under certain conditions (Melson et al., 2015). A thorough re-investigation of the neutrino rates may thus be worthwhile.

In recent studies the role of initial asymmetries in supernova progenitors has been investigated. These perturbations naturally arise during convective silicon and oxygen

---

[1] In our own simulations of silicon burning in 2D and 3D we also see clear signs of an inverted turbulent energy cascade in 2D (see Fig. 5.21).

shell burning (Arnett and Meakin, 2011; Müller et al., 2016). The role of asymmetries in the progenitor has previously been analyzed with parametrized initial conditions (Couch and Ott, 2013, 2015; Müller and Janka, 2015; Burrows et al., 2016), however, the results of these studies remained inconclusive. It was found that the impact of such perturbations on shock revival largely depends on the Mach numbers that develop during convective shell burning and the spatial scale of the perturbations. Therefore, 3D modeling of the progenitor during the last minutes before the onset of core collapse is imperative to obtain a realistic 3D structure of the progenitor, which can be used as starting point for simulations of the core collapse in 3D.

First attempts to model the 3D structure of the progenitor have been carried out by Couch et al. (2015). To this end, they evolved a $15M_\odot$ star from the pre-main sequence to the onset of core collapse in the 1D stellar evolution code MESA (Paxton et al., 2011, 2013, 2015). Afterwards, the model from the stellar evolution code was mapped to a 3D hydrodynamics code to follow the last 160 s of silicon burning. However, as discussed in Müller et al. (2016), they used several approximations that might affect the structure and magnitude of the perturbations. They simulated only one octant of the whole star and more importantly they artificially enhanced the electron capture rates to speed up the evolution towards collapse. The spatial scale of the perturbations might be affected by the restriction of the physical domain to one octant, which would have further implications on the dynamics of the core collapse. In addition to that, intermediate mass-elements in the silicon shell are burned to iron group elements within only 160 s in the 3D model of Couch et al. (2015) due to the modified reaction rates. This process takes roughly 1000 s in the corresponding stellar evolution model, which suggests that the convective velocities in the 3D model are artificially increased by more than 80 % (Müller et al., 2016). Moreover, convective activity might be artificially prolonged in shells that are almost fully burned. Couch et al. (2015) found that the shock expands slightly faster for the model with full 3D initial conditions than for the reference model with spherically averaged initial conditions. However, both models lead to an explosion and the dynamics of the core collapse might be modified due to the leakage scheme that was used for the neutrino transport. As such, it is still not conclusive, whether seed perturbations in the progenitor can enable shock revival.

The impact of seed perturbations on shock revival has been further investigated by Müller et al. (2017). They previously simulated the last five minutes of oxygen shell burning in 3D (Müller et al., 2016) for a $18M_\odot$ star in the mass region between $1.68M_\odot$ and $4.07M_\odot$. In their recent work they use these results as input for the simulation of the core collapse. For comparison they also simulated the core-collapse for a spherically symmetric model and another 3D progenitor model with reduced convective velocities, where they assumed a fixed temperature during the last five minutes of oxygen burning. They obtained shock revival for both models with 3D initial conditions, but the model with spherically symmetric initial conditions did not explode until the end of their simulation and the critical timescale ratio $\tau_{\mathrm{adv}}/\tau_{\mathrm{heat}}$ was well below the runaway threshold. This suggests that large scale perturbations in the progenitor may be beneficial for shock revival in some cases. Müller et al. (2017) inferred a reduction of the critical luminosity by about 22 % for the full 3D initial conditions compared to the model with 1D initial conditions. For the model with reduced convective velocities they still obtained a reduction of the critical luminosity by 16 %.

They continued the simulation of the explosion for the model with full 3D initial conditions until 2.35 s after bounce. However, the cycle of accretion and mass ejection was still ongoing, so they were unable to obtain final values for the explosion energy, nickel mass and the properties of the proto-neutron star. At the end of their simulation the diagnostic explosion energy reaches a value of 7.7×10^{50} erg. They conclude that 5×10^{50} erg is a relatively safe lower limit for the final explosion energy, which would be within the range of observed explosion energies, but slightly below average. As such, their simulation still does not represent a typical Type II supernova, but it is a step in the right direction. They emphasize that their $18 M_\odot$ progenitor has particularly suitable conditions for shock revival through seed perturbations. Such conditions are not necessarily fulfilled in all supernova progenitors and there is still a large parameter space to explore.

Since the oxygen shell burning simulations of Müller et al. (2016) only cover the mass region between $1.68 M_\odot$ and $4.07 M_\odot$, data from the 1D stellar evolution model is used outside of this domain. Müller et al. (2017) mention that patching together the 3D simulation of the oxygen shell burning with the core of the 1D stellar evolution model leads to hydrostatic readjustments, which slightly reduce the mass accretion rate and the electron flavor luminosity. This might lead to an reduction of the explosion energy, which could be avoided by consistent modeling of the whole core of the supernova progenitor including both the silicon burning region and the oxygen burning region.

In a recent publication of Collins et al. (2018) the properties of convective oxygen and silicon burning shells in supernova progenitors are studied for a large set of non-rotating progenitors with zero-age main sequence masses ranging from $9.45 M_\odot$ to $35 M_\odot$. They find that thick convective shells (angular wave number $l \leq 5$) with large convective Mach numbers are generally more common for convective oxygen shells than for convective silicon burning shells. Favorable conditions for perturbation-aided explosions, i.e. high convective Mach numbers ~ 0.15 and medium-scale convective modes with $l < 10$, are rarely found in convective silicon burning shells. However, most of these results are based on spherically symmetric stellar evolution models and the convective properties are inferred from mixing-length theory. Full 3D simulations of the most promising regimes for perturbation-aided explosions may thus be still worthwhile.

Changes in the progenitor structure due to convective boundary mixing during advanced burning stages may also influence the details of the core collapse (Müller, 2016). Thin, unburnt shells with a small density contrast might be swallowed up by neighboring convective zones due to entrainment. However, details on long term effects of entrainment and their impact on supernova progenitors are still rather uncertain.

3.3 Convection in stars

Small perturbations in a star may grow and give rise to macroscopic local motions under certain circumstances. We have discussed in the previous section that the growth of such perturbations might be crucial for shock revival in 3D simulations of core collapse supernovae. In this section, we are going to derive criteria for the stability of hydrostatic atmospheres in stars (based on Kippenhahn et al., 2012, Sec. 6.1). For this purpose we define a fluctuation DA of a quantity A as

$$DA := A_\mathrm{e} - A_\mathrm{s}, \tag{3.14}$$

where the subscript e denotes a small “element” that has constant, but somewhat different values than in the “surroundings” (subscript s). We impose that $DP = 0$, since elements with $DP \neq 0$ would either expand or contract immediately to restore pressure balance with the surroundings.

Now we consider an element that has been lifted from its original position at r to $r + \Delta r$ with $\Delta r > 0$. The density difference of that element in comparison to the surroundings then yields

$$D\rho = \left[\left(\frac{d\rho}{dr} \right)_\mathrm{e} - \left(\frac{d\rho}{dr} \right)_\mathrm{s} \right] \Delta r, \tag{3.15}$$

where $(d\rho/dr)_\mathrm{e}$ describes the change of the element’s density while it rises by dr and $(d\rho/dr)_\mathrm{s}$ is the spatial density gradient in the surroundings. A non-vanishing $D\rho$ leads to a buoyancy force $\boldsymbol{F}_\mathbf{b} = -gD\rho\mathbf{e}_{\boldsymbol{r}}$. As we can see the buoyancy force points upwards, if $D\rho < 0$, i.e. the element is lighter than its surroundings, which means this layer is unstable. Therefore, we have to impose $D\rho > 0$ to obtain a buoyancy force that points downwards and achieve stability. This leads to the following stability criterion:

$$\left(\frac{d\rho}{dr} \right)_\mathrm{e} - \left(\frac{d\rho}{dr} \right)_\mathrm{s} > 0. \tag{3.16}$$

To obtain a more practical stability criterion we assume that the element rises adiabatically, i.e. there is no exchange of energy with the surroundings. This is a good approximation deep in the interior of stars. We use the equation of state $\rho = \rho(P, T, \mu)$ in differential form

$$\frac{d\rho}{\rho} = \alpha \frac{dP}{P} - \delta \frac{dT}{T} + \varphi \frac{d\mu}{\mu} \tag{3.17}$$

to express the gradients in terms of T instead of ρ with the following partial derivatives

$$\alpha = \left(\frac{\partial \ln \rho}{\partial \ln P} \right), \qquad \delta = -\left(\frac{\partial \ln \rho}{\partial \ln T} \right), \qquad \varphi = \left(\frac{\partial \ln \rho}{\partial \ln \mu} \right). \tag{3.18}$$

These coefficients are defined in such a way that the partial derivatives are $\alpha = \delta = \varphi = 1$ in the case of an ideal gas with $\rho \sim P\mu/T$.

Eq. (3.17) can be used to rewrite Eq. (3.16) in the following way:

$$\left(\frac{\alpha}{P} \frac{dP}{dr} \right)_\mathrm{e} - \left(\frac{\delta}{T} \frac{dT}{dr} \right)_\mathrm{e} - \left(\frac{\alpha}{P} \frac{dP}{dr} \right)_\mathrm{s} + \left(\frac{\delta}{T} \frac{dT}{dr} \right)_\mathrm{s} - \left(\frac{\varphi}{\mu} \frac{d\mu}{dr} \right)_\mathrm{s} > 0. \tag{3.19}$$

The change of the chemical composition for the element has been omitted, since the moving element carries its composition along. The two terms involving pressure gradients are identical due to pressure equilibrium and cancel each other. The remaining terms are multiplied with the pressure scale height

$$H_P = -\frac{dr}{d\ln P} = -P\frac{dr}{dP}, \tag{3.20}$$

which yields the following condition for stability:

$$\left(\frac{d\ln T}{d\ln P}\right)_{\mathrm{s}} < \left(\frac{d\ln T}{d\ln P}\right)_{\mathrm{e}} + \frac{\varphi}{\delta}\left(\frac{d\ln \mu}{d\ln P}\right)_{\mathrm{s}}. \tag{3.21}$$

At this point we introduce the following derivatives:

$$\nabla = \left(\frac{d\ln T}{d\ln P}\right)_{\mathrm{s}}, \qquad \nabla_{\mathrm{e}} = \left(\frac{d\ln T}{d\ln P}\right)_{\mathrm{e}}, \qquad \nabla_{\mu} = \left(\frac{d\ln \mu}{d\ln P}\right)_{\mathrm{s}}. \tag{3.22}$$

In the case of an adiabatic process ∇_{e} can be replaced by ∇_{ad}, which leads to the *Ledoux criterion* of stability:

$$\nabla < \nabla_{\mathrm{ad}} + \frac{\varphi}{\delta}\nabla_{\mu}. \tag{3.23}$$

In the case of homogeneous chemical composition $\nabla_{\mu} = 0$, so that Eq. (3.23) reduces to the *Schwarzschild criterion*

$$\nabla < \nabla_{\mathrm{ad}}. \tag{3.24}$$

If the left-hand side is larger than the right-hand side in Eq. (3.23) and (3.24), the layer is dynamically unstable. In this case, small perturbations will increase and the region becomes convective.

The stability criteria in Eq. (3.23) and (3.24) can also be expressed in terms of the *Brunt-Väisälä frequency* N, which describes the adiabatic oscillation of an element in a dynamically stable layer:

$$N^2 = \frac{g\delta}{H_P}\left(\nabla_{\mathrm{ad}} - \nabla + \frac{\varphi}{\delta}\nabla_{\mu}\right) > 0. \tag{3.25}$$

If the stability criterion is not fulfilled the Brunt-Väisälä frequency is imaginary, which means that the element moves away exponentially instead of oscillating. Consequently, the layer is dynamically unstable.

4 Numerical methods

4.1 Computational fluid dynamics on curvilinear grids

In this section, we discuss the discretization of the hydrodynamic equations in *SLH* (see Sec. 1.1.3). The finite-volume discretization is introduced in Sec. 4.1.1, while the time-stepping methods in *SLH* are discussed in Sec. 4.1.2. A particular noteworthy feature in *SLH* is the transformation of the hydrodynamic equations to a general curvilinear grid, which allows us to adapt the physical grid to the geometry of the problem while keeping the computational grid uniform and Cartesian. An overview of this method is given in Sec. 4.1.3.

4.1.1 Finite-volume methods

For simplicity the following derivation of the finite-volume discretization is discussed in 1D. The extension of this approach to more spatial dimensions is straightforward. In *SLH* the hydrodynamic equations are discretized with a finite-volume approach (see e.g. LeVeque, 1998). This means that Q_i^n is not an approximation to a single value $q(x_i, t^n)$, but rather an average of q over an interval $C_i = [x_{i-1/2}, x_{i+1/2}]$ of length $h = \Delta x$:

$$Q_i^n \approx \frac{1}{h} \int\limits_{x_{i-1/2}}^{x_{i+1/2}} q(x, t^n) dx \equiv \frac{1}{h} \int_{C_i} q(x, t^n) dx \tag{4.1}$$

In this notation, integer-values refer to cell centers, whereas half-integer indices represent cell interfaces. The formulation of the finite-volume scheme is now obtained by using the integral form of the balance equation (2.2):

$$\int_{C_i} q(x, t^{n+1}) dx = \int_{C_i} q(x, t^n) dx - \int\limits_{t^n}^{t^{n+1}} f\left[q\left(x_{i+1/2}, t\right)\right] - f\left[q\left(x_{i-1/2}, t\right)\right] dt. \tag{4.2}$$

Dividing by h and applying the definition of the cell averages from Eq. (4.1) leads to the *flux differencing form*:

$$Q_i^{n+1} = Q_i^n - \frac{k}{h}\left(F_{i+1/2}^n - F_{i-1/2}^n\right), \quad k = \Delta t, \tag{4.3}$$

where $F_{i\pm1/2}^n$ is an approximation of the flux over the interface at $i \pm 1/2$ during the time interval between t^n and t^{n+1}. This *numerical flux function* is computed via:

$$F_{i\pm1/2}^n \approx \frac{1}{k} \int\limits_{t^n}^{t^{n+1}} f\left[q\left(x_{i\pm1/2}, t\right)\right] dt, \tag{4.4}$$

which requires an estimate of the flux over the interfaces, since $q(x_{i\pm1/2}, t)$ is not known exactly in the interval from t^n to t^{n+1}. This finite-volume discretization guarantees that the quantities on the numerical grid are conserved in the same way as the true solution, except for fluxes over the grid boundary and source terms.

Estimates of the flux functions $F^n_{i\pm1/2}$ at $t > t^n$ are often based on solving the *Riemann problem* at the cell interfaces. The Riemann problem is an initial-value problem for a conservation law with piecewise constant left and right states q_l and q_r (cf. LeVeque, 1998):

$$q(x,0) = \begin{cases} q_\mathrm{l} & \text{for } x < 0 \\ q_\mathrm{r} & \text{for } x > 0 \end{cases}. \tag{4.5}$$

For example in *Godunov's method* (Godunov, 1959) a piecewise constant function is reconstructed using the cell averages. In numerical applications the resulting Riemann problem is typically solved using approximate Riemann solvers instead of the full solution to the Riemann problem. A popular method that is also used in *SLH* is the so called *Roe solver* (Roe, 1981), which assumes a linear system of conservation laws with constant coefficients instead of the full nonlinear problem. With suitable preconditioning this scheme can be extended to treat low Mach number flows (Miczek, 2013; Miczek et al., 2015). The accuracy of the scheme can be improved by replacing the piecewise constant reconstruction with higher-order reconstructions (e.g. linear or parabolic). However, those higher-order methods often lead to oscillations and overshooting at discontinuities (Press et al., 2007).

In the presence of gravity source terms, it is not trivial to preserve the hydrostatic equilibrium exactly, because the gravity source term $\rho\boldsymbol{g}$ is discretized independently of the pressure gradient $\boldsymbol{\nabla}P$ in the Euler equations (2.4). Consequently, the condition for hydrostatic equilibrium $\boldsymbol{\nabla}P = \boldsymbol{g}$ (cf. Eq. 2.31) is not fulfilled on a discrete level, which means that unstabilities may still develop even if the initial setup was in perfect hydrostatic equilibrium. For instance in Miczek (2013) and Edelmann (2014), it was shown that a checkerboard-like instability develops, if low Mach preconditioning (see Sec. 1.1.3) is switched on. Those problems with the gravity source term can be avoided by using a *well-balanced* scheme, which guarantees precise balance of the pressure gradient and the gravitational forces in case of the hydrostatic solution. Examples of well-balanced schemes are given in Cargo and Le Roux (1994), Edelmann (2014) and Chandrashekar and Klingenberg (2015).

4.1.2 Time-Stepping

In *SLH* the time discretization is performed separately from the spatial discretization,[1] which was described in the previous section. This approach allows flexible combinations of spatial discretization techniques with different time-stepping methods. In particular it is possible to choose either an *explicit* or an *implicit* time-discretization depending on the expected Mach numbers. To illustrate some concepts of time discretization schemes

---

[1] This separation of the spatial and temporal discretization is also known as *method of lines*.

we are going to consider the following differential equation:

$$\frac{\partial u}{\partial t} = h(u, x, t), \tag{4.6}$$

where $h(u, x, t)$ represents the spatial discretization.

Explicit time-stepping

Explicit numerical schemes only use information from previous time-steps, i.e. $t \leq t^n$, to obtain the solution $u(x, t^{n+1})$ for the next time-step. A straightforward time integration of Eq. (4.6) leads to the explicit forward Euler method, which is first-order accurate in time:

$$u_j^{n+1} = u_j^n + h^n \Delta t. \tag{4.7}$$

Higher-order explicit methods can be constructed by evaluating the spatial derivatives at additional trial points within Δt (*Runge-Kutta methods*). With a proper choice of these trial points the lower order error terms can be eliminated successively and higher-order methods are obtained (see e.g. Press et al., 2007). The explicit Runge-Kutta method that is implemented in *SLH* is third order accurate (RK3) and is described in Shu and Osher (1988).

The major drawback of explicit numerical schemes is that their time-step is restricted by the Courant-Friedrichs-Lewy (CFL) criterion for stability (Courant et al., 1928). This criterion requires that the physical domain of dependence at the new time-step must be included in the numerical domain of dependence. In the case of the hydrodynamic Euler equations the CFL criterion is given by:

$$\Delta t_{\mathrm{CFL}} = \mathrm{CFL}_{uc} \frac{\Delta x}{|u| + c_{\mathrm{s}}}, \tag{4.8}$$

where u is the local fluid velocity, c_{s} is the speed of sound, Δx is the grid spacing and CFL_{uc} is a constant of order one that depends on the specific spatial and temporal discretization. In the case of low Mach numbers ($|u| \ll c_{\mathrm{s}}$) the CFL criterion is dominated by the sound speed, however, the physical processes of interest occur at the timescale of the fluid velocity. As such, explicit methods are impractical in the low-Mach regime.

Implicit time-stepping

The restrictions of the CFL criterion can be avoided by using implicit time-integration. In this case, the spatial residual $h(u, x, t)$ of Eq. (4.6) is evaluated at the new time-step t^{n+1}, for which the solution $u(x, t^{n+1})$ is not yet known. The simplest implicit scheme is the *backward Euler* method:

$$u_j^{n+1} = u_j^n + h^{n+1} \Delta t. \tag{4.9}$$

However, since h^{n+1} depends on the solution at the new time-step t^{n+1} a straightforward computation of u_j^{n+1} is not possible for time-implicit schemes. Therefore, implicit time-stepping for the Euler equations involves the solution of a non-linear system of equations. In *SLH* a Newton-Raphson method is applied to obtain the solution at t^{n+1}. This reduces the solution of the original non-linear system of equations to an iterative solution of linear systems of equations. The resulting linear systems of equations are then solved with iterative sparse-matrix solvers such as BiCGSTAB(l) or GMRES(r) (see Sec. 4.3.5).

Consequently, implicit methods are considerably more demanding than explicit methods in terms of computation time and memory usage per time-step. However, in the case of low Mach numbers this is compensated due to the fact that the time-step is not restricted by the CFL criterion (4.8). Since the spatial residual h^{n+1} is computed from the neighboring cells at time-step t^{n+1}, which in turn depend on their neighboring cells, the whole domain is now connected. Therefore, the numerical domain of dependence is the whole computational domain.

The backward Euler method (4.9) is again only first-order accurate in time, but higher-order Runge-Kutta methods can also be constructed for implicit time-integration. However, in this case several non-linear systems of equations need to be solved for each implicit stage in the Runge-Kutta scheme, which would be prohibitively expensive. Fortunately, the coefficients of the implicit Runge-Kutta scheme can be chosen in a way that the computational effort can be reduced significantly. More detailed descriptions of those *Explicit first stage Singly Diagonally Implicit Runge-Kutta* (ESDIRK) methods and the time-stepping in *SLH* can be found in Miczek (2013) and Edelmann (2014). The following implicit Runge-Kutta schemes are implemented in *SLH*: ESDIRK23, ESDIRK34, ESDIRK46, ESDIRK58, where the first digit is the formal order of the method and the second digit is the number of stages in the Runge-Kutta scheme.

Even though the CFL criterion (4.8) does not need to be fulfilled for implicit methods, there are still constraints on the time-step. A common choice is the *advective CFL criterion*, which imposes that the fluid moves roughly one cell per time-step:

$$\Delta t_{\mathrm{CFL}} = \mathrm{CFL}_u \frac{\Delta x}{|u|}, \tag{4.10}$$

This criterion ensures that the properties of the flow are well resolved, but especially for low Mach number flows it is not as restrictive as the CFL_{uc} criterion, since it only depends on the local fluid flow and not on the sound speed. This implies that sound waves are damped, but in the case of low Mach number flows it is usually not necessary to resolve them due to the decoupling of sound waves and advective flow.

In the presence of gravity an alternative CFL-like criterion can be formulated using the free-fall time scale (Miczek, 2013). Considering that in hydrostatic equilibrium gravity is almost completely balanced by pressure, this CFL_{ug} criterion is a bit too restrictive. However, this criterion also gives reasonable time-steps for initial conditions with zero velocity, where the CFL_u criterion would be singular.

4.1.3 Transformation to general coordinates

So far we have only considered a uniform Cartesian grid for the formulation of the hydrodynamic equations. However, in *SLH* it is possible to construct arbitrary curvilinear grids with a coordinate transformation to general coordinates. With this feature the physical grid can be adapted to the geometry of the problem, e.g. polar or spherical coordinates, while all the numerical computations are computed on a uniform Cartesian grid by taking the metric terms from the coordinate transformation into account. This concept of transforming the hydrodynamic Euler equations to general curvilinear coordinates was first used in an astrophysical context by Kifonidis and Müller (2012) and adopted into *SLH* by Miczek (2013).

For this purpose the curvilinear coordinates $\xi(x, y, z)$, $\eta(x, y, z)$ and $\zeta(x, y, z)$ are defined via a coordinate transformation between the global Cartesian coordinate system (x, y, z) and the general curvilinear grid (ξ, η, ζ). Without loss of generality the coordinate transformation may always be chosen so that the cell widths on the computational grid are uniform, i.e $\Delta\xi = \Delta\eta = \Delta\zeta = 1$. The functions that define this transformation can be chosen arbitrarily. The only constraints are that the transformation needs to be invertible and the second derivatives of the transformation have to commute. In principle the new coordinate system does not even need to be orthogonal. The existence of second derivatives implies that the transformation needs to be continuously differentiable. However, this requirement is not necessarily fulfilled for all grid geometries, which may lead to numerical artifacts (cf. Sec. 5.1.2).

The Jacobian determinants for this coordinate transformation and the corresponding inverse transformation are then defined as:

$$J = \begin{vmatrix} \xi_{,x} & \xi_{,y} & \xi_{,z} \\ \eta_{,x} & \eta_{,y} & \eta_{,z} \\ \zeta_{,x} & \zeta_{,y} & \zeta_{,z} \end{vmatrix}, \qquad J^{-1} = \begin{vmatrix} x_{,\xi} & x_{,\eta} & x_{,\zeta} \\ y_{,\xi} & y_{,\eta} & y_{,\zeta} \\ z_{,\xi} & z_{,\eta} & z_{,\zeta} \end{vmatrix}. \tag{4.11}$$

In the following, we illustrate the transformation of the Euler equations to general coordinates. For this purpose we reformulate Eq. (2.4) as (cf. Kifonidis and Müller, 2012; Miczek, 2013):

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{F}_x(\boldsymbol{Q})}{\partial x} + \frac{\partial \boldsymbol{F}_y(\boldsymbol{Q})}{\partial y} + \frac{\partial \boldsymbol{F}_z(\boldsymbol{Q})}{\partial z} = \boldsymbol{S}(\boldsymbol{Q}), \tag{4.12}$$

where $\boldsymbol{Q}$ is the vector of the conserved variables, $\boldsymbol{F}_x$, $\boldsymbol{F}_y$ and $\boldsymbol{F}_z$ are the flux functions in the corresponding coordinate direction and $\boldsymbol{S}$ is a vector of source terms that are assumed to be non-stiff. In the case of the homogeneous Euler equations these vectors are given by:

$$\boldsymbol{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_{\text{tot}} \end{pmatrix}, \quad \boldsymbol{F}_x(\boldsymbol{Q}) = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho u v \\ \rho u w \\ \rho u e_{\text{tot}} + uP \end{pmatrix}, \quad \boldsymbol{F}_y(\boldsymbol{Q}) = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + P \\ \rho v w \\ \rho v e_{\text{tot}} + vP \end{pmatrix}, \tag{4.13}$$

with $\boldsymbol{S}(\boldsymbol{Q}) = 0$ and $\boldsymbol{F}_z(\boldsymbol{Q})$ is analogous to the other two flux functions $\boldsymbol{F}_x(\boldsymbol{Q})$ and $\boldsymbol{F}_y(\boldsymbol{Q})$. With the chain rule of differentiation we obtain expressions for the derivatives of the transformed fluxes, e.g.:

$$
\begin{aligned}
J^{-1}\frac{\partial \boldsymbol{F}_x}{\partial x} =& J^{-1}\frac{\partial \boldsymbol{F}_x}{\partial \xi}\xi_{,x} + J^{-1}\frac{\partial \boldsymbol{F}_x}{\partial \eta}\eta_{,x} + J^{-1}\frac{\partial \boldsymbol{F}_x}{\partial \zeta}\zeta_{,x} \\
=& \frac{\partial}{\partial \xi}\left(J^{-1}\xi_{,x}\boldsymbol{F}_x\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\eta_{,x}\boldsymbol{F}_x\right) + \frac{\partial}{\partial \zeta}\left(J^{-1}\zeta_{,x}\boldsymbol{F}_x\right) \\
& -\boldsymbol{F}_x \cdot \left(\frac{\partial}{\partial \xi}\left(J^{-1}\xi_{,x}\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\eta_{,x}\right) + \frac{\partial}{\partial \zeta}\left(J^{-1}\zeta_{,x}\right)\right) \\
=& \frac{\partial}{\partial \xi}\left(J^{-1}\xi_{,x}\boldsymbol{F}_x\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\eta_{,x}\boldsymbol{F}_x\right) + \frac{\partial}{\partial \zeta}\left(J^{-1}\zeta_{,x}\boldsymbol{F}_x\right),
\end{aligned}
\tag{4.14}
$$

and analogous terms for $\boldsymbol{F}_y$ and $\boldsymbol{F}_z$. Inserting these terms into Eq. (4.12) and rearranging leads to:

$$
J^{-1}\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{F}_\xi(\boldsymbol{Q})}{\partial \xi} + \frac{\partial \boldsymbol{F}_\eta(\boldsymbol{Q})}{\partial \eta} + \frac{\partial \boldsymbol{F}_\zeta(\boldsymbol{Q})}{\partial \zeta} = J^{-1}\boldsymbol{S}(\boldsymbol{Q}),
\tag{4.15}
$$

with

$$
\begin{aligned}
\boldsymbol{F}_\xi &= J^{-1}\left(\xi_{,x}\boldsymbol{F}_x + \xi_{,y}\boldsymbol{F}_y + \xi_{,z}\boldsymbol{F}_z\right), \\
\boldsymbol{F}_\eta &= J^{-1}\left(\eta_{,x}\boldsymbol{F}_x + \eta_{,y}\boldsymbol{F}_y + \eta_{,z}\boldsymbol{F}_z\right), \\
\boldsymbol{F}_\zeta &= J^{-1}\left(\zeta_{,x}\boldsymbol{F}_x + \zeta_{,y}\boldsymbol{F}_y + \zeta_{,z}\boldsymbol{F}_z\right).
\end{aligned}
\tag{4.16}
$$

This allows us to formulate the transformed flux functions in a direction-independent way:

$$
\boldsymbol{F}_\chi = \begin{pmatrix} \rho Q_\chi \\ \rho u Q_\chi + J^{-1}\chi_{,x}P \\ \rho v Q_\chi + J^{-1}\chi_{,y}P \\ \rho w Q_\chi + J^{-1}\chi_{,z}P \\ \rho(e_{\text{tot}} + P/\rho)Q_\chi \end{pmatrix}, \qquad \chi \in \{\xi, \eta, \zeta\},
\tag{4.17}
$$

where $Q_\chi = J^{-1}\chi_{,x}u + J^{-1}\chi_{,y}v + J^{-1}\chi_{,z}w$. The metric terms $\chi_{,x}$, $\chi_{,y}$ and $\chi_{,z}$ can be computed analytically for certain geometries. However, in *SLH* all metric terms are usually computed with finite differences to ensure that the second derivatives of the transformation functions commute. If this is not exactly fulfilled, spurious geometric source terms may arise as shown by Thompson et al. (1982, pp. 84-88) and Thompson et al. (1985, pp. 158-166). In the case of a two-dimensional transformation (Kifonidis and Müller, 2012) this leads to the following constraints:

$$
\begin{aligned}
(y_{,\eta})_{i+1/2,j} - (y_{,\eta})_{i-1/2,j} &= (y_{,\xi})_{i,j+1/2} - (y_{,\xi})_{i,j-1/2}, \\
(x_{,\eta})_{i+1/2,j} - (x_{,\eta})_{i-1/2,j} &= (x_{,\xi})_{i,j+1/2} - (x_{,\xi})_{i,j-1/2},
\end{aligned}
\tag{4.18}
$$

where half-valued indices once again denote values at the cell interfaces. Since analytical expressions typically do not fulfill the above consistency relation, it is recommended that those derivatives are always computed with appropriate finite differences. It is also noteworthy that J^{-1} corresponds to the cell volume V. However, due to discretization errors in the finite differences the sum of the resulting cell volumes does not necessarily add up to the total volume of the domain. To avoid this problem the cell volume in *SLH* is instead computed by connecting the cell's corners with straight lines (Vinokur and Kordulla, 1983).

Aside from Cartesian geometry in 1D, 2D and 3D several other grid geometries are implemented in *SLH*. The 2D polar and 3D spherical geometries are suitable for simulations of stars that do not include the center of the star, since coordinate singularities are encountered at the poles and at the origin of the coordinate system in this case. Logically rectangular grids for circular and spherical domains can be constructed in a way that they are almost Cartesian in the center. Different types of such *cubed sphere* grids are presented in Calhoun et al. (2008). The interpolated grids described in Section 3.3 of Calhoun et al. (2008) and the smoother circle mapping from Section 3.2 are both implemented in *SLH*. However, these grids often lead to discontinuities in the derivatives of the coordinate transformation. Therefore, they are not suitable for finite difference methods. In addition to that, a uniform Cartesian grid with sinusoidal perturbations was implemented during this thesis for numerical testing purposes. In comparison to other curvilinear grids this particular grid has the advantage that the derivatives of the transformation are varying continuously. The mapping for this grid is generated by

$$x_d = \xi_d + \alpha_d \prod_{p=1}^{D} \sin\left(2\pi\xi_p\right), \qquad d = 1, \cdots, D, \tag{4.19}$$

following the description in Colella et al. (2011). To prevent grid tangling, α_d should fulfill the requirement $0 \leq 2\pi\alpha_d \leq 1$ in all dimensions. The default choice in *SLH* is $\alpha_d = 0.1$ in all three dimensions. An illustration of this grid with 64 cells in each direction is shown in Fig. 4.1.

4.2 Nuclear reaction networks

In the following subsections, we first give a quick overview on how to solve stiff sets of ODEs, which often occur during the solution of nuclear networks (see Sec. 2.2). In addition to that, we discuss operator splitting techniques for the coupling of the nuclear reactions to the hydrodynamics. Finally, we introduce an α chain reaction network with effective reaction rates. Since this specific reaction network was used for the stellar evolution calculations, we are also going to use it for the hydrodynamic simulations in Sec. 5.3 and 5.4. This ensures that the chemical composition is consistent between the stellar evolution code and *SLH*.

4.2.1 Numerical integration of stiff ODEs

As mentioned in Sec. 2.2 the equations for the nuclear network form a *stiff* set of differential equations due to the differences in the reaction timescales. This makes it particularly challenging to solve this problem numerically. Since explicit methods typically require a time step on the order of the fastest reaction of the network, implicit methods are in general more suitable in this case. A more detailed discussion of the issues with integrating stiff sets of equations is given in Press et al. (2007).

Bader and Deuflhard (1983) developed a semi-implicit extrapolation method for stiff sets of differential equations, which is based on the explicit midpoint rule by Bulirsch and

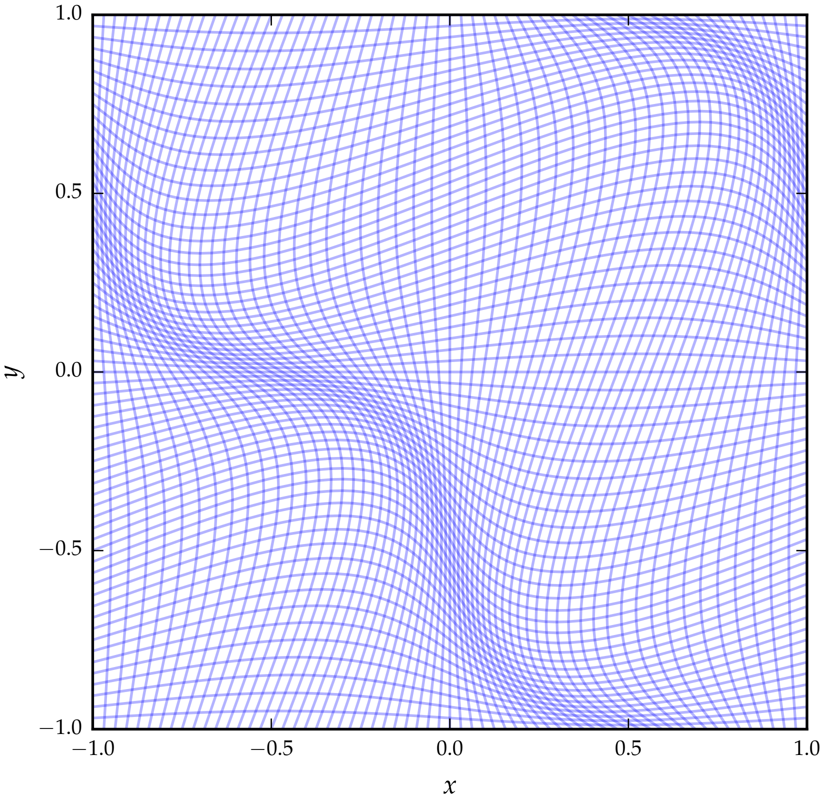

Figure 4.1 | A sinusoidal grid for numerical testing purposes ranging from −1.0 to 1.0 with $N = 64$ cells in each direction. This grid was implemented into *SLH* according to the description in Colella et al. (2011). In this thesis, it is used for the benchmarking of the gravity solvers in Sec. 5.1.

Stoer (1966). The general idea behind those methods is the polynomial extrapolation of the step size towards zero by using a sequence of non-zero step sizes h. The number of intermediate steps is increased successively until the desired convergence for the step size $h = 0$ is achieved.

For large nuclear nuclear reaction networks the Jacobian matrix is typically very sparse, since most species in the network are only connected via reactions that involve captures of light particles such as neutrons, protons, α particles and photodisintegrations, as we have already seen in the subsections of Sec. 3.1. Nuclear reactions of two colliding heavy ions, such as ^{12}C or ^{16}O are very rare. Consequently, the matrix for a network with 300 species has 90 000 matrix elements, but only less than 5000 are non-zero (Hix and Meyer, 2006). Timmes (1999) did a detailed study on the efficiency of matrix solvers depending on the size of the nuclear network. He found that for small nuclear networks (fewer than 60 isotopes) it is usually more efficient to use direct LU decomposition methods (e.g. LAPACK, GIFT), however, sparse matrix solvers such as UMFPACK or

BiCG should be used for larger nuclear networks. More details on sparse matrix solvers for linear systems of equations are given in Sec. 4.3.5.

In *SLH* the nuclear network is coupled to the hydrodynamics via *Godunov splitting*, i.e. the new hydrodynamic state and the nuclear reaction network are evaluated in an alternating sequence. The energy release and the changes in the composition due to nuclear reactions define the new state that is used for the next hydrodynamic time step. In principle a higher-order operator splitting method can be obtained by using *Strang splitting*. This is achieved by first evaluating one of the operators $\mathcal{A}$ over a half time step $\Delta t/2$, then the other operator $\mathcal{B}$ over a full time step Δt and finally again the first operator $\mathcal{A}$ over a half time step $\Delta t/2$. However, in practice the difference between those two methods is often negligible and is only relevant in the limit $\Delta t \to 0$ (LeVeque, 2002). A full coupling of the hydrodynamics to the reaction network is also possible with an *unsplit method*, however, this is only feasible for very small nuclear networks, since the Jacobian grows very rapidly with the number of species as discussed in Sec. 2.1.2. Since the Godunov splitting error is barely noticeable in comparison to unsplit methods (Edelmann, 2014), Strang operator splitting is currently not implemented in *SLH*.

4.2.2 Alpha chain reaction networks with effective rates

Most of the simulations in this thesis were carried out with a 21-isotope α-chain reaction network. The 21-isotope network is based on a 13-isotope α-chain network ranging from ^{4}He to ^{56}Ni, which is described in Timmes et al. (2000). Aside from (α, γ) and the corresponding reverse reaction the 13-isotope network also includes effective rates for the $(\alpha, \mathrm{p})(\mathrm{p}, \gamma)$ links and the reverse reactions by assuming that the intermediate isotopes are in steady-state with their neighbors. This is necessary to obtain reasonably accurate energy generation rates for temperatures $T \gtrsim 2.5 \times 10^{9}\,\mathrm{K}$. This 13-isotope network is further extended with ^{1}H, ^{3}He, ^{14}N, ^{54}Fe, protons (from photodisintegration) and neutrons to account for pp-chains and steady-state CNO cycles during hydrogen burning and photodisintegration into ^{54}Fe. The resulting 19-isotope nuclear network is described in Weaver et al. (1978).

In the 21-isotope network ^{56}Fe and ^{56}Cr are added to account for the neutronization during silicon burning. This neutronization leads to a decrease in the electron fraction Y_e and is crucial for the contraction of the iron core. The reaction rate for the electron captures from ^{56}Fe to ^{56}Cr is obtained from the electron capture rate on ^{56}Ni. It should be noted that the isotope ^{56}Cr in this network does not correspond to the physical isotope ^{56}Cr. It is merely a proxy for the neutron enriched material in the network. As such, a conversion between the composition for this effective rate nuclear network and other nuclear networks is not straightforward.

This 21-isotope nuclear network is the default network for the 1D stellar evolution code *MESA* (Paxton et al., 2011, 2013, 2015), which was used as initial model and as reference model for the 2D and 3D hydrodynamic simulations presented in this thesis (Sec. 5.3 and 5.4). For consistency of the chemical composition we use the same 21-isotope nuclear network as in the stellar evolution code, unless otherwise mentioned. This so-called *approx21* network was reimplemented into *SLH* from the Microphysics module of the open source hydrodynamic codes *CASTRO* and *MAESTRO* (see Sec. 1.1.2).

During the reimplementation a substantial bug in this version of the *approx21* network was discovered, which prevented the appropriate neutronization during silicon burning[2]. The reaction rates in this implementation of the network are also not fully consistent with the rates used in *MESA*. For instance the implementation of the network in the Microphysics module originally used the electron capture rate from Mazurek (1973) and Mazurek et al. (1974)[3]. During the course of this thesis the more recent electron capture rate from Langanke and Martinez-Pinedo (2000) was implemented into the *SLH* version of the network (see Sec. 5.2.2). However, even with the updated electron capture rate there are still other discrepancies in the reaction rates, which modify the energy generation of the network.

4.3 Gravity solvers

The gravity solvers described in this section have been implemented into *SLH* during the course of this PhD thesis. In the following subsections, we discuss the implementation details for each of them. The benchmarking process for these gravity solvers is described in Sec. 5.1. Note that most of the gravity solvers are only implemented for the 3D case, since additional assumptions about the geometry have to be made in a 2D setup.

4.3.1 Monopole solver

The physical motivation behind the monopole solver was already discussed in Sec. 2.3.2. Eq. (2.30) can be integrated by dividing the computational domain into radial bins. The implementation of the monopole solver in *SLH* was adopted from the *LEAFS* code (Reinecke, 2001) with some tweaks to improve the accuracy of the method. For instance the monopole gravity solver in *SLH* first computes the center of mass, whereas in *LEAFS* it is assumed that the center of mass is at the origin of the coordinate system. The radial binning is then carried out with respect to the computed center of mass in an iterative binning procedure to ensure that all radial bins are filled. Afterwards, the gravitational potential is computed according to Eq. (2.30) by integrating the first part from the center outwards and the second part from the outer boundary inwards. The gravitational potential of the neighboring bins is then used to interpolate towards the cells on the grid, which is another improvement in comparison to the method in *LEAFS*. In addition to that, it is possible to add an external potential, which accounts for mass that is not part of the grid. For example in polar or spherical coordinates this can be used to include the center of the star, which is not on the grid. Aside from that, it could also be used to account for the gravitational potential of another star in a binary system. This solver works reasonably well for all grid geometries, as long as the deviations from spherical symmetry in the mass distribution are not too large (see Sec. 5.1.4). It can also be used to obtain boundary conditions and an initial guess for

---

[2] This bug-fix has been pushed to the Microphysics repository in the meantime.

[3] Meanwhile the more recent electron capture rate from Langanke and Martinez-Pinedo (2000) was also implemented in the Microphysics module of *CASTRO* and *MAESTRO*.

more advanced solvers, e.g. the mixed method Poisson solver described in Sec. 4.3.4. As we shall see in Sec. 5.1.5 a good initial guess can drastically reduce the required computing time.

4.3.2 Poisson solver in Cartesian coordinates

The Poisson solver in Cartesian coordinates is an established numerical method that is well understood and often described in the literature (e.g. Press et al., 2007). The goal of this method is to solve Poisson's equation (2.25), which is a prototypical elliptic equation, i.e. a *boundary value problem*. In the following, we illustrate the discretization of Poisson's equation on a Cartesian grid in two dimensions with uniform grid spacing Δ and compute the second derivatives with finite differences. The extension to three spatial dimensions is straightforward.

$$\begin{aligned} \frac{\Phi_{j+1,k} - 2\Phi_{j,k} + \Phi_{j-1,k}}{\Delta^2} + \frac{\Phi_{j,k+1} - 2\Phi_{j,k} + \Phi_{j,k-1}}{\Delta^2} &= 4\pi G\rho_{j,k} \\ \Leftrightarrow \Phi_{j+1,k} + \Phi_{j-1,k} + \Phi_{j,k+1} + \Phi_{j,k-1} - 4\Phi_{j,k} &= \Delta^2 4\pi G\rho_{j,k}. \end{aligned} \tag{4.20}$$

This can be written in matrix form by defining an index $m := j(K+1) + k$ with a one-dimensional sequence.

$$\Phi_{m+K+1} + \Phi_{m-K-1} + \Phi_{m+1} + \Phi_{m-1} - 4\Phi_m = \Delta^2 4\pi G\rho_m, \tag{4.21}$$

which leads to a matrix equation of the form

$$A_{mn}\Phi_n = \Delta^2 4\pi G\rho_m. \tag{4.22}$$

In the case of Poisson's equation the shape of this matrix is "tridiagonal with fringes" (see Press et al. 2007 for an illustration). For a grid with 100 × 100 cells this matrix already has 10^8 entries, but only very few entries are different from zero. Therefore, it is strongly recommended that the algorithm to solve this matrix equation takes advantage of the sparseness. For instance it can be solved with iterative methods for sparse linear systems (see Sec. 4.3.5). Other alternatives to deal with this linear system of equations are *relaxation methods* and so-called *rapid methods* (e.g. Fourier methods). Those methods are described in more detail in Press et al. (2007). In this thesis, the iterative matrix solver BiCGSTAB(5) is used for the benchmarking of the Poisson solver in Sec. 5.1.1. Other iterative solvers that have been implemented into *SLH* for solving the hydrodynamic equations, are also available for the Poisson solver. However, only the BiCGSTAB(5) solver has been thoroughly tested in this thesis.

The structure of this matrix and the methods are well known for the Cartesian case. However, a formulation of Poisson's equation in general coordinates would involve second order derivatives of the metric terms for the transformation between the physical grid and the computational grid. Not only is it challenging to formulate the discretized version of Poisson's equation in this case, but the discontinuities in the metric terms in certain grid geometries might render it completely useless. Therefore, this method is currently only implemented for Cartesian grids and does not work on curvilinear grids. In the case of curvilinear grids the mixed method Poisson solver in Sec. 4.3.4 should be used instead.

4.3.3 Evaluating gradients in curvilinear coordinates

Before we move on to the discussion of the mixed method Poisson solver on curvilinear coordinates, we will first take a look at the computation of the gravitational force from a given gravitational potential in curvilinear coordinates, i.e. computing the gradient of the gravitational potential. This constitutes already the first part of the discretization of the mixed method Poisson solver discussed in Sec. 4.3.4.

The derivation of the gradient of the gravitational potential is in principle analogous to the evaluation of the flux terms for the Euler equations in Sec. 4.1.3. For the following derivation we only consider two spatial dimensions. Extending the treatment to three spatial dimensions is straightforward:

$$\begin{aligned} J^{-1}g_x &= - J^{-1}\frac{\partial \Phi}{\partial x} = - \left[J^{-1}\frac{\partial \Phi}{\partial \xi}\frac{\partial \xi}{\partial x} + J^{-1}\frac{\partial \Phi}{\partial \eta}\frac{\partial \eta}{\partial x}\right] \\ &= - \left[\frac{\partial}{\partial \xi}\left(J^{-1}\frac{\partial \xi}{\partial x}\Phi\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\frac{\partial \eta}{\partial x}\Phi\right) \right. \\ &\qquad \left. -\Phi \left(\frac{\partial}{\partial \xi}\left(J^{-1}\frac{\partial \xi}{\partial x}\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\frac{\partial \eta}{\partial x}\right)\right)\right] \\ &= - \left[\frac{\partial}{\partial \xi}\left(J^{-1}\frac{\partial \xi}{\partial x}\Phi\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\frac{\partial \eta}{\partial x}\Phi\right) \right]. \end{aligned} \tag{4.23}$$

Once again the chain rule of differentiation was used to obtain this expression. The terms in the third row cancel due to $J^{-1}\frac{\partial \xi}{\partial x} = \frac{\partial y}{\partial \eta}$ and $J^{-1}\frac{\partial \eta}{\partial x} = -\frac{\partial y}{\partial \xi}$ (Kifonidis and Müller, 2012). For a more compact formulation we now introduce the transformed gravitational potential

$$\tilde{\mathbf{\Phi}}_n = J^{-1} \begin{pmatrix} \frac{\partial \xi}{\partial n} \\ \frac{\partial \eta}{\partial n} \end{pmatrix} \Phi, \qquad n \in \{x, y\}, \tag{4.24}$$

which allows us to write the gradient of the gravitational potential in the following way:

$$J^{-1}g_n = -\mathbf{\nabla}_{\xi,\eta}\tilde{\mathbf{\Phi}}_n, \qquad n \in \{x, y\}, \tag{4.25}$$

where $\mathbf{\nabla}_{\xi,\eta}$ is the gradient on the uniform Cartesian grid. The expression for the gravity component g_n can now be discretized using finite differences:

$$(g_n)_{ij} = -\frac{1}{2V_{ij}} \left[\left(\tilde{\Phi}^{\xi}_n\right)_{i+1,j} - \left(\tilde{\Phi}^{\xi}_n\right)_{i-1,j} + \left(\tilde{\Phi}^{\eta}_n\right)_{i,j+1} - \left(\tilde{\Phi}^{\eta}_n\right)_{i,j-1} \right], \tag{4.26}$$

where $\tilde{\Phi}^{\xi}_n = J^{-1}\frac{\partial \xi}{\partial n}\Phi$ denotes the ξ-component of the $\tilde{\mathbf{\Phi}}_n$ vector and V_{ij} is the cell volume. This computation of the gradient requires that the coordinate transformation from the physical grid to the computational grid is continuously differentiable. However, in practical applications the above requirement is not necessarily fulfilled in all grid geometries. As such, numerical artifacts might develop during the computation of the gradients, if this condition is not fulfilled (see Sec. 5.1.2).

4.3.4 Mixed method Poisson solver in curvilinear coordinates

In curvilinear coordinates a straightforward solution of Poisson's equation is not feasible, since second-order derivatives of the coordinate transformation are required. This makes it very difficult to obtain a consistent discretization of Poisson's equation and it is not clear, whether a direct discretization of Poisson's equation with second-order derivatives would work as expected in the case of strong discontinuities in the coordinate transformation.

This problem can be partially avoided by using a *mixed method Poisson solver*, which only requires first-order derivatives of the coordinate transformation. The general idea behind this approach is presented for instance in Duncan and Jones (1993). In the framework of the mixed method Poisson solver, second-order derivatives of the coordinate transformation are avoided by splitting Poisson's equation into a set of four first-order differential equations, as shown in Eq. (2.26). The formulation with this set of first-order differential equations has the advantage that the gravitational potential and the gravitational force are computed self-consistently. As such, a separate computation of the gravitational field with finite differences is not necessary. The method discussed in Duncan and Jones (1993) is only applicable to Cartesian grid geometries and uses a finite element discretization. In the following, we are going to extend this treatment to curvilinear coordinates with a finite difference method.

The discretization of the gradient in curvilinear coordinates was already discussed in Sec. 4.3.3. The divergence of the gravitational force can be discretized in a similar fashion. For this purpose we first consider the derivative of the x-component of the gravitational force g_x in x-direction in a two-dimensional setup. Analogous to Eq. (4.23) this leads to:

$$\begin{aligned} J^{-1}\frac{\partial \mathrm{g}_x}{\partial x} &= \left[J^{-1}\frac{\partial g_x}{\partial \xi}\frac{\partial \xi}{\partial x} + J^{-1}\frac{\partial g_x}{\partial \eta}\frac{\partial \eta}{\partial x}\right] \\ &= \left[\frac{\partial}{\partial \xi}\left(J^{-1}\frac{\partial \xi}{\partial x}\mathrm{g}_x\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\frac{\partial \eta}{\partial x}g_x\right)\right]. \end{aligned} \tag{4.27}$$

Together with the corresponding term for g_y this can be used to derive the divergence of the gravitational force in the transformed coordinates:

$$\begin{aligned} \boldsymbol{\nabla}_x \cdot \boldsymbol{g} &= \frac{\partial \mathrm{g}_x}{\partial x} + \frac{\partial \mathrm{g}_y}{\partial y} \\ &= \frac{1}{J^{-1}}\left[\frac{\partial}{\partial \xi}\left(J^{-1}\frac{\partial \xi}{\partial x}\mathrm{g}_x\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\frac{\partial \eta}{\partial x}g_x\right)\right] \\ &+ \frac{1}{J^{-1}}\left[\frac{\partial}{\partial \xi}\left(J^{-1}\frac{\partial \xi}{\partial y}\mathrm{g}_y\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\frac{\partial \eta}{\partial y}g_y\right)\right] \\ &= \frac{1}{J^{-1}}\left[\frac{\partial}{\partial \xi}\left(J^{-1}\frac{\partial \xi}{\partial x}\mathrm{g}_x + J^{-1}\frac{\partial \xi}{\partial y}\mathrm{g}_y\right) + \frac{\partial}{\partial \eta}\left(J^{-1}\frac{\partial \eta}{\partial x}g_x + J^{-1}\frac{\partial \eta}{\partial y}g_y\right)\right]. \end{aligned} \tag{4.28}$$

This is once again discretized by using central finite differences. Together with Eq. (4.26) we obtain the following system of equations for the mixed method Poisson solver in two dimensions:

$$
\begin{aligned}
&\left[2J^{-1}g_x\right]_{i,j} + \left[\tilde{\mathbf{\Phi}}^{\xi}_{x}\right]_{i+1,j} - \left[\tilde{\mathbf{\Phi}}^{\xi}_{x}\right]_{i-1,j} + \left[\tilde{\mathbf{\Phi}}^{\eta}_{x}\right]_{i,j+1} - \left[\tilde{\mathbf{\Phi}}^{\eta}_{x}\right]_{i,j-1} = 0 \\
&\left[2J^{-1}g_y\right]_{i,j} + \left[\tilde{\mathbf{\Phi}}^{\xi}_{y}\right]_{i+1,j} - \left[\tilde{\mathbf{\Phi}}^{\xi}_{y}\right]_{i-1,j} + \left[\tilde{\mathbf{\Phi}}^{\eta}_{y}\right]_{i,j+1} - \left[\tilde{\mathbf{\Phi}}^{\eta}_{y}\right]_{i,j-1} = 0 \\
&\left[J^{-1}\frac{\partial\xi}{\partial x}g_x + J^{-1}\frac{\partial\xi}{\partial y}g_y\right]_{i+1,j} - \left[J^{-1}\frac{\partial\xi}{\partial x}g_x + J^{-1}\frac{\partial\xi}{\partial y}g_y\right]_{i-1,j} + \\
&\left[J^{-1}\frac{\partial\eta}{\partial x}g_x + J^{-1}\frac{\partial\eta}{\partial y}g_y\right]_{i,j+1} - \left[J^{-1}\frac{\partial\eta}{\partial x}g_x + J^{-1}\frac{\partial\eta}{\partial y}g_y\right]_{i,j-1} = \left[-8\pi G\rho J^{-1}\right]_{i,j}.
\end{aligned}
\tag{4.29}
$$

In principle this matrix equation can be solved with similar methods as Poisson's equation. However, the properties of this matrix are drastically different, which makes it much harder to solve this system efficiently. For a given grid size in 3D this matrix has 16 times more entries than the matrix for Poisson's equation, since we have four equations (gravitational potential + three components of the gravitational force) for each grid cell instead of one. The sparsity pattern of the matrix for an 8^3 grid is illustrated in Fig. 4.2. In this case, the matrix already has around 4×10^6 entries, however, only 57 344 of them are non-zero (i.e. 1.4 %). Unfortunately the condition number

$$
cond(\mathcal{A}) = \|\mathcal{A}\|_2 \|\mathcal{A}^{-1}\|_2 = |\lambda_n| \,/\, |\lambda_1| \tag{4.30}
$$

of this matrix is very high. In this equation, λ_n and λ_1 are the eigenvalues of the matrix $\mathcal{A}$ with the highest and lowest absolute value, respectively. For example in the case of a sinusoidal grid with 8^3 cells the condition number is already 1.10×10^{14} with similar values in other grid geometries. This is in part due to vastly different orders of magnitude of the matrix entries. For comparison the condition number for the Poisson equation in Cartesian coordinates on a 8^3 grid is 32 and the corresponding matrix is also considerably smaller with only 262 144 entries. Therefore, solving the linear system for the mixed method Poisson solver is a lot more challenging than solving the linear system for the standard Poisson solver in the Cartesian case, which means that suitable preconditioning is definitely necessary for the mixed method Poisson solver. Details of the iterative solvers and the preconditioning are discussed in the next section.

4.3.5 Iterative solvers for sparse linear systems and preconditioning

In this section, we give a short introduction to iterative solvers and preconditioning for sparse linear systems. As such, we only explain the general concepts behind those methods, details of specific iterative methods are discussed in Meister (2011).

In the following, we are concerned with solving a linear system of equations with the shape

$$
\mathcal{A}\boldsymbol{x} = \boldsymbol{b}, \tag{4.31}
$$

where $\mathcal{A}$ is a matrix that is defined by the discretization of the gravity solver, $\boldsymbol{b}$ is the right hand side vector and $\boldsymbol{x}$ is the vector of the unknown variables (cf. Eq. 4.22 for the Poisson solver in Cartesian coordinates). In our case, $\boldsymbol{b}$ contains the density ρ and information about the boundary conditions of the grid, whereas $\boldsymbol{x}$ always contains the

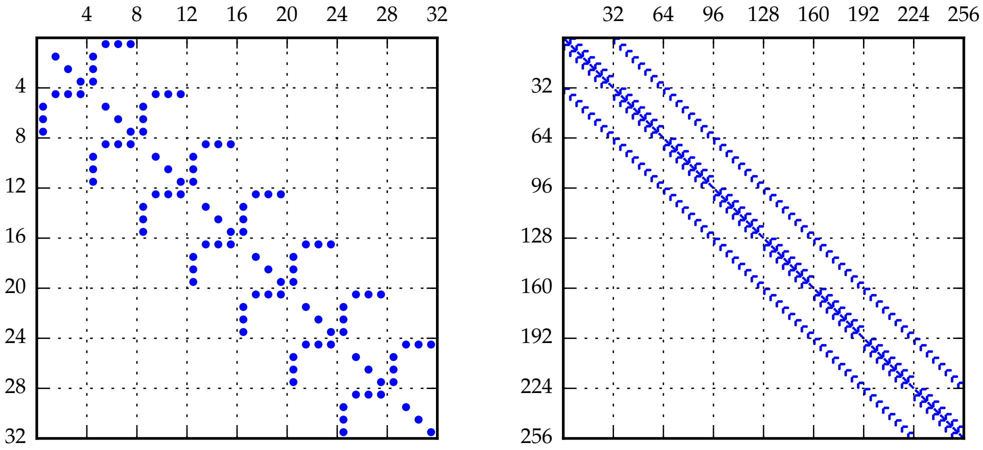

Figure 4.2 | Sparsity pattern of the matrix for the mixed method Poisson solver for a grid with 8^3 cells. The left plot shows a zoomed in region of the matrix with eight grid cells in x-direction that are each separated by dashed lines. The right plot shows a larger part of the matrix with 64 cells, corresponding to eight cells in x-direction and y-direction. The pattern from the left plot is repeated on the diagonal blocks of the matrix. This pattern is repeated in a similar fashion for the cells in z-direction for a total of 2048 matrix entries.

gravitational potential $\mathbf{\Phi}$ and in the case of the mixed method Poisson solver also the three components of the gravitational acceleration $\boldsymbol{g}$.

Iterative methods are defined by a prescription that successively updates the current guess for the solution vector $\boldsymbol{x}_m$ with a new guess $\boldsymbol{x}_{m+1}$ until sufficient convergence of the method is achieved. In general, there are two types of iterative linear solvers. In the first one, the matrix $\mathcal{A}$ is split into $\mathcal{A} = \mathcal{B} + (\mathcal{A} - \mathcal{B})$, where $\mathcal{B}$ should ideally be a good approximation of the matrix $\mathcal{A}$, but also easily invertible (*matrix splitting*). A suitable choice of the matrix $\mathcal{B}$ then leads to the Jacobi method or the Gauss-Seidel method. Unfortunately those methods are not practical for solving the linear system obtained from the mixed method Poisson solver due to the constraint that all matrix entries on the diagonal need to be non-zero, which is clearly not the case for the matrix shown in Fig. 4.2.

The second type of iterative solvers are the so-called *projection methods* or *Krylov-subspace methods*. Those methods typically involve the minimization of the residual $\boldsymbol{r} = \mathcal{A}\boldsymbol{x} - \boldsymbol{b}$ over a subspace, which is formed by the search directions $\boldsymbol{p}_m$. The search directions are defined by the specific method and they determine the convergence properties of the method. The most promising approaches for solving the linear system from the mixed method Poisson solver are the BiCGSTAB(l) method (Van der Vorst, 1992; Sleijpen and Fokkema, 1993) and the GMRES(r) method (Saad and Schultz, 1986). In contrast to most methods for non-symmetric matrices those two directly minimize the residual, which leads to a smoother convergence history. Furthermore, those methods are already implemented into *SLH*, as they are used as iterative solvers for the implicit time stepping of the hydrodynamics (Miczek, 2013). Therefore, with some modifications they can also be used for solving the sparse linear systems that are defined by the gravity solvers.

In particular the high condition numbers (cf. Eq. 4.30) of the matrix for the mixed method Poisson solver make it somewhat challenging to solve this system of equations with iterative linear solvers. In such cases the convergence of the scheme is usually either very slow or it does not converge at all. Therefore, suitable preconditioning of these matrices is necessary.

In general a preconditioned system for Eq. (4.31) has the following form:

$$\begin{aligned} \mathcal{P}_\mathrm{L}\mathcal{A}\mathcal{P}_\mathrm{R}\boldsymbol{y} &= \mathcal{P}_\mathrm{L}\boldsymbol{b} \\ \boldsymbol{x} &= \mathcal{P}_\mathrm{R}\boldsymbol{y}, \end{aligned} \tag{4.32}$$

where $\mathcal{P}_\mathrm{L}$ is the matrix for left preconditioning and $\mathcal{P}_\mathrm{R}$ is the matrix for right preconditioning. In an ideal case the preconditioned matrix $\mathcal{P}_\mathrm{L}\mathcal{A}\mathcal{P}_\mathrm{R}$ should be close to the identity matrix $\mathcal{I}$, because the condition number of the identity matrix is 1. This requires that the matrices $\mathcal{P}_\mathrm{L}$ and $\mathcal{P}_\mathrm{R}$ are close to the inverse of $\mathcal{A}$. However, at the same time the preconditioning matrices $\mathcal{P}_\mathrm{L}$ and $\mathcal{P}_\mathrm{R}$ should be easy to compute and sparse. Since those two goals cannot be achieved achieved at the same time a trade-off is often necessary.

A simple type of preconditioning is the scaling of the matrix $\mathcal{A}$ with a diagonal matrix $\mathcal{D} = \mathrm{diag}\{d_{11}, \ldots, d_{nn}\}$. However, diagonal matrices are usually not a good approximation for the inverse of matrix $\mathcal{A}$ and therefore the scaling preconditioning often provides only a small speed-up of the convergence. Commonly used variants of scaling preconditioning are the scaling with the diagonal element and the row-wise and column-wise scaling with respect to a vector norm. The first variant does not work for the mixed method Poisson solver, since some of the diagonal entries are zero. However, the row-wise scaling from the left and column-wise scaling from the right with the 1-norm (i.e. the sum of the absolute values) provides a suitable preconditioning for the matrix defined by the mixed method Poisson solver. For instance in the case of a sinusoidal grid with 8^3 cells the condition number is reduced from 1.10×10^{14} to 120 for the preconditioned system, which means the system of equations can now be solved with reasonable computational effort. In this thesis, we solve the preconditioned system of the mixed method Poisson solver with BiCGSTAB(5) (see Sec. 5.1.3). Other iterative solvers in *SLH* may also work, but so far only BiCGSTAB(5) has been thoroughly tested.

Splitting associated preconditioning methods (see e.g. Meister, 2011) once again do not work in the case of the mixed method Poisson solver, since some of the diagonal entries are zero. Preconditioning of the matrix with an incomplete LU factorization has been attempted within the *PETSc* framework, however, this did not lead to noteworthy improvements of the convergence.

In context with iterative solvers for sparse linear systems it should also be mentioned that the number of iterations until convergence depends heavily on the quality of the initial guess. Since the density distribution usually does not change very rapidly from one time step to the other, we can reuse the gravitational potential and the gravitational acceleration from the previous time step as initial guess for the solution at the next time step. This significantly reduces the required number of iterations, as we shall see in Sec. 5.1.5. In addition to that, we can use the monopole solution as initial guess for the mixed method Poisson solver in the first time step.

5 Results

5.1 Benchmarking of the gravity solvers

In the following, we are going to assess the accuracy and performance of the newly implemented gravity solvers. For this purpose we define an analytical test setup with an exponentially decreasing density distribution

$$\rho(r) = \rho_\mathrm{c} \mathrm{e}^{-kr}, \tag{5.1}$$

where ρ_c is the central density and r is the distance from the center. The parameter $k = \frac{1}{1.5 \times 10^7\,\mathrm{cm}} \ln(1/0.7)$ is chosen such that the density decreases to 70 % of the central density at a distance of $r = 1.5 \times 10^7\,\mathrm{cm}$. Analytical expressions for the gravitational potential $\Phi(r)$ and the gravitational acceleration $\boldsymbol{g}(r)$ can be derived in this case by using Eq. (2.30) and evaluating the gradient of the potential afterwards

$$\begin{aligned} \Phi(r) &= \frac{4\pi G\rho_\mathrm{c}}{k^3}\left(k\mathrm{e}^{-kr} + \frac{2}{r}\mathrm{e}^{-kr} - \frac{2}{r}\right) + C, \\ \boldsymbol{g}(r) &= \frac{4\pi G\rho_\mathrm{c}}{k^3 r^2}\mathrm{e}^{-kr}\left(k^2r^2 + 2kr + 2 - 2\mathrm{e}^{kr}\right)\boldsymbol{e}_r. \end{aligned} \tag{5.2}$$

For convenience the integration constant C of the gravitational potential is chosen so that the gravitational potential is zero at the origin of the coordinate system, unless otherwise noted. In the following, we consider a three-dimensional test setup with the density distribution given by Eq. (5.1) on a grid with 128^3 cells with MPI parallelization on 8 cores. For each of our numerical methods, we compute the gravitational acceleration $\boldsymbol{g}$ and the gravitational potential Φ and compare it with the analytical results given by Eq. (5.2). We ran the code in parallel to verify that the parallelized version is also working correctly. The scaling behavior of the implementation is discussed in Sec. 5.1.5.

For the purpose of analyzing the deviations between the numerical methods developed in this thesis and the analytical solution, we define the *local relative error* of the gravitational acceleration as

$$e_\mathrm{rel}(\boldsymbol{x}) = \frac{\|\boldsymbol{g}_\mathrm{num}(\boldsymbol{x}) - \boldsymbol{g}_\mathrm{ana}(\boldsymbol{x})\|_2}{\|\boldsymbol{g}_\mathrm{ana}(\boldsymbol{x})\|_2}, \tag{5.3}$$

where $\boldsymbol{g}_\mathrm{num}(\boldsymbol{x})$ is the gravitational acceleration as computed by the specified numerical method and $\boldsymbol{g}_\mathrm{ana}(\boldsymbol{x})$ is the corresponding analytical solution as derived from Eq. (5.2). The local relative error for the gravitational potential Φ is computed in a similar fashion, but the Euclidean norm in the above expression is replaced with absolute values since the gravitational potential is a scalar quantity. This local relative error is a measure for the accuracy at a given grid coordinate. For the following analysis, we define the maximum value of the corresponding local relative error as

$$e_{\max} = \max_{\text{cells}} \left[e_{\text{rel}}(\boldsymbol{x}) \right] . \tag{5.4}$$

This maximum includes all values on the entire grid. In the following illustrations, we only show a cut through the central plane with $z = 0$ (cf. Fig. 5.1). Therefore, the global maximum value $e_{\max}$ is not necessarily shown in our illustration of the grid. In addition to that, we define a *globally averaged relative error* as

$$\epsilon = \frac{1}{NGC} \sum_{\text{cells}} e_{\text{rel}}(\boldsymbol{x}). \tag{5.5}$$

This sum of the local relative errors goes over all grid cells and therefore gives a global measure for the quality of the solution. For comparison of different resolutions it is normalized with the number of grid cells NGC. For the following test setups the number of grid cells is usually $128^3 \approx 2 \times 10^6$. This globally averaged relative error also allows us to compare the quality of the numerical solution for different geometries and different numerical methods.

Obviously the accuracy of the Poisson solver and the mixed method Poisson solver also depends on the iterative solver that is used for solving the resulting linear system of equations and on the convergence threshold of the iterative solver. Unless otherwise noted we are using BiCGSTAB(5) with a convergence threshold of 10^{-8} with respect to the norm of the right hand side (cf. Eq. 4.31). This should give a sufficiently accurate solution in most cases, but the convergence threshold can be further decreased to improve the accuracy of the solvers. Obviously this requires more iterations until convergence is achieved, but this is usually not a major concern since the gravity solver only occupies a small fraction of the overall computing time, especially if the solution from the previous time step is used as initial guess for the gravity solver. In the case of the Poisson solver in Cartesian coordinates preconditioning is not necessary. For the mixed method Poisson solver on curvilinear grids we use the row-wise and column-wise scaling preconditioning described in Sec. 4.3.5.

In the following sections, our goal is to compare to the accuracy of the numerical methods for different grid geometries. For this purpose we consider the grid geometries illustrated in Fig. 5.1. In all of these cases we show a cut through the central plane, where the z-coordinate is zero. The Cartesian geometry in the top left panel and the sinusoidal geometry in the top right panel look the same, since the sinusoidal contribution vanishes for $z = 0$, however, the derivatives of the metric terms are still different from the Cartesian case (the difference between the Cartesian geometry and the sinusoidal geometry will be more obvious in the following figures). Another illustration of the sinusoidal grid, where the sinusoidal contributions are clearly visible, is shown in Fig. 4.1. The bottom left panel in Fig. 5.1 shows the cubed sphere geometry from Calhoun et al. (2008) and the bottom right panel shows a grid with spherical geometry. In the latter case the center of the object is not on the computational grid to avoid the coordinate singularity at the origin of the coordinate system. However, we do account for the missing mass in the center in all of our numerical schemes.

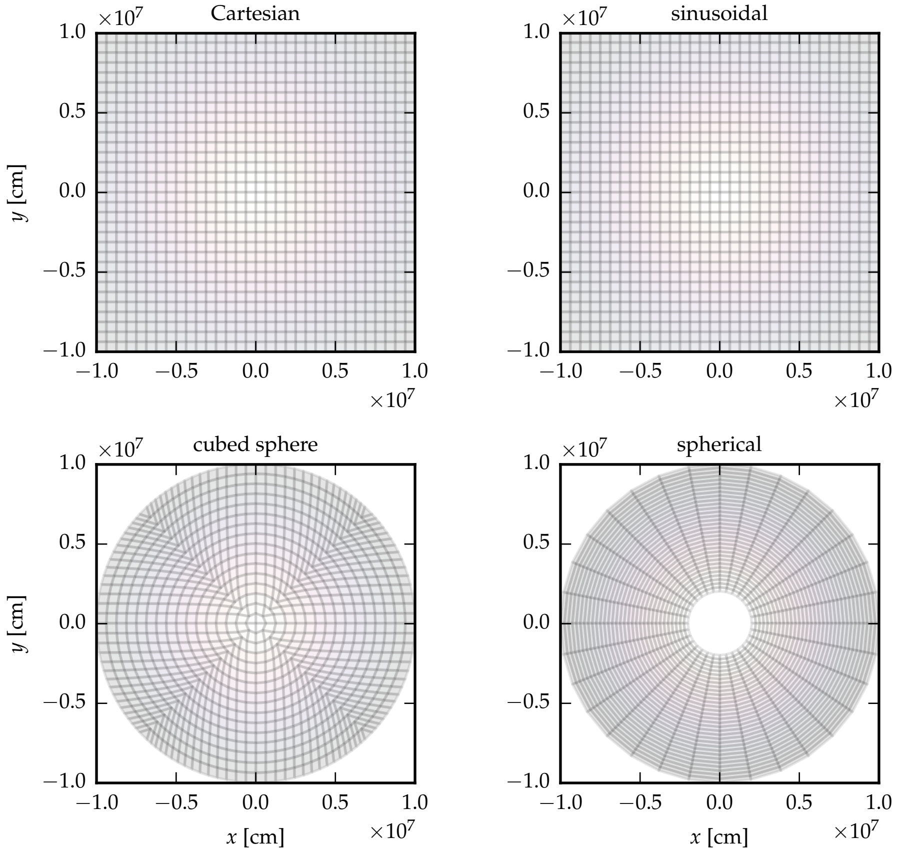

Figure 5.1 | Illustration of the grid geometries that are used for the benchmarking of the gravity solvers in the following sections. For the purpose of this illustration the resolution of the grid has been reduced to 32 cells in each coordinate direction. In all four cases we show a cut through the central plane with $z = 0$.

5.1.1 Poisson solver in Cartesian coordinates

As mentioned in Sec. 4.3.2 the Poisson solver in Cartesian coordinates is already a well established numerical method. Therefore, in this section we are only going to verify that our implementation of the Poisson solver meets those high expectations. Additionally, we are going to use the results for the Poisson solver in Cartesian coordinates as reference for the other numerical methods that are discussed in the following sections. It should be noted that the Poisson solver only gives a solution for the gravitational potential $\Phi(\boldsymbol{x})$. This means that the components of the gravitational acceleration $\boldsymbol{g}(\boldsymbol{x})$ have to be obtained in a separate step by computing the gradient of the gravitational potential with finite differences, which may lead to additional deviations from the analytic solution. However, in comparison to other grid geometries the evaluation of the gradient in Cartesian coordinates is very accurate (cf. Sec. 5.1.2). The distribution of the local relative error for the gravitational potential obtained from the Poisson solver in Cartesian coordinates is shown in the left panel of Fig. 5.2, the right panel shows the local relative error for the gravitational acceleration computed from the resulting potential via central, first-order finite differences.

It is clearly visible that the local relative error increases substantially towards the center. This is mainly due to the fact that both the gravitational potential and the gravitational acceleration have a root at the origin of the coordinate system, as can be seen from Eq. (5.2). For analyzing purposes the gravitational potential has been shifted so that the minimum value of the numerical solution coincides with the analytical solution. Consequently, the relative error for the innermost cells is almost zero ($\lesssim 1 \times 10^{-14}$), which is indicated by the black colors in the center. Overall the local relative error in the gravitational potential never exceeds a value of $e_{\max} = 3.06 \times 10^{-4}$. However, it should be noted that the relative error in the gravitational potential is heavily influenced by the choice of the constant C in Eq. (5.2).

Therefore, it is more conclusive to look at the gravitational acceleration that is obtained by evaluating the gradient from this gravitational potential. We can see that the local relative error in the gravitational acceleration never exceeds a value of $e_{\max} = 8.02 \times 10^{-4}$, even in the center of the grid, where the absolute value of the gravitational acceleration is close to zero. In addition to that, the global average of the relative error over all cells on the grid is $\epsilon = 1.21 \times 10^{-5}$. We conclude that the Poisson solver in Cartesian coordinates is very accurate as expected. It should further be noted that the accuracy of the Poisson solver depends on the convergence threshold of the iterative solver for the linear system of equations. However, the precise correlation between the convergence threshold and the relative error on the grid is not obvious. As mentioned in the previous section we are using a convergence threshold of 10^{-8} with a BiCGSTAB(5) method for these tests, which should be accurate enough for most practical applications.

5.1.2 Evaluating gradients in curvilinear coordinates

During the derivation of the gradients in curvilinear coordinates we already discussed that this treatment breaks down if the coordinate transformation is not continuously differentiable. Unfortunately this requirement is not necessarily fulfilled in general curvilinear coordinates. Since this would most likely also impact the accuracy of the mixed method Poisson solver in curvilinear coordinates, we are obviously concerned with the numerical artifacts that arise if the coordinate transformation is not continuously differentiable. To analyze these artifacts we compute the gravitational acceleration $\boldsymbol{g}(\boldsymbol{x})$ from the analytic gravitational potential $\Phi(\boldsymbol{x})$ given by Eq. (5.2) and compare it with the analytical expression for the gravitational acceleration. The local relative error for the different geometries is shown in Fig. 5.3.

In the Cartesian case (top left panel) we show the relative errors resulting from central, first-order finite differences on a Cartesian grid. This result is very similar to the relative error of the gravitational acceleration for the standard Poisson solver in Cartesian coordinates, which suggests that the deviations from the analytical solution in Fig. 5.2 are mainly due to discretization errors in the finite differences. In fact, the

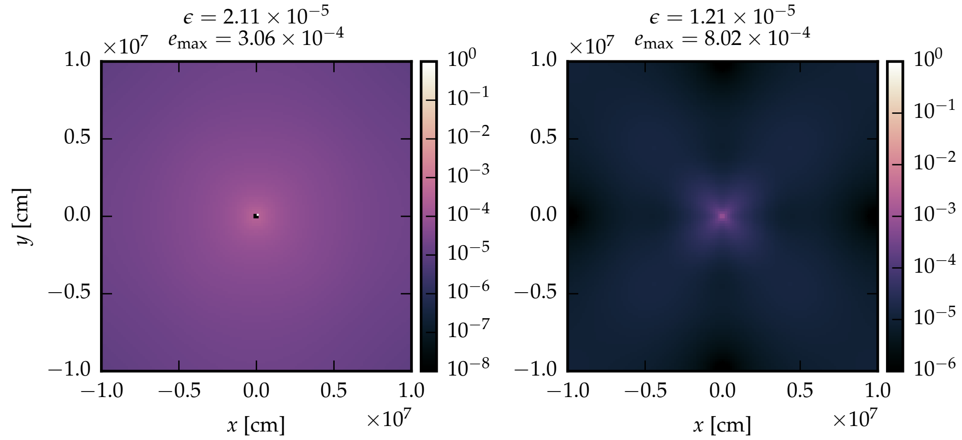

Figure 5.2 | Distribution of the local relative error $e_{\text{rel}}(\boldsymbol{x})$ of the gravitational potential $\boldsymbol{\Phi}(\boldsymbol{x})$ (left panel) for the standard Poisson solver in Cartesian coordinates and for the gravitational acceleration $\boldsymbol{g}(\boldsymbol{x})$ computed from this potential via standard finite differences (right panel). The globally averaged relative error ϵ is a measure for the quality of the solution in comparison to other numerical methods and grid geometries and $e_{\max}$ denotes the maximum value of the corresponding local relative error.

relative error of the gravitational acceleration is smaller, if the gradient is computed from the gravitational potential of the Poisson solver and not from the analytical potential given by Eq. (5.2). This may seem counter-intuitive at first, however, the Poisson solver by construction computes a gravitational potential that fulfills the discretized version of Poisson's equation. Due to that the potential resulting from the Poisson solver might be more suited for computing the gravitational acceleration via finite differences.

However, we are more interested in the numerical artifacts that are caused by coordinate transformations with discontinuous metric terms. These numerical artifacts are particularly pronounced for the cubed sphere geometry in the bottom left panel. In the strongly deformed cells along the diagonals with $x = y$ the relative error in the evaluation of the gradient regularly exceeds values of 0.2. This effect is even stronger along diagonals with $x = y = z$ (not shown in the plot), where the relative error can be as large as $e_{\max} = 0.4$. The relative errors for evaluating the gradients on the sinusoidal grid or the spherical grid are much smaller and only reach values of $e_{\max} = 3.52 \times 10^{-3}$ and $e_{\max} = 1.85 \times 10^{-3}$, which is not that much larger than for the standard finite differences in the Cartesian case. On the spherical grid the largest deviations in the gravitational acceleration are encountered at the inner boundary of the grid, whereas in sinusoidal geometry the largest deviations from the analytical solution are close to the central plane shown in Fig. 5.3 (top right panel) at the coordinates

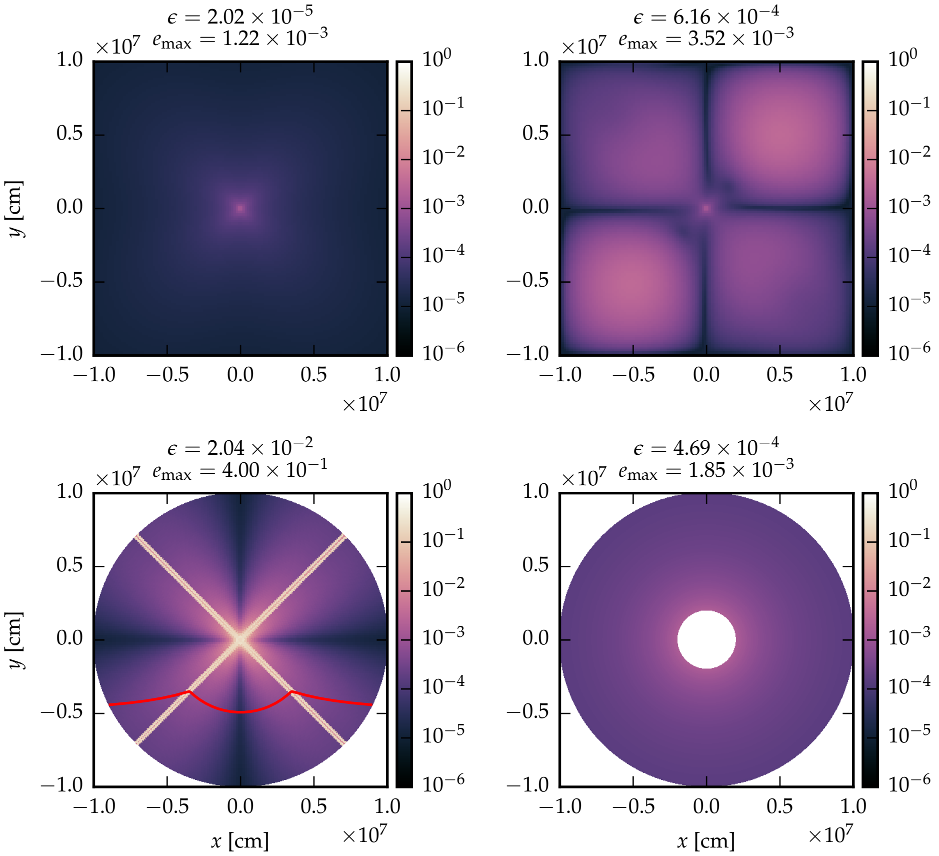

Figure 5.3 | Distribution of the local relative error $e_{\mathrm{rel}}(\boldsymbol{x})$ for the evaluation of the gravitational acceleration $\boldsymbol{g}(\boldsymbol{x})$ from the analytic gravitational potential $\Phi(\boldsymbol{x})$ via finite differences on different grid geometries. The different grid geometries that are used for this test setup are illustrated in Fig. 5.1. The definition of ϵ and $e_{\max}$ is the same as in Fig. 5.2. The red line in the bottom left panel is used for further analysis in Fig. 5.4.

$x = y = \pm 5.38 \times 10^7$ cm, where the sinusoidal contributions reach a maximum. In all three cases the largest errors in the finite differences manifest themselves in locations where the grid is strongly deformed.

In the case of the cubed sphere grid we want to take a closer look at the metric terms and their correlation with the errors in the finite differences. For this purpose we plot the relative error and the relevant metric terms along the red line shown in the bottom left panel of Fig. 5.3. The results are shown in Fig. 5.4.

We can clearly see that the largest errors in the evaluation of the gradients coincide with pronounced jumps in the metric terms $J^{-1}\xi_x$ and $J^{-1}\xi_y$. However, it is also noteworthy that there is almost no change in $J^{-1}\xi_z$. This is due to the fact that we are very close to the central plane with $z = 0\,\mathrm{cm}$. Consequently, we would expect that the largest deviations of the numeric result to the analytical solution are in the x- and

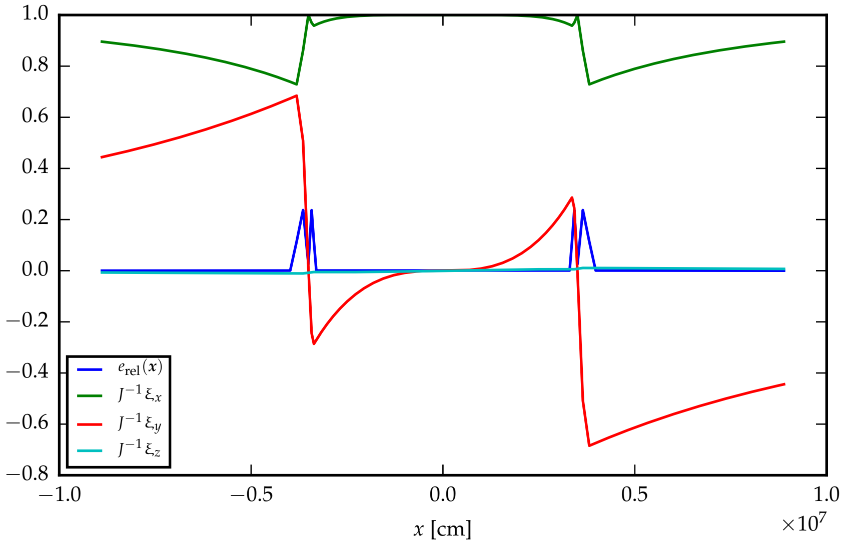

Figure 5.4 | This plot illustrates the correlation between discontinuities in the metric terms and errors in the evaluation of gradients on curvilinear grids. It shows the local relative error of the gravitational acceleration $\boldsymbol{g}(\boldsymbol{x})$ with the relevant metric terms along the red line in the bottom left panel of Fig. 5.3.

y-components of the gravitational acceleration, which is indeed the case. In the case of the sinusoidal grid and the spherical grid the coordinate transformation is much smoother. This is also the reason why the evaluation of the gradient is less problematic in those two cases.

Obviously the numerical artifacts in the computation of finite differences on curvilinear grids with strong discontinuities in the metric terms are a major problem and we expect that similar artifacts will appear in the mixed method Poisson solver. Unfortunately there is not much we can do about these discontinuities in the derivatives of the coordinate transformation. We could either try to construct coordinate transformations with smoother metric terms, however, this is not always possible. Alternatively we could use numerical methods that do not make explicit use of the metric terms, for example the monopole solver in Sec. 5.1.4.

5.1.3 Mixed method Poisson solver

It was discussed in the previous section that numerical artifacts in the evaluation of the gradients due to discontinuities in the metric terms would probably lead to similar errors for the mixed method Poisson solver. Nevertheless it is worthwhile to investigate whether this is indeed the case. For this purpose we compute the local relative error for the gravitational acceleration that was obtained from the mixed method Poisson solver. It is noteworthy that the mixed method Poisson solver computes a self-consistent solution for the gravitational potential and the gravitational acceleration at the same time. Therefore, a separate computation of the gravitational acceleration with the finite differences from the previous section is not necessary. The distribution of the local relative errors for the gravitational acceleration $\boldsymbol{g}(\boldsymbol{x})$ is shown in Fig. 5.5. The top left panel shows the result for the standard Poisson solver in Cartesian coordinates for comparison. We are once again using BiCGSTAB(5) with a convergence threshold of 10^{-8} for the iterative solution of the matrix. In addition to that, row-wise and column-wise scaling preconditioning is used to speed up the convergence.

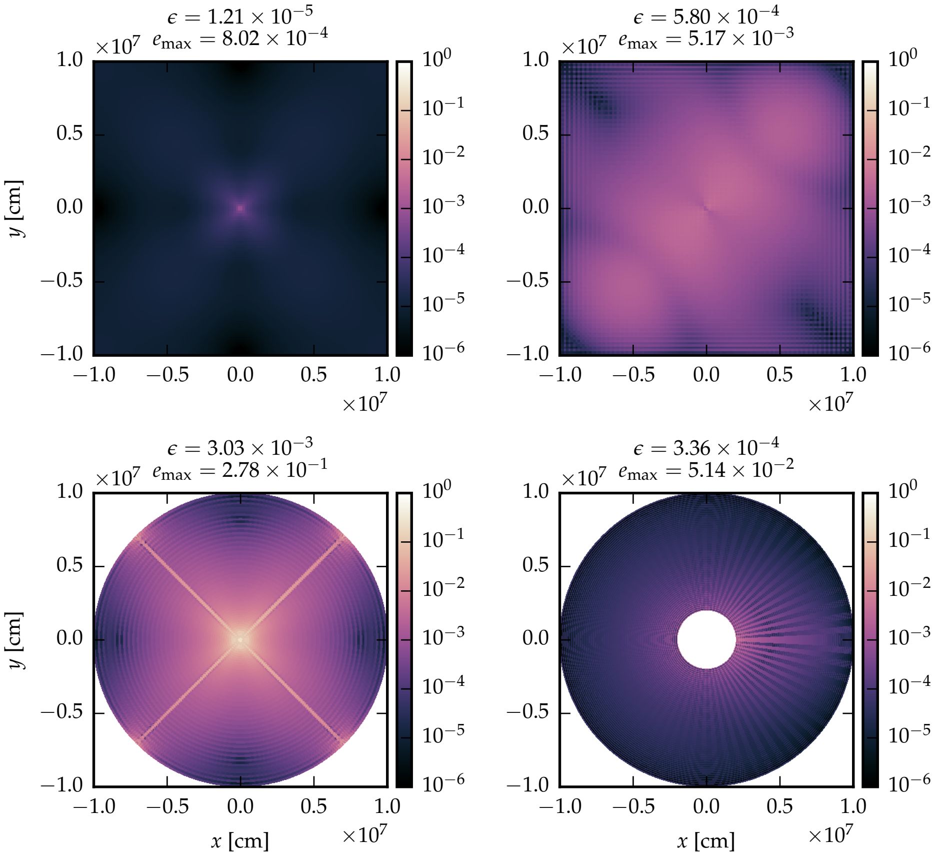

Figure 5.5 | Distribution of the local relative error $e_{\mathrm{rel}}(\boldsymbol{x})$ for the gravitational acceleration $\boldsymbol{g}(\boldsymbol{x})$ resulting from the mixed method Poisson solver on different grid geometries. The different grid geometries that are used for this test setup are illustrated in Fig. 5.1. The definition of ϵ and $e_{\max}$ is the same as in Fig. 5.2.

We can see in all four cases that the relative error of the gravitational acceleration increases substantially towards the center. One reason for this is of course the decreasing absolute value of the gravitational acceleration towards the center. But in addition to that the errors in the iterative solution also accumulate towards the center, since we give the values at the boundaries of the grid and then iterate towards the center. In the cubed sphere geometry the artifacts along the diagonals are once again clearly visible and the local relative error peaks at $e_{\max} = 0.278$ in the center. Obviously this is again due to the discontinuities in the metric terms, which we showed in Fig. 5.4. However, in comparison to Fig. 5.3 the globally averaged error in the gravitational acceleration is almost one order of magnitude smaller than for the computation of the gradients from a given gravitational potential. This is better than we originally expected after analyzing the errors in the evaluation of the gradients, but obviously still not good enough for practical applications.

It looks much better in the case of smoothly varying metric terms. For the sinusoidal geometry the relative error in the gravitational acceleration peaks at $e_{\max} = 5.17 \times 10^{-3}$ near the center of the grid, which is slightly higher than the maximum error for the evaluation of the gradients from the analytic solution of the gravitational potential ($e_{\max} = 3.52 \times 10^{-3}$) in the previous section. However, the globally averaged error for the gravitational acceleration from the mixed method Poisson solver ($\epsilon = 5.80 \times 10^{-4}$) is actually slightly lower than the globally averaged error for the evaluation of the gradients ($\epsilon = 6.16 \times 10^{-4}$). Obviously this sinusoidal geometry is only a numerical test setup and does not have any relevance for practical applications, but it shows that the implementation of the mixed method Poisson solver works in principle, if there are no abrupt changes in the metric terms.

The setup with spherical geometry is a test case with smoothly varying metric terms that is more relevant for practical applications. In this case, the error in the gravitational acceleration peaks at the inner boundary in the radial coordinate for maximum and minimum values of θ (i.e. furthest from the central xy-plane, which is shown in Fig. 5.5) and close to the periodic boundary in φ-direction at $e_{\max} = 5.14 \times 10^{-2}$. The globally averaged error $\epsilon = 3.36 \times 10^{-4}$ is lower than for the sinusoidal geometry and slightly lower than for the evaluation of the gradients from the analytic solution of the gravitational potential ($\epsilon = 4.69 \times 10^{-4}$). However, there is clearly a periodic pattern in the φ-direction that originates near the periodic boundary. This is most likely due to the fact that periodic boundaries are not yet implemented in the matrix of the mixed method Poisson solver. We expect that the errors should decrease further with a proper implementation of the periodic boundaries. Practical applications of the mixed method Poisson solver in spherical geometry might be possible afterwards.

5.1.4 Monopole solver

In Sec. 5.1.2 we have shown that evaluating gradients on a curvilinear grid is a challenging task, if the metric terms are not continuously differentiable. In those cases it might be better to use a gravity solver that does not make explicit use of the metric terms. The simplest way to achieve this is using a monopole gravity solver, since it allows a direct computation of the gravitational acceleration without the use of finite differences. For this computation only the enclosed mass in the first integral of Eq. (2.30) is needed. However, this treatment breaks down if part of the mass is not on the numerical grid. In this case, we add an external potential, which accounts for mass that is not part of the grid. In the presence of such an external gravitational potential the gravitational acceleration has to be computed with the finite differences from the previous section.

Among the four test cases that we use for benchmarking this treatment with the external potential is only necessary for the spherical geometry, since the central region with the origin of the coordinate system is not part of this grid. Therefore, we need to evaluate the gravitational acceleration with finite differences in this case. The resulting distribution of the local relative error in the gravitational acceleration $\boldsymbol{g}(\boldsymbol{x})$ is shown in Fig. 5.6.

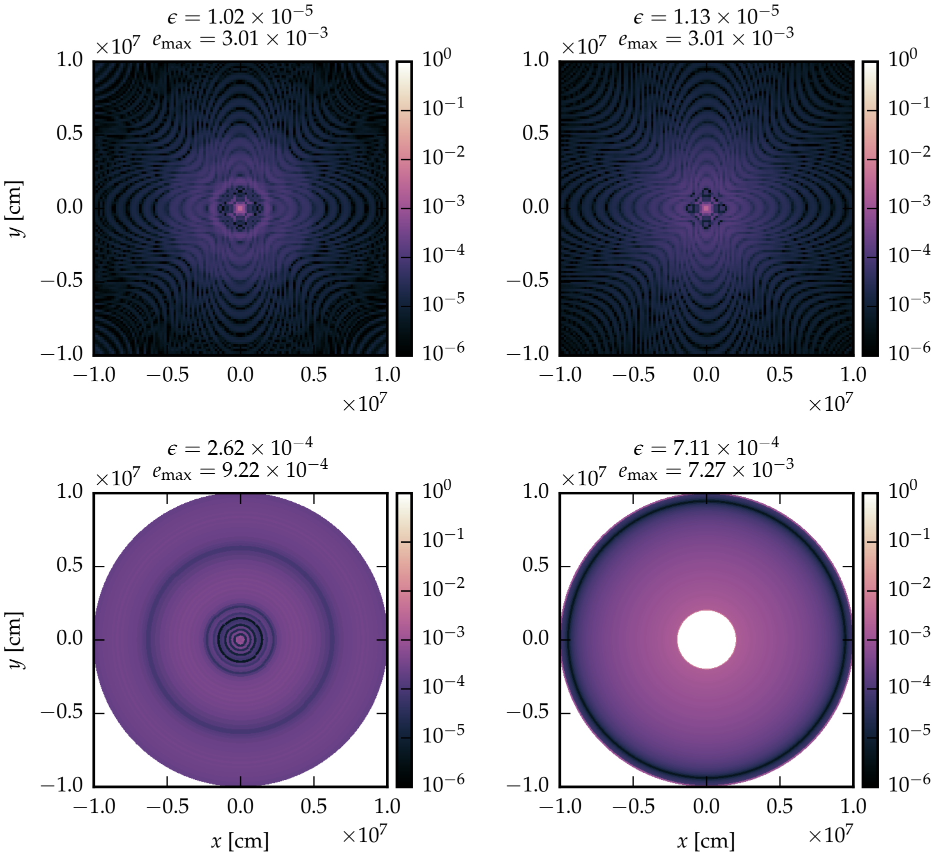

Figure 5.6 | Distribution of the local relative error $e_{\mathrm{rel}}(\boldsymbol{x})$ for the gravitational acceleration $\boldsymbol{g}(\boldsymbol{x})$ resulting from the monopole gravity solver on different grid geometries. The different grid geometries that are used for this test setup are illustrated in Fig. 5.1. The definition of ϵ and $e_{\max}$ is the same as in Fig. 5.2.

It is noteworthy that the results for the Cartesian geometry and the sinusoidal geometry are very similar in this case. The maximum value of the relative error at the origin of the coordinate system is the same in both cases and the globally averaged error ϵ is only marginally different. In addition to that, the monopole solver also works reasonably well in the case of the cubed sphere geometry, since derivatives of the coordinate transformation are not involved in this calculation. The result for the gravitational acceleration is only slightly worse than for the Cartesian and the sinusoidal geometry. As expected the result for the spherical geometry is the worst out of those four test cases, since finite differences are required to compute the gradient of the gravitational potential in this case. However, even in this case the relative error of the gravitational acceleration does not exceed $e_{\max} = 7.27 \times 10^{-3}$, which should be good enough for most practical applications. It is also noteworthy that the globally averaged error for spherical geometry is larger than for the mixed method Poisson solver, even though there is this problem with the periodic boundary in the latter case.

The main caveat with the monopole solver is that this only works reasonably well, if the deviations from a spherically symmetric density distribution are not too large. The exponentially decreasing density distribution from Eq. (5.1) is obviously spherically symmetric. Hence the monopole solver works very well for this test setup. However, during late burning stages in stars large asymmetries may develop (Arnett and Meakin, 2011; Müller et al., 2016). In those cases the monopole solver might not be accurate enough and more advanced methods should be used.

An analytical test setup that is not spherically symmetric can easily be constructed by overlapping two exponentially decreasing density distributions that are shifted against each other. The analytical solution for this setup can be computed by using the superposition principle. In this case the quality of the monopole solution should depend on the distance between the peaks and the overlap of the two exponentially decreasing density distributions. Note that simply shifting the density distribution from the origin of the coordinate system, which has been done in Schrauth (2015) for the monopole gravity solver in *LEAFS*, produces almost the same results as shown in Fig. 5.6, since our implementation of the monopole solver does not assume that the center of mass is at the origin of the coordinate system (see Sec. 4.3.1).

5.1.5 Performance and scaling

Up until now we were mainly concerned with the accuracy of the newly implemented gravity solvers. In this section, we are going to analyze the performance and scaling for some of those gravity solvers. For this purpose we use once again the analytical test setup defined in Sec. 5.1 with a resolution of 128^3 cells. To ensure a homogeneous environment all of these tests were performed on a single Ivy Bridge node (i.e. 16 cores) on the local cluster at the Heidelberg Institute for Theoretical Studies (HITS). Same as before, we are using BiCGSTAB(5) with a convergence threshold of 10^{-8} for the iterative solution of the linear system of equations.

In the following, we compare the convergence behavior for the Poisson solver in Cartesian geometry and for the mixed method Poisson solver in sinusoidal geometry. In Fig. 5.7 we show the convergence history for the Poisson solver (blue lines) and the mixed method Poisson solver (red lines). Solid lines start from value zero for both the gravitational acceleration and the gravitational force, whereas dotted lines start from the solution of the monopole solver, which is a much better approximation of the actual solution. As we can see the iteration with zero-valued initial guess starts with a residual of 1 for both the mixed method Poisson solver and the standard Poisson solver, whereas the iteration with the monopole solver as initial guess starts at 2.74×10^{-5} in the case of the mixed method Poisson solver and 2.19×10^{-7} in the case of the standard Poisson solver.

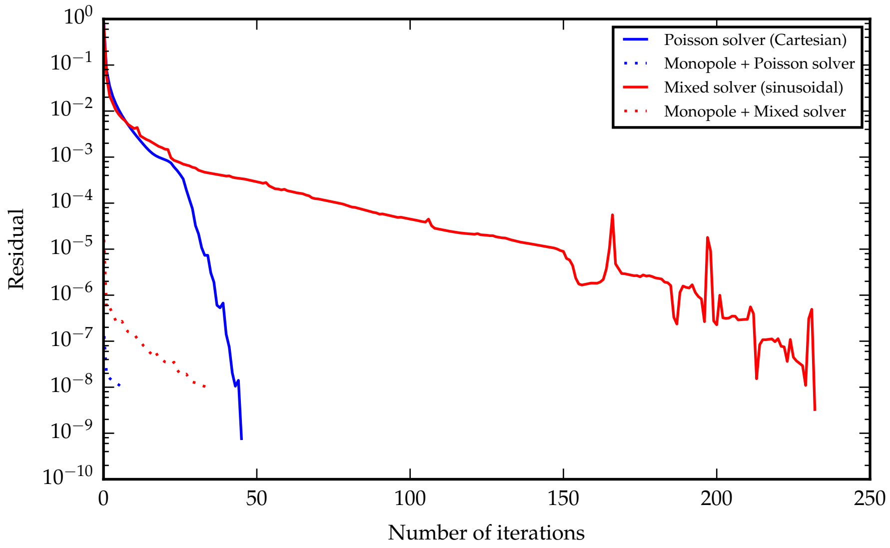

Figure 5.7 | Convergence history of the Poisson solver in Cartesian geometry and for the mixed method Poisson solver in sinusoidal geometry. Using the monopole solver as initial guess for the iterative solver significantly reduces the number of iterations in both cases.

In the case of the mixed method Poisson solver in sinusoidal geometry it requires 233 iterations starting from a zero-valued initial guess to reach the convergence threshold of 10^{-8}. This takes roughly 350 s on a single Ivy Bridge node. With the monopole solver as initial guess convergence is reached after only 35 iterations, which corresponds to 17 s with the above configuration. It should be noted that there is almost no difference in the quality of solution, both solutions are practically identical with the solution shown in the upper right panel of Fig. 5.5.

For the standard Poisson solver in Cartesian coordinates we can see a similar reduction from 46 iterations with zero-valued initial guess to only seven iterations, if the monopole solution is used as initial guess. The computing time is reduced from roughly 7 s to 1 s. However, in this case the quality of the solution with the monopole solver is slightly worse due to the low number of iterations. This could be prevented by decreasing the threshold for convergence. We want to emphasize that there is no unique relation between the residual and the relative errors on the grid. Therefore, it can be very misleading to compare different grid geometries based on the residual. This is especially the case for the comparison of the residual between the standard Poisson solver in Cartesian coordinates and the mixed method Poisson solver in a different geometry, since the definition of those two residuals is inherently different. Obviously the residual also scales with the number of grid cells. Therefore, it might be necessary to readjust the convergence threshold after changing the grid resolution.

We conclude that the quality of the initial guess has a big impact on the convergence behavior. A good initial guess (e.g. the monopole solution for a spherically symmetric density distribution) significantly reduces the number of iterations and thus the required computing time. In a similar fashion we can also use the solution from the previous time step as initial guess for the next time step. Usually the density distribution does not change very much between two time steps, so this should be an even better initial guess than the monopole solution in most cases. This also has the advantage that the gravity solver consumes almost no computing time, if the density distribution does not change between two time steps, since the initial guess should already be below the convergence threshold. With this approach, the computing time of the gravity solver is almost negligible in comparison to the hydrodynamics and the nuclear network. During the 3D simulation of silicon burning in Sec. 5.4 the Poisson gravity solver consumed only around 0.2 % of the total computing time.

In the following, we want to analyze the scaling of the monopole solver and the mixed method Poisson solver in sinusoidal geometry. For this purpose the same analytic setup was run by Philipp Edelmann during the JUQUEEN Extreme Scaling Workshop 2016 at Forschungszentrum Jülich on up to 24 racks (i.e. 393 216 cores) with a resolution of 1920^3 cells. The monopole solver was used as described in Sec. 5.1.4, whereas the mixed method Poisson solver was used with the monopole solution as initial guess. The solution of the resulting linear system of equations was then obtained using BiCGSTAB(5) with a convergence threshold of 10^{-8}, which is a rather large value for this resolution. In fact this threshold is already reached after two iterations. However, the scaling behavior should not be much different even for such a low number of iterations.

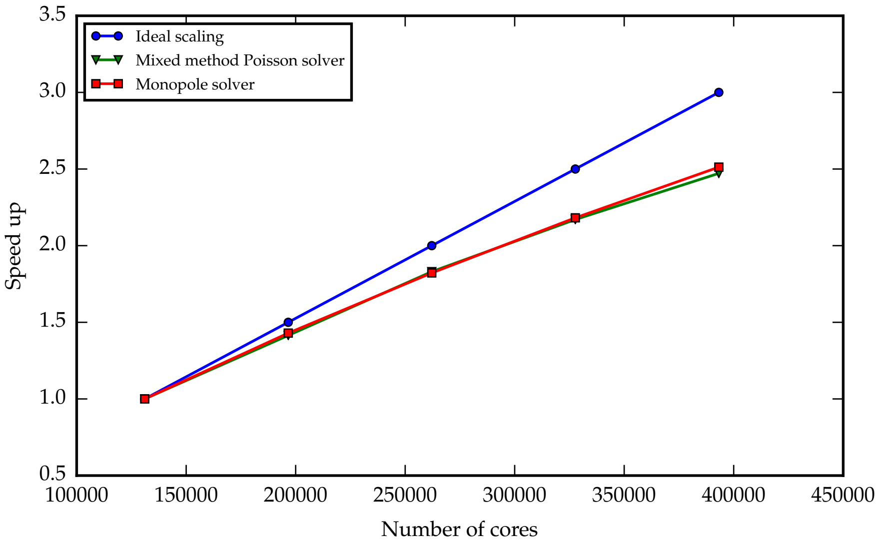

Figure 5.8 | Strong scaling of the mixed method Poisson solver and the monopole solver on a 1920^3 grid. The lowest number of cores (i.e. 131 072) is used as reference point for the scaling. The data for this plot were provided by Philipp Edelmann.

The results of the scaling tests are shown in Fig. 5.8. We can see that both the monopole solver and the mixed method Poisson solver scale reasonably well up to 24 racks with an efficiency of 82.4 % in the case of the mixed method Poisson solver and 83.8 % in the case of the monopole solver. This confirms that the parallelization of the gravity solvers is working even in extreme scenarios. In fact this speed up is very close to the scaling for the linear solver and the fluxes reported in Edelmann and Röpke (2016). Given the fact that the gravity solvers constitute only a small fraction of the total computing time in most cases, this should be sufficient for most practical applications.

5.2 Setup of the initial model for Si burning

In the following sections, we are going to describe the initial setup that is used for the simulations of silicon burning in 2D (Sec. 5.3) and 3D (Sec. 5.4). The initial model was obtained from the stellar evolution code *MESA* version 7623 (Paxton et al., 2011, 2013, 2015) by evolving a non-magnetic, non-rotating $15M_\odot$ star from the pre-main sequence phase to the onset of core collapse, when the velocity of the in-falling core reaches $1000\,\mathrm{km\,s^{-1}}$. The *MESA* parameters inlist for this model was provided by Couch et al. (2015).

The stellar evolution model initially uses a basic 8-isotope network. This network is successively extended during the course of the simulation. Towards the end of the simulation it uses a 21 isotope network with effective rates (*approx21*). This network was briefly described in Sec. 4.2.2. The network uses a very simplified composition for the treatment of core neutronization. Therefore, it is not trivial to translate the composition of this nuclear network to a larger network, which does not use the same approximate treatment. For simplicity we use the same 21-isotope nuclear network in our simulations, as it allows us to directly compare our results with the stellar evolution model. As we show in Sec. 5.2.2 the reaction rates in our version of the network are not fully consistent with the ones used in *MESA*, which leads to slight discrepancies in the energy generation rate.

During the late stages of silicon burning, roughly 1000 s before core collapse the model is mapped to the hydrodynamics code *SLH* (Sec. 1.1.3). At this point the mass of the iron core in the stellar evolution model is already $1.3M_\odot$. The abundance profiles for the stellar evolution model at this time are shown in Fig. 5.9. The whole central region consists mainly of iron-group isotopes, such as ^{54}Fe, ^{56}Fe and ^{56}Cr. Due to the approximate treatment of the network, those species do not represent the actual physical isotopes. They are mainly used to represent the increasing neutronization of the iron core towards the center. The bottom of the silicon burning layer is located at around 1.7×10^8 cm, where the ^{28}Si mass fraction decreases below 10^{-2}. Regions of oxygen burning and neon burning are also visible further outside in the model. The inner border of those burning regions can roughly be identified by a sharp decline of the corresponding burning fuel. A more detailed analysis of the burning zones is given in Sec. 5.3.2. The carbon burning shell is even further outside and not visible in this plot.

Ideally one would start with a stellar model at the beginning of silicon burning, when

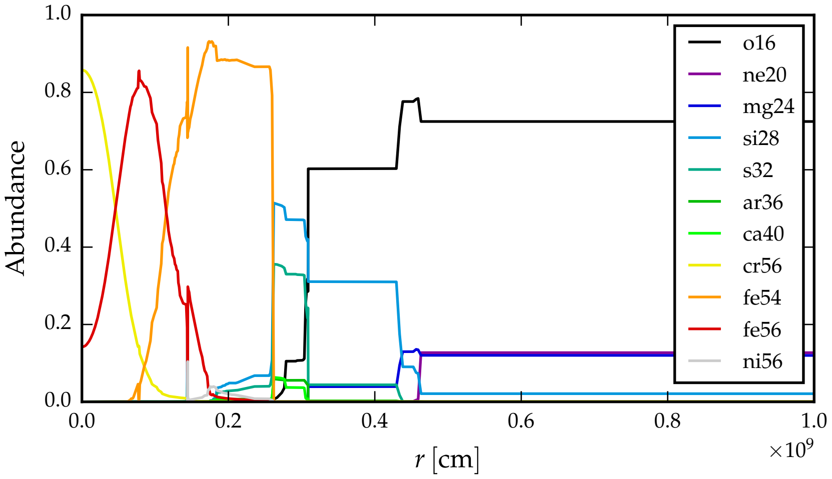

Figure 5.9 | Abundance profiles of the *MESA* model for all isotopes that exceed a mass fraction of 0.05 between the center and an outer radius of 10^9 cm.

the iron core is not yet present. However, this is not feasible with the currently available computational resources. In fact it is already quite challenging to reach the onset of core collapse starting with the initial model above. Couch et al. (2015) were only able to simulate the evolution until core collapse, because they artificially enhanced the electron capture rate, which was not done for the simulations presented here. In addition to that, they only simulated one octant of the whole star, whereas we cover the whole core in full 4π steradians.

During the stellar evolution modeling in *MESA* the Helmholtz equation of state from Timmes and Swesty (2000) with Coulomb corrections was used. As mentioned in Sec. 1.1.3, this equation of state is also implemented in *SLH*. In addition to that, we have verified the accuracy of the equation of state by recomputing the temperature profile with the density and pressure profiles from *MESA* as input values. In the relevant parts of the model the relative difference between this recomputed temperature profile and the original temperature profile from *MESA* stays below 10^{-8}. Therefore, we conclude that differences in the implementation of the equation of state are negligible.

5.2.1 Maintaining hydrostatic equilibrium

It is particularly important to maintain hydrostatic equilibrium during the mapping of the 1D stellar evolution model from *MESA* to the hydrodynamic grid. This is achieved by using the hydrostatic integration method described in Zingale et al. (2002). For this integration, one thermodynamic quantity from the original stellar evolution model is used along with the nuclear composition. Since reaction rates are very sensitive to the temperature we take the temperature profile $T(r)$ and the gravitational acceleration $g(r)$ from the initial model and obtain initial profiles for the density $\rho(r)$ and pressure $P(r)$

through the hydrostatic integration. This is achieved by reformulating the equation for the hydrostatic equilibrium (2.32) as a differential equation for the pressure

$$P'(r) = \rho[P(r), T(r)] \cdot g(r). \tag{5.6}$$

The resulting profiles for temperature, density and pressure after the hydrostatic integration and the mapping to the hydrodynamic grid in *SLH* are shown in Fig. 5.10. The original profile from the stellar evolution code is also shown for comparison.

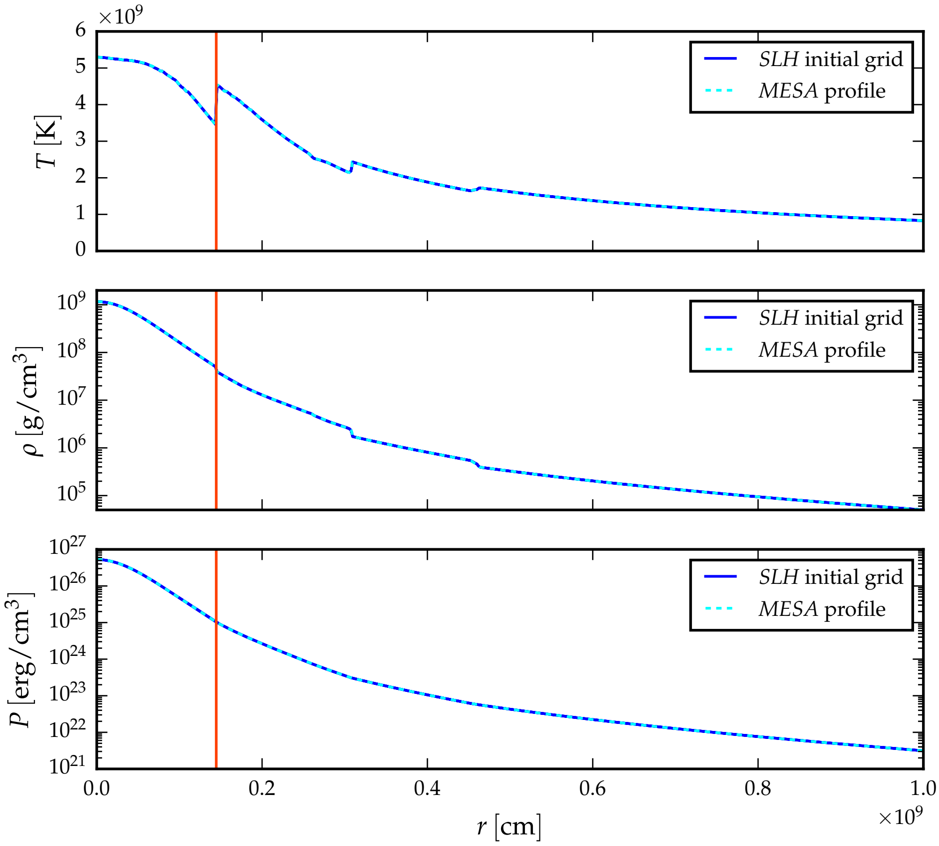

Figure 5.10 | Initial profiles of temperature $T(r)$, density $\rho(r)$ and pressure $P(r)$ after mapping from the 1D stellar evolution code *MESA* to the hydrodynamics code *SLH*. The red line indicates the starting point of the hydrostatic integration. The dashed line shows the corresponding model from the stellar evolution code.

In principle the starting point of the integration can be chosen arbitrarily. However, for our initial model there is a sharp jump in the temperature profile at $r = 1.45 \times 10^8$ cm, where silicon core burning takes place. This is an important feature of the initial model and should be reproduced as accurately as possible. Therefore, we start the integration of the hydrostatic profile at this temperature jump indicated by the red line in Fig. 5.10 and integrate the profile towards the inner and outer boundaries of the model. The resulting initial model accurately represents the original profiles from the corresponding stellar evolution model.

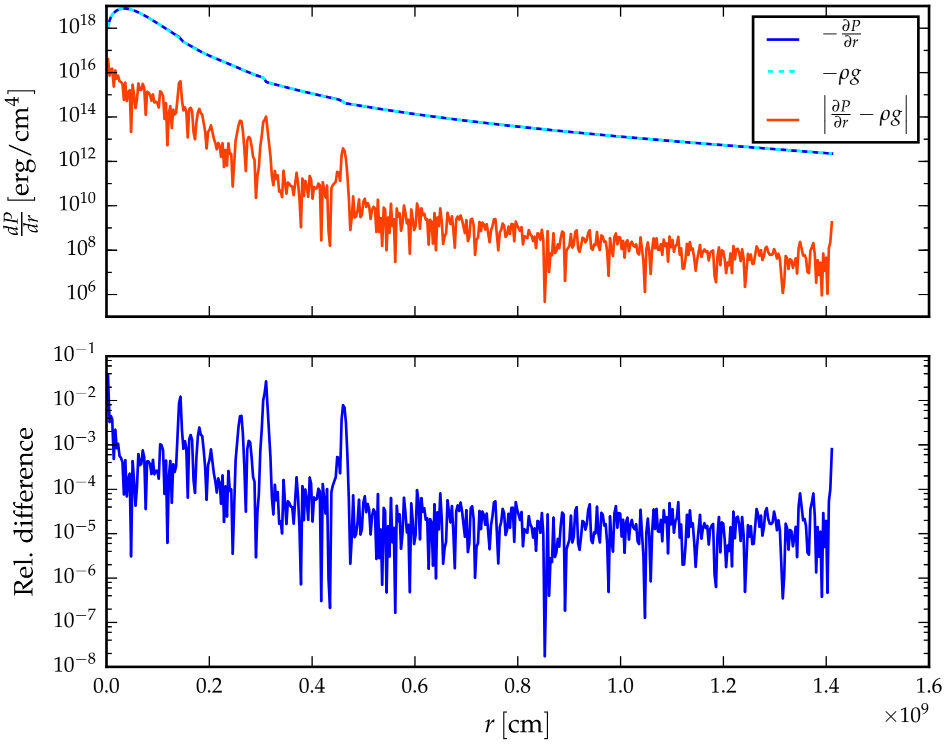

Figure 5.11 | Quality of the hydrostatic equilibrium after the hydrostatic integration and the mapping to the hydrodynamics code *SLH*. The upper panel shows the absolute error and the individual contributions of the pressure gradient $\partial P/\partial r$ and the gravitational force ρg. The lower panel shows the relative difference with respect to the pressure gradient.

Fig. 5.11 shows the quality of the hydrostatic equilibrium that is obtained by this integration process. The upper panel shows the contributions from the gravitational force ρg and the pressure gradient $\partial P/\partial r$ to the hydrostatic equilibrium. In addition to that, it also shows the absolute difference between those two contributions. The relative error is shown in the lower panel. Across the whole model the relative error rarely exceeds 10^{-3}. The largest deviations from hydrostatic equilibrium are found in the center of the model, which is mainly due to the sharply decreasing value of the gravitational acceleration towards the center. Starting a hydrodynamic simulation from this initial model leads to slowly increasing Mach numbers in the convective regions while the Mach numbers in radiative regions typically stay below 10^{-3} during the whole simulation. A detailed analysis of the convective pattern and a comparison with the Brunt-Väisälä frequency of the original stellar evolution model is given in Sec. 5.3.1.

5.2.2 Reaction rates

Another important ingredient of the model are the reaction rates that are used in the nuclear network. Unfortunately many reaction rates are somewhat uncertain and the reaction rates used in our version of the network are not fully consistent with the ones used in the stellar evolution code *MESA*. In this section, we are going to analyze the impact of those discrepancies in the reaction rates on the neutronization rate and the energy generation rate of the nuclear network.

As mentioned in Sec. 4.2.2 the original implementation of the nuclear network in *CASTRO* and *MAESTRO* still uses the ^{56}Ni electron capture rate from Mazurek (1973); Mazurek et al. (1974)[1], whereas the nuclear network in *MESA* uses the more recent electron capture rate from Langanke and Martinez-Pinedo (2000, henceforth LMP (2000)). During this project the electron capture rate from LMP (2000) was also implemented in our version of the nuclear network[2]. A comparison of those two electron capture rates for our initial *SLH* model is shown in Fig. 5.12. The dashed red line once again indicates the abrupt jump in our temperature profile near the base of the silicon core burning region. The abrupt drop of the Mazurek rate near 3.1×10^8 cm is due to the cutoff at low densities.

The more recent electron capture rate from LMP (2000) is higher than the electron capture rate from Mazurek (1973); Mazurek et al. (1974) almost across the whole model. Especially in the central region, where electron captures are most important, the rate from LMP (2000) is roughly a factor of 7 higher. This should have a large impact on the neutronization rate and hence lead to a significant speed up of the evolution towards core collapse. In addition to that, we can also see that the abrupt change in the temperature profile near the base of the silicon burning region has a significant impact on the electron capture rate from LMP (2000). However, this feature is not visible for the reaction rate from Mazurek (1973); Mazurek et al. (1974).

To assess the impact of the reaction rates on the neutronization rate and energy generation rate of the network, we did a network only test run of the nuclear network in *SLH* for both cases (LMP rate and Mazurek rate) starting from the initial profiles shown in Fig. 5.10. For this purpose we used a network time step of 0.01 s, which roughly corresponds to the time step that is used for the simulations of silicon burning in Sec. 5.3 and 5.4. Since our version of the nuclear network is not fully consistent with the initial composition from the stellar evolution model, there is an initial transient where the chemical composition adjusts to the reaction rates in our version of the network. Therefore, we run the network for 50 time steps each before we analyze the neutronization rate and the energy generation rate. At this point the neutronization rate and the energy generation rate of the network have stabilized. Fig. 5.13 shows the resulting rate of change in the electron fraction dY_e/dt in the upper panel and the

---

[1] Meanwhile the reaction rate from Langanke and Martinez-Pinedo (2000) was also implemented in *CASTRO* and *MAESTRO*.

[2] Our implementation of the nuclear network is based on the nuclear network in the Microphysics module of *CASTRO* and *MAESTRO*. During the reimplementation of the network into *SLH* a substantial bug was discovered that prevented the proper neutronization during silicon burning. The bug-fix for the proper neutronization has been pushed to the *CASTRO* and *MAESTRO* repository.

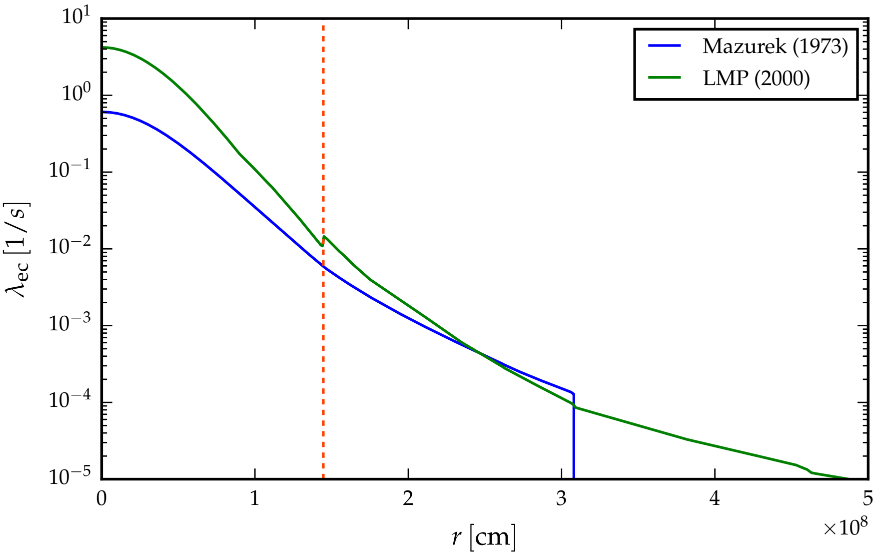

Figure 5.12 | Comparison of the ^{56}Ni electron capture rate λ_{ec} between Langanke and Martinez-Pinedo (2000) and Mazurek (1973); Mazurek et al. (1974) for the initial profiles shown in Fig. 5.10. The dashed red line indicates the sharp jump in the temperature profile near the base of the silicon core burning region.

energy release rate dE/dt in the lower panel. For comparison we also show the energy release rate from the network in the stellar evolution code *MESA*.

As expected the electron fraction Y_{e} decreases much faster in the case of the LMP (2000) rate than for the Mazurek rate, especially in the central region inside of $r = 1.4 \times 10^{8}$ cm. In addition to that, the energy loss rate in the center is almost one order of magnitude higher for the rate from LMP (2000). Further outside than 2.0×10^{8} cm there is almost no difference in the energy generation rate, since the difference between the reaction rates is much smaller in this region. However, there are still large discrepancies in the energy generation rate between our version of the nuclear network and the nuclear network in *MESA*. These differences are mainly due to the fact that we are using the rates that were provided by F. Timmes as part of the *approx21* network, whereas the network in *MESA* mainly uses reaction rates from the JINA Reaclib Database (Cyburt et al., 2010). We attempted to also implement the JINA Reaclib rates into our version of the network, however, the network did not converge with the JINA Reaclib rates. Unfortunately we were unable to resolve these issues with the JINA Reaclib rates. Therefore, we still use the reaction rates provided by F. Timmes for our production runs in Sec. 5.3 and 5.4, even though they are not fully consistent with the reaction rates in *MESA*. Another reason for those discrepancies could be the different treatment of neutrino losses between the network in *SLH* and the network in *MESA*. This is further explained in Sec. 5.3.4.

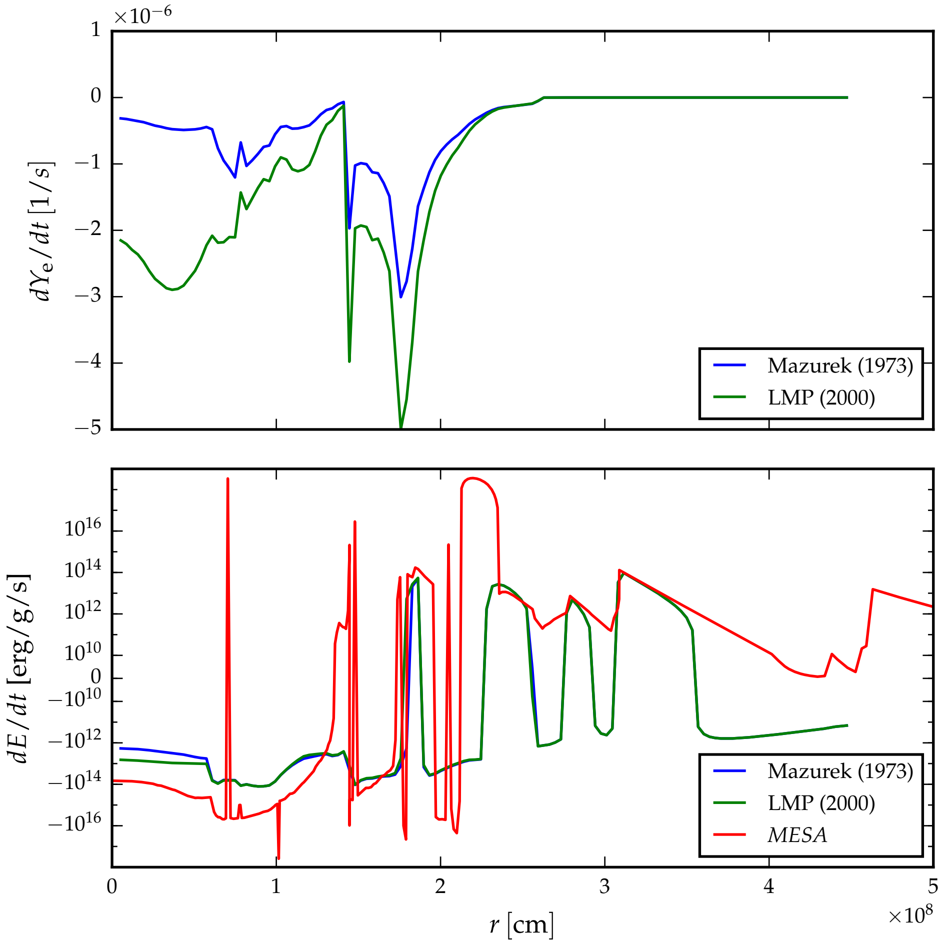

Figure 5.13 | Comparison of the Mazurek rate and the LMP (2000) rate for a network only run (see text for a more detailed description). The upper panel shows the rate of change in the electron fraction dY_e/dt, while the lower panel shows the energy release rate dE/dt.

5.3 Modeling of Si burning in 2D

After a detailed discussion of the initial setup in the previous section, we are now going to present the results of the 2D simulations of silicon burning. The main goal of those 2D simulations was to verify that our modeling approach is working as intended. In comparison to the 3D simulation in Sec. 5.4 those 2D simulations are much cheaper, which means we were able to run a number of different test setups with the computing resources that are available locally at the Heidelberg Institute for Theoretical Studies (HITS). Since most of the gravity solvers from Sec. 5.1 are not fully implemented in 2D, we do not use any gravity solvers for the following 2D simulations and assume that the gravitational potential is constant in time. In addition to that, we used the CFL_{ug} time-step criterion (see Sec. 4.1.2) for all these simulations, because we start our simulations with zero initial velocity and let convection develop. This means we can not use the less restrictive CFL_u, at least not at the start of the simulation.

5.3.1 Development of convection zones

In the first 2D test setup the nuclear reaction network was disabled. The main goal of this test run was to verify that the *SLH* model develops convection as predicted by the Brunt-Väisälä frequency of the initial model from the stellar evolution code *MESA*. This 2D setup was also used to test different grid geometries, boundary conditions, hydrodynamic fluxes, hydrodynamic reconstruction schemes and well-balancing (cf. Sec. 4.1.1).

In the following, we will only show the results for the most promising model setup. This model was run on a *Cartesian grid* geometry with 1024^2 cells and *constant ghost cell* boundary conditions, i.e. the ghost cells at the outer boundary of the grid are initialized with values from the initial profile and never changed during the whole simulation. These boundaries have the advantage that they accurately fulfill the hydrostatic equilibrium at the beginning. However, this also means that they do not adjust to changes in the stellar structure, which could lead to problems if the structure of the star changes drastically during the simulation. The model uses a standard *Roe solver* (Roe, 1981) without the low-Mach preconditioning that is also available in *SLH*. *Parabolic reconstruction* was used, since this resulted in a slower growth of the Mach numbers and also conserved the initial hydrostatic equilibrium more accurately than the linear MUSCL reconstruction scheme. Cargo-Leroux well-balancing (Cargo and Le Roux, 1994; Edelmann, 2014) was also tried with this setup, however, this destabilized our model and was therefore disregarded. The recently implemented Chandrashekar-$\alpha\beta$ well-balancing based on Chandrashekar and Klingenberg (2015) was not yet suitable for realistic stellar profiles, however, this might be something to look into in the future. Therefore, we *do not use any well-balancing* for the simulations presented here. The computational domain for this simulation was 1.0×10^9 cm. This is much larger than most of the other simulations presented in this thesis. The resulting spatial distribution of the Mach number after $t = 203.90$ s, which corresponds to 22 203 hydrodynamic time steps, is shown in Fig. 5.14.

It is remarkable that the distribution of the Mach number is still almost completely symmetric to the coordinate axes after more than 20 000 hydrodynamic time steps. This artificial symmetry of the model could be broken by using initial perturbations in either the velocity (Zingale et al., 2009) or the temperature (Gilet et al., 2013), which would also speed up the development of convection. However, this symmetry also indicates that the initial hydrostatic equilibrium is preserved very well. At this point of the simulation several convective zones have developed, while the core region inside of 2.0×10^8 cm does not show any signs of convection. To verify that those convective features are consistent with the stellar evolution model we compare with the Brunt-Väisälä frequency of the *MESA* profile that was used for the initial setup in *SLH*. The Brunt-Väisälä frequency for this *MESA* profile, as well as the Brunt-Väisälä frequency of the initial *SLH* model[3]

---

[3] The oscillations in the Brunt-Väisälä frequency of the *SLH* model are due to inaccurate evaluation of the temperature gradients.

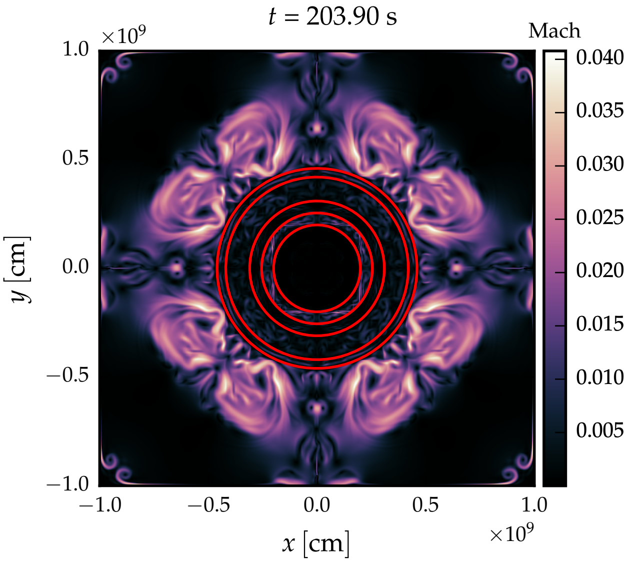

Figure 5.14 | Distribution of the Mach number for an *SLH* run without chemical species and nuclear network at $t = 203.90\,\mathrm{s}$. Convection develops in the regions predicted by the Brunt-Väisälä frequency (see Fig. 5.15). The red circles, which indicate the borders between convective and non-convective zones, correspond to the blue lines in Fig. 5.15.

are shown in Fig. 5.15. Regions that are expected to be convective are shaded in blue, whereas white regions are expected to be stable. The blue lines that mark the borders of the convective zones correspond to the red circles in Fig. 5.14.

Overall the Brunt-Väisälä frequency of the initial *SLH* matches reasonably well with the Brunt-Väisälä frequency of the stellar evolution model. The convective features of the *MESA* profile as predicted by the Brunt-Väisälä frequency are accurately reproduced in our hydrodynamic setup (cf. red lines in Fig. 5.14). Therefore, we conclude that in terms of the convective properties our *SLH* model is a good representation of the stellar evolution model that was used for the mapping. In the following sections, we can now start to put more realistic physics into our model.

5.3.2 Identification of the hydrostatic burning zones

In this section, we present the first 2D test run that also includes the *approx21* nuclear network, which was reimplemented into *SLH* from the *CASTRO* and *MAESTRO* hydrodynamics codes (see Sec. 1.1.2). It should be noted that this simulation was run with the original version of the reimplemented network, which means that it does not yet contain the bug fix that enables the neutronization in the central region of the model. In addition to that, it still uses the electron capture rate from Mazurek (1973); Mazurek et al. (1974) and not the more recent one from Langanke and Martinez-Pinedo (2000). However, this is not a major issue for the purpose of identifying the hydrostatic burning zones. Follow-up simulations with the updated nuclear network are shown in later parts

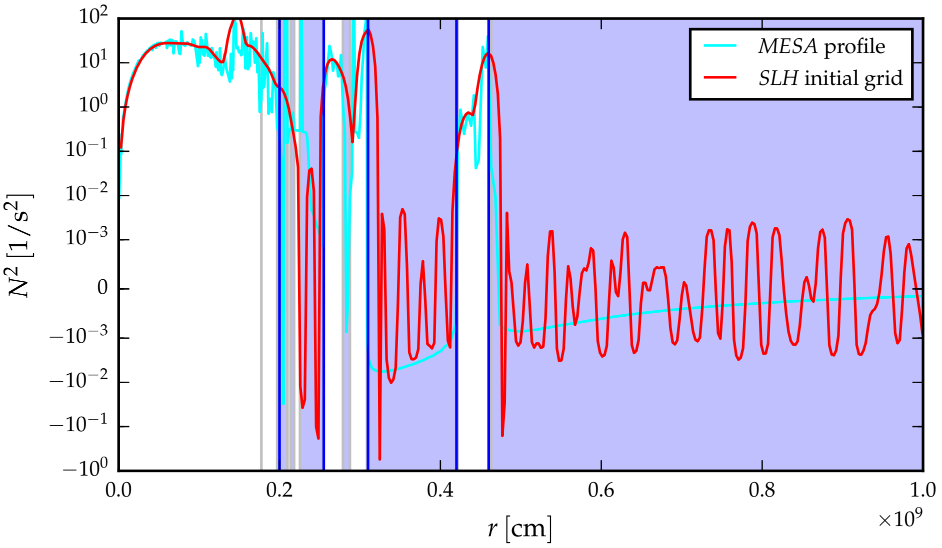

Figure 5.15 | Brunt-Väisälä frequency of the *MESA* profile that is used as starting point in the *SLH* simulations. For comparison the Brunt-Väisälä frequency for the initial *SLH* grid file is also shown. The differences in the Brunt-Väisälä frequency are mainly due to inaccuracies in the evaluation of the temperature gradients. White regions are expected to be stable, blue shaded regions are expected to be convective. The blue lines that separate the major convective regions correspond to the red lines in Fig. 5.14.

of the thesis.

This simulation was performed on a 512^2 *Cartesian grid* with a computational domain of 5.0×10^8 cm. As established in the previous section we are using *constant ghost cell* boundary conditions and a standard *Roe hydro flux* with a *parabolic reconstruction* scheme and *no well-balancing*. The main purpose of this simulation was to identify the convective silicon burning region, so that we can restrict the computational domain appropriately. Fig. 5.16 shows the initial temperature profile at $t = 0$ s in the first panel[4] and the distribution of selected chemical isotopes at $t = 347.45$ s in the following three panels. From this point onward, ^{28}Si + ^{32}S refers to the compound abundance of ^{28}Si and ^{32}S.

The convective oxygen burning shell can clearly be identified in the region from 3.1×10^8 cm to 4.5×10^8 cm, which typically takes place at temperatures between 1.5 GK and 2.7 GK (see Sec. 3.1.3). In this region, the isotope ^{16}O is consumed and intermediate mass elements such as ^{28}Si and ^{32}S are produced. There are also some small traces of even heavier isotopes such as ^{36}Ar in this burning region. In the region outside of 4.5×10^8 cm neon shell burning takes place at temperatures between 1.2 GK and 1.8 GK. The carbon burning region would be even further outside and is not visible in this plot. The convective silicon burning shell can be identified in the region between 1.7×10^8 cm

---

[4] In all our simulations, $t = 0$ s refers to the initial *SLH* model after mapping to the hydrodynamic grid.

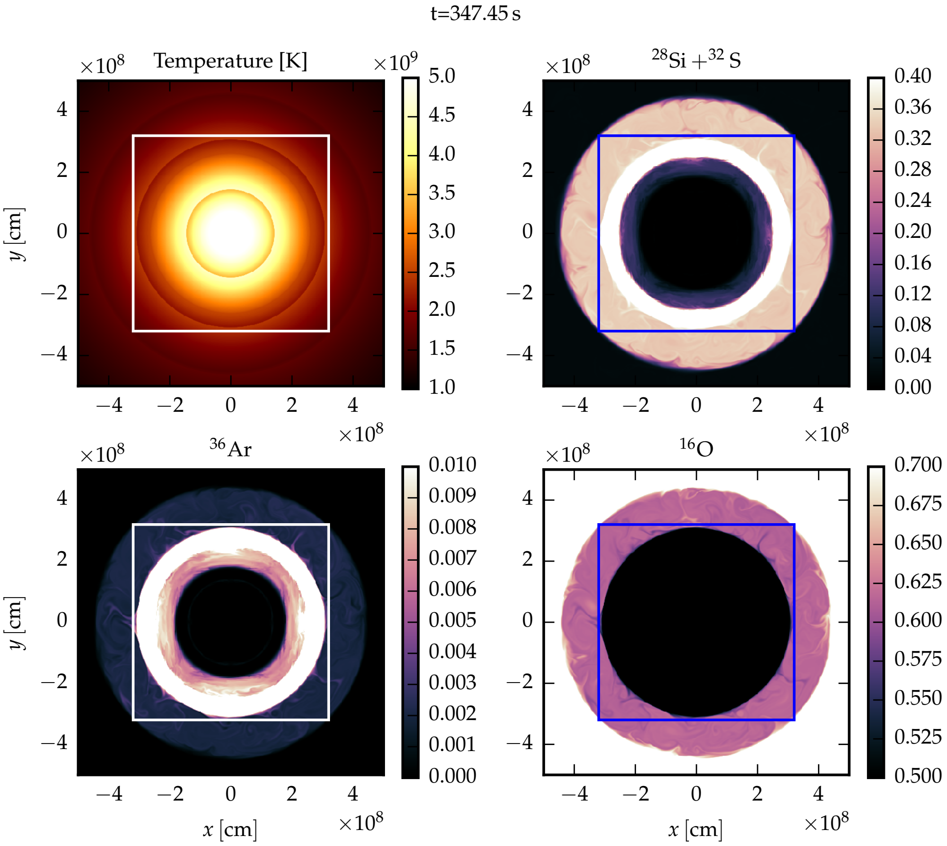

Figure 5.16 | The first panel shows the temperature profile at the start of the simulation in *SLH*. The following panels show the abundance distributions for selected chemical isotopes at $t = 347.34\,\mathrm{s}$. The restricted physical domain at $r = 3.2 \times 10^8\,\mathrm{cm}$, which is used for follow-up simulations is indicated by white/blue lines.

and 2.5×10^8 cm, where the abundances of ^{28}Si, ^{32}S and ^{36}Ar abundances get almost completely depleted in favor of iron group elements such as ^{54}Fe and ^{56}Fe. This also coincides very well with the expected temperature range of silicon burning between 2.8 GK and 4.1 GK. Even further inside at temperatures above 5.0 GK full NSE of the strong and electromagnetic interactions is established (see Sec. 2.2.1). This regime is only very roughly approximated in our nuclear reaction network with few representative species with varying neutron excess.

Now that we have identified the major burning regimes, we are going to further restrict the computational domain as indicated by the white and blue lines at 3.2×10^8 cm in Fig. 5.16. This restricted computational domain covers the entire convective silicon burning region and also large parts of the oxygen burning shell. Decreasing the computational domain to the relevant parts of the simulation has the advantage of improving the resolution in the silicon burning, while keeping the number of grid cells constant.

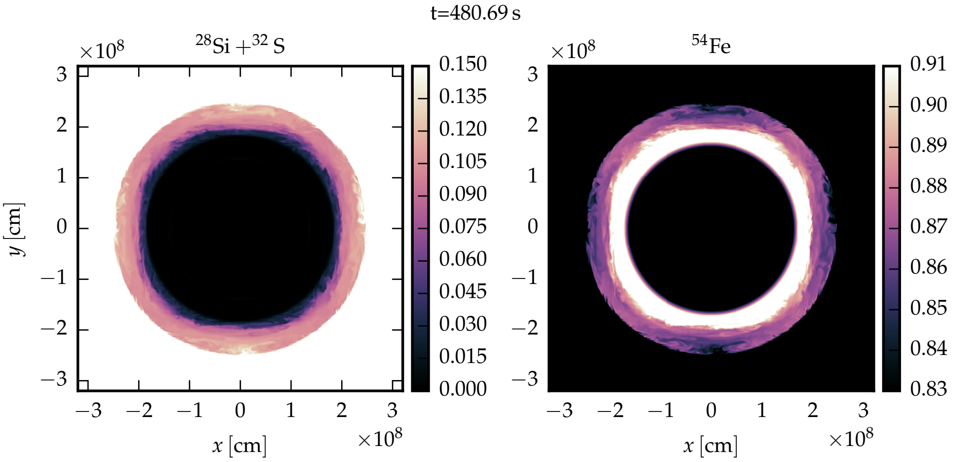

Figure 5.17 | Abundance distributions of ^{28}Si + ^{32}S and ^{54}Fe for the restricted physical domain as indicated by the white/blue lines in Fig. 5.16.

This is illustrated in Fig. 5.17, which shows abundance distributions of ^{28}Si + ^{32}S and ^{54}Fe for a simulation using the restricted computational domain of 3.2×10^8 cm with 512^2 grid cells. In addition to that, this simulation already uses the updated nuclear reaction network with the bug fix that enables proper neutronization in the central region of the model and the more recent electron capture rates from Langanke and Martinez-Pinedo (2000). This setup once again uses *constant ghost cell* boundary conditions and the standard *Roe hydro flux* with a *parabolic reconstruction* scheme. As before, *low Mach number preconditioning and well-balancing are disabled.* In addition to that, a constant opacity of $\kappa = 0.05\,\mathrm{cm^2\,g^{-1}}$ was assumed for this simulation. This value is an upper limit for the opacity as inferred from the *MESA* profile that was used for the mapping to the hydrodynamic grid.

As we can see in Fig. 5.17 the convective silicon burning region is well resolved in this simulation. The combined abundances of ^{28}Si and ^{32}S decrease towards the base of the convective silicon burning region at 1.7×10^8 cm, whereas ^{54}Fe and other iron group elements are produced. We have now established a 2D setup that accurately captures the nuclear processes and the hydrodynamic evolution. The properties of this model are discussed in more detail in the following sections.

5.3.3 Frequency of network evaluations

In this section, we continue the analysis of the model from the previous section with the computational domain of 3.2×10^8 cm. In addition to that, we also present a model that only evaluates the nuclear reaction network every 50 hydrodynamical time steps, which corresponds to an evaluation of the network roughly every $\Delta t_{\mathrm{net}} = 0.37$ s in physical time. Between two network evaluations the energy generation rate and the rate of change in the network species is stored. These stored values are used for updates of the composition and the energy until the next full evaluation of the network.

This method significantly reduces the amount of computational resources that are spent on the reaction network, while still being reasonably accurate. Evaluating the network in every hydrodynamical time step consumes roughly 50 % to 60 % of the computational resources, depending on the other details of the setup. With a very low frequency of network evaluations this fraction could be reduced to being completely negligible, however, this could lead to a significant reduction in the accuracy of the network. For a moderate frequency of network evaluations (e.g. every 50 hydrodynamical time steps, as above) the results are usually still reasonably accurate and the contribution of the network to the overall computing time can be reduced below 10 %. Obviously the values given here are specific to the setup presented here and it might already be different, if another stellar evolution model is used for the mapping to the hydrodynamic grid. Therefore, it is advised that the frequency of network evaluations is readjusted when the initial setup is changed.

Fig. 5.18 shows changes with respect to the initial *SLH* model (subscript 0) for radially averaged profiles of density ρ, temperature T, electron fraction Y_e and energy E after $t \approx 481\,\mathrm{s}$ for the two models mentioned above. In addition to that, changes for the corresponding *MESA* model are also shown for comparison. We can see that there is almost no difference between the two *SLH* models, which means that evaluating the nuclear network every 0.37 s should be good enough for this particular initial model. However, in both *SLH* models there are no signs of core contraction as it would be expected from the stellar evolution model. This is most obvious in the first panel, where we can see that the central density in the *MESA* model has increased by more than 30 %, whereas there is almost no change in the central density of the *SLH* models. There are also obvious differences between the *SLH* models and the *MESA* model in the temperature, energy and Y_e profiles, however, this is in part also due to the missing contraction of the *SLH* models.

This missing contraction in the 2D models is due to problems with the geometry of the setup in 2D. In Sec. 5.4.2 we will see that the 3D model, which contains the correct geometric information, shows contraction similar to the stellar evolution model from *MESA*. Consequently, the profiles for this 3D simulation are overall much more similar to the stellar evolution model (see Fig. 5.22). A detailed discussion of the problems with the chosen 2D setup and a possible solution for this problem are presented in Sec. 5.4.3 after analyzing the properties of the 3D model.

5.3.4 Network properties

In this section, we take a look at the spatial distribution of the energy generation rate dE/dt and the rate of change in the electron fraction dY_e/dt for the 2D model with the reduced frequency of network evaluations from the previous section after 480.99 s of simulation time (see Fig. 5.19). The results are comparable to the radially averaged profiles shown in Fig. 5.13, however, this time we show the spatial distribution on the grid, which gives us further insights.

The base of the oxygen burning shell at $r = 3.1 \times 10^8\,\mathrm{cm}$ is once again clearly visible

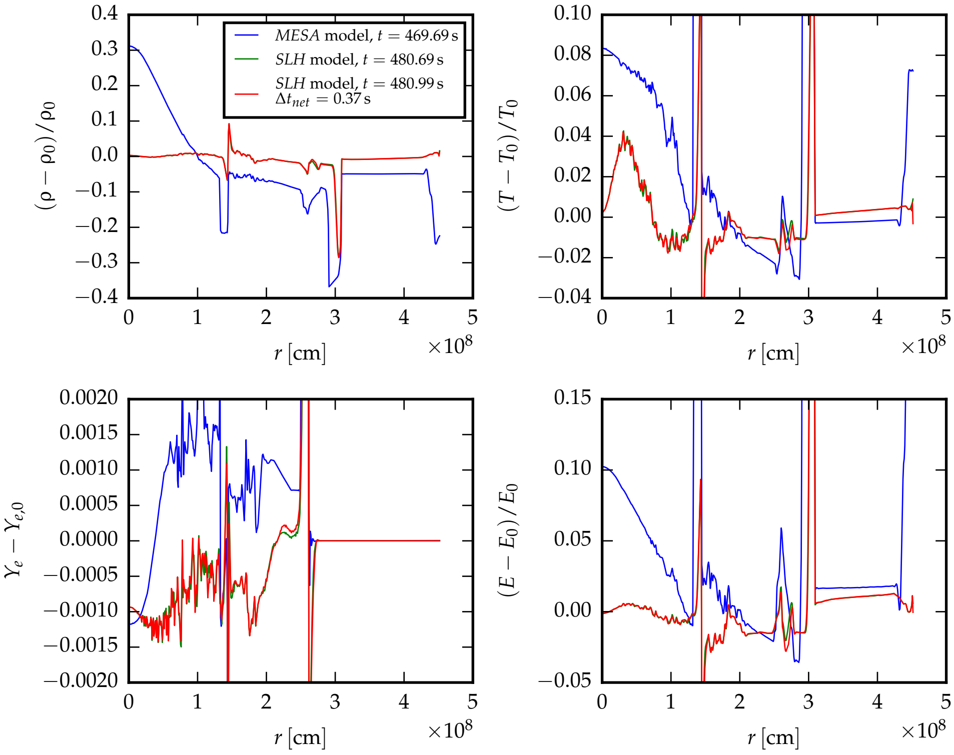

Figure 5.18 | Changes in the density, temperature, electron fraction and energy profiles with respect to the corresponding initial model (subscript 0). The green line shows a 2D *SLH* model where the network is evaluated in every hydrodynamical time step. The red line shows a 2D *SLH* model with a reduced frequency of network evaluations. For comparison the profile for the corresponding model from the stellar evolution code *MESA* is also shown.

with large positive contributions to the energy generation rate. Similarly, a thin region with positive energy generation can be identified around $r = 1.8 \times 10^8$ cm near the base of the convective silicon burning region. Between those two regions is the convective silicon shell, where the energy generation is predominantly negative due to energy losses from neutrino cooling. However, some spots with positive energy generation are sprinkled in, where nuclear burning still takes place. In the NSE region at the center of the model the energy generation rate is again negative due to neutrino losses. At this point it should be noted that only thermal neutrino losses according to Itoh et al. (1996) are implemented in the current version of the nuclear network. Neutrino losses due to weak nuclear reactions are not considered in our version of the network. This could be one of the reasons for the discrepancy in the energy generation rate in the lower panel of Fig. 5.13.

In addition to that, it is somewhat interesting that outside of a radius of 1.9×10^8 cm there is no decrease in the electron fraction. This is due to the fact that the network only contains symmetric α-chain species up to ^{52}Fe. Therefore, a decrease in Y_e in

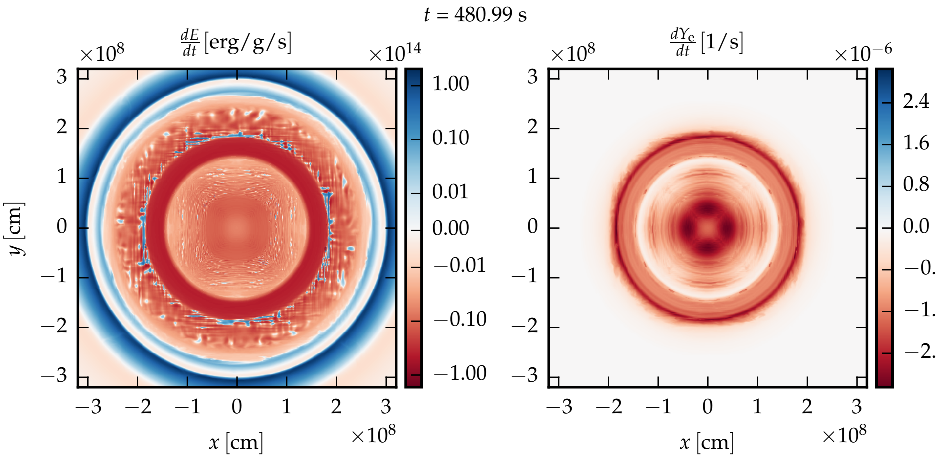

Figure 5.19 | Spatial distribution of the energy generation rate dE/dt and the rate of change in the electron fraction dY_e/dt for the 2D model with the reduced frequency of network evaluations from Sec. 5.3.3 after $t = 480.99\,\mathrm{s}$.

this network is only possible after iron group elements have been formed during silicon burning. As discussed in Sec. 3.1.3 there should already be a substantial neutron excess during oxygen burning due to β^+-decays and electron captures. However, those reactions are not included in this nuclear network. This leads to a rapid decrease in the electron fraction at the base of the silicon burning region, where substantial amounts of iron group elements are produced.

Another rapid decrease in the electron fraction takes place near the center of the NSE region. This is the region, where the electron captures on ^{56}Ni from Sec. 5.2.2 become important. Note that the neutronization in the core is only roughly approximated in this network by successively converting iron group elements into more neutron rich isotopes with ^{56}Cr being the most neutron rich species (see Sec. 4.2.2).

5.4 Modeling of Si burning in 3D

In the previous sections, we have defined and thoroughly tested our modeling approach in 2D with somewhat satisfying results. The main problem with our 2D simulations was that the models do not contract as it would be expected from the stellar evolution model. Unfortunately we were unable to resolve this issue with our modeling approach in 2D. Nevertheless we decided to perform a simulation of silicon burning in 3D with the computing time that was allocated for this project on JUQUEEN at Forschungszentrum Jülich. Since 3D simulations are much more expensive than the 2D runs from the previous section we were only able to run a single model in 3D after doing some initial tests with the setup. Perhaps the most interesting result of this 3D run was the fact that the 3D model contracts in a similar fashion as the stellar evolution model (see

Sec. 5.4.2), even though we never managed to obtain contraction in 2D. As it turns out, we only noticed a deficiency of our 2D setup after we successfully simulated the contraction of the core in 3D. The problems with our 2D modeling setup and possible solutions are discussed in Sec. 5.4.3.

The setup for our 3D run is based on the 2D model that was described towards the end of Sec. 5.3.2. This means we are using *constant ghost cell* boundary conditions and the standard *Roe hydro flux* with a *parabolic reconstruction* scheme once again. In addition to that, we use the restricted computational domain of 3.2×10^8 cm, which was established in Sec. 5.3.2. A full evaluation of the nuclear network was performed every 35 hydrodynamical time steps. With a CFL_{ug} time step of roughly 0.01 s this corresponds to a network evaluation every 0.35 s, which is comparable to the 2D setup with the reduced frequency of network evaluations shown in Fig. 5.18. Therefore, the treatment of nuclear reactions should be reasonably accurate. In contrast to the 2D setup, we are using the *standard Poisson solver* on a Cartesian grid for this 3D simulation. The solution of the monopole solver is used as initial guess for the Poisson solver in the first time step. This also serves as a first test case for the performance of the gravity solver in a realistic physical application.

Originally we wanted to use a grid with 512^3 grid cells, so that we have the same resolution as in 2D. However, it turned out that we could not afford this resolution with the given computational resources. Therefore, we had to reduce the number of grid cells to 256^3 and even then we only managed to simulate 154.41 s of silicon burning in 3D with 3.9×10^6 core-hours. With this setup a full simulation of the roughly 1000 s until core-collapse would require around 25.4×10^6 core-hours in total, which is rather expensive and not possible within the current computing time proposal. Therefore, this 3D setup presented here only serves as a proof of concept. Substantial improvements of our 3D modeling approach are clearly necessary to reduce the required computational resources. Suggestions for future improvements of the 3D modeling, which hopefully reduce the computing time, are given in Sec. 5.4.4.

5.4.1 Development of convection

To get a first impression on the quality of the 3D model, we once again look at the convective features of the model and compare it to the 2D model with the reduced frequency of network evaluations from Sec. 5.3.3. This comparison is particularly interesting, because the turbulent energy cascade is inherently inverted in 2D. Same as in Fig. 5.17 we use the isotope ^{54}Fe as tracer for the convective silicon burning region. The resulting Fig. 5.20 shows that the convective pattern is indeed quite different from the convective pattern of the 2D simulation at the same time. Convection in 2D develops much faster and the mixing is a lot more violent than in the 3D case. However, this could partially also be due to the higher resolution of the 2D run.

For a more detailed analysis of the differences between the 2D simulation and the 3D simulation we show the power spectrum for both cases in Fig. 5.21. In both cases the expected $k^{-5/3}$ Kolmogorov spectrum (Kolmogorov, 1941) develops, where k is the wavenumber obtained via Fourier transformation of the length scales. However, in 2D

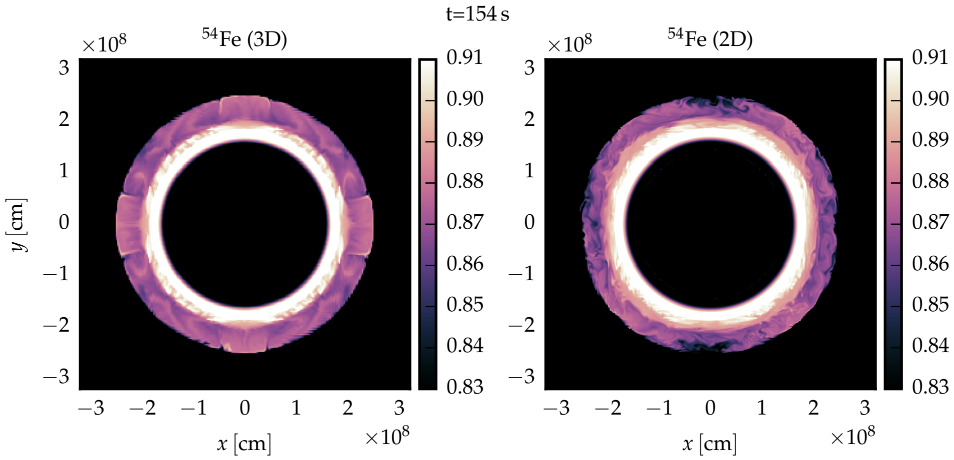

Figure 5.20 | Abundance distribution of ^{54}Fe as tracer for the convective silicon burning region at $t = 154$ s. The left panel shows the 3D model, which is discussed in the current section. The right panel shows the 2D model with the reduced frequency of network evaluations from Sec. 5.3.3 for comparison. Convection clearly develops differently in those two cases.

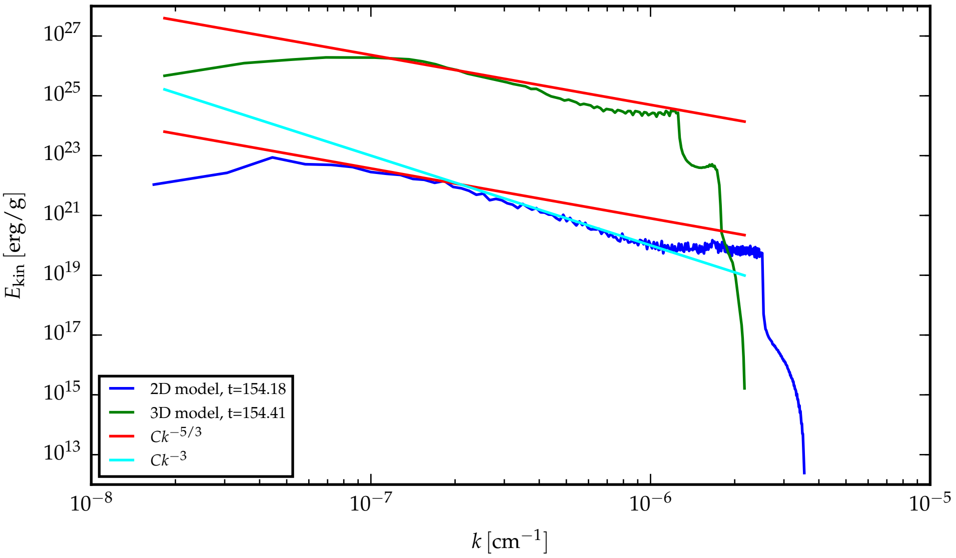

Figure 5.21 | Turbulent energy spectrum for the 2D and 3D models shown in Fig. 5.20. In addition to that, we also show the lines for the $k^{-5/3}$ spectrum as predicted by Kolmogorov (1941) and for the k^{-3} power-law from the forward enstrophy (i.e. the squared vorticity of the velocity field) cascade.

the peak of the spectrum is shifted towards smaller k values as a consequence of the inverted turbulent energy cascade. At around $2.0 \times 10^{-7}\,\mathrm{cm}^{-1}$ the power-law in 2D is broken and changes from $k^{-5/3}$ to k^{-3}. This is due to the fact that only enstrophy (i.e. the squared vorticity of the velocity field) is transported in a forward cascade in this case. Similarly, the power-law in 3D is broken for $k < 1.0 \times 10^{-7}\,\mathrm{cm}^{-1}$, which suggests that kinetic energy is transferred to small scales in a forward energy cascade. This crucial difference in the turbulent energy transport is further discussed in Hanke et al. (2012) for a power-spectrum in spherical harmonics. As discussed in Sec. 3.2.1 this inverted turbulent energy cascade is particularly important for follow-up simulations of the core-collapse supernova, since it leads to larger turbulent structures in 2D and artificially enhances shock revival.

5.4.2 Contraction and neutronization of the 3D model

In the previous section we discussed the differences in the convective properties between the 3D model and the 2D model from Sec. 5.3.3. Now we are concerned with the contraction and neutronization of our 3D model. In Fig. 5.18 we have shown that the 2D model did not contract as it would be expected from the stellar evolution model. We show similar plots for the 3D model in Fig. 5.22. For comparison we also added a 2D model that uses a monopole gravity solver, but is otherwise identical to the models discussed in Sec. 5.3.3. This treatment of using a monopole solver with our 2D setup is not very consistent with the geometry imposed by the 2D simulation (see Sec. 5.4.3). Nevertheless, we used the monopole gravity solver in this case to demonstrate that the missing gravity solver in the 2D simulations from Sec. 5.3.3 is not the reason for the missing contraction in 2D.

Even though the 2D simulations from Sec. 5.3 have progressed much further in time, we can see that this 3D simulation behaves significantly different from all 2D simulations in terms of core contraction and neutronization. This is most obvious in the first panel of Fig. 5.22, which once again shows the relative difference in the density with respect to the density of the initial *SLH* model. Clearly in contrast to the 2D models there is now a substantial increase of the density in the central region, which is almost on par with the corresponding model from the stellar evolution code *MESA*. This suggests that our 3D model is in fact contracting on a similar time scale as the *MESA* model. In contrast, the 2D model with the monopole gravity solver barely shows any signs of contraction at this stage of the simulation. The contraction in the innermost center of the *MESA* model is still slightly faster than for our 3D model, which could be due to the remaining discrepancies in the reaction rates and neutrino losses. Since temperature and density are related via the equation of state, it comes as no surprise that the higher densities in *MESA* lead to lower core temperatures than for our 3D model. The temperature profile for the 2D simulation with the monopole solver is completely different due to the missing contraction of the model.

Up to a radius of $1.7 \times 10^{8}\,\mathrm{cm}$ the changes of the electron fraction Y_e in our 3D model are in good agreement with the stellar evolution data. This confirms that at least some of the differences observed in Fig. 5.18 can be attributed to the missing contraction in

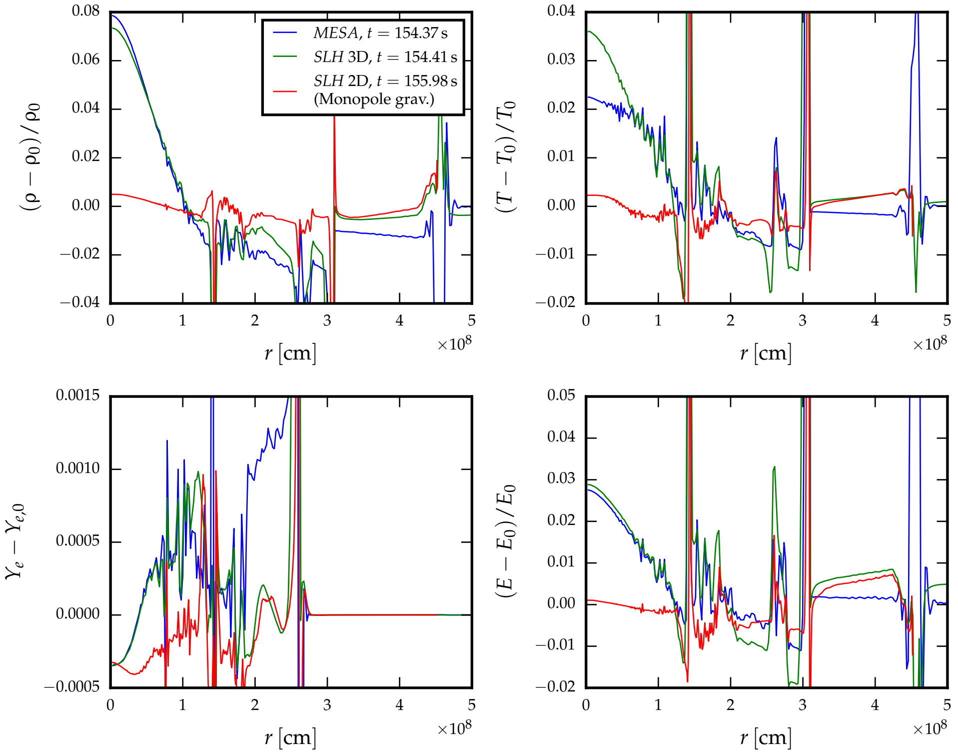

Figure 5.22 | Changes in the density, temperature, electron fraction and energy profiles with respect to the corresponding initial model (subscript 0; cf. Fig. 5.18 for the same plot in the 2D case). For comparison, the profiles for a 2D model with monopole gravity solver and for the model from the stellar evolution code *MESA* are also shown.

the 2D models. In the region from 1.7×10^8 cm to 2.5×10^8 cm the differences between our 3D model and the *MESA* model are more pronounced. This is the region where convective silicon burning takes place (see Sec. 5.3.2). However, our full 3D treatment of convection is clearly much more sophisticated than the parametrized mixing-length theory (see Sec. 1.1.1), which is used as a rough approximation for convection in the 1D stellar evolution code since a full treatment of multidimensional phenomena is not possible. It is very likely that the different treatment of convection leads to changes in the composition, which are then reflected in the Y_e profile. Due to the missing contraction the 2D model with the monopole solver once again has a completely different profile.

A very similar behavior is found for the changes in the internal energy. The *MESA* model and our 3D model are overall in good agreement. Some differences are present in the convective silicon burning region from 1.7×10^8 cm to 2.5×10^8 cm and in the convective oxygen burning region from 3.1×10^8 cm to 4.5×10^8 cm. The increase of the internal energy towards the center is almost completely missing in the 2D model with the monopole solver.

We conclude that our 3D model is a much better representation of the stellar evolution model from *MESA* than the 2D models discussed in Sec. 5.3.3. It would be interesting to follow the 3D model until the onset of core collapse and compare it with the profiles from the stellar evolution model. Unfortunately this is not feasible with our current 3D setup due to limited computational resources. With some of the improvements discussed in Sec. 5.4.4 it might be possible to follow the model until core collapse with significantly reduced computational resources.

5.4.3 Problems with the 2D modeling approach and possible solutions

In Sec. 5.3.3 we identified a common issue of all 2D simulations in this thesis. None of our 2D models show the contraction of the core, which is predicted by the stellar evolution code *MESA*. For a long time we suspected that this might be due to a mismatch in the reaction rates, which is why we tried to include the same reaction rates in our nuclear network as in the nuclear network from *MESA*. Eventually we decided to run a 3D simulation, even though there were still some discrepancies in the reaction networks and we realized that this 3D model did in fact contract on a similar time scale as predicted by the *MESA* model (see Sec. 5.4.2). Obviously this raises the question, what was actually missing in the 2D simulations that we did so far. Apparently the discrepancies in the reaction rates are not the main reason for the missing contraction in the 2D models, since the 3D model shows clear signs of contraction with the same reaction rates.

Another difference between this 3D model and the previous 2D modeling attempts was the treatment of gravity. Since most of our gravity solvers do not work in 2D without further assumptions about the geometry, all of the simulations in Sec. 5.3 were performed under the assumption that the gravitational potential is constant in time. In contrast to that, the 3D simulation in Sec. 5.4 uses the standard Poisson solver on a Cartesian grid (see Sec. 4.3.2 and 5.1.1). Unfortunately with the current implementation of the Poisson solver it is impossible to use it for our 2D setup. Therefore, we tried using the monopole gravity solver with our 2D setup from Sec. 5.3.3 and compared it with the data from our 3D model and the predictions from the stellar evolution code (see Sec. 5.4.2). However, even with the monopole gravity solver the 2D model clearly failed to reproduce the expected contraction.

This lead us to the conclusion that there is a deficiency in the geometrical setup of our 2D simulations. For the 2D simulations in this thesis we assumed that the simulated domain corresponds to a cut through the equatorial plane of a sphere. However, this geometrical setup actually corresponds to an infinite cylinder in z-direction, which is obviously not a very realistic model for a star. This also explains why imposing monopole gravity on our 2D model did not work as intended, since the gravity solver was not consistent with the geometry of our setup. We did not expect that this mismatch in the geometry would have such a large impact on the contraction of the model.

This issue with our 2D setup could be resolved by using a similar setup as in the *LEAFS* hydrodynamics code (Reinecke, 2001, section 5.3), which is mainly used for simulations of Type Ia supernovae. 2D simulations in *LEAFS* are typically performed under the assumption that the simulated domain corresponds to a slice in the rz-plane in cylindrical coordinates. This setup imposes axial symmetry around the z-axis, i.e. the model is rotationally symmetric around the z-axis. With this setup we can construct a 2D model that does in fact correspond to a sphere in 3D. However, this setup requires additional corrections in the flux terms to account for the different size of the cells. Unfortunately these corrections for the flux terms are not implemented in *SLH*, which means that we can not test this hypothesis right now.

Implementing those corrections into *SLH* would hopefully allow us to perform 2D simulations that are more comparable to our 3D model in Sec. 5.4.2. At the moment our 2D simulations are obviously not suited to make predictions for the behavior in 3D. Therefore, in the current state we always rely on 3D simulations, if we want to make statements about the contraction of the model. However, with the current 3D setup we clearly cannot afford to do a large number of 3D simulations. Therefore, it would be useful to have a 2D model that makes accurate predictions for the contraction of the model in 3D. This would make it a lot easier to identify suitable setups for 3D simulations.

5.4.4 Future improvements of our 3D modeling

In Sec. 5.4 we briefly discussed that our 3D simulation is very computationally expensive and that it would require roughly 21.5×10^6 core-hours to continue the simulation until the onset of core-collapse. This is rather expensive and substantial improvements of our 3D setup would be very useful, if we want to cover a longer period of time with our simulations. For this purpose we are first going to analyze, which parts of the code consumed the most computing time in this 3D run.

With the current 3D setup around 80 % of the total computing time is spent on solving the system of hydrodynamic equations, which consists mainly of the solution of the linear system of equations for the implicit solver. Since this is by far the largest fraction of the computing time, we are going to take a closer look at the details of the hydrodynamics after analyzing the contributions of the remaining components of the code.

In Sec. 5.3.3 we mentioned that the nuclear network consumes roughly 50 % to 60 %, if we evaluate the network in every hydrodynamical time step. This means that evaluating the network in every time step would severely limit the time frame that we could cover with our simulation. However, as we showed in Sec. 5.3.3, it is not necessary to evaluate the network in every hydrodynamical time step. In this 3D run, we evaluated the nuclear network only every 35 hydrodynamical time steps, which reduces the required computing time for the network significantly.

In the 3D setup from Sec. 5.4 the network with the reduced frequency of evaluations consumes only around 2 % of the total computing time. In principle this means that further optimization of the nuclear network is not necessary at the moment. However, improving the performance of the network would allow us to evaluate the network more frequently. Therefore, improving the network performance is not our main focus for now, but it might be worthwhile after the issues with the hydrodynamics have been addressed. A particular problem of the nuclear network is the load imbalance in parallelized code, since more steps in the iterative solver are required in regions where the network is more active. However, those regions are typically on the same radius, which means a traditional domain decomposition on a Cartesian grid is not very effective in this case. This decreases the overall performance of the network, because other domains have to wait until the network evaluations in the most expensive domain are finished.

The standard Poisson gravity solver, which was used for this 3D run, consumed around 0.2 % of the total computing time. Therefore, we conclude that the computing time for the gravity solver is almost completely negligible for this setup and further performance optimization of the Poisson solver is not necessary.

We do not know, where the remaining 18 % of the computing time is consumed, since we only tracked the contributions of the components listed above. Other noteworthy contributions could be from the preconditioner of the iterative solver or data input and output (e.g. writing output and restart files). However, since all of this together only contributes less than 18 % to the overall computing time, it is not relevant for us right now.

It should be noted that the timings for the components listed above are not completely accurate, because the timing was only done for the processor with rank 0. Due to load imbalance it is possible that the corresponding routine is still carried out on other processors after the execution of the routine has concluded on the processor with rank 0. Therefore, the contributions of the above components to the overall computing time might actually be slightly larger than what we have measured here.

After analyzing the contributions of each component to the overall computing time, we have now established that most of the computing time in our 3D run is consumed by the iterative solution of the hydrodynamic equations. In the following, we will give several suggestions on how to reduce the contribution of the hydrodynamics to the overall computing time.

Since we start our 3D simulation with zero velocity we are using the CFL_{ug} time step. This is a good choice at the start of the simulation, because the CFL_u criterion is not suitable for simulations with zero velocities (see Sec. 4.1.2). However, in principle it would be possible to switch to the less restrictive CFL_u criterion after convection has fully developed. For our current 3D setup the CFL_u criterion would result in a hydrodynamic time step of $\Delta t_u = 0.026\,\mathrm{s}$. Under the assumption that the contribution of each component to the overall computing time does not change, we could reach the onset of core-collapse with only 9.79×10^6 core-hours. Obviously this is only a rough estimate, since the contributions of each component to the overall time would change, if we reduce the time step.

Another reason why the solution of the hydrodynamics equation is currently so expensive in our 3D simulation is the large number of passive scalars. For each of those passive scalars another system of equations has to be solved in a subsequent step. For a small number of passive scalars this is almost negligible, however, in our 3D setup we have 21 passive scalars for the species that are used in the nuclear network. Some of those species such as ^{1}H, ^{3}He or ^{14}N are at around 10^{-10} during the whole course of the simulation. We could simply omit those species from our passive scalars and set them to a low value during the calls to the nuclear network.

In addition to that, it could be worthwhile to look into the settings of the iterative solver and the time integration. For our 3D setup we have used ESDIRK23 with a BiCGSTAB(5) iterative solver during the Newton-Raphson method. This configuration was optimized for the 2D simulations, but a different setting might be more appropriate for the 3D simulation due to the larger number of grid cells. In particular it might be worthwhile to look into multigrid solvers available in *SLH*, since they perform a lot better for a large number of grid cells.

Recently a faster reconstruction for EoS variables has been implemented into the *SLH* code, which has been shown to improve the performance by a factor of four in an explicit 2D pure hydro test setup. Unfortunately this functionality was not yet fully working for our 3D setup due to some bugs, which have been fixed in the meantime. Therefore, this feature could be used in future 3D runs to further increase the performance of the hydrodynamic part in our simulations.

With the suggestions above it should be possible to substantially improve the performance of our 3D simulations. This would allow us to cover a longer period of silicon burning with our simulations and possibly we could even reach the onset of core-collapse starting from our initial model, which is around 1000 s away from the collapse. It might also be possible to increase the resolution once again to 512^3 so that we have the same resolution as in 2D. We conclude that with some modifications to increase the performance our 3D setup looks very promising. Ideally we would want to run a large number of 3D simulations for stellar evolution models with different mass, however, this requires major improvements in the performance of our modeling approach.

In addition to those improvements for the hydrodynamic part of the simulation, it might also be worthwhile to look into improvements for the nuclear network part. A QSE-reduced nuclear network based on Hix et al. (2007) might be a better choice for the simulations presented here than the effective rate α-chain network, which was used in our simulations. However, implementing such a QSE-reduced network into *SLH* would require considerable effort. Moreover, this would require that the stellar evolution calculations are rerun with this network to ensure that the composition of the stellar evolution model is consistent with our network. Therefore, we conclude that this is a more long-term project after the issues with the hydrodynamic part of the simulation have been addressed.

6 Conclusions and Outlook

The two main goals of this thesis were the implementation of a gravity solver on curvilinear grids into the hydrodynamics code *SLH* and the simulation of silicon burning in 2D and 3D with *SLH*. In the following, we give a short recap to which degree those two goals have been achieved and suggestions for future improvements and applications.

6.1 Implementation of the gravity solvers

During the course of this thesis several gravity solvers have been implemented into *SLH* and thoroughly tested. The most important results of the gravity solver benchmarking in Sec. 5.1 are summarized here.

The implementation of the standard Poisson solver in Cartesian coordinates works very well (see Sec. 5.1.1). This comes as no surprise, since the standard Poisson solver is a well established numerical method. As such, the method itself is well understood and thoroughly tested. In this thesis, we merely verified that our implementation of the Poisson solver in Cartesian coordinates fulfills those high expectations. In addition to that, it also served as a reference for the quality and performance of the other numerical methods in this thesis. This gravity solver was also successfully used in a realistic 3D setup during our simulations of silicon burning (see Sec. 5.4). We conclude that our implementation is robust and suitable for physical applications on Cartesian grids in 3D.

However, implementing a gravity solver that also works on arbitrary curvilinear grids is very challenging. In Sec. 5.1.2 we have shown that this is mainly due to numerical artifacts that manifest itself in the case of discontinuous coordinate transformations during the evaluation of the gradient of the gravitational potential with finite differences. This problem is especially pronounced in the case of cubed sphere grids due to abrupt changes in the derivatives of the metric terms. We already identified this potential problem in our derivation of the finite differences on curvilinear grids in Sec. 4.3.3, however, we did not expect that it would be such a huge issue.

This problem with the metric terms also persists for our newly developed mixed method Poisson solver on curvilinear grids (Sec. 5.1.3), since it is based on the same discretization of the finite differences. Our mixed method Poisson solver still works reasonably well in the case of smoothly varying metric terms, e.g. on sinusoidal grids or spherical grids. This proves that our implementation works in principle, as long as the changes in the metric terms are not too abrupt. However, it is not suitable for grids with strong discontinuities in the coordinate transformation, such as the cubed sphere grid. It should also be noted that the periodic boundary conditions for the spherical grid are not yet implemented, which leads to a periodic pattern in the relative error of the gravitational acceleration originating from the periodic boundary. If this issue is resolved it might be possible to use the mixed method Poisson solver for physical applications on spherical grids. However, this requires more testing of the gravity solver with a realistic physical setup.

To avoid these problems with discontinuous metric terms it might be better to use a gravity solver that does not make explicit use of the finite differences. A cheap gravity solver that works reasonably well regardless of metric terms is a monopole gravity solver (see Sec. 5.1.4), however, it assumes that the gravitational potential is spherically symmetric. This should be fulfilled to a reasonable approximation for most stellar astrophysics applications and more sophisticated gravity solvers are only necessary in the case of large deviations from spherical symmetry. Therefore, this implementation of the monopole solver is in principle suitable for physical applications on arbitrary curvilinear grids as long as the deviations from spherical symmetry are not too large. However, previous simulations of silicon burning (Arnett and Meakin, 2011) and oxygen burning (Müller et al., 2016) suggest that substantial asymmetries develop during late nuclear burning stages. Consequently, the monopole gravity solver might not be accurate enough in this particular case.

Other alternatives that do not explicitly use the discretization of the finite differences are for example higher-order multipole gravity solvers or FFT gravity solvers. The former also considers gravitational quadrupole moments during the expansion of the gravitational potential into spherical harmonics in Eq. (2.28). The latter could be implemented by first mapping to a Cartesian grid and then using a standard FFT gravity solver in Cartesian coordinates. This approach was already used successfully by Schrauth (2015) for the implementation of a gravity solver into the hydrodynamics code *LEAFS*. One of those two alternatives could be implemented into *SLH* during a future project if necessary.

Scaling and performance tests for some of the newly implemented gravity solvers were performed in Sec. 5.1.5. One important result is that a good initial guess for the gravitational acceleration and the gravitational potential substantially reduces the number of iterations for both the mixed method Poisson solver on curvilinear grids and for the standard Poisson solver on Cartesian grids. Usually it is enough to simply use the solution from the previous time step as initial guess for the next time step. At the start of the simulation the solution from the monopole solver can be used as initial guess. Another noteworthy result of the scaling tests is that our implementations of the mixed method Poisson solver and the monopole solver scale very well up to 24 racks (i.e. 393 216 cores) on JUQUEEN, which means that they are also suitable for large scale simulations on massively parallel systems.

6.2 Simulations of silicon burning in 2D and 3D

The second part of the thesis focused on the hydrodynamic simulation of silicon burning in 2D and 3D. For this purpose a model of the 1D stellar evolution code *MESA* was used as initial model for our hydrodynamic simulations with *SLH*. The stellar evolution model that we use is roughly 1000 s away from the onset of core collapse and the inlist for this *MESA* model was provided by Couch et al. (2015).

During the mapping between the 1D stellar evolution model and the 2D and 3D *SLH* models it is particularly important that the models are still in hydrostatic equilibrium after the mapping process. Otherwise unphysical readjustments of the initial profiles may occur right after the mapping. For the hydrostatic integration of our profiles we adopted the integration procedure described in Zingale et al. (2002). The quality of the resulting hydrostatic equilibrium after the mapping to the *SLH* grid is discussed in Sec. 5.2.1 and was found to be satisfying.

Another crucial ingredient for our modeling is the effective rate α-chain reaction network that was also used for the stellar evolution calculations. This reaction network was reimplemented into *SLH* from the open source hydrodynamics codes *CASTRO* and *MAESTRO*, which uses a slightly different implementation of the reaction network than the stellar evolution code *MESA*. This leads to slight, but still acceptable discrepancies in the energy generation and neutronization rates of the network due to differences in the reaction rates. A detailed analysis and discussion of the discrepancies in the reaction network is given in Sec. 5.2.2.

After successful mapping from the stellar evolution code to our hydrodynamic code, we first performed 2D simulations to confirm that our setup is working correctly. For this purpose, we verified that the convective properties of our 2D model are in good agreement with predictions of the Brunt-Väisälä frequency based on mixing-length theory (see Sec. 5.3.1). Afterwards, we identified the hydrostatic burning regions in our 2D model based on temperatures and characteristic isotope abundances in Sec. 5.3.2. This allowed us to further restrict the simulated domain in order to improve the resolution in the silicon burning region. In addition to that, we studied the impact of reducing the frequency of nuclear network evaluations with this setup to improve the overall performance of the simulation. We found that the model with a reduced frequency of network evaluations behaves very similar to the model that evaluates the network in every hydrodynamic time step (see Sec. 5.3.3). Reducing the frequency of network evaluations is especially useful for our 3D simulation, since the nuclear network would otherwise consume a large fraction of the overall computing time. The properties of the nuclear network in our 2D simulations are further analyzed in Sec. 5.3.4.

However, the main deficit of our 2D modeling approach is that none of our 2D models contract as it would be expected from the stellar evolution calculations. This is mainly due to inconsistent treatment of the geometry in 2D. Those issues with our 2D modeling approach could be addressed by introducing a 2.5D. For this purpose it is assumed that the simulated domain represents a slice in the rz plane and the model is rotationally symmetric around the z-axis, which leads to a consistent spherical geometry in 2D. Further details of this geometrical setup are also discussed in Sec. 5.4.3. The implementation of this feature into *SLH* would require corrections for the flux functions, which are not yet included in *SLH*. With this modeling approach it might be possible to accurately predict the contraction of our models in 2D, which could be useful to identify suitable setups for 3D simulations.

In addition to that, we also performed one 3D simulation of silicon burning on JUQUEEN, which, to our knowledge, is the first model simulated in full 4π volume. A first comparison of the convective features between our 2D models and this 3D model is shown in Sec. 5.4.1. As expected the turbulent energy cascade is artificially inverted in the 2D case. This inverted energy cascade in 2D transfers energy to larger scales and makes the 2D models more prone to explosions in the subsequent modeling of core-collapse supernovae (see Sec. 3.2.1). As mentioned before our 3D model successfully contracts on a similar time scale as predicted by the stellar evolution model (see Sec. 5.4.2). This confirms that there is something wrong with our modeling approach in 2D.

However, as discussed in Sec. 5.4 our 3D model consumes a large amount of computing time, which currently makes it unfeasible to run many different 3D models. Our current 3D model consumed 3.9×10^6 core-hours and only covers 154.41 s of silicon burning, i.e. only a fraction of the 1000 s from the initial model to the onset of core collapse. Suggestions for future improvements to speed up our modeling in 3D are given in Sec. 5.4.4. With those improvements a considerable reduction of the required computational resources is expected. This would enable us to cover a longer period of silicon burning and possibly even reach the onset of core collapse with our hydrodynamic simulations within a reasonable amount of computing time. Ideally it would be possible to do several 3D simulations with different initial models, but this is a more long-term goal and requires substantial improvements in our modeling. The recent publication of Collins et al. (2018) could be used as a guideline to identify promising silicon burning models.

Bibliography

Almgren, A. S., Beckner, V. E., Bell, J. B., Day, M. S., Howell, L. H., Joggerst, C. C., Lijewski, M. J., Nonaka, A., Singer, M., and Zingale, M. (2010). CASTRO: A New Compressible Astrophysical Solver. I. Hydrodynamics and Self-gravity. *ApJ*, 715:1221–1238, 1005.0114.

Almgren, A. S., Bell, J. B., and Zingale, M. (2007). Maestro: A low mach number stellar hydrodynamics code. *Journal of Physics Conference Series*, 78(1):012085.

An, Z.-D., Ma, Y.-G., Fan, G.-T., Li, Y.-J., Chen, Z.-P., and Sun, Y.-Y. (2016). New Astrophysical Reaction Rate for the ^{12}C(α, γ)^{16}O Reaction. *ApJ*, 817:L5, 1602.01692.

Arnett, D. (1994). Oxygen-burning hydrodynamics. 1: Steady shell burning. *ApJ*, 427:932–946.

Arnett, W. D. and Meakin, C. (2011). Toward realistic progenitors of core-collapse supernovae. *ApJ*, 733:78, 1101.5646.

Asida, S. M. and Arnett, D. (2000). Further adventures: Oxygen burning in a convective shell. *The Astrophysical Journal*, 545(1):435.

Bader, G. and Deuflhard, P. (1983). A semi-implicit mid-point rule for stiff systems of ordinary differential equations. *Numerische Mathematik*, 41(3):373–398.

Baraffe, I., Pratt, J., Goffrey, T., Constantino, T., Folini, D., Popov, M. V., Walder, R., and Viallet, M. (2017). Lithium Depletion in Solar-like Stars: Effect of Overshooting Based on Realistic Multi-dimensional Simulations. *ApJ*, 845:L6, 1707.09498.

Barsukow, W., Edelmann, P. V. F., Klingenberg, C., Miczek, F., and Röpke, F. K. (2017). A numerical scheme for the compressible low-mach number regime of ideal fluid dynamics. *Journal of Scientific Computing*, pages 1–24, 1612.03910.

Batchelor, G. K. (2000). *An introduction to fluid dynamics*. Cambridge University Press, Cambridge.

Bazán, G. and Arnett, D. (1994). Convection, nucleosynthesis, and core collapse. *ApJ*, 433:L41–L43.

Bazán, G. and Arnett, D. (1997). Large nuclear networks in presupernova models. *Nuclear Physics A*, 621(1):607 – 610. Nuclei in the Cosmos.

Bazán, G. and Arnett, D. (1998). Two-dimensional Hydrodynamics of Pre–Core Collapse: Oxygen Shell Burning. *ApJ*, 496:316–332, astro-ph/9702239.

Bethe, H. A. (1990). Supernova mechanisms. *Reviews of Modern Physics*, 62:801–866.

Bethe, H. A. and Wilson, J. R. (1985). Revival of a stalled supernova shock by neutrino heating. *ApJ*, 295:14–23.

Blondin, J. M., Mezzacappa, A., and DeMarino, C. (2003). Stability of Standing Accretion Shocks, with an Eye toward Core-Collapse Supernovae. *ApJ*, 584:971–980, astro-ph/0210634.

Böhm-Vitense, E. (1958). Über die Wasserstoffkonvektionszone in Sternen verschiedener Effektivtemperaturen und Leuchtkräfte. Mit 5 Textabbildungen. *ZAp*, 46:108.

Bolaños Rosales, A. (2016). *Low Mach Number Simulations of Convective Boundary Mixing in Classical Novae*. Dissertation, Julius-Maximilians-Universität Würzburg.

Botto Poala, A. (2017). A differential study of the bulk-richardson number for astrophysical application. Master's thesis, Universität Heidelberg.

Bouvier, A. and Wadhwa, M. (2010). The age of the Solar System redefined by the oldest Pb-Pb age of a meteoritic inclusion. *Nature Geoscience*, 3:637–641.

Bulirsch, R. and Stoer, J. (1966). Numerical treatment of ordinary differential equations by extrapolation methods. *Numerische Mathematik*, 8(1):1–13.

Burrows, A., Hayes, J., and Fryxell, B. A. (1995). On the Nature of Core-Collapse Supernova Explosions. *ApJ*, 450:830, astro-ph/9506061.

Burrows, A., Vartanyan, D., Dolence, J. C., Skinner, M. A., and Radice, D. (2016). Crucial Physical Dependencies of the Core-Collapse Supernova Mechanism. *ArXiv e-prints*, 1611.05859.

Calhoun, D. A., Helzel, C., and Leveque, R. J. (2008). Logically Rectangular Grids and Finite Volume Methods for PDEs in Circular and Spherical Domains. *SIAM Review*, 50:723–752.

Canuto, V. M. and Mazzitelli, I. (1991). Stellar turbulent convection - A new model and applications. *ApJ*, 370:295–311.

Cargo, P. and Le Roux, A. (1994). Un schéma équilibre adapté au modèle d'atmosphère avec termes de gravité. *Comptes rendus de l'Académie des sciences. Série 1, Mathématique*, 318(1):73–76.

Chandrashekar, P. and Klingenberg, C. (2015). A second order well-balanced finite volume scheme for euler equations with gravity. *SIAM Journal on Scientific Computing*, 37(3):B382–B402.

Colella, P., Dorr, M. R., Hittinger, J. A., and Martin, D. F. (2011). High-order, finite-volume methods in mapped coordinates. *Journal of Computational Physics*, 230(8):2952–2976.

Colella, P. and Woodward, P. R. (1984). The Piecewise Parabolic Method (PPM) for gas-dynamical simulations. *Journal of Computational Physics*, 54:174–201.

Collins, C., Müller, B., and Heger, A. (2018). Properties of convective oxygen and silicon burning shells in supernova progenitors. *MNRAS*, 473:1695–1704, 1709.00236.

Couch, S. M., Chatzopoulos, E., Arnett, W. D., and Timmes, F. X. (2015). The Three-dimensional Evolution to Core Collapse of a Massive Star. *ApJ*, 808:L21, 1503.02199.

Couch, S. M. and Ott, C. D. (2013). Revival of the stalled core-collapse supernova shock triggered by precollapse asphericity in the progenitor star. *ApJ*, 778:L7, 1309.2632.

Couch, S. M. and Ott, C. D. (2015). The role of turbulence in neutrino-driven core-collapse supernova explosions. *ApJ*, 799:5, 1408.1399.

Courant, R., Friedrichs, K. O., and Lewy, H. (1928). Über die partiellen Differentialgleichungen der mathematischen Physik. *Math. Ann.*, 100:32–74.

Cristini, A., Meakin, C., Hirschi, R., Arnett, D., Georgy, C., Viallet, M., and Walkington, I. (2017). 3D hydrodynamic simulations of carbon burning in massive stars. *MNRAS*, 471:279–300, 1610.05173.

Cyburt, R. H., Amthor, A. M., Ferguson, R., Meisel, Z., Smith, K., Warren, S., Heger, A., Hoffman, R. D., Rauscher, T., Sakharuk, A., Schatz, H., Thielemann, F. K., and Wiescher, M. (2010). The JINA REACLIB Database: Its Recent Updates and Impact on Type-I X-ray Bursts. *ApJS*, 189:240–252.

Duncan, C. and Jones, J. (1993). A mixed method Poisson solver for three-dimensional self-gravitating astrophysical fluid dynamical systems. In *6th Copper Mountain Conference on Multigrid Methods*.

Edelmann, P. V. F. (2014). *Coupling of Nuclear Reaction Networks and Hydrodynamics for Application in Stellar Astrophysics*. Dissertation, Technische Universität München.

Edelmann, P. V. F., Roepke, F. K., Hirschi, R., Georgy, C., and Jones, S. (2017). Testing a one-dimensional prescription of dynamical shear mixing with a two-dimensional hydrodynamic simulation. *ArXiv e-prints*, 1704.06261.

Edelmann, P. V. F. and Röpke, F. K. (2016). SLH Seven-League Hydro code. In Brömmel, D., Frings, W., and Wylie, B. J. N., editors, *JUQUEEN Extreme Scaling Workshop 2016*, number FZJ-JSC-IB-2016-01 in JSC Internal Report, pages 63–67.

Foglizzo, T., Galletti, P., Scheck, L., and Janka, H.-T. (2007). Instability of a Stalled Accretion Shock: Evidence for the Advective-Acoustic Cycle. *ApJ*, 654:1006–1021, astro-ph/0606640.

Fryxell, B. A., Müller, E., and Arnett, W. D. (1989). Hydrodynamics and nuclear burning. MPA Green Report 449, Max-Planck-Institut für Astrophysik, Garching.

Gilet, C., Almgren, A. S., Bell, J. B., Nonaka, A., Woosley, S. E., and Zingale, M. (2013). Low Mach Number Modeling of Core Convection in Massive Stars. *ApJ*, 773:137.

Godunov, S. K. (1959). Finite difference method for numerical computation of discontinous solution of the equations of fluid dynamics. *Matematicheskii Sbornik*, 47:271.

Goffrey, T., Pratt, J., Viallet, M., Baraffe, I., Popov, M. V., Walder, R., Folini, D., Geroux, C., and Constantino, T. (2017). Benchmarking the Multidimensional Stellar Implicit Code MUSIC. *A&A*, 600:A7, 1610.10053.

Grimm-Strele, H., Kupka, F., and Muthsam, H. J. (2014). Curvilinear Grids for WENO Methods in Astrophysical Simulations. *Computer Physics Communications*, 185:764–776, 1308.3066.

Guilet, J. and Foglizzo, T. (2012). On the linear growth mechanism driving the standing accretion shock instability. *MNRAS*, 421:546–560, 1112.1427.

Hanke, F., Marek, A., Müller, B., and Janka, H.-T. (2012). Is strong SASI activity the key to successful neutrino-driven supernova explosions? *ApJ*, 755:138, 1108.4355.

Herant, M., Benz, W., Hix, W. R., Fryer, C. L., and Colgate, S. A. (1994). Inside the supernova: A powerful convective engine. *ApJ*, 435:339–361, astro-ph/9404024.

Hix, W. R. and Meyer, B. S. (2006). Thermonuclear kinetics in astrophysics. *Nuclear Physics A*, 777:188–207, arXiv:astro-ph/0509698.

Hix, W. R., Parete-Koon, S. T., Freiburghaus, C., and Thielemann, F.-K. (2007). The QSE-Reduced Nuclear Reaction Network for Silicon Burning. *ApJ*, 667:476–488.

Horst, L. (2016). On the path to multidimensional simulations of shear instabilities in stellar evolution: Mapping 1d initial data into 3d space. Master's thesis, Julius-Maximilians-Universität Würzburg.

Iliadis, C. (2015). *Nuclear Physics of Stars.* Wiley-VCH, 2nd edition.

Itoh, N., Hayashi, H., Nishikawa, A., and Kohyama, Y. (1996). Neutrino energy loss in stellar interiors. vii. pair, photo-, plasma, bremsstrahlung, and recombination neutrino processes. *ApJS*, 102:411.

Jackson, J. D. (1999). *Classical Electrodynamics.* WILEY.

Jacobs, A. M., Zingale, M., Nonaka, A., Almgren, A. S., and Bell, J. B. (2016). Low mach number modeling of convection in helium shells on sub-chandrasekhar white dwarfs. II. Bulk properties of simple models. *ApJ*, 827:84, 1507.06696.

Janka, H., Langanke, K., Marek, A., Martínez-Pinedo, G., and Müller, B. (2007). Theory of core-collapse supernovae. *Phys. Rep.*, 442:38–74, arXiv:astro-ph/0612072.

Janka, H.-T., Hanke, F., Hüdepohl, L., Marek, A., Müller, B., and Obergaulinger, M. (2012). Core-collapse supernovae: Reflections and directions. *Progress of Theoretical and Experimental Physics*, 2012(1):01A309, 1211.1378.

Janka, H.-T., Melson, T., and Summa, A. (2016). Physics of Core-Collapse Supernovae in Three Dimensions: a Sneak Preview. *ArXiv e-prints*, 1602.05576.

Janka, H.-T. and Müller, E. (1996). Neutrino heating, convection, and the mechanism of Type-II supernova explosions. *A&A*, 306:167.

Jones, S., Andrassy, R., Sandalski, S., Davis, A., Woodward, P., and Herwig, F. (2017). Idealized hydrodynamic simulations of turbulent oxygen-burning shell convection in 4π geometry. *MNRAS*, 465:2991–3010, 1605.03766.

Kifonidis, K. and Müller, E. (2012). On multigrid solution of the implicit equations of hydrodynamics. experiments for the compressible euler equations in general coordinates. *A&A*, 544:A47.

Kippenhahn, R., Weigert, A., and Weiss, A. (2012). *Stellar Structure and Evolution*. Springer-Verlag, Berlin Heidelberg.

Kolmogorov, A. N. (1941). The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *Dokl. Akad. Nauk SSSR*, 30:299–303. in Russian.

Kunz, R., Fey, M., Jaeger, M., Mayer, A., Hammer, J. W., Staudt, G., Harissopulos, S., and Paradellis, T. (2002). Astrophysical Reaction Rate of $^{12}\mathrm{C}(\alpha, \gamma)^{16}\mathrm{O}$. *ApJ*, 567:643–650.

Kupka, F. and Muthsam, H. J. (2017). Modelling of stellar convection. *Living Reviews in Computational Astrophysics*, 3:1.

Kupka, F., Zaussinger, F., and Montgomery, M. H. (2018). Mixing and overshooting in surface convection zones of DA white dwarfs: first results from ANTARES. *MNRAS*, 474:4660–4671, 1712.00641.

Kwatra, N., Su, J., Grétarsson, J. T., and Fedkiw, R. (2009). A method for avoiding the acoustic time step restriction in compressible flow. *Journal of Computational Physics*, 228:4146–4161.

Lach, F. (2016). Two- and three-dimensional simulations of classical novae. Master's thesis, Julius-Maximilians-Universität Würzburg.

Laming, J. M. (2007). Analytic Approach to the Stability of Standing Accretion Shocks: Application to Core-Collapse Supernovae. *ApJ*, 659:1449–1457, astro-ph/0701264.

Landau, L. D. and Lifschitz, E. M. (1991). *Hydrodynamik*, volume 6 of *Lehrbuch der theoretischen Physik*. Akademie Verlag, Berlin.

Langanke, K. and Martinez-Pinedo, G. (2000). Shell-model calculations of stellar weak interaction rates. II. Weak rates for nuclei in the mass range A = 45 - 65 in supernovae environments. *Nuclear Physics A*, 673:481–508, arXiv:nucl-th/0001018.

Lentz, E. J., Bruenn, S. W., Hix, W. R., Mezzacappa, A., Messer, O. E. B., Endeve, E., Blondin, J. M., Harris, J. A., Marronetti, P., and Yakunin, K. N. (2015). Three-dimensional Core-collapse Supernova Simulated Using a 15 $M_\odot$ Progenitor. *ApJ*, 807:L31, 1505.05110.

LeVeque, R. J. (1998). Nonlinear conservation laws and finite volume methods. In Steiner, O. and Gautschy, A., editors, *Computational Methods for Astrophysical Flows*, Saas-Fee Advanced Course 27, pages 1–159. Springer, Berlin Heidelberg New York.

LeVeque, R. J. (2002). *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, Cambridge.

Liebendörfer, M., Messer, O. E. B., Mezzacappa, A., Bruenn, S. W., Cardall, C. Y., and Thielemann, F.-K. (2004). A finite difference representation of neutrino radiation hydrodynamics in spherically symmetric general relativistic spacetime. *ApJS*, 150:263–316, arXiv:astro-ph/0207036.

Liebendörfer, M., Mezzacappa, A., Thielemann, F.-K., Messer, O. E., Hix, W. R., and Bruenn, S. W. (2001). Probing the gravitational well: No supernova explosion in spherical symmetry with general relativistic Boltzmann neutrino transport. *Phys. Rev. D*, 63(10):103004, astro-ph/0006418.

Liebendörfer, M., Rampp, M., Janka, H.-T., and Mezzacappa, A. (2005). Supernova Simulations with Boltzmann Neutrino Transport: A Comparison of Methods. *ApJ*, 620:840–860, astro-ph/0310662.

Liu, X.-D., Osher, S., and Chan, T. (1994). Weighted Essentially Non-oscillatory Schemes. *Journal of Computational Physics*, 115:200–212.

Mazurek, T. J. (1973). PhD thesis, Yeshiva University.

Mazurek, T. J., Truran, J. W., and Cameron, A. G. W. (1974). Electron capture in carbon dwarf supernovae. *Astrophysics and Space Science*, 27(2):261–291.

Meakin, C. A. and Arnett, D. (2007a). Anelastic and Compressible Simulations of Stellar Oxygen Burning. *ApJ*, 665:690–697, astro-ph/0611317.

Meakin, C. A. and Arnett, D. (2007b). Turbulent convection in stellar interiors. i. hydrodynamic simulation. *ApJ*, 667:448–475, arXiv:astro-ph/0611315.

Meister, A. (2011). *Numerik linearer Gleichungssysteme*. Vieweg+Teubner, 4th edition.

Melson, T., Janka, H.-T., Bollig, R., Hanke, F., Marek, A., and Müller, B. (2015). Neutrino-driven Explosion of a 20 Solar-mass Star in Three Dimensions Enabled by Strange-quark Contributions to Neutrino-Nucleon Scattering. *ApJ*, 808:L42, 1504.07631.

Miczek, F. (2013). *Simulation of low Mach number astrophysical flows.* Dissertation, Technische Universität München.

Miczek, F., Röpke, F. K., and Edelmann, P. V. F. (2015). New numerical solver for flows at various mach numbers. *A&A*, 576:A50, 1409.8289.

Müller, B. (2016). The Status of Multi-Dimensional Core-Collapse Supernova Models. *PASA*, 33:e048, 1608.03274.

Müller, B. and Janka, H.-T. (2015). Non-radial instabilities and progenitor asphericities in core-collapse supernovae. *MNRAS*, 448:2141–2174, 1409.4783.

Müller, B., Melson, T., Heger, A., and Janka, H.-T. (2017). Supernova Simulations from a 3D Progenitor Model – Impact of Perturbations and Evolution of Explosion Properties. *ArXiv e-prints*, 1705.00620.

Müller, B., Viallet, M., Heger, A., and Janka, H.-T. (2016). The last minutes of oxygen shell burning in a massive star. *ApJ*, 833:124, 1605.01393.

Müller, E. (1998). Simulation of astrophysical fluid flow. In Steiner, O. and Gautschy, A., editors, *Computational Methods for Astrophysical Flows*, Saas-Fee Advanced Course 27, pages 343–494. Springer, Berlin Heidelberg New York.

Mundprecht, E., Muthsam, H. J., and Kupka, F. (2013). Multidimensional realistic modelling of Cepheid-like variables - I. Extensions of the ANTARES code. *MNRAS*, 435:3191–3205, 1209.2952.

Mundprecht, E., Muthsam, H. J., and Kupka, F. (2015). Multidimensional realistic modelling of Cepheid-like variables - II. Analysis of a Cepheid model. *MNRAS*, 449:2539–2552, 1502.00449.

Muthsam, H. J., Kupka, F., Löw-Baselli, B., Obertscheider, C., Langer, M., and Lenz, P. (2010). ANTARES - A Numerical Tool for Astrophysical RESearch with applications to solar granulation. *New Astronomy*, 15:460–475, 0905.0177.

Nakamura, K., Kuroda, T., Takiwaki, T., and Kotake, K. (2014). Impacts of Rotation on Three-dimensional Hydrodynamics of Core-collapse Supernovae. *ApJ*, 793:45, 1403.7290.

Nakamura, K., Takiwaki, T., Kuroda, T., and Kotake, K. (2015). Systematic features of axisymmetric neutrino-driven core-collapse supernova models in multiple progenitors. *PASJ*, 67:107, 1406.2415.

Nonaka, A., Almgren, A. S., Bell, J. B., Lijewski, M. J., Malone, C. M., and Zingale, M. (2010). MAESTRO: An Adaptive Low Mach Number Hydrodynamics Algorithm for Stellar Flows. *ApJS*, 188:358–383, 1005.0112.

O'Connor, E. and Couch, S. (2015). Two Dimensional Core-Collapse Supernova Explosions Aided by General Relativity with Multidimensional Neutrino Transport. *ArXiv e-prints*, 1511.07443.

Paxton, B., Bildsten, L., Dotter, A., Herwig, F., Lesaffre, P., and Timmes, F. (2011). Modules for experiments in stellar astrophysics (mesa). *ApJS*, 192:3, 1009.1622.

Paxton, B., Cantiello, M., Arras, P., Bildsten, L., Brown, E. F., Dotter, A., Mankovich, C., Montgomery, M. H., Stello, D., Timmes, F. X., and Townsend, R. (2013). Modules for Experiments in Stellar Astrophysics (MESA): Planets, Oscillations, Rotation, and Massive Stars. *ApJS*, 208:4, 1301.0319.

Paxton, B., Marchant, P., Schwab, J., Bauer, E. B., Bildsten, L., Cantiello, M., Dessart, L., Farmer, R., Hu, H., Langer, N., Townsend, R. H. D., Townsley, D. M., and Timmes, F. X. (2015). Modules for Experiments in Stellar Astrophysics (MESA): Binaries, Pulsations, and Explosions. *ApJS*, 220:15, 1506.03146.

Pratt, J., Baraffe, I., Goffrey, T., Constantino, T., Viallet, M., Popov, M. V., Walder, R., and Folini, D. (2017). Extreme value statistics for two-dimensional convective penetration in a pre-main sequence star. *A&A*, 604:A125, 1706.04852.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing*, volume 3. Cambridge University Press.

Rampp, M. and Janka, H.-T. (2000). Spherically Symmetric Simulation with Boltzmann Neutrino Transport of Core Collapse and Postbounce Evolution of a 15 M_{solar} Star. *ApJ*, 539:L33–L36, astro-ph/0005438.

Rampp, M. and Janka, H.-T. (2002). Radiation hydrodynamics with neutrinos. Variable Eddington factor method for core-collapse supernova simulations. *A&A*, 396:361–392, astro-ph/0203101.

Reinecke, M. A. (2001). *Modeling and simulation of turbulent combustion in Type Ia supernovae.* PhD thesis, Technical University of Munich. available at http://tumb1.biblio.tu-muenchen.de/publ/diss/allgemein.html.

Roe, P. L. (1981). Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357 – 372.

Saad, Y. and Schultz, M. H. (1986). Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869.

Schrauth, M. (2015). Massively parallel non-uniform possion solver and its application to thermonuclear supernova modeling. Master's thesis, Julius-Maximilians-Universität Würzburg.

Shu, C.-W. and Osher, S. (1988). Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439 – 471.

Sleijpen, G. L. and Fokkema, D. R. (1993). Bicgstab (l) for linear equations involving unsymmetric matrices with complex spectrum. *Electronic Transactions on Numerical Analysis*, 1(11):2000.

Thompson, J., Warsi, Z., and Mastin, C. (1985). *Numerical grid generation: foundations and applications*. North-Holland.

Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W. (1982). Boundary-fitted coordinate systems for numerical solution of partial differential equations – a review. *Journal of Computational Physics*, 47:1.

Thompson, M. J. (2006). *An Introduction to Astrophysical Fluid Dynamics*. Imperial College Press.

Timmes, F. X. (1999). Integration of Nuclear Reaction Networks for Stellar Hydrodynamics. *ApJS*, 124:241–263.

Timmes, F. X., Hoffman, R. D., and Woosley, S. E. (2000). An Inexpensive Nuclear Energy Generation Network for Stellar Hydrodynamics. *ApJS*, 129:377–398.

Timmes, F. X. and Swesty, F. D. (2000). The accuracy, consistency, and speed of an electron-positron equation of state based on table interpolation of the Helmholtz free energy. *ApJS*, 126:501–516.

Van der Vorst, H. A. (1992). Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13(2):631–644.

Viallet, M., Baraffe, I., and Walder, R. (2011). Towards a new generation of multi-dimensional stellar evolution models: development of an implicit hydrodynamic code. *A&A*, 531:A86, 1103.1524.

Viallet, M., Baraffe, I., and Walder, R. (2013a). Comparison of different nonlinear solvers for 2D time-implicit stellar hydrodynamics. *A&A*, 555:A81, 1305.6581.

Viallet, M., Goffrey, T., Baraffe, I., Folini, D., Geroux, C., Popov, M. V., Pratt, J., and Walder, R. (2016). A jacobian-free newton-krylov method for time-implicit multidimensional hydrodynamics. physics-based preconditioning for sound waves and thermal diffusion. *A&A*, 586:A153, 1512.03662.

Viallet, M., Meakin, C., Arnett, D., and Mocák, M. (2013b). Turbulent Convection in Stellar Interiors. III. Mean-field Analysis and Stratification Effects. *ApJ*, 769:1, 1212.6365.

Vinokur, M. and Kordulla, W. (1983). Efficient computation of volume in flow predictions. *AIAA Journal*, 21:917.

Weaver, T. A., Zimmerman, G. B., and Woosley, S. E. (1978). Presupernova evolution of massive stars. *ApJ*, 225:1021–1029.

Wilson, J. R. (1985). Supernovae and Post-Collapse Behavior. In Centrella, J. M., Leblanc, J. M., and Bowers, R. L., editors, *Numerical Astrophysics*, page 422.

Woodward, P. R., Herwig, F., and Lin, P.-H. (2015). Hydrodynamic Simulations of H Entrainment at the Top of He-shell Flash Convection. *ApJ*, 798:49.

Zingale, M., Almgren, A. S., Bell, J. B., Nonaka, A., and Woosley, S. E. (2009). Low Mach Number Modeling of Type IA Supernovae. IV. White Dwarf Convection. *ApJ*, 704:196–210, 0908.2668.

Zingale, M., Dursi, L. J., ZuHone, J., Calder, A. C., Fryxell, B., Plewa, T., Truran, J. W., Caceres, A., Olson, K., Ricker, P. M., Riley, K., Rosner, R., Siegel, A., Timmes, F. X., and Vladimirova, N. (2002). Mapping Initial Hydrostatic Models in Godunov Codes. *ApJS*, 143:539–565, astro-ph/0208031.

Zingale, M., Nonaka, A., Almgren, A. S., Bell, J. B., Malone, C. M., and Orvedahl, R. J. (2013). Low mach number modeling of convection in helium shells on sub-chandrasekhar white dwarfs. I. Methodology. *ApJ*, 764:97, 1212.4380.

Danksagung

Zunächst möchte ich Professor Friedrich Röpke dafür danken, dass er mir dieses interessante Thema vorgeschlagen hat und diese Promotion ermöglicht hat. Er hat sich immer wieder Zeit genommen, um den weiteren Verlauf des Projektes zu besprechen und mich stets mit hilfreichen Ratschlägen unterstützt. Auch während seinem Umzug von Würzburg nach Heidelberg hat er sich regelmäßig nach dem aktuellen Stand der Arbeit erkundigt und bei Problemen geholfen. Weiterhin möchte ich bei Philipp Edelmann und Sam Jones bedanken, weil sie wesentlich zu dem Gelingen dieser Arbeit beigetragen haben. Bei Philipp möchte ich mich insbesondere dafür bedanken, dass er mich bei der Arbeit mit dem *SLH*-Code und bei der Implementierung der Gravitations-Löser unterstützt hat und bei Fragen jederzeit erreichbar war. Bei Sam möchte ich mich für seine Expertise mit Sternentwicklungs-Codes und nuklearen Netzwerken bedanken. Nur durch seine Mithilfe konnte ein gravierender Bug in der Implementierung des nuklearen Netzwerks gefunden werden.

Außerdem geht mein Dank auch an alle, die an der Entwicklung des *SLH*-Codes beteiligt waren. Insbesondere bedanke ich mich bei Fabian Miczek, weil er mit seiner Arbeit an *SLH* die Grundlage für dieses Projekt geschaffen hat. Weiterhin geht mein Dank auch an Alejandro Bolaños, weil er mir bei dem Übertragen des *MESA*-Modells in den *SLH*-Code wichtige Hinweise gegeben hat. Bei Leonhard Horst bedanke ich mich für seine Mitarbeit an *SLH* und dafür, dass er die Simulationen zum Siliziumbrennen nun weiterführt. Bei Jonas Berberich möchte ich mich für sein Feedback zur mathematischen Formulierung des “mixed method Poisson solvers” bedanken. Ebenfalls bedanke ich mich bei Philipp Edelmann, Leonhard Horst und Florian Lach für das Korrekturlesen der vorliegenden Arbeit. Bei Prof. Christian Klingenberg bedanke ich mich für seinen entscheidenden Hinweis auf den “mixed method Poisson solver”.

Ich möchte mich außerdem bei den übrigen Kollegen von der PSO-Gruppe und der TAP-Gruppe am Heidelberger Institut für Theoretische Studien (HITS) für die interessanten Gespräche während der Mittagspause und den Kaffeepausen bedanken, insbesondere bei Andreas Bauswein, Markus Kromer und Rüdiger Pakmor.

Weiterhin bedanke ich mich bei dem (HITS) und bei der Klaus Tschira Stiftung für die Finanzierung dieses Projektes und für die angenehme Arbeitsatmosphäre. Beim HITS bedanke ich mich außerdem für die Bereitstellung der lokalen Rechencluster, auf denen ein Großteil der Simulationen durchgeführt wurde. Darüber hinaus bedanke ich mich bei dem Gauss Centre for Supercomputing (GCS) für die Bereitstellung der Rechenzeit auf dem Großrechner JUQUEEN am Forschungszentrum Jülich.

Außerhalb des Instituts geht mein Dank vor allem an Stephan Hachinger, weil er mir mit seinen persönlichen Erfahrungen stets weiter geholfen hat und trotz der räumlichen Trennung den Kontakt aufrecht erhalten hat. Er war für mich auch in schwierigen Zeiten

immer ein wertvoller Ansprechpartner. Über die gemeinsamen Wochenenden in München und die regelmäßigen Telefongespräche habe ich mich sehr gefreut.

Zum Abschluss bedanke ich mich bei meinen Eltern Harald und Birgit für die Unterstützung während dem Studium und der Doktorarbeit. Im Kreise der Familie geht mein Dank außerdem auch an meine Geschwister Elena und Jonas, sowie Felix Uthoff.

Eigenständigkeitserklärung

Hiermit erkläre ich, dass es sich bei der vorliegenden Arbeit um meine eigenständig erbrachte Leistung handelt und ich keine Quellen oder Hilfsmittel außer den angegebenen verwendet habe.

Heidelberg, den 12.10.2018,
Aron Michel