2025

Modeling Relativistic Fluids in Dynamical Spacetimes

Terrence Alphonse Pierre Jacques

Modeling Relativistic Fluids in Dynamical Spacetimes

Terrence Alphonse Pierre Jacques

Dissertation submitted to
The Eberly College of Arts and Sciences
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Physics

Maura McLaughlin, Ph.D., Chair
Zachariah B. Etienne, Ph.D.
Paul Cassak, Ph.D.
Maria C. Babiuc Hamilton, Ph.D.

Department of Physics and Astronomy
Morgantown, West Virginia
2025
Keywords: neutron stars, numerical relativity, hydrodynamics

Abstract

Modeling Relativistic Fluids in Dynamical Spacetimes

Terrence Alphonse Pierre Jacques

Multi-messenger astrophysics opens a new era in our understanding of the most dynamic and energetic systems in the Universe. Correlating gravitational-wave and electromagnetic signals in space and time enables stringent tests of models for core-collapse supernovae, merging supermassive black-hole binaries with accretion disks and jets, and mergers of compact object binaries such as binary neutron stars (BNS) and white dwarfs. Comparisons between models and multi-messenger observations may be used to constrain the neutron-star equation of state (EOS), formation channels for compact-object binaries, and emission mechanisms behind short gamma-ray bursts.

In modeling such astrophysical systems, great success has been achieved by treating the matter in the perfect-fluid limit, and evolving it with the equations of general relativistic hydrodynamics (GRHD), a covariant generalization of the Newtonian Euler equations. To capture the full dynamics when the spacetime itself changes rapidly, i.e.,*dynamical* spacetimes, the GRHD equations must be coupled to an evolution of the spacetime. We describe the spacetime using Einstein's equations, but these are a set of coupled, nonlinear partial differential equations that admit no closed-form solutions for generic sources; instead, they are solved numerically using *numerical relativity*. By evolving the fluid equations and Einstein's equations together, with communication between the two schemes, one obtains a self-consistent description of relativistic fluids in dynamical spacetimes.

In this dissertation we construct such a modeling framework, and use it to model a variety of systems. We begin with the key ingredients to develop such a numerical framework, drawing parallels to solving Maxwell's equations, and sketching out the core components needed to solve the GRHD equations. We then extend the GRHD equations to singular curvilinear coordinates using the reference-metric approach, and derive the full set of modified equations, which are suitable for numerical implementation. We solve these equations in our newly developed GRHD code, GRoovy, which evolves matter configurations self-consistently with dynamical spacetimes. We present results from a suite of rigorous tests on the GRoovy code, presenting results from two- and three-dimensional shock tests. We then show a number of results from tests in which we model non-rotating and rotating neutron stars (NSs), in both fixed and dynamical spacetimes. We also test our code's ability to model NSs using realistic equations of state, and neutrino emission in curved spacetimes. Notably, we show that when using a reference metric, we can simulate NSs using spherical and cylindrical coordinates, leading to computational savings on the order of up to $\sim 10^4$, in the case of spherical coordinates, as compared to Cartesian coordinates.

Finally, we describe further improvements to GRoovy, including Charm++-based parallelization and increased robustness when using realistic equations of state. Using this upgraded version, we model non-axisymmetric instabilities in rapidly rotating NSs that produce transient, elongated "bar" deformations and emit gravitational waves. We present preliminary constraints on the threshold for this instability and examine how that threshold depends on uncertainties in the nuclear EOS.

Acknowledgements

I will be forever grateful to my PhD advisor Zach Etienne for helping me pursue my goals and interests throughout my time in his research group, and for cultivating an environment where everyone can be heard. I am also thankful for his support beyond research, training me in the soft skills required to be a successful researcher. Under his careful and considerate guidance I realized my years-long goal of becoming a computational astrophysicist.

I am grateful to Maura McLaughlin for always having an open door, and for being a steady source of support and advice. I also thank her, Duncan Lorimer, and Nihan Pol for working hard with me to publish our NSBH paper. Special thanks to my committee members. I am also thankful to Viola and Miranda for handling a number of bureaucratic hurdles on my behalf.

I am eternally thankful for the many friendships I developed at WVU, especially through the COVID pandemic. Greg, Katey, Nick, and Ripu, I appreciate you all. I want to especially thank Nick for the countless nights of food and wine. I also want to recognize the lifelong friends I made at Oberlin: Anna, Cormac, James, Lele, Linh, and Peter. Finally, a special thanks to Chris, who inspires me to be unapologetically Black in academia.

I am forever grateful to the many people who supported my journey in physics, which began in high school. I want to thank Mr. Abugana, Mr. Mingo, Mr. Klein, Mazi, and Mr. Husbands for providing me with a strong foundation in math and physics. The coursework and research I have completed to this point would not have been possible without the patient instruction I received from them within an underfunded New York City public school. Special thanks to Mr. Burnett and Dr. Wiltshire for cultivating an atmosphere centered on academic excellence, and for encouraging me to see myself as a young scholar. I also thank Dan Stinebring and Rob Owen for helping to propel my career in academia, and all the other faculty at Oberlin College for guiding me during my undergraduate studies. Special thanks to Dan Styer for meeting with me as a prospective student and convincing me that an education at a small liberal arts school in Ohio was more than worth it.

I want to express my deep gratitude to Maryelizabeth. Thank you for moving across the country with me. It was with your unyielding love and support that this dissertation was made possible. Special thanks to Pepper and Maxwell for showing me the importance of a good midday nap.

I also thank my aunt Titine and my brother Tyler for their never-ending love and support. Finally, I am grateful to my father, Jean, who has always supported my dreams — thank you for instilling in me the importance of hard work and a good education.

Contents

List of Figures

Chapter 1

Introduction

Binary neutron star (BNS) mergers and other compact-object transients stand among the most dynamic and energetic events in the Universe. Representing extreme laboratories of relativistic astrophysics, they can offer new insights into the some of the most elusive problems in nuclear astrophysics. For example, neutron stars (NSs) are composed of the densest form of matter known, reaching similar density to that of the nucleus within an atom. Lacking the resources to directly probe this extreme state of matter in experimental laboratories, we must resort to astrophysical observations of NSs to understand how this ultra-relativistic matter behaves. Doing so provides an ever-evolving understanding of the NS equation of state (EoS). Further, given that the stellar evolution of massive stars only produces chemical elements up to iron, at which point it is no longer energetically variable to continue nuclear fusion, BNS mergers provide the only means to create heavier elements such as gold. This will occur in the dynamical ejecta resulting from the merger, where nucleosynthesis will take place as subatomic particles interact.

Within BNS mergers there is also the intense interaction between magnetized plasmas and strong gravity. Depending on the mass ratio and spins of the NSs, and the NS EoS, the merger remnant can be either a NS, short- or long-lived, or a black hole. Matter will then accrete onto this remnant, which in turn will drive relativistic jets and outflows driven from magnetic fields and neutrino irradiation.

GW170817 was one such BNS merger event, with subsequent emission ranging the electromagnetic (EM) spectrum [1], and gravitational wave (GW) emission observed by the LIGO/VIRGO observatories [2]. Apart from the short γ-ray burst (sGRB) detected approximately 1.7 s after the GW detection, e.g.,by INTEGRAL [3], these EM observations were associated with a kilonova, powered by r-process nucleosynthesis [4, 5]. Kilonovae are composed of at least two components: an early blue component, arising from dynamical

ejecta shock heated by the contact boundary at merger, and a red component associated with material ejected from tidal tail interactions or spherical accretion in the case of prompt collapse [4]. Alternatively, the blue component can also emerge from accretion outflows or magnetized winds from a hyper-massive NS remnant [5, 6]. Observations of these components along with those of the GWs may be used to place constraints on the NS EoS when compared against numerical models [e.g., 7, 8].

GW170817 will almost certainly not be the only multi-messenger detection in our lifetimes, however, as existing GW observatories will be further improved and new ones are planned in the coming decades. Further, BNS mergers will not be the only events to feature GW and EM emission, as other candidates include core-collapse supernova, merging supermassive black hole binaries with accretion disks, and mergers of white dwarfs. However, modeling such systems demands evolving relativistic matter coupled with dynamical spacetimes.

A popular approach for modeling these astrophysical systems is to treat matter distributions as relativistic, collisional fluids, coupled to rapidly changing spacetimes. In other words, these systems are modeled by self-consistently evolving relativistic fluids alongside a dynamical spacetime. Previous studies have focused on simulating such systems using numerical techniques, often employing Cartesian coordinates with adaptive mesh refinement (AMR) to model scenarios like the mergers of compact object binaries. For single compact objects, studies have also used curvilinear coordinates, which are more efficient, particularly when near-symmetries exist. However, many of these codes either lack the necessary physical realism to fully capture the dynamics or assume a fixed spacetime while only evolving the matter configurations. While the latter approach is valid for systems where matter dynamics do not significantly influence the spacetime, it fails in scenarios where the feedback between matter and spacetime is non-negligible.

In this dissertation, we address both of these needs by developing a code that efficiently models systems with near-symmetries using singular curvilinear coordinates, while also self-consistently evolving the matter and spacetime. We begin in Chapter 2 with an introduction to the numerical methods employed throughout this work, drawing parallels between the numerical evolution of Maxwell's equations in vacuum with Einstein's equations.

In Chapter 3 we describe the components necessary to evolve relativistic matter configurations within the perfect fluid approximation. We then present a formulation of the general relativistic hydrodynamics (GRHD) equations, which are well-suited for numerical integration in singular curvilinear coordinates. In Chapter 4 we introduce our newly developed GRHD code, GRoovy, and reproduce our recent code announcement paper, demonstrating its ability to solve the GRHD equations while simultaneously evolving Einstein's equations. Finally, in Chapter 5 we outline ongoing efforts to apply GRoovy to model non-axisymmetric instabilities in rapidly rotating neutron stars, showcasing preliminary results and describing improvements made to the code, including more robust primitive recovery algorithms and Charm++ parallelization. We use geometrized units throughout this dissertation, where $G = c = M_\odot = 1$, with G as the gravitational constant, c as the speed of light, and $M_\odot$ representing a solar mass. Throughout this dissertation Latin indices indicate spatial components, while Greek indices indicate spacetime components.

The work presented in this dissertation comprises the bulk of my PhD research, but some projects are omitted for the sake of a more cohesive presentation. In particular, I do not include an ongoing study in which I work to constrain the number and Galactic merger rate of black hole—NS binaries. In this project, collaborators and I combine population-synthesis and pulsar-evolution models with a Bayesian statistical framework. I also omit related work on modeling relativistic magnetospheres in singular curvilinear coordinates. Despite these exclusions, the results and methods shown here provide a solid foundation for future studies that will employ GRoovy to model relativistic astrophysical systems.

Chapter 2

Numerical Relativity

2.1 Basic Numerics

Before describing the numerical methods used throughout this dissertation, it is useful to first define some terms and concepts. We start with the two types of numerical error we encounter in the coming Chapter s. The first, *round-off error*, results from numbers being represented by a finite number of bits in a given computer language. While there some numbers that can be exactly representable, such as most reasonable integers, in general a floating-point number will have some associated precision. Typical computer languages have machine precision $\epsilon \approx 10^{-16}$, defined as the smallest number such that $1 + \epsilon = 1$. This then implies that if any two numbers are more than 16 orders of magnitude apart, their sum or difference will yield the larger of the two, hence the term round-off error. Put another way, ϵ represents the threshold at which we say a floating-point is essentially zero.

For example, if we were to measure the *relative difference* between two numbers A and B,

$$E_{\text{diff}} = 2 \times \frac{A - B}{A + B}, \tag{2.1}$$

and find that $E_{\text{diff}} \sim \epsilon$, then we would conclude that the two numbers agree to round-off error. Alternatively, if we know B to be some trusted value, and A represents our numerical approximation to B, then we may also compute the *relative error* using

$$E_{\text{rel}} = \frac{A - B}{B}. \tag{2.2}$$

Eq. (2.2) is particularly useful because taking $\log_{10} E_{\text{diff}}$ yields the number of significant digits that are equivalent between the two numbers. In this dissertation we consider numbers

below 10^{-13} to be at round-off, since in general we expect these rounding errors to accumulate following a random walk process, roughly of the form $E \approx 10^{-16}\sqrt{N}$, where N is the number of numerical operations. Note however that these errors can be positive or negative, therefore there will be ample cancellations with some gradual accumulation. For our errors to accumulate from machine precision to 10^{-13}, we would need well over 10^6 operations over the course of a simulation. This dissertation will not include such simulations.

The second type of error we frequently come across is *truncation error*. Given a continuous and differentiable function $f(x)$ of some continuous real number x, we approximate it using a Taylor series expansion about some point a using

$$f(x) = \sum_{n=0}^{\infty} \frac{f^n(a)(x-a)^n}{n!}, \tag{2.3}$$

where $f^n(x)$ represents the nth-derivative of $f(x)$. In practice we must *truncate* the series at the nth term, thus making our approximation nth order accurate. Now, suppose we discretize x onto a uniform numerical grid as x_i for $i = 0, N-1$, for N points, with grid spacing Δx. We take linear combinations of Eq. (2.3) at different points to make numerical approximations of $f^1(x) = \partial_x f$ or $f^2(x) = \partial^2{}_x f$. This gives us *finite difference* approximations to $f(x)$. For example, a second-order accurate finite difference approximation of $\partial_x f$ has the form

$$\partial_x f(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2\Delta x} + O\left(\Delta x^2\right). \tag{2.4}$$

Here we see that the error associated with this approximation scales with Δx^2. Of course, we may use a higher order approximation to $\partial_x f(x_i)$, but the stencil size (i.e.,how many neighboring points we include in the calculation) increases. However, we can in general gain some computational savings when increasing the order of the approximation as compared to decreasing the grid spacing on our numerical grid. This can be particularly useful when working in three spatial dimensions. For example, for a fourth-order accurate approximation the error term scales as Δx^4. Therefore, a second-order approximation needs about twice as many points to achieve similar accuracy. In three dimensions the computational cost increases to eight times as many points.

To provide some grounding for the above discussion, we now briefly go through an example of numerical calculation representative of most calculations done for the bulk of this dissertation. Consider the one-dimensional advection equation

$$\partial_t u\left(t,x\right) = c\partial_x u\left(t,x\right), \tag{2.5}$$

where $u\left(t,x\right)$ is a scalar function, which could be initialized as a Gaussian for example, and c is the propagation speed, which we take to be constant in space and time. By solving this equation we completely describe how $u\left(t,x\right)$ behaves for all space and time. This partial differential equation essentially states that at a given position and time, the change $u\left(t,x\right)$ undergoes in the next instant in time is proportional to the spatial deviation of $u\left(t,x\right)$ about x. As we previously discussed, we approximate $\partial_x u\left(t,x\right)$ using a finite-difference approximation, at some order. For simplicity let's use Eq. (2.5). Now we just need an approximation for the time derivative. In general we can actually use any standard time integration algorithm from the large class of solvers for ordinary differential equations (ODEs). This is known as applying the method of lines. For example, we can use the second-order accurate midpoint method (e.g.,see [9]), a Runge-Kutta scheme. Consider an ODE of the form

$$\frac{dy\left(t\right)}{dt} = g\left(t,y\right). \tag{2.6}$$

If we discretize the coordinate t such that $t = ndt$ for iteration n and discretization size dt, we are therefore sampling the function $y\left(t\right)$ at iteration n, giving us $y_n = y\left(t_n\right)$. Given initial data $y\left(0\right)$, we approximate y_n at following iterations using

$$k_1 = dtg\left(t_n, y_n\right), \tag{2.7}$$

$$k_2 = dtg\left(t_n + \frac{1}{2}dt, y_n + \frac{1}{2}k_1\right), \tag{2.8}$$

$$y_{n+1} = y_n + k_2. \tag{2.9}$$

The method of lines simply entails using Eq. (2.9) to approximate the time evolution of Eq. (2.5).

One thing to note when discretizing such equations is choosing the size of the time-step. Given the wave speed c we can immediately place a constraint on the time-step as $\Delta t \leq \Delta x/c$, otherwise we would be taking time-steps further in time than the numerical solution can traverse a single grid cell, which would lead to instability. This is of course an optimistic estimate, and in general one has to take even smaller steps to ensure stability of the numerical scheme.

2.2 3+1 Decomposition of Spacetime

The bulk of this dissertation is focused on modeling relativistic fluids with time-dependent spacetimes. Several decades of research have been devoted to developing numerical methods that can robustly simulate these systems. To create reliable and robust models of these complex process, we need valid descriptions for the dynamics of the rapidly changing spacetime and matter. This Chapter focuses on how to model rapidly changing spacetimes, using a formulation of Einstein's field equations suited for numerical computation. Before diving in, it is important to specify the applicability of these methods, which fall under the research area of numerical relativity. A useful figure produced by the LISA collaboration, taken from Ref. [10] and reproduced in Fig. 2.1, shows the most popular approaches used to model gravitational wave sources in the case of binaries. The figure shows, on the vertical axis, system compactness $\mathcal{C} = (m_1 + m_2)/r$, where m_1, m_2 are the constituent masses, m_1 for the more massive object, and r is their separation, against the horizontal axis showing mass ratio $q = m_2/m_1$. We see that for fairly compact binary systems with constituents of similar mass, in the lower left region of Fig. 2.1, this is the regime where numerical relativity is most useful. Beyond this region, using numerical relativity becomes prohibitively expensive. For example, for systems with mass ratios $q > 1000$, one has to ensure that the compact objects are sufficiently resolved, thereby drastically increasing computational costs. As shown in the figure, it is therefore more useful, and computationally tractable, to instead use black hole perturbation theory. Thus, we must only consider using numerical relativity for problems where the time and length scales are suitable for numerical integration. In the remainder of this Chapter , we briefly build up the foundations of numerical relativity from

general relativity.

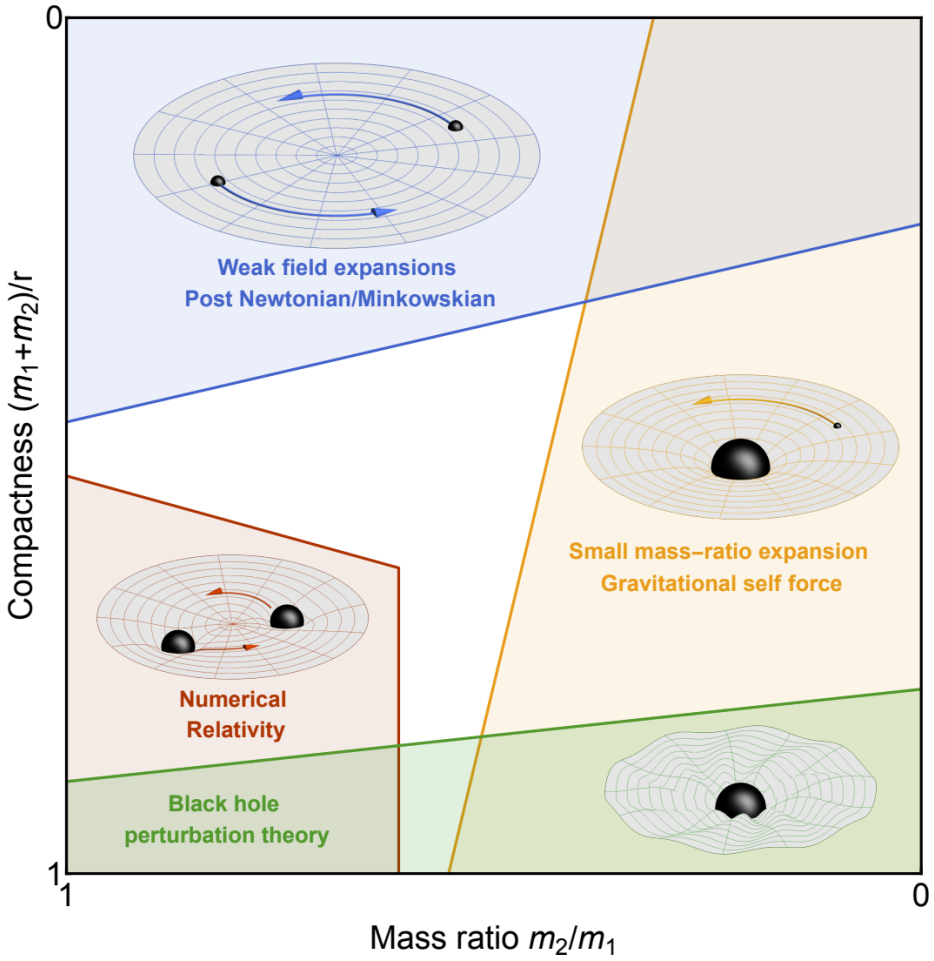

FIG. 2.1. Figure showing applicability of numerical relativity, taken from Figure 6.1 of Ref. [10].[1] Numerical relativity is best utilized for systems of high compactness and mass ratios of $\sim 1-0$.

We consider a four-dimensional differentiable manifold $\mathcal{M}$, with a set of four coordinates x^μ to label each point. Spacetime can be described by such a manifold, with each event representing a single point, with a time coordinate and three spatial coordinates. To measure the distance between any two events, we integrate the line element defined as

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu, \tag{2.10}$$

where $g_{\mu\nu}$ is the metric tensor, which describes our spacetime, and dx^μ are infinitesimal dis-

[1] Licensed under Creative Commons Attribution 4.0 International, and adapted here without alteration.

placements along our coordinates. For example, in Minkowski spacetime (i.e.,no curvature, or complete vacuum without any gravitational sources) and in Cartesian coordinates, we have $g_{\mu\nu} = \text{diag}\,(-1, 1, 1, 1)$, referred to as the flat metric. In spherical coordinates the flat metric takes the form $g_{\mu\nu} = \text{diag}\,(-1, 1, r, r\sin\theta)$. In general, however, $g_{\mu\nu}$ will take on less trivial forms in curved spacetimes.

The geometric information required to fully describe our spacetime is encoded within $g_{\mu\nu}$. To describe how the spacetime changes with our coordinates, whether spatially or temporally, we have Einstein's equations

$$G_{\mu\nu} = 8\pi T_{\mu\nu}, \tag{2.11}$$

in units of $c = G = 1$, where $G_{\mu\nu}$ is a function of $g_{\mu\nu}$, and $T_{\mu\nu}$ encodes the matter source terms. Eq. (2.11) encodes how spacetime and gravity, arising from the curvature of spacetime, changes. Because $G_{\mu\nu}$ and $T_{\mu\nu}$ are symmetric tensors, Eq. (2.11) represents ten coupled, partial differential equations, which does not possess a general analytic solution. Thus, if we wish to describe how matter behaves in dynamical spacetimes, as well as make predictions for the gravitational waves we expect to observe from such relativistic systems, we must resort to using numerical methods to find solutions. We turn to numerical relativity for the answer.

The problem in front of us is to reformulate Einstein's equations such that they are suitable for numerical integration. However, given the gauge freedom within general relativity, we need to specify coordinates such that we describe gravitational fields and matter distributions at one instance of *coordinate time*. The goal is then to formulate Einstein's equation as an initial value, or Cauchy, problem. In such problems, one specifies an initial state of a numerical solution, and then, using prescribed evolution equations, the solution is marched forward in time.

To arrive at such a formulation, we begin by foliating the four-dimensional manifold along the time coordinate into a set of non-overlapping or intersecting hypersurfaces. Each hypersurface then represents a three-dimensional space-like surface ($ds^2 > 0$). This is the core idea of the so-called 3+1 decomposition. Each hypersurface is separated by proper

time α, known as the lapse function. This only covers the time coordinate, however, and in general the spatial coordinates need not be the same for each hypersurface. Thus, if we have some point P on one hypersurface, that same point at some proper time later will have undergone a shift or change in coordinates. We represent this shift in spatial coordinates using the shift vector function β^i. The projection of $g_{\mu\nu}$ onto each spatial slice is then its purely spatial components, $\gamma_{ij} = g_{ij}$. Finally, we define the extrinsic curvature K_{ij}, which essentially encodes how deformed each hypersurface is as compared to neighboring surfaces.

We then find that the line element takes the form

$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij}\left(dx^i + \beta^i dt\right)\left(dx^j + \beta^j dt\right), \tag{2.12}$$

and the metric tensor is now

$$g_{\mu\nu} = \begin{pmatrix} -\alpha^2 + \beta_i\beta^i & \beta_j \\ \beta_i & \gamma_{ij} \end{pmatrix}. \tag{2.13}$$

On each spatial hypersurface we also define the normal vector n^μ as

$$n^\mu = \left(\frac{1}{\alpha}, -\frac{\beta^i}{\alpha}\right), \tag{2.14}$$

$$n_\mu = (-\alpha, 0). \tag{2.15}$$

Einstein's equations now take the form

$$\partial_t \gamma_{ij} = -2\alpha K_{ij} + D_i\beta_j + D_j\beta_i, \tag{2.16}$$

$$\partial_t K = -D^2\alpha + \alpha\left(K_{ij}K^{ij} + 4\pi\left(\rho + S\right)\right) + \beta^i D_i K, \tag{2.17}$$

where $K = \gamma^{ij}K_{ij}$ is the trace of the extrinsic curvature, D_i is the covariant derivative, and

ρ and S represent contractions of the stress-energy tensor given by

$$\rho = n_\mu n_\nu T^{\mu\nu}, \tag{2.18}$$

$$S_{ij} = \gamma_{i\mu}\gamma_{j\nu}T^{\mu\nu}, \tag{2.19}$$

$$S = \gamma^{ij}S_{ij}. \tag{2.20}$$

The covariant derivative has the form

$$D_k A^i = A^i_{,k} + \Gamma^i_{mk}A^m, \tag{2.21}$$

for some tensor A^i, where Γ^i_{jk} is the metric connection associated with a specific metric, in this case the spatial metric γ_{ij}, but in general four-dimensional spacetime it is given by

$$\Gamma^\alpha_{\mu\nu} = \frac{1}{2}g^{\alpha\delta}\left(\partial_\nu g_{\delta\mu} + \partial_\mu g_{\delta\nu} - \partial_\delta g_{\mu\nu}\right). \tag{2.22}$$

More details are found in Ref. [11]. This particular reformulation of Einstein's equations is known as the Arnowitt-Deser-Misner (ADM) formulation. Unfortunately, this formulation of Einstein's equations has actually been found to be numerically unstable. To gain some insight into why this is the case, we use the following subsection to consider an analogy between the ADM equations and Maxwell's equations.

2.3 Parallels with Maxwell's Equations

One of the breakthroughs to solving Einstein's equations numerically was developing a formulation that was well suited for numerical evolution. To illustrate this point, we first examine Maxwell's equations in vacuum, i.e.,no source terms, in flat space and in Gaussian

and $c = 1$ units, given by

$$\vec{\nabla} \cdot \vec{E} = 0, \tag{2.23}$$

$$\vec{\nabla} \cdot \vec{B} = 0, \tag{2.24}$$

$$\frac{\partial \vec{B}}{\partial t} = -\vec{\nabla} \times \vec{E}, \tag{2.25}$$

$$\frac{\partial \vec{E}}{\partial t} = \vec{\nabla} \times \vec{B}. \tag{2.26}$$

We also have the associated vector and scalar potentials, $\vec{A}$ and φ, defined by

$$\frac{\partial \vec{A}}{\partial t} = -\vec{E} - \vec{\nabla}\varphi, \tag{2.27}$$

$$\vec{B} = \vec{\nabla} \times \vec{A}. \tag{2.28}$$

Now, replacing $\vec{B}$ with $\vec{\nabla} \times \vec{A}$ in Ampere's law, and using the standard identity

$$\vec{\nabla} \times \left(\vec{\nabla} \times \vec{A}\right) = \vec{\nabla}\left(\vec{\nabla} \cdot \vec{A}\right) - \nabla^2 \vec{A}, \tag{2.29}$$

we write

$$\vec{\nabla} \times \vec{B} = \vec{\nabla} \times \left(\vec{\nabla} \times \vec{A}\right) = \vec{\nabla}\left(\vec{\nabla} \cdot \vec{A}\right) - \nabla^2 \vec{A}. \tag{2.30}$$

Thus, our time evolution equations become

$$\frac{\partial \vec{A}}{\partial t} = -\vec{E} - \vec{\nabla}\varphi, \tag{2.31}$$

$$\frac{\partial \vec{E}}{\partial t} = \vec{\nabla}\left(\vec{\nabla} \cdot \vec{A}\right) - \nabla^2 \vec{A}. \tag{2.32}$$

Using index notation, in Cartesian coordinates we have

$$\partial_t A^i = -E^i - \partial^i \varphi, \tag{2.33}$$

$$\partial_t E^i = \partial^i \partial_j A^j - \partial_j \partial^j A^i. \tag{2.34}$$

Note the presence of the mixed second derivative above, and that in our system of equations we have six equations, but seven unknowns. Thus, we add a time evolution equation to φ as well, which amounts to choosing a gauge. In particular we choose the Lorenz gauge,

$$\partial_t \varphi = -\partial_i A^i. \tag{2.35}$$

Furthermore, note that because we are working in vacuum, we have the constraints

$$\partial_i E^i = 0, \tag{2.36}$$

$$\partial_i B^i = 0. \tag{2.37}$$

Since $\vec{B} = \vec{\nabla} \times \vec{A}$, the divergence-free condition on $\vec{B}$ is automatically satisfied. The evolution equations are thus

$$\partial_t A^i = -E^i - \partial^i \varphi, \tag{2.38}$$

$$\partial_t E^i = \partial^i \partial_j A^j - \partial_j \partial^j A^i, \tag{2.39}$$

$$\partial_t \varphi = -\partial_i A^i, \tag{2.40}$$

subject to the constraint

$$\mathcal{C} \equiv \partial_i E^i = 0. \tag{2.41}$$

Tracking the departure of our numerical results from this constraint helps us keep track of the numerical error. We refer to this system of equations as System I.

Note that in choosing to evolve the vector potential instead of the magnetic field, we guarantee the absence of magnetic monopoles to machine precision. In typical applications which model interactions between plasmas and electromagnetic fields, maintaining a divergence-less magnetic field is crucial so as to not introduce un-physical dynamics caused by magnetic monopoles, such as the advection of field-lines.

Note the first term on the right hand side of Eq. (2.39), which contains mixed second derivatives of the vector potential. While it looks fairly innocuous, it actually poses major

numerical issues. To illustrate this, consider the one-dimensional wave equation,

$$\partial_t^2 u\,(t,x) = c^2 \partial_i \partial^i u\,(t,x)\,. \tag{2.42}$$

This equation, which models the propagation of some scalar field u across space and time at some wave speed c, belongs to a family of partial differential equations classified as "hyperbolic" partial differential equations (PDEs). These are identified by their nature of being second order in space and time, and contain wave-like solutions. This immediately implies that Eqs. (2.38-2.40) is merely *weakly* hyperbolic, as it contains mixed spatial derivatives of the vector potential. To be more concrete, we outline how to analyze the hyperbolicity of Eqs. (2.38-2.40), following the discussion presented in Ref. [11]. First, one recasts the system of equations to be first order in space, making substitutions such that all second-order partial derivatives become first-order. One then writes in compact form

$$\partial_t \mathbf{u} + \mathbf{A}^i \cdot \partial_i \mathbf{u} = \mathbf{S} \tag{2.43}$$

where $\mathbf{u}$ is the solution vector with 7 components, $\mathbf{S}$ is the source vector, and $\mathbf{A}^i$ is a 7×7 matrix. One then considers an arbitrary unit vector n^i and constructs a new matrix $\mathbf{P} = \mathbf{A}^i n_i$, which is known as the characteristic matrix. Eq. (2.43) is said to strongly hyperbolic if, for all n^i, $\mathbf{P}$ possesses real-valued eigenvalues and orthogonal eigenvectors which span the vector space, i.e.,they form a complete set. It $\mathbf{P}$ has real-valued eigenvalues but not a complete set of eigenvectors, then the system is weakly hyperbolic.

The equations arising from the ADM formulation were similarly found to be weakly hyperbolic, which results in nodes of accumulated errors to remain stationary on numerical grids. In parallel with the fix that was applied to arrive at a numerically stable formulation of Einstein's equations, later known as the BSSN equations, we also *fix* the above system using a similar idea. Namely, to maintain numerical stability and accuracy we remove the

mixed second derivative term $\partial^i \partial_j A^j$ by introducing

$$\Gamma \equiv \partial_i A^i, \tag{2.44}$$

$$\partial_t \Gamma = \partial_i \partial_t A^i = -\partial_i E^i - \partial_i \partial^i \varphi \tag{2.45}$$

$$= -\partial_i \partial^i \varphi. \tag{2.46}$$

Thus, our evolution equations are

$$\partial_t A^i = -E^i - \partial^i \varphi, \tag{2.47}$$

$$\partial_t E^i = \partial^i \Gamma - \partial_j \partial^j A^i, \tag{2.48}$$

$$\partial_t \Gamma = -\partial_i \partial^i \varphi, \tag{2.49}$$

$$\partial_t \varphi = -\Gamma, \tag{2.50}$$

subject to the constraints

$$\mathcal{G} \equiv \Gamma - \partial_i A^i = 0, \tag{2.51}$$

$$\mathcal{C} \equiv \partial_i E^i = 0. \tag{2.52}$$

We refer to this system of equations as System II.

Now that we have these two systems of equations, we wish to compare their numerical results over time. First, however, we must construct initial data that will then be time evolved. Beginning from the analytic solution to System I given by Eq. 16 of Ref. [12], we have

$$A^{\hat{\phi}} = \mathcal{A} \sin\theta \left(\frac{e^{-\lambda v^2} - e^{-\lambda u^2}}{r^2} - 2\lambda \frac{v e^{-\lambda v^2} - u e^{-\lambda u^2}}{r} \right), \tag{2.53}$$

where $A^{\hat{\phi}} = A^{\phi} r \sin\theta$, $\mathcal{A}$ is the amplitude, λ describes the size of the wave packet, $u = t + r$, and $v = t - r$. Other components of the vector potential are 0. Note that these expressions represent the exact solution to both systems of equations at any time $t \geq 0$, at all points on our numerical grid. Thus, to get initial data we set $t = 0$.

For System II, we also need to set initial data for Γ. Since $\Gamma = -\partial_t \varphi$ and we have chosen $\varphi(t=0) = 0$, $\Gamma(t=0) = 0$. We calculate E^i using

$$E^i = -\partial_t A^i. \tag{2.54}$$

These data have free parameters A, λ and t.

Using the Einstein Toolkit [13], a collection of numerical relativity codes, we model both systems in Cartesian coordinates, with our results shown in Fig. 2.2. We see that in System I the errors associated with the divergence of the electric field, while convergent with grid resolution, are nonetheless constant in time. This is of course sub-optimal, since these errors, at this level far above machine precision, will eventually contaminate the global solution. Consider now System II, where these errors actually propagate away from our finite grid. In fact, at later times these errors, which we compute as a sum over the entire numerical grid, reduce down to machine precision levels.

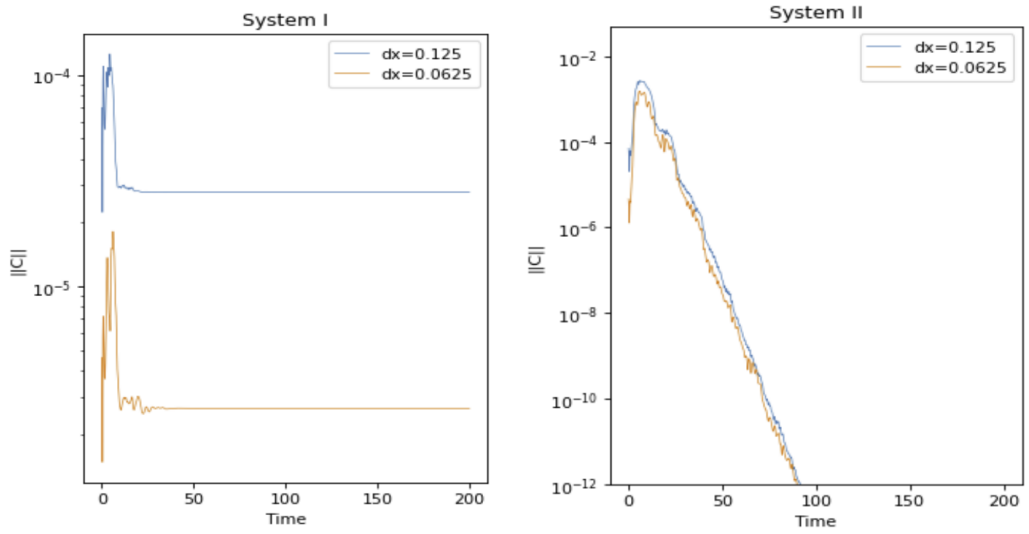

FIG. 2.2. Time evolution of constraint violation $\mathcal{C}$ in System I (**left**) and System II (**right**). Simulations were done using the Einstein Toolkit.

Please see Ref. [11] for further discussion on deriving the new full set of equations, known as the Baumgarte-Shapiro-Shibata-Nakamara (BSSN) formulation. These equations have been widely successful in modelling single, binary, and even multiple black hole systems. They have also been used when evolving non-vacuum spacetimes (e.g., [14–19], among many

others).

2.4 Maxwell's Equations in Curvilinear Coordinates

The previous Section served as motivation for the BSSN formulation, but the main goal of this dissertation is to efficiently model compact objects in dynamical spacetimes. To do so, we need to ensure that our evolution equations for the spacetime are well suited for integration in curvilinear coordinates. For example, if we were to model a non-rotating neutron star, it would be much more efficient to model the system using spherical coordinates. Therefore, we once again turn to Maxwell's equations to get a feel for what is required to evolve the BSSN equations in curvilinear coordinates.

Our current formulation of Maxwell's equations remains non-covariant, i.e.,the forms of the equations are altered if we move to curvilinear coordinates. To address this issue, and thus allow us to write the equations down in any arbitrary coordinate system, we must rewrite the equations to be manifestly covariant. We begin this process by employing the so-called "comma goes to semi-colon rule," replacing all partial derivatives, which do not transform as tensorial quantities, with covariant derivatives, which do. Rewriting Maxwell's equations with only tensorial quantities yields a covariant formulation of the equations,

$$\partial_t A^i = -E^i - \partial^i \varphi \rightarrow -E^i - \hat{\nabla}^i \varphi = -E^i - \hat{g}^{ij} \hat{\nabla}_j \varphi, \tag{2.55}$$

$$\partial_t E^i = \partial^i \Gamma - \partial_j \partial^j A^i \rightarrow \hat{\nabla}^i \Gamma - \hat{\nabla}_j \hat{\nabla}^j A^i = \hat{g}^{ij} \hat{\nabla}_j \Gamma - \hat{g}^{jk} \hat{\nabla}_j \hat{\nabla}_k A^i, \tag{2.56}$$

$$\partial_t \Gamma = -\partial_i \partial^i \varphi \rightarrow -\hat{\nabla}_i \hat{\nabla}^i \varphi = -\hat{g}^{ij} \hat{\nabla}_i \hat{\nabla}_j \varphi, \tag{2.57}$$

$$\partial_t \varphi = -\Gamma, \tag{2.58}$$

where $\hat{\nabla}_i$ is the covariant derivative associated with the reference metric $\hat{g}_{ij}$, which describes a flat space background in an arbitrary coordinate system. Expanding the above equations

using the standard covariant derivative definitions, we arrive at

$$\partial_t A^i = -E^i - \hat{g}^{ij}\partial_j \varphi, \tag{2.59}$$

$$\partial_t E^i = \hat{g}^{ij}\partial_j \Gamma - \hat{g}^{jk}\hat{\nabla}_j \hat{\nabla}_k A^i, \tag{2.60}$$

$$\partial_t \Gamma = -\hat{g}^{ij}\left(\partial_i \partial_j \varphi - \hat{\Gamma}^k_{ji}\partial_k \varphi\right), \tag{2.61}$$

$$\partial_t \varphi = -\Gamma, \tag{2.62}$$

subject to constraints

$$\mathcal{G} \equiv \Gamma - \partial_i A^i - \hat{\Gamma}^i_{ji} A^j = 0, \tag{2.63}$$

$$\mathcal{C} \equiv \partial_i E^i + \hat{\Gamma}^i_{ji} E^j = 0, \tag{2.64}$$

where $\hat{\Gamma}^i_{jk}$ is the metric connection associated with $\hat{g}_{ij}$. Since $\hat{\nabla}_k A^i$ is a tensor, its covariant derivative has the same indexing as a tensor T^i_k:

$$\hat{\nabla}_j T^i_k = T^i_{k,j} + \hat{\Gamma}^i_{dj} T^d_k - \hat{\Gamma}^d_{kj} T^i_d. \tag{2.65}$$

Therefore,

$$\hat{\nabla}_j\left(\hat{\nabla}_k A^i\right) = \left(A^i_{,k} + \hat{\Gamma}^i_{mk} A^m\right)_{,j} + \hat{\Gamma}^i_{dj}\left(A^d_{,k} + \hat{\Gamma}^d_{mk} A^m\right) - \hat{\Gamma}^d_{kj}\left(A^i_{,d} + \hat{\Gamma}^i_{md} A^m\right) \tag{2.66}$$

$$= A^i_{,kj} + \hat{\Gamma}^i_{mk,j} A^m + \hat{\Gamma}^i_{mk} A^m_{,j} + \hat{\Gamma}^i_{dj} A^d_{,k} + \hat{\Gamma}^i_{dj}\hat{\Gamma}^d_{mk} A^m - \hat{\Gamma}^d_{kj} A^i_{,d} - \hat{\Gamma}^d_{kj}\hat{\Gamma}^i_{md} A^m \tag{2.67}$$

$$= A^i_{,kj} + \hat{\Gamma}^i_{mk,j} A^m + \hat{\Gamma}^i_{mk} A^m_{,j} + \hat{\Gamma}^i_{dj} A^d_{,k} - \hat{\Gamma}^d_{kj} A^i_{,d} + \hat{\Gamma}^i_{dj}\hat{\Gamma}^d_{mk} A^m - \hat{\Gamma}^d_{kj}\hat{\Gamma}^i_{md} A^m. \tag{2.68}$$

Now, note that Eq. (2.63) requires partial derivatives of the vector potential. If we take spatial partial derivatives of the initial data defined by Eq. (2.53), we would mostly likely observe divergent behavior. In spherical coordinates, for example, radial derivatives of terms with a $1/r$ dependence are ill-defined along the z-axis and at the origin. Thus, to evolve these equations in curvilinear coordinates we explore the following.

Consider an arbitrary vector Λ^i, with smooth, continuous Cartesian components Λ^x, Λ^y, and Λ^z. Transforming Λ^i to, e.g. spherical coordinates, introduces terms that spoil the smoothness of Λ^i,

$$\Lambda^\phi = \frac{1}{r\sin\theta} \times [\text{smooth part}]\,. \tag{2.69}$$

Evolving Λ^ϕ will introduce instabilities along the z-axis. To avoid this, we instead evolve the rescaled quantity λ^i, defined by

$$\bar{\Lambda}^i = \frac{\lambda^i}{\text{scalefactor}[i]}. \tag{2.70}$$

where no sums are implied by the repeated indices, and the scale factors are the square root of the non-zero terms of the flat metric. Thus, we evolve the smoothed variable λ^i via

$$\lambda^i = \bar{\Lambda}^i \text{scalefactor}[i]. \tag{2.71}$$

Defining ReU[i] = 1/scalefactor[i], we have

$$\lambda^i = \frac{\bar{\Lambda}^i}{\text{ReU}[i]}. \tag{2.72}$$

We now define the rescaled quantities a^i and e^i as

$$a^i = \frac{A^i}{\text{ReU}[i]}, \tag{2.73}$$

$$\tag{2.74}$$

$$e^i = \frac{E^i}{\text{ReU}[i]}, \tag{2.75}$$

and rewrite our formulation of Maxwell's equations in curvilinear coordinates,

$$\partial_t a^i = \frac{\partial_t A^i}{\text{ReU}[i]} = \frac{-E^i - \hat{g}^{ij}\partial_j\varphi}{\text{ReU}[i]} = -e^i - \frac{\hat{g}^{ij}\partial_j\varphi}{\text{ReU}[i]}, \tag{2.76}$$

$$\partial_t e^i = \frac{\partial_t E^i}{\text{ReU}[i]} = \frac{\hat{g}^{ij}\partial_j\Gamma - \hat{g}^{jk}\hat{\nabla}_j\left(\hat{\nabla}_k A^i\right)}{\text{ReU}[i]} = \frac{\hat{g}^{ij}\partial_j\Gamma}{\text{ReU}[i]} - \frac{\hat{g}^{jk}\hat{\nabla}_j\left(\hat{\nabla}_k\left(a^i\text{ReU}[i]\right)\right)}{\text{ReU}[i]}. \tag{2.77}$$

The remainder of Maxwell's equations are unchanged:

$$\partial_t \Gamma = -\hat{g}^{ij} \left(\partial_i \partial_j \varphi - \hat{\Gamma}^k_{ji} \partial_k \varphi \right), \partial_t \varphi = -\Gamma, \tag{2.78}$$

subject to constraints

$$\mathcal{G} \equiv \Gamma - \partial_i A^i - \hat{\Gamma}^i_{ji} A^j = 0, \tag{2.79}$$

$$\mathcal{C} \equiv \partial_i E^i + \hat{\Gamma}^i_{ji} E^j = 0. \tag{2.80}$$

2.5 Radiation Boundary Conditions

Before moving on, we must discuss our implementation of boundary conditions. In practice we cannot sample our numerical solutions at an infinite number of points, and we must truncate numerical grids eventually. This immediately introduces the problem of how to treat points beyond the outer boundary of our numerical grid, called ghost points or ghost zones. There are many examples of reliable solutions to such a problem. One simple option is to copy data from the interior to the boundary points. This is commonly employed when evolving hydrodynamic fields, where material is likely to be very diffuse. This of course relies on the boundary either being very far away, or little to no fluid flows reaching the boundaries. In the general case however, especially when evolving gravitational fields, this approach tends to be numerically unstable over long timescales, when information from the boundaries has time to propagate throughout the entire numerical domain and contaminate all our data. For example, given a Cartesian grid with side-length L, we expect poor convergence of errors and numerical instabilities after a few light-crossing times $\tau_{\rm LC} = L/c$, where c is the propagation speed of the solution. $\tau_{\rm LC}$ therefore represents the time required for information to traverse the entire grid along one axis. Another common option is to extrapolate data to our ghost zones, which in turn is used to update data in the interior later. This too has the issue of eventual contamination of interior data. If we therefore wish to evolve systems for much longer than $\tau_{\rm LC}$, we need numerically stable boundary conditions that minimizes contamination of the numerical solution in the interior. A robust scheme we have implemented is the Sommerfeld boundary condition.

When numerically solving an initial value problem, appropriate initial data must be provided, coupled to a technique for evolving the initial data forward in time (to construct the solution at $t > 0$), and we must impose boundary conditions. The essential idea of a transparent boundary is creating a boundary in which wave fronts can pass through with minimal reflections. In other words, the boundary condition acts to map our numerical solution to outside our numerical domain in a smooth fashion. Because this mapping is assumed to be linear, this treatment occurs at the same time-level as calculations within our numerical domain.

Suppose we have a dynamical variable f; i.e., a variable in our hyperbolic system of PDEs that satisfies the equation

$$\partial_t f = \text{something}. \tag{2.81}$$

In general we refer to "something" as the "right-hand side" or "RHS" of the hyperbolic PDE, where we formulate the PDE such that the RHS contains no explicit time derivatives. However, it may contain spatial derivatives, which are computed using finite differences.

To construct the solution at times after the initial data, we adopt the Method of Lines (MoL) approach, which integrates the equations forward in time using standard explicit techniques typically used when solving ordinary differential equations. In doing so, MoL evaluates the RHS of the PDE at all points in our numerical domain, except at the ghost zones. Our Sommerfeld boundary condition implementation assumes f behaves as a spherically symmetric, outgoing wave at the boundaries. Under these assumptions, the waves satisfy the wave equation in spherical coordinates with angular parts set to zero, i.e., the solution $u(r,t)$ satisfies

$$\begin{aligned} 0 = \Box\, u &= \hat{D}^\nu \hat{D}_\nu u, && (2.82)\\ &= \hat{g}^{\mu\nu} \hat{D}_\mu \hat{D}_\nu u, && (2.83)\\ &= \hat{g}^{\mu\nu} \hat{D}_\mu \partial_\nu u, && (2.84)\\ &= \hat{g}^{\mu\nu} \left[\partial_\mu(\partial_\nu u) - \hat{\Gamma}^\alpha_{\mu\nu}(\partial_\alpha u) \right], && (2.85) \end{aligned}$$

where the hatted metric is defined as $\hat{g}^{tt} = -1/v^2$ (where v is the wave speed), $\hat{g}^{rr} = 1$, $\hat{g}^{\theta\theta} =$

$1/r^2$, and $\hat{g}^{\phi\phi} = 1/(r^2 \sin^2\theta)$, and are the only nonzero terms for the metric. However, the fact that $u = u(r,t)$ does not depend on the angular pieces greatly simplifies the expression to

$$\Box\, u = \hat{g}^{\mu\nu}\left[\partial_\mu(\partial_\nu u) - \hat{\Gamma}^\alpha_{\mu\nu}(\partial_\alpha u)\right], \tag{2.86}$$

$$= \left(-\frac{1}{v^2}\partial_t^2 + \partial_r^2\right) u - \hat{\Gamma}^\alpha_{\mu\nu}(\partial_\alpha u), \tag{2.87}$$

$$= \left(-\frac{1}{v^2}\partial_t^2 + \partial_r^2\right) u - \hat{g}^{\mu\nu}\left[\hat{\Gamma}^t_{\mu\nu}\partial_t + \hat{\Gamma}^r_{\mu\nu}\partial_r\right] u. \tag{2.88}$$

After removing terms implying angular variation in u, one obtains the wave equation for spherically symmetric waves,

$$\frac{1}{v^2}\partial_t^2 u = \partial_r^2 u + \frac{2}{r}\partial_r u, \tag{2.89}$$

which has the general solution

$$u(r,t) = A\frac{u(r+vt)}{r} + B\frac{u(r-vt)}{r}, \tag{2.90}$$

where the (left) right term represents an (ingoing) outgoing wave.

Inspired by the solution to the scalar wave equation, our Sommerfeld boundary condition will assume the solution $f(r,t)$ acts as an outgoing spherical wave ($A = 0$), with an asymptotic value f_0 at $r \to \infty$ and a correction term for transient non-wavelike behavior at the boundaries, with r^n falloff (ignoring higher-order radial falloffs).

$$f = f_0 + \frac{u(r-vt)}{r} + \frac{c(t,r,\theta,\phi)}{r^n}, \tag{2.91}$$

Here, c is some function of all four spacetime coordinates. Further, we do not apply Sommerfeld boundary conditions to f but to $\partial_t f$ instead,

$$\partial_t f = -v\frac{u'(r-vt)}{r}. \tag{2.92}$$

To get a better understanding of the $u'(r-vt)$ term, we compute the radial partial derivative,

$$\partial_r f = \frac{u'(r-vt)}{r} - \frac{u(r-vt)}{r^2} - n\frac{c}{r^{n+1}} \tag{2.93}$$

$$\implies \frac{u'(r-vt)}{r} = \partial_r f + \frac{u(r-vt)}{r^2} + n\frac{c}{r^{n+1}}. \tag{2.94}$$

Thus we get

$$\partial_t f = -v\frac{u'(r-vt)}{r} \tag{2.95}$$

$$= -v\left[\partial_r f + \frac{u(r-vt)}{r^2} + n\frac{c}{r^{n+1}}\right]. \tag{2.96}$$

To address the unknown $\frac{u(r-vt)}{r^2}$ term, notice our ansatz is

$$f = f_0 + \frac{u(r-vt)}{r} + \frac{c}{r^n}, \tag{2.97}$$

which implies that

$$\frac{f-f_0}{r} = \frac{u(r-vt)}{r^2} + \frac{c}{r^{n+1}} \tag{2.98}$$

$$\implies \frac{u(r-vt)}{r^2} = \frac{f-f_0}{r} - \frac{c}{r^{n+1}}, \tag{2.99}$$

so we have

$$\partial_t f = -v\left[\partial_r f + \frac{u(r-vt)}{r^2} + n\frac{c}{r^{n+1}}\right], \tag{2.100}$$

$$= -v\left[\partial_r f + \frac{f-f_0}{r} - \frac{c}{r^{n+1}} + n\frac{c}{r^{n+1}}\right], \tag{2.101}$$

$$= -v\left[\partial_r f + \frac{f-f_0}{r}\right] + \frac{k}{r^{n+1}}, \tag{2.102}$$

where $k = k(t,r,\theta,\phi) = -vc(n-1)$ is just another function of all four spacetime coordinates. Thus we have derived our boundary condition,

$$\boxed{\partial_t f = -\frac{v}{r}\left[r\partial_r f + (f-f_0)\right] + \frac{k(t,r,\theta,\phi)}{r^{n+1}},} \tag{2.103}$$

which gives the core equation

$$\boxed{\partial_t f = -\frac{v}{r}\left[r\partial_r f + (f - f_0)\right] + \frac{k}{r^{n+1}}.} \tag{2.104}$$

We expand the radial derivative in the previous equation to arrive at

$$\partial_t f = -\frac{v}{r(x^i)}\left[r\frac{\partial x^i}{\partial r}\partial_i f + (f - f_0)\right] + \frac{k}{r^{n+1}}. \tag{2.105}$$

The term containing $\partial x^i/\partial r$ can be impossible to compute directly, as we typically expect to be given $r(x^i)$ but not necessarily $x^i(r)$. The key here is to note that if we are given $x^j_{\text{Sph}} = (r(x^i), \theta(x^i), \phi(x^i))$ for all coordinate systems, then we can define the Jacobian

$$J = \frac{\partial x^j_{\text{Sph}}(x^i)}{\partial x^i}, \tag{2.106}$$

and inverting this matrix gives

$$J^{-1} = \frac{\partial x^i}{\partial x^j_{\text{Sph}}}. \tag{2.107}$$

J is an invertible matrix if it represents a transformation between two orthogonal coordinate systems and has a non-zero determinant.

Chapter 3

General Relativistic Hydrodynamics using a Finite Volume Method

The previous Chapter gave us some of the basic numerical tools we need to study systems with dynamical spacetimes. Using the NRPy infrastructure, we have access to the various time-stepping and finite-differencing algorithms needed to model such systems using high-order numerical methods. However, while we have viable and remarkably stable ways to evolve the spacetime using the BSSN formulation of Einstein's equations within NRPy/BlackHoles@Home, we still do not have a prescription for evolving the matter configurations.

The codes developed in this dissertation are employed for self-consistent simulations, where Eq. (2.11) is fully modeled. By self-consistent, we mean that we fully model the coupled evolution of the matter and spacetime dynamics. Of course, there are two alternatives to this rigorous approach. The first is where one simply evolves the spacetime, while keeping the matter distribution fixed. This typically serves as a code test, like what is done for the hydro-without-hydro test in Ref. [20], ensuring that the coupling of matter to the spacetime evolution via source terms appearing in the BSSN evolution equations are correct. This is of course inconsistent, and long term evolution of such a system is expected to become unstable. The second, and much more popular approach, is the Cowling approximation, in which the matter distribution is evolved on top of a fixed spacetime, and self-gravity of the matter is ignored. This is especially pertinent for systems where the spacetime is expected to be fairly stationary, such as the black hole - accretion disk systems studied in Refs. [21–25]. Thus, given that the black hole is much more massive than the disk, the dynamics or even accretion of the disk are not expected to drastically affect the spacetime.

In this Chapter we focus on how to model non-magnetized and magnetized fluids in a

general relativistic framework. We cover the equations which we use to model these fluids, and the various computational methods or algorithms needed to ensure stable evolutions of such fluids.

3.1 3+1 GRHD Equations

Before we write down and describe the equations we'll use to model fluids in general relativity, it is useful to be precise about our assumptions for the fluid. In this dissertation we assume that the fluid is perfect, meaning that we neglect viscosity, and further assume that the fluid is an ideal conductor, implying that there are no electric fields in the frame co-moving with the fluid. We begin with a covariant formulation of the conservation of stress-energy $T^{\mu\nu}$ and matter current density J^{μ}, which correspond to the conservation of momentum-energy and the continuity equation, taking the form

$$\nabla_{\mu} T^{\mu\nu} = 0, \tag{3.1}$$

$$\nabla_{\mu} J^{\mu} = 0, \tag{3.2}$$

where $J^{\mu} = \rho u^{\mu}$, ρ is the proper rest-mass density, and u^{μ} is the fluid four-velocity. In the perfect, non-magnetized fluid limit, ignoring heat conduction and viscosity, $T^{\mu\nu}$ takes the form

$$T^{\mu\nu} = \rho h u^{\mu} u^{\nu} + p g^{\mu\nu}, \tag{3.3}$$

where $h = 1 + \epsilon + p/\rho$ is the enthalpy, ϵ is the specific internal energy, and p is the pressure. Similar to what was required for Einstein's equations, we must recast Eq. (3.1) and Eq. (3.2) into 3+1 form, so that we may evolve fluid configurations from one hypersurface to the next. In doing so, we also wish to cast them into a *conservative form*, for two main reasons. First, so that we may take advantage of so-called high-resolution shock capturing techniques, where we ensure that our code accurately models shocks without coming across numerical issues. Second, by casting the equations in a conservative form, we ensure that in some instances our code is guaranteed to conserve rest-mass density, energy, and momentum to machine precision. Such instances include fluid flows without shocks or discontinuities, as

well as flows within spacetimes that are analytical solutions to Einstein's equations. In the general case, however, when considering systems with relativistic fluid flows coupled to a rapidly changing, non-trivial spacetime, numerical approximations will spoil the conservation properties. We write down the evolution equations in compact form as

$$\partial_t \boldsymbol{C} + \partial_i \boldsymbol{F}^i = \boldsymbol{S}. \tag{3.4}$$

$\boldsymbol{C}$ is the vector form of our *conserved*, or evolved, variables. The first is the densitized conserved baryonic density $\tilde{D} = \sqrt{\gamma} W \rho_b$, where $W = \alpha u^0$ is the observed Lorentz factor between observers in the fluid frame and the frame normal to the spatial hypersurface. We also have the densitized conserved momentum $\tilde{S}_i = \alpha\sqrt{\gamma} T^0_i$, and finally densitized conserved energy $\tilde{\tau} = \alpha^2\sqrt{\gamma} T^{00} - \tilde{D}$. These conserved variables are functions of the *primitive* variables $\boldsymbol{P} = \{\rho, p, u^\mu, \epsilon\}$.

The flux terms $\boldsymbol{F}^i$ are defined as

$$\boldsymbol{F}^i = \begin{bmatrix} \tilde{D} \\ \tilde{\tau} \\ \tilde{S}_j \end{bmatrix} \equiv \sqrt{\gamma} \begin{bmatrix} W\rho \\ \alpha^2 T^{00} - W\rho \\ \alpha T^0_{\ j} \end{bmatrix} , \tag{3.5}$$

while the source terms are

$$\boldsymbol{S} = \begin{bmatrix} 0 \\ \alpha\sqrt{\gamma}\left[\Theta^{ij} K_{ij} - \left(T^{00}\beta^i + T^{0i}\right)\partial_i \alpha\right] \\ \frac{1}{2}\alpha\sqrt{\gamma} T^{\mu\nu} g_{\mu\nu,i} \end{bmatrix} , \tag{3.6}$$

where $\Theta^{ij} \equiv T^{00}\beta^i\beta^j + 2T^{0i}\beta^j + T^{ij}$.

3.2 Magnetohydrodynamics

In many astrophysical systems magnetic fields play a crucial role in the dynamics. Whether through the magneto-rotational instability (MRI), which introduces turbulence in accretion disks, or powering relativistic jets in black hole - accretion disk systems via the

Blandford-Znajek mechanism, or the acceleration of charged particles in compact object magnetospheres, one must couple the dynamics of the system to the evolution of the electromagnetic (EM) field in order to fully capture the system dynamics. To do so, we must then include evolution of the EM fields in our hydrodynamic evolution equations. In the ideal MHD limit, where one assumes the fluid is a perfect conductor, we write the evolution of the conserved magnetic field $\tilde{B}^i = \sqrt{\gamma} B^i$, where B^i is the magnetic field as observed by a normal observer, using the induction equation [26]

$$\partial_t \tilde{B}^i + \partial_j \left(v^j B^i - v^i B^j \right) = 0. \tag{3.7}$$

To couple the evolution of the fluid, or plasma in this case, we need to add EM contributions to $T^{\mu\nu}$, which then takes the form

$$T^{\mu\nu} = \left(\rho h + b^2 \right) u^\mu u^\nu + \left(p + \frac{b^2}{2} \right) g^{\mu\nu} - b^\mu b^\nu, \tag{3.8}$$

where $b^\mu = B^\mu_{(u)}$ is the magnetic field as measured by an observer co-moving with the fluid, given that the electric field vanishes in the ideal MHD in this frame, defined as [27]

$$B^0_{(u)} = u_i B^i / \alpha, \tag{3.9}$$

$$B^i_{(u)} = \left(B^i / \alpha + u^i B^0_{(u)} \right) / u^0. \tag{3.10}$$

In general solving Eq. (3.7) using finite differences violates the divergence constraint of Maxwell's equations, which results in un-physical magnetic monopoles. Some popular methods for solving Eq. (3.7) can be found in Ref. [28], but their discussion is outside the scope of this dissertation.

3.3 Numerical Methods for Evolving Relativistic Fluids

In this Section we review all the ingredients needed to evolve non-magnetized relativistic fluids. One is required to use specialized algorithms because in general astrophysical fluids exhibit shocks, contact discontinuities, and other sharp discontinuities in the primitive

variables. If one was to naively apply finite differences to the GRHD equations while shocks are present, undesirable numerical effects would manifest, such as un-physical oscillations in the fluid profile.

3.3.1 Finite Volume Discretization

Instead of using finite differences to solve Eq. (3.4), we instead use a finite volume method. This is mainly because of the desirable property of finite-volume methods, which guarantee the preservation of conservation laws Eq. (3.1) and Eq. (3.2) to machine precision. In the BlackHoles@Home infrastructure [29] we sample quantities at cell-centers, avoiding coordinate-singularities when using singular curvilinear coordinates, using a uniform grid discretization for coordinates $x^i = (x^1, x^2, x^3)$ of the form

$$x^i(j) = \Delta x^i \left(j - N_G + \frac{1}{2} \right), \tag{3.11}$$

where j is the grid index, Δx^i is the grid spacing, and N_G is the number of ghost cells, used for finite differences, interpolations, and reconstruction. On our numerical grid we then have infinitesimal cell volumes $dV = dxdydz$. Following the discussion presented in [30], if we take a volume integral of Eq. (3.4) over a single grid cell with extents $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$,

$\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$, and so on, we get

$$\partial_t \int_V \boldsymbol{C} dV + \int_V dV \partial_i \boldsymbol{F}^i = \int_V dV \boldsymbol{S}, \tag{3.12}$$

$$V\partial_t \bar{\boldsymbol{C}} + \int_V dV \partial_i \boldsymbol{F}^i = V\bar{\boldsymbol{S}},$$

$$V\partial_t \bar{\boldsymbol{C}} + \int_V dV \left(\partial_x \boldsymbol{F}^1 + \partial_y \boldsymbol{F}^2 + \partial_z \boldsymbol{F}^3\right) = V\bar{\boldsymbol{S}},$$

$$V\partial_t \bar{\boldsymbol{C}} + \int_A dydz \boldsymbol{F}^1 + \int_A dxdz \boldsymbol{F}^2 + \int_A dxdy \boldsymbol{F}^3 = V\bar{\boldsymbol{S}},$$

$$V\partial_t \bar{\boldsymbol{C}} + \int_A dydz \boldsymbol{F}^1 \Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} + \int_A dxdz \boldsymbol{F}^2 \Big|_{y_{i-\frac{1}{2}}}^{y_{i+\frac{1}{2}}} + \int_A dxdy \boldsymbol{F}^3 \Big|_{z_{i-\frac{1}{2}}}^{z_{i+\frac{1}{2}}} = V\bar{\boldsymbol{S}},$$

$$V\partial_t \bar{\boldsymbol{C}} + \Delta y \Delta z \breve{\boldsymbol{F}}^1 \Big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} + \Delta x \Delta z \breve{\boldsymbol{F}}^2 \Big|_{y_{i-\frac{1}{2}}}^{y_{i+\frac{1}{2}}} + \Delta x \Delta y \breve{\boldsymbol{F}}^3 \Big|_{z_{i-\frac{1}{2}}}^{z_{i+\frac{1}{2}}} = V\bar{\boldsymbol{S}},$$

$$\partial_t \bar{\boldsymbol{C}} + \frac{\breve{\boldsymbol{F}}^1_{x_{i+\frac{1}{2}}} - \breve{\boldsymbol{F}}^1_{x_{i-\frac{1}{2}}}}{\Delta x} + \frac{\breve{\boldsymbol{F}}^2_{y_{i+\frac{1}{2}}} - \breve{\boldsymbol{F}}^2_{y_{i-\frac{1}{2}}}}{\Delta y} + \frac{\breve{\boldsymbol{F}}^3_{z_{i+\frac{1}{2}}} - \breve{\boldsymbol{F}}^3_{z_{i-\frac{1}{2}}}}{\Delta z} = \bar{\boldsymbol{S}}, \tag{3.13}$$

where $V = \Delta x \Delta y \Delta z$ is the cell volume, $\bar{Q} \equiv \frac{1}{V}\int_V dVQ$ represents the volume average of a point-valued quantity Q, and $\breve{Q} \equiv \frac{1}{A}\int_A daQ$ is the average over a surface area A. Notice that while this result looks very similar to a finite difference operation, similar to Eq. (2.4), Eq. (3.13) is actually an exact expression, in the sense that we haven't applied any approximations. To solve this equation, we need to prescribe a way to compute the volume and surface averages. This sets the approximation of our finite volume method. Most GRHD and GRMHD codes approximate these averages using the value of the quantity at the volume centroid, which is a second order accurate approximation. In Cartesian coordinates the volume centroid coincides with the cell-center, but in general curvilinear this is not the case (see Refs. [31, 32] for more details.) Of course, one can move beyond second order accuracy, but then two main issues arise. The first is specifying how to compute the averages at high-order, and to do so in such a way that is computationally efficient. Second, one must take extreme care in the computation of the flux and source terms, making sure to make clear distinctions between point-wise and averaged quantities, as mixing up the two anywhere in the computations will degrade the overall order of the scheme.

We note that at second order accuracy, the prescription Eq. (3.13) for integrating the evolution equations is equivalent to simply using a second order finite difference approxima-

tion to Eq. (3.4). Therefore, the main difference between a finite volume and finite difference method in the context of GRHD or GRMHD only occurs in higher order schemes.

3.3.2 High Resolution Shock-Capturing

Integrating Eq. (3.4) in time requires us to evaluate the surface area averaged fluxes at the cell interfaces $x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}$. To simplify our discussion, however, we approximate these averages with the face-centered values. This means that in our approximation of averages with face-centered values, we need to interpolate values from the cell-centers to the cell faces, or interfaces. Specifically, we need to interpolate the primitive values. However, as with taking finite differences, we cannot apply the usual interpolation algorithms, such as Lagrangian or Hermite interpolation, where we approximate the primitives, where we use local polynomial approximations of the primitive variables and evaluate these polynomials at cell-interfaces. The issue lies in the occurrence of shocks or other sharp discontinuities in the profiles of the primitive variables. If we apply these interpolation algorithms in the presence of sharp discontinuities, we are sure to encounter non-physical oscillatory behavior. Moreover, when a shock is present, we want to ensure that the Rankine-Hugoniot jump conditions are satisfied, which essentially demand that our conservation laws are satisfied in the presence of shocks.

Thus, to evaluate the fluxes at the interfaces, we must use high resolution shock-capturing (HRSC) methods. In essence, these methods allow us to resolve and evolve shocks and other sharp fluid discontinuities, as well as provide a solution to the Riemann problem, and thereby provide an estimate for the fluxes at the interfaces. In the following sections we describe the key ingredients of an HRSC scheme.

3.3.2.1 Solving for Fluxes

When solving the GRHD equations, we need to ensure that rest-mass, momentum and energy are conserved. It is therefore common practice in the literature to use a *Riemann solver*. One of the most common solvers is the Harten, Lax, and van Leer (HLL) Riemann

solver [33], which has the form

$$F^{\rm HLL} = \frac{c_{\rm min} F_{\rm R} + c_{\rm max} F_{\rm L} - c_{\rm min} c_{\rm max} (P_{\rm R} - P_{\rm L})}{c_{\rm min} + c_{\rm max}}. \tag{3.14}$$

The R, L subscripts denote primitive variables and flux terms, defined in Eq. (3.5), evaluated on the left and right sides of the cell-interface, meaning at $x_{i+\frac{1}{2}+\epsilon}$ and $x_{i+\frac{1}{2}-\epsilon}$. The HLL solver only considers the minimum and maximum characteristic speeds, $c_{\rm min}$ and $c_{\rm max}$, of the GRHD equations. In pure hydrodynamics, these can be computed by solving the dispersion relation for MHD waves with $B^i = 0$, leading to the quadratic equation $ac_{\pm}^2 + bc_{\pm} + c = 0$, where

$$a = (1 - c_s^2) (u^0)^2 - c_s^2 g^{00}, \tag{3.15}$$

$$b = 2c_s^2 g^{i0} - 2u^i u^0 (1 - c_s^2), \tag{3.16}$$

$$c = (1 - c_s^2) (u^i)^2 - c_s^2 g^{ii}, \tag{3.17}$$

$$c_+ = \max \left(\tfrac{-b \pm \sqrt{b^2 - 4ac}}{2a} \right), \tag{3.18}$$

$$c_- = \min \left(\tfrac{-b \pm \sqrt{b^2 - 4ac}}{2a} \right), \tag{3.19}$$

where c_s is the sound speed. One can also consider more elaborate solvers by including additional characteristics, such as the Harten, Lax, van Leer - Contact (HLLC) [34, 35] solver.

3.3.2.2 Primitive Reconstruction

As we alluded to earlier, we need a more robust, "shock-aware" interpolation algorithm that avoids using high-order polynomials to approximate the primitives at cell-interfaces. One of the most popular methods to do so is by introducing a so-called *slope-limiter*. Slope-limiters act to smear out sharp discontinuities across several grid cells, such that it can then be adequately resolved. By applying a slope-limiter, we are therefore creating a scheme that is *total variation diminishing* (TVD). If we define the total variation of scalar function $f(x)$ as

$$TV(f) = \int_{-\infty}^{\infty} |f'(x)| \, dx, \tag{3.20}$$

then we must require that over the course of an evolution from iteration n to iteration $n+1$

$$TV\left(f^{n+1}\right) \leq TV\left(f^{n}\right). \tag{3.21}$$

By enforcing this property in our numerical solution, we ensure un-physical oscillations do not grow in amplitude. One such kind of slope-limiter is the minmod limiter, defined as

$$\sigma_i = \text{minmod}\left(\frac{f_i - f_{i-1}}{\Delta x}, \frac{f_{i+1} - f_i}{\Delta x}\right) \tag{3.22}$$

where

$$\text{minmod}\left(a, b\right) = \begin{cases} a & ; \quad \text{if } |a| < |b| \text{ and } ab > 0 \\ b & ; \quad \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & ; \quad \text{if } ab \leq 0. \end{cases} \tag{3.23}$$

Thus, to solve the right and left sides of the cell-interfaces, on say the left interface of grid cell i, as required by the HLL solver Eq. (3.14), we use the stencil $f_{i-2}, f_{i-1}, f_i, f_{i+1}$ to obtain values of $f_r = f_{i-\frac{1}{2}+\epsilon}$, $f_l = f_{i-\frac{1}{2}-\epsilon}$ using

$$f_r \quad = f_i - \tfrac{1}{2}\sigma_i, \tag{3.24}$$

$$f_l \quad = f_{i-1} + \tfrac{1}{2}\sigma_{i-1}. \tag{3.25}$$

This slope limiter is second-order accurate for smooth flows and reduces to first order at shocks or other discontinuities. However, it would be desirable to have higher-order methods for smooth-flows. Such methods are the piece-wise parabolic method (PPM), the fifth-order monotinicity-preserving reconstruction (MP5) method, and fifth-order weighted essentially non-oscillatory reconstruction (WENO5).

PPM approximates the underlying function using a parabola, which is third order accurate, but uses a shock-detecting algorithm to deduce if there is a shock present within the grid cell. The shock-detection algorithm considers gradients in the pressure, and applies a slope-limiter if one is present. Note, however, that without additional geometric factors the pressure gradient is incorrect in curvilinear coordinates. MP5 also has a similar

issue in curvilinear coordinates, which most likely explains the failure of the reconstruction methods in spherical coordinates as noted by Ref. [36]. WENO5 reconstruction (e.g.,[37] and references therein) considers a convex combination of third-order trial polynomials, and applies weights to the stencils based on a smoothness indicator. This approach is attractive in curvilinear coordinates because of its independence from considering any gradients of the underlying function.

3.3.3 Conservative-to-Primitive Solver

Given an EoS, one can compute the conservative variables from the primitive variables. The inverse is non-trivial, as in general there does not exist a closed-form solution for solving for primitives, at least in the context of GRMHD, as the conservatives depend non-linearly on the primitives. Therefore, there have been several algorithms that have been developed to solve this problem. The basic idea is to reduce the dimensionality of the algebraic problem to one or two dimensions, and use a root-finding algorithm to solve for the primitives. Examples are found in Refs. [38–41].

A common distinction between these solvers, beyond their dimensionality, is the specific combination of conservative variables they use for the root-finding. Typically solvers can use either the conserved energy or conserved entropy. The entropy follows a conservation law of the form

$$\nabla_\nu \left(S u^\mu\right) = 0, \tag{3.26}$$

$$\implies \partial_t \tilde{S} + \partial_i \left(\tilde{S} v^i\right) = 0. \tag{3.27}$$

However, note that at shocks entropy is not conserved, so the conservation law is no longer valid.

After solving for the primitives at a given time-step, one measures the accuracy of the solution by computing the relative error, as given by Eq. (2.2), between the conservatives computed after a time step, and the conservatives computed using the new primitive variables recovered from the conservative-to-primitive routine. In ideal situations the error would be at machine precision, meaning that the recovered primitive variables are entirely consistent

with the evolved conservatives. In our tests we find that a relative error of 10^{-8} is satisfactory, and we also see this error decrease with increasing grid resolution.

3.3.4 Equation of State

The GRMHD equations are closed once an EoS is assumed. Of course, such a choice is problem-dependent. Given that these equations can be reliably applied to model a wide-array of astrophysical systems, once must similarly have a form for the EoS that is readily applicable. One of the most common analytic forms is the polytropic EoS, which takes the form

$$P = K\rho^{\Gamma}, \tag{3.28}$$

where K is the polytropic gas constant, while Γ governs the "stiffness" of the EoS. This EoS is usually used for cold models, as heating, e.g.,from shocks, is not modeled. Therefore it is typically used to construct cold initial data models. In the isentropic limit this is equivalent to the ideal fluid EoS, which takes the form

$$P = (\Gamma - 1)\,\rho\epsilon, \tag{3.29}$$

where ϵ is the specific internal energy. For example, numerical studies of black hole - accretion disk systems typically adopt $\Gamma = 5/3$ (e.g.,Ref. [42]), which models non-relativistic, non-interacting degenerate matter [11], while for simulations involving neutron stars a common choice is $\Gamma = 2$, which models cold, relativistic, degenerate matter.

While these analytic equations of state (EoSs) are quite convenient for numerical simulations, they nonetheless lack realism, at least in the case of approximating nuclear EoSs. More specifically, to include more realism, one must also model finite-temperature effects, as well as neutrino emission and absorption. It has therefore become popular to approximate the nuclear EoS using EoS tables. These tables define thermodynamic and compositional quantities in tabular form for finite ranges. In the stellarcollapse format [1], each quantity Q is a function of density, temperature, and electron fraction, defined as the ratio between the lepton and baryon number densities, $Q = Q(\rho, T, Y_e)$. Thermodynamic quantities in-

[1] see http://stellarcollapse.org.

clude pressure, entropy, specific internal energy, and sound speed. Compositional quantities include neutron, proton, and electron chemical potentials. Thus, given the combination (ρ, T, Y_e), one computes their desired quantity, employing interpolation to acquire data at any point within the ranges of (ρ, T, Y_e).

Chapter 4

General Relativistic Hydrodynamics using a Reference Metric Formulation

In this Section we derive the evolution equations of general relativistic hydrodynamics (GRHD) in curvilinear coordinates, for hybrid equations of state (EoSs), using a reference metric approach. While the original equations as implemented in other codes in Cartesian coordinates are written to be covariant and coordinate-free, one could indeed just directly evolve the equations in curvilinear coordinates. However, as shown in Ref. [43], there are numerical advantages to using a reference metric formulation of the equations. While there is some overlap with the brief derivations presented in Chapter 5, which reproduces Ref. [44], here we show the full derivations.

To set the stage, we consider that our spatial metric γ_{ij} has the form

$$\gamma_{ij} = e^{4\phi}\bar{\gamma}_{ij}, \tag{4.1}$$

where $\bar{\gamma}_{ij}$ is the conformal metric, $\psi \equiv e^{\phi}$ is the conformal factor, and ϕ is the conformal exponent. We further decompose the conformal metric using

$$\bar{\gamma}_{ij} = \hat{\gamma}_{ij} + \epsilon_{ij}, \tag{4.2}$$

where $\hat{\gamma}_{ij}$ is the flat space reference metric in the chosen coordinate system, and ϵ_{ij} represents deviations from flat space that are not necessarily small. To derive the GRHD equations within a reference metric approach, we begin by following the derivation given in [43], Eqs. (9), and (12-15), considering the continuity equation

$$\nabla_a \left(\rho_0 u^a\right) = 0, \tag{4.3}$$

where $a, b, c, \cdots$ are spacetime indices. We use the identity

$$\nabla_a V^a = \frac{1}{\sqrt{|g|}} \partial_a \left(\sqrt{|g|} V^a \right), \tag{4.4}$$

which holds for *any* metric and its associated covariant derivative, to obtain

$$\nabla_a \left(\rho_0 u^a \right) = \frac{1}{\sqrt{-g}} \partial_a \left(\sqrt{-g} \rho_0 u^a \right) \tag{4.5}$$

$$= \frac{1}{\sqrt{-g}} \left(\partial_t \left(\sqrt{-g} \rho_0 u^t \right) + \partial_i \left(\sqrt{-g} \rho_0 u^i \right) \right) = 0. \tag{4.6}$$

Now, using the result $\sqrt{-g} = \alpha \sqrt{\gamma} = \alpha e^{6\phi} \sqrt{\bar{\gamma}}$, we rewrite the spatial components of Eq. (4.6) as

$$\partial_i \left(\alpha e^{6\phi} \sqrt{\bar{\gamma}} \rho_0 u^i \right) = \partial_i \left(\sqrt{\hat{\gamma}} \alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \rho_0 u^i \right), \tag{4.7}$$

but since the identity Eq. (4.4) for any metric, we also have

$$\hat{\nabla}_i \left(\alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \rho_0 u^i \right) = \frac{1}{\sqrt{\hat{\gamma}}} \partial_i \left(\sqrt{\hat{\gamma}} \alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \rho_0 u^i \right), \tag{4.8}$$

therefore,

$$\partial_i \left(\alpha e^{6\phi} \sqrt{\bar{\gamma}} \rho_0 u^i \right) = \sqrt{\hat{\gamma}} \hat{\mathcal{D}}_i \left(\alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \rho_0 u^i \right), \tag{4.9}$$

where Eq. (4.4) was used again, but this time with respect to the reference metric $\hat{\gamma}_{ij}$. Dividing all terms by $\sqrt{\hat{\gamma}}$, we arrive at

$$\partial_t \left(\alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \rho_0 u^t \right) + \hat{\mathcal{D}}_i \left(\alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \rho_0 u^i \right) = 0. \tag{4.10}$$

If we define the densitized baryonic density as $D = \alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \rho_0 u^t$, and the flux as $(f_D)^i = \alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \rho_0 u^i = D \left(v^i - \frac{\beta^i}{\alpha} \right)$, where $v^a \equiv \gamma^a_b \left(\frac{u^b}{W} + \frac{\beta^b}{\alpha} \right)$ is the fluid velocity measured by a normal observer and $W \equiv \alpha u^t$, we then have in compact form

$$\partial_t D + \hat{\mathcal{D}}_i \left(f_D \right)^i = 0. \tag{4.11}$$

Here we make the critical observation that in the original formulation, recovered when we define the reference metric in Cartesian coordinates, there are no source terms, but in curvilinear coordinates, e.g.,spherical coordinates, source terms appear from connection coefficients associated with the reference metric. This leads to mass being conserved up to truncation error, instead of to round-off error in Cartesian coordinates, as shown and discussed in Refs. [43, 44].

To derive the momentum equation, from conservation of energy—momentum, we begin with the identity (Eq. 22 in Ref. [43])

$$\nabla_b A^b_a = \frac{1}{\sqrt{-g}}\partial_b\left(\sqrt{-g}A^b_a\right) - A^{b\,(4)}_c\Gamma^c_{ba}, \tag{4.12}$$

where ${}^{(4)}\Gamma^c_{ba}$ is the metric connection associated with the spacetime metric $g_{\mu\nu}$. We start with taking a spatial projection of $T^{\mu\nu}$:

$$\begin{aligned} 0 = \gamma_{ib}\nabla_a T^{ab} = g_{ib}\nabla_a T^{ab} &= \nabla_a\left(g_{ib}T^{ab}\right), && (4.13)\\ &= \frac{1}{\sqrt{-g}}\partial_a\left(\sqrt{-g}T^a_i\right) - T_a{}^{b\,(4)}\Gamma^a_{ib}, && (4.14)\\ &= \frac{1}{\sqrt{-g}}\left(\partial_t\left(\sqrt{-g}T^t_i\right) + \partial_j\left(\sqrt{-g}T^j_i\right)\right) - T_a{}^{b\,(4)}\Gamma^a_{ib}. && (4.15)\end{aligned}$$

Similar to what was done with the continuity equation, we expand the terms with spatial derivatives:

$$\begin{aligned} \partial_j\left(\sqrt{-g}T^j_i\right) &= \partial_j\left(\sqrt{\hat{\gamma}}\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}T^j_i\right) && (4.16)\\ &= \sqrt{\hat{\gamma}}\hat{\mathcal{D}}_j\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}T^j_i\right) + \alpha e^{6\phi}\sqrt{\bar{\gamma}}T^j_k\hat{\Gamma}^k_{ij}. && (4.17)\end{aligned}$$

Now, plugging this into the previous equation, replacing $\sqrt{-g}$, dividing through by $\sqrt{\hat{\gamma}}$, and assuming $\partial_t\sqrt{\hat{\gamma}} = 0$ we have

$$\begin{aligned} \partial_t\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}}T^t_i\right) &= -\sqrt{\hat{\gamma}}\hat{\mathcal{D}}_j\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}T^j_i\right) - \alpha e^{6\phi}\sqrt{\bar{\gamma}}T^j_k\hat{\Gamma}^k_{ij} + \alpha e^{6\phi}\sqrt{\bar{\gamma}}T_a{}^{b\,(4)}\Gamma^a_{ib}, && (4.18)\\ \partial_t\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}T^t_i\right) &= -\hat{\mathcal{D}}_j\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}T^j_i\right) - \alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\left(T^j_k\hat{\Gamma}^k_{ij} - T_a{}^{b\,(4)}\Gamma^a_{ib}\right). && (4.19)\end{aligned}$$

Our final task is to express ${}^{(4)}\Gamma^a_{ib}$, which is associated with g_{ab}, in terms of 3+1 variables. We follow Eqs. 29-34 in Ref. [43], expanding

$$T_a{}^{b\,(4)}\Gamma^a_{bi} - T_k{}^j\hat{\Gamma}^k_{ji} = T^{cb\,(4)}\Gamma_{cbi} - T^{cj}g_{kc}\hat{\Gamma}^k_{ji}, \tag{4.20}$$

into terms that contain only the time component T^{00}, those that contain only the mixed-component terms T^{0j}, and those that contain only spatial component terms T^{jk}. Beginning with terms that contain only T^{00},

$$T^{00\,(4)}\Gamma_{00i} = \frac{1}{2}T^{00}\partial_i g_{00} = \frac{1}{2}T^{00}\partial_i\left(-\alpha^2 + \gamma_{jk}\beta^j\beta^k\right). \tag{4.21}$$

Note that in the last expression, the term in the parentheses is a scalar on each hypersurface, so can we replace ∂_i with $\hat{\mathcal{D}}_i$, giving

$$T^{00\,(4)}\Gamma_{00i} = \frac{1}{2}T^{00}\hat{\mathcal{D}}_i\left(-\alpha^2 + \gamma_{jk}\beta^j\beta^k\right) \tag{4.22}$$

$$= \frac{1}{2}T^{00}\left(-2\alpha\hat{\mathcal{D}}_i\alpha + 2\beta_k\hat{\mathcal{D}}_i\beta^k + \beta^j\beta^k\hat{\mathcal{D}}_i\gamma_{jk}\right). \tag{4.23}$$

For the terms involving T^{0j} we have

$$T^{0j\,(4)}\Gamma_{0ji} + T^{j0\,(4)}\Gamma_{j0i} - T^{0j}g_{k0}\hat{\Gamma}^k_{ji} = T^{0j}\left({}^{(4)}\Gamma_{0ji} + {}^{(4)}\Gamma_{j0i} - \beta_k\hat{\Gamma}^k_{ji}\right), \tag{4.24}$$

$$= T^{0j}\left(\partial_i\beta_j - \beta_k\hat{\Gamma}^k_{ji}\right) = T^{0j}\hat{\mathcal{D}}_i\beta_j, \tag{4.25}$$

$$= T^{0j}\hat{\mathcal{D}}_i\left(\gamma_{jk}\beta^k\right) = T^{0j}\left(\gamma_{jk}\hat{\mathcal{D}}_i\beta^k + \beta^k\hat{\mathcal{D}}_i\gamma_{jk}\right), \tag{4.26}$$

$$= T^{0j}\left(g_{jk}\hat{\mathcal{D}}_i\beta^k + \beta^k\hat{\mathcal{D}}_i\gamma_{jk}\right). \tag{4.27}$$

Note that $T^{0\mu}g_{\mu\nu} = T^0_\nu = T^{00}g_{0\nu} + T^{0j}g_{j\nu}$, thus, $T^0_k = T^{00}g_{0k} + T^{0j}g_{jk}$, which gives

$$T^{0j}\left(g_{jk}\hat{\mathcal{D}}_i\beta^k + \beta^k\hat{\mathcal{D}}_i\gamma_{jk}\right) = T^0_k\hat{\mathcal{D}}_i\beta^k - T^{00}\beta_k\hat{\mathcal{D}}_i\beta^k + T^{0j}\beta^k\hat{\mathcal{D}}_i\gamma_{jk} \tag{4.28}$$

Now moving to just the spatial terms,

$$T^{jk}\left({}^{(4)}\Gamma_{jki}-\gamma_{kl}\hat{\Gamma}^{l}_{ji}\right)=T^{jk}\left(\Gamma_{jki}-\gamma_{kl}\hat{\Gamma}^{l}_{ji}\right), \tag{4.29}$$

$$=T^{jk}\left(\frac{1}{2}\left(\partial_i\gamma_{jk}+\partial_k\gamma_{ji}-\partial_j\gamma_{ki}\right)-\gamma_{kl}\hat{\Gamma}^{l}_{ji}\right), \tag{4.30}$$

$$=T^{jk}\left(\frac{1}{2}\partial_i\gamma_{jk}-\gamma_{kl}\hat{\Gamma}^{l}_{ji}\right), \tag{4.31}$$

$$=T^{jk}e^{4\phi}\left(2\bar{\gamma}_{jk}\partial_i\phi+\frac{1}{2}\partial_i\bar{\gamma}_{jk}-\bar{\gamma}_{kl}\hat{\Gamma}^{l}_{ji}\right), \tag{4.32}$$

$$=T^{jk}e^{4\phi}\left(2\bar{\gamma}_{jk}\partial_i\phi+\bar{\gamma}_{jl}\left(\bar{\Gamma}^{l}_{ki}-\hat{\Gamma}^{l}_{ki}\right)\right), \tag{4.33}$$

$$=T^{jk}e^{4\phi}\left(2\bar{\gamma}_{jk}\partial_i\phi+\frac{1}{2}\hat{\mathcal{D}}_i\bar{\gamma}_{jk}\right), \tag{4.34}$$

$$=\frac{1}{2}T^{jk}\hat{\mathcal{D}}_i\gamma_{jk}, \tag{4.35}$$

where we use

$$T^{jk}\bar{\mathcal{D}}_i\bar{\gamma}_{jk}=T^{jk}\left(\partial_i\bar{\gamma}_{jk}-\bar{\Gamma}^{l}_{ik}\bar{\gamma}_{lj}-\bar{\Gamma}^{l}_{ij}\bar{\gamma}_{lk}\right)=0, \tag{4.36}$$

$$T^{jk}\left(\partial_i\bar{\gamma}_{jk}\right)=2T^{jk}\left(\bar{\Gamma}^{l}_{ik}\bar{\gamma}_{lj}\right), \tag{4.37}$$

as well as

$$\Delta\Gamma^{i}_{jk}\equiv\bar{\Gamma}^{i}_{jk}-\hat{\Gamma}^{i}_{jk} \tag{4.38}$$

$$=\frac{1}{2}\bar{\gamma}^{il}\left(\hat{\mathcal{D}}_j\bar{\gamma}_{lk}+\hat{\mathcal{D}}_k\bar{\gamma}_{lj}-\hat{\mathcal{D}}_l\bar{\gamma}_{jk}\right). \tag{4.39}$$

Combining all terms, we finally have

$$T_a{}^{b(4)}\Gamma^a_{bi} - T_k{}^j\hat{\Gamma}^k_{ji} = \frac{1}{2}T^{00}\left(-2\alpha\hat{\mathcal{D}}_i\alpha + 2\beta_k\hat{\mathcal{D}}_i\beta^k + \beta^j\beta^k\hat{\mathcal{D}}_i\gamma_{jk}\right) \tag{4.40}$$

$$+ T^0_k\hat{\mathcal{D}}_i\beta^k - T^{00}\beta_k\hat{\mathcal{D}}_i\beta^k + T^{0j}\beta^k\hat{\mathcal{D}}_i\gamma_{jk} \tag{4.41}$$

$$+ \frac{1}{2}T^{jk}\hat{\mathcal{D}}_i\gamma_{jk} \tag{4.42}$$

$$= \frac{1}{2}T^{00}\left(-2\alpha\hat{\mathcal{D}}_i\alpha + \beta^j\beta^k\hat{\mathcal{D}}_i\gamma_{jk}\right) + T^0_k\hat{\mathcal{D}}_i\beta^k + T^{0j}\beta^k\hat{\mathcal{D}}_i\gamma_{jk} + \frac{1}{2}T^{jk}\hat{\mathcal{D}}_i\gamma_{jk} \tag{4.43}$$

$$= -T^{00}\alpha\hat{\mathcal{D}}_i\alpha + T^0_k\hat{\mathcal{D}}_i\beta^k + \frac{1}{2}\hat{\mathcal{D}}_i\gamma_{jk}\left(T^{00}\beta^j\beta^k + 2T^{0j}\beta^k + T^{jk}\right). \tag{4.44}$$

Using this result, we now have

$$\partial_t\left(e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\mathcal{S}_i\right) = -\hat{\mathcal{D}}_j\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}T^j_i\right) + \alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\left(-T^{00}\alpha\hat{\mathcal{D}}_i\alpha + T^0_k\hat{\mathcal{D}}_i\beta^k\right. \tag{4.45}$$

$$\left. + \frac{1}{2}\hat{\mathcal{D}}_i\gamma_{jk}\left(T^{00}\beta^j\beta^k + 2T^{0j}\beta^k + T^{jk}\right)\right). \tag{4.46}$$

Defining the densitized momentum as $\tilde{\mathcal{S}}_i \equiv e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\mathcal{S}_i$, the flux term as $(f_{\mathcal{S}})^j_i \equiv \alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}T^j_i$, and the source term as

$$(s_{\mathcal{S}})_i \equiv \alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\left(-T^{00}\alpha\hat{\mathcal{D}}_i\alpha + T^0_k\hat{\mathcal{D}}_i\beta^k + \frac{1}{2}\hat{\mathcal{D}}_i\gamma_{jk}\left(T^{00}\beta^j\beta^k + 2T^{0j}\beta^k + T^{jk}\right)\right), \tag{4.47}$$

we now have in compact form

$$\partial_t\tilde{\mathcal{S}}_i = -\hat{\mathcal{D}}_j\left(f_{\mathcal{S}}\right)^j_i + \left(s_{\mathcal{S}}\right)_i. \tag{4.48}$$

To derive the energy equation, we begin by again using the relation defined in Eq. (4.4). However, here we apply it to the equation

$$n_a\nabla_b T^{ab} - \nabla_a\left(\rho_0 u^a\right) = 0, \tag{4.49}$$

where $n_a = (-\alpha, 0, 0, 0)$ is the vector normal to each hypersurface. We begin with this equation because it is equivalent to what is done in the Cartesian Valencia formulation (see

Ref. [45] for an excellent review), and results in an equation that is well suited for numerical integration. We can rewrite the previous equation as

$$\nabla_b \left(n_a T^{ab} - \rho_0 u^b \right) = T^{ab} \nabla_b n_a. \tag{4.50}$$

For convenience we define the variable $\eta^b \equiv n_a T^{ab} - \rho_0 u^b$, giving

$$\nabla_b \left(\eta^b \right) = T^{ab} \nabla_b n_a. \tag{4.51}$$

Continuing in exactly the same fashion as before with the continuity equation, we arrive at

$$\partial_t \left(\alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \eta^t \right) + \hat{\mathcal{D}}_i \left(\alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \eta^i \right) = \left(\alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \right) T^{ab} \nabla_b n_a. \tag{4.52}$$

Considering the time-component first,

$$\eta^t = n_a T^{at} - \rho_0 u^t = -\alpha \left(\rho_0 h u^t u^t - \frac{P}{\alpha^2} \right) - \rho_0 u^t, \tag{4.53}$$

$$\partial_t \left(\alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \eta^t \right) = \partial_t \left(\alpha e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \left(-\alpha \left(\rho_0 h u^t u^t - \frac{P}{\alpha^2} \right) - \rho_0 u^t \right) \right), \tag{4.54}$$

$$= \partial_t \left(e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \left(-\alpha^2 \rho_0 h \left(u^t \right)^2 + P - \alpha \rho_0 u^t \right) \right), \tag{4.55}$$

$$= \partial_t \left(e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \left(-W^2 \rho_0 h + P \right) - D \right), \tag{4.56}$$

$$= \partial_t \left(e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \left(-W^2 \rho_0 h + P \right) + D \right) - \partial_t \left(2D \right), \tag{4.57}$$

$$= -\partial_t \left(\tilde{\tau} \right) - \partial_t \left(2D \right), \tag{4.58}$$

where $\tilde{\tau} = e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} \left(W^2 \rho_0 h - P \right) - D = \alpha^2 e^{6\phi} \sqrt{\bar{\gamma}/\hat{\gamma}} T^{tt} - D$ is the densitized energy. Now,

expanding out the spatial terms, substituting $\partial_t\left(2D\right)$, and multiplying all terms by -1,

$$\implies -\hat{\mathcal{D}}_i\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\eta^i\right)+\partial_t\left(2D\right), \tag{4.59}$$

$$=-\hat{\mathcal{D}}_i\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\left(-\alpha\left(\rho_0 h u^t u^i+\frac{P\beta^i}{-\alpha^2}\right)-\rho_0 u^i\right)\right)-2\hat{\mathcal{D}}_i\left(f_D\right)^i, \tag{4.60}$$

$$=-\hat{\mathcal{D}}_i\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\left(-\alpha\left(\rho_0 h u^t u^i+\frac{P\beta^i}{-\alpha^2}\right)-\rho_0 u^i\right)\right)-2\hat{\mathcal{D}}_i\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\rho_0 u^i\right), \tag{4.61}$$

$$=\hat{\mathcal{D}}_i\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\left(\alpha\rho_0 h u^t u^i+\frac{P\beta^i}{\alpha}-\rho_0 u^i\right)\right), \tag{4.62}$$

$$=\hat{\mathcal{D}}_i\left(f_\tau\right)^i, \tag{4.63}$$

$$\left(f_\tau\right)^i\equiv\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\left(\alpha\rho_0 h u^t u^i+\frac{P\beta^i}{\alpha}-\rho_0 u^i\right), \tag{4.64}$$

$$=\alpha\left(\tilde{\tau}\left(v^i-\frac{\beta^i}{\alpha}\right)+e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}Pv^i\right). \tag{4.65}$$

We then have

$$\partial_t\left(\tilde{\tau}\right)+\hat{\mathcal{D}}_i\left(f_\tau\right)^i=-\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\right)T^{ab}\nabla_b n_a. \tag{4.66}$$

All that remains is to expand out the source terms. Using Eqs. 2 and 7 from Ref. [43],

$$\gamma_{ab}=g_{ab}+n_a n_b, \tag{4.67}$$

$$K_{ij}=-\gamma_i^k\gamma_j^l\nabla_k n_l, \tag{4.68}$$

where K_{ij} is the extrinsic curvature, we write

$$T^{ab}\nabla_b n_a=T^{ab}g_a^c\ g_b^d\nabla_c n_d, \tag{4.69}$$

$$=\left(\gamma_a^c-n_a n^c\right)\left(\gamma_b^d-n_b n^d\right)\nabla_c n_d, \tag{4.70}$$

$$=T^{ab}\nabla_c n_d\left(\gamma_a^c\ \gamma_b^d+n_a n^c n_b n^d-\gamma_a^c\ n_b n^d-\gamma_b^d\ n_a n^c\right). \tag{4.71}$$

Note that since $n^d \nabla_c n_d = 0$, we therefore have

$$\implies T^{ab}\nabla_c n_d \left(\gamma_a^c\ \gamma_b^d + n_a n^c n_b n^d - \gamma_a^c\ n_b n^d - \gamma_b^d\ n_a n^c\right), \tag{4.72}$$

$$= T^{ab}\left(-K_{ab} - \gamma_b^d\ n_a n^c \nabla_c n_d\right), \tag{4.73}$$

$$= T^{ab}\left(-K_{ab} - \gamma_b^d\ n_a \left(n^c \nabla_c n_d\right)\right), \tag{4.74}$$

$$= T^{ab}\left(-K_{ab} - \gamma_b^d\ n_a \left(\gamma_d^c \partial_c \ln\alpha\right)\right), \tag{4.75}$$

$$= T^{ab}\left(-K_{ab} - n_a \partial_c \ln\alpha \left(\gamma_d^c\ \gamma_b^d\right)\right), \tag{4.76}$$

$$= T^{ab}\left(-K_{ab} - \gamma_b^c\ n_a \partial_c \ln\alpha\right), \tag{4.77}$$

where we used Eq. 44 in Ref. [43]:

$$n^c \nabla_c n_d = \gamma_d^c \partial_c \ln\alpha. \tag{4.78}$$

Focusing on the second term of the last equation, and using the definition $\gamma_b^c = \delta_b^c + n^c n_b$,

$$T^{ab}\gamma_b^c\ n_a \partial_c \ln\alpha = T^{ab} n_a \partial_c \ln\alpha \left(\delta_b^c + n^c n_b\right), \tag{4.79}$$

$$= T^{ac} n_a \partial_c \ln\alpha + T^{ab} n^c n_b n_a \partial_c \ln\alpha, \tag{4.80}$$

$$= -T^{tc}\partial_c\alpha + \alpha T^{tt} n^c \partial_c \alpha, \tag{4.81}$$

$$= -T^{ti}\partial_i\alpha - T^{tt}\beta^i \partial_i \alpha, \tag{4.82}$$

where the $c = 0$ terms cancel and $n^a = -\frac{1}{\alpha}(-1, \beta^i) = -\frac{1}{\alpha}(-1, \beta^1, \beta^2, \beta^3)$. Now focusing on the first term,

$$T^{ab}K_{ab} = T^{ab} g_{ac} g_{bd} K^{cd}, \tag{4.83}$$

$$= T^{00}\beta_i\beta_j K^{ij} + 2T^{0i}\beta_j \gamma_{ik} K^{jk} + T^{jk} K_{jk}. \tag{4.84}$$

Collecting all source terms, we now define

$$s_\tau \equiv \alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\left(T^{00}\left(\beta_i\beta_j K^{ij} - \beta^i\partial_i\alpha\right) + T^{0i}\left(2\beta_j\gamma_{ik}K^{jk} - \partial_i\alpha\right) + T^{ij}K_{ij}\right), \tag{4.85}$$

and we finally have

$$\partial_t \left(\tilde{\tau}\right) + \hat{\mathcal{D}}_i \left(f_\tau\right)^i = s_\tau. \tag{4.86}$$

In the preceding derivations, we wrote the equations in terms of the fluid velocity measured by a normal observer, but since our code wishes to build upon the Cartesian implementation of the IllinoisGRMHD code, here we rewrite them using the velocity used by IllinoisGRMHD, namely $v^i = u^i/u^0$. Beginning with the continuity equation

$$\partial_t \left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\rho_0 u^t\right) + \hat{\mathcal{D}}_i \left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}}\rho_0 u^i\right) = 0, \tag{4.87}$$

we define the flux as $\left(f_D\right)^i = Dv^i$, and we again write

$$\partial_t D + \hat{\mathcal{D}}_i \left(f_D\right)^i = 0. \tag{4.88}$$

The momentum equation remains unchanged;

$$\partial_t \tilde{\mathcal{S}}_i = -\hat{\mathcal{D}}_j \left(f_{\mathcal{S}}\right)^j_i + \left(s_{\mathcal{S}}\right)_i. \tag{4.89}$$

Focusing on the flux term of the energy equation, from our previous derivation and recalling that $T^{\mu\nu} = \rho_0 h u^\mu u^\nu + P g^{\mu\nu}$, we have

$$\implies \hat{\mathcal{D}}_i \left(\alpha e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}} \left(\alpha\rho_0 h u^t u^i + \frac{P\beta^i}{\alpha} - \rho_0 u^i\right)\right), \tag{4.90}$$

$$= \hat{\mathcal{D}}_i \left(e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}} \left(\alpha^2\rho_0 h u^t u^i + P\beta^i - \alpha\rho_0 u^i\right)\right), \tag{4.91}$$

$$= \hat{\mathcal{D}}_i \left(e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}} \left(\alpha^2 T^{ti}\right) - Dv^i\right), \tag{4.92}$$

$$= \hat{\mathcal{D}}_i \left(f_\tau\right)^i, \tag{4.93}$$

$$\left(f_\tau\right)^i \equiv e^{6\phi}\sqrt{\bar{\gamma}/\hat{\gamma}} \left(\alpha^2 T^{ti}\right) - Dv^i. \tag{4.94}$$

We again write

$$\partial_t \left(\tau\right) + \hat{\mathcal{D}}_i \left(f_\tau\right)^i = s_\tau. \tag{4.95}$$

With these equations we can now model relativistic fluid flows in curvilinear coordinates,

where we apply spherical- or axi-symmetry to enable faster and more efficient simulations for matter configurations containing such symmetries.

Chapter 5

General Relativistic Hydrodynamics Code for Dynamical Spacetimes with Curvilinear Coordinates, Tabulated Equations of State, and Neutrino Physics

This Chapter reproduces the paper "General relativistic hydrodynamics code for dynamical spacetimes with curvilinear coordinates, tabulated equations of state, and neutrino physics", published in Physical Review D [44].

Abstract

Many astrophysical systems of interest to numerical relativity—such as rapidly rotating stars, black hole accretion disks, and core-collapse supernovae—exhibit near-symmetries. These systems generally consist of a strongly gravitating central object surrounded by an accretion disk, debris, and ejecta. Simulations can efficiently exploit the near-axisymmetry of these systems by reducing the number of points in the angular direction around the near-symmetry axis, enabling efficient simulations over seconds-long timescales with minimal computational expense. In this chapter, we introduce GRoovy, a novel code capable of modeling astrophysical systems containing compact objects by solving the equations of general relativistic hydrodynamics (GRHD) in full general relativity using singular curvilinear (spherical-like and cylindrical-like) and Cartesian coordinates. We demonstrate the code's robustness through a battery of challenging GRHD tests, ranging from flat, static spacetimes to curved, dynamical

Contributing authors: Samuel Cupp, Leonardo R. Werneck, Samuel D. Tootle, Maria C. Babiuc Hamilton, Zachariah B. Etienne

spacetimes. These tests further showcase the code's capabilities in modeling systems with realistic, finite-temperature equations of state and neutrino cooling via a leakage scheme. GRoovy extensively leverages GRHayL, an open-source, modular, and infrastructure-agnostic general relativistic magnetohydrodynamics library built from the highly robust algorithms of IllinoisGRMHD. Long-term simulations of binary neutron star and black hole–neutron star post-merger remnants will benefit greatly from using a future `Charm++`-parallelized version of GRoovy to study phenomena such as remnant stability, gamma-ray bursts, and nucleosynthesis.

5.1 Introduction

The gravitational wave (GW) event GW170817 [2, 46] marked the first simultaneous direct detection of GWs and electromagnetic (EM) radiation from a binary neutron star (BNS) merger [47], launching a new era of multi-messenger astrophysics. This event, observed across X-ray, radio, and optical bands [48–52], provided an unparalleled opportunity to test and refine theoretical models for the EM counterparts to BNS mergers [5, 53–59]. Future observations are expected to further constrain the neutron star equation of state [60, 61], refine models of short gamma-ray burst (sGRB) emission, and deepen our understanding of neutrino physics and r-process nucleosynthesis [62–64].

While GW170817 remains the sole GW+EM event, the frequency of BNS and black hole-neutron star (BHNS) mergers, as inferred from current event rates observed by the LIGO-VIRGO-KAGRA collaboration [65–67], underscores the need for compact object merger simulation catalogs. Given the non-linear nature of Einstein's field equations coupled to the equations of general relativistic magnetohydrodynamics (GRMHD), these catalogs must be built using self-consistent numerical relativity models that span both observational and theoretical uncertainties.

After the inspiral and merger phases, BNS and BHNS systems become nearly axisymmetric, with Keplerian fluid flows largely following grid lines in spherical or cylindrical coordinates, centered on the compact remnant at $r = 0$. When these flows are modeled on Cartesian numerical grids, they cross grid lines obliquely, resulting in spurious angular momentum loss and hindering the accuracy of post-merger simulations at low-to-moderate

resolutions. Spherical or cylindrical coordinate systems enable more reliable post-merger simulations over timescales of seconds at lower resolutions. To date, several codes have been developed that adopt spherical coordinates [21–23, 36, 38, 39, 68–74] for more accurate modeling of GRMHD and GRHD flows near compact objects.

Recently, "hand-off" scenarios have become popular, in which the inspiral and merger are modeled using codes with more general-purpose grids and Cartesian adaptive mesh refinement. Once the post-merger spacetime becomes largely stationary, the metric and GRMHD data are interpolated into a code that adopts spherical coordinates [24, 25, 75]. These specialized codes typically focus on static-spacetime black hole accretion disk simulations. Further work is still needed, however, as future studies will need to self-consistently model finite temperature effects and neutrino cooling of post-merger remnants across a vast parameter space of BNS and BHNS systems. Incorporating these processes will be essential for studying magnetized winds from accretion disks [76–78] and understanding polar fluid flows that may evolve into ultra-relativistic jets over secular timescales.

Compact object mergers are not the only strongly gravitating astrophysical systems that are largely spherically or axially symmetric. Core-collapse supernova studies (e.g., [79–85]) have been conducted using spherical coordinates, either in 1D, 2D, or full 3D. Similarly, studies of direct collapse black holes [86–89], which couple hydrodynamics with a rapidly evolving gravitational field, may also benefit from using curvilinear coordinates. We also acknowledge the vast amount of research in modeling black hole accretion disks (e.g., [39, 90–93]) that often adopt spherical geometries to model accretion flows and jets.

To accurately and efficiently model this wide range of strongly gravitating and highly relativistic astrophysical systems that exhibit some degree of symmetry, we have developed GRoovy, a new code within the BlackHoles@Home/NRPy framework [29, 94]. GRoovy implements the equations of general relativistic hydrodynamics (GRHD) in full general relativity and supports spherical, cylindrical, and Cartesian coordinate systems.

Our code leverages the newly developed, open-source General Relativistic Hydrodynamics Library (GRHayL) [95, 96], a highly modular redesign of the IllinoisGRMHD code [26, 27, 97]. Through its straightforward, infrastructure-agnostic interface, GRHayL provides all the GRMHD algorithms of IllinoisGRMHD, with each algorithm validated to roundoff error

against the established IllinoisGRMHD code. This modularity allows users to easily integrate only the algorithms needed for specific applications.

GRHayL's flexibility has allowed us to integrate GRoovy's curvilinear-coordinate GRHD implementation with IllinoisGRMHD's well-tested, realistic finite-temperature equations of state (EOS) and neutrino-leakage algorithms [27], enabling a wide range of physics within the GRHD regime. This integration facilitates modeling of neutrino-irradiated disk winds [98–100] and outflows [101, 102], bulk transport [103, 104], thermal transport [105–108], and the post-merger lifetime and stability of remnant neutron stars [109–113]. The future goal of the `Charm++`-based version of GRoovy is to efficiently simulate the evolution of these systems over time, independent of magnetic field evolution.

Here, we showcase the capabilities of GRoovy by modeling various hydrodynamical flows in different coordinate systems. Building upon the reference metric formulation of the GRHD equations presented in [36, 43, 69, 114], we incorporate the conservation of lepton number and source terms to model finite temperature effects and neutrino emission via a leakage scheme. We then numerically integrate these equations forward in time using a finite volume method, conducting a series of standard yet challenging numerical tests. These tests include two- and three-dimensional shock tests in spherical coordinates, neutrino leakage tests, and demonstrations of GRoovy's ability to accurately model both non-rotating and rotating isolated neutron stars. Our results indicate that GRoovy is highly robust in modeling flows within flat spacetimes, strongly curved static spacetimes, and fully dynamical spacetimes.

The remainder of this chapter is organized as follows. Section 5.2 outlines the GRHD equations our codes approximately solves. Section 5.3 details our numerical techniques and algorithms. In Section 5.4, we describe our setups and present the results of various code tests. We conclude in Section 5.5.

5.2 Basic Equations

In this Section, we present the core equations used to model astrophysical systems with our new code. We use geometrized units throughout this chapter, where $G = c = M_\odot = 1$,

with G as the gravitational constant and c as the speed of light. Latin indices indicate spatial components, while Greek indices represent spacetime components. We express Einstein's equations in $3+1$ form, starting with the standard line element [115]

$$ds^2 = \left(-\alpha^2 + \beta^i \beta_i\right) dt^2 + 2\beta_i \, dt \, dx^i + \gamma_{ij} \, dx^i \, dx^j \ , \tag{5.1}$$

where α is the lapse function, β^j is the shift vector, and $\gamma_{ij} = g_{ij}$ is the spatial part of the spacetime metric $g_{\mu\nu}$. We adopt the following conformal decomposition of the spatial metric [11]:

$$\gamma_{ij} = e^{4\phi} \bar{\gamma}_{ij}, \tag{5.2}$$

where $\bar{\gamma}_{ij}$ is the conformal metric, $\psi \equiv e^{\phi}$ is the conformal factor, and ϕ is the conformal exponent. To evolve the spacetime in curvilinear coordinates, we further decompose the conformal metric as

$$\bar{\gamma}_{ij} = \hat{\gamma}_{ij} + \epsilon_{ij}, \tag{5.3}$$

where $\hat{\gamma}_{ij}$ is the flat space reference metric in the chosen coordinate system, and ϵ_{ij} represents deviations from flat space that are not necessarily small. For example, in spherical coordinates, the flat reference metric is given by $\hat{\gamma}_{ij} = \mathrm{diag}\left(1, r^2, r^2 \sin^2\theta\right)$. To simplify future expressions, we also define the quantity

$$\xi = e^{6\phi} \sqrt{\frac{\bar{\gamma}}{\hat{\gamma}}}, \tag{5.4}$$

where $\bar{\gamma}$ is the determinant of the conformal metric, and $\hat{\gamma}$ is the determinant of the reference metric. Following the Lagrangian BSSN formulation of Ref. [116], we set $\bar{\gamma}$ to remain constant in time. For numerical convenience, we choose this constant to be $\hat{\gamma}$, and enforce $\bar{\gamma} = \hat{\gamma}$ at each time step. As described in Section IV. D of Ref. [29], we ensure this condition by applying the algebraic correction

$$\bar{\gamma}_{ij} \to \left(\frac{\hat{\gamma}}{\bar{\gamma}}\right)^{1/3} \bar{\gamma}_{ij}, \tag{5.5}$$

to the conformal metric components at the end of each Runge-Kutta (RK) substep in our numerical evolution. In the following equations we retain the factor ξ to maintain consistency with previous literature.

GRoovy solves for the spacetime using the BSSN formulation [116–118] of Einstein's equations with a reference metric prescription [29, 70, 119] implemented within the NRPy code-generation framework. The NRPy auto-generated BSSN reference-metric code has been validated in spherical coordinates against the independent code of Ref. [119] (see Ref. [29]) and in Cartesian coordinates against `ML_BSSN` in the Einstein Toolkit using the Baikal thorn.

To couple the fluid and spacetime dynamics, we must also solve the GRHD equations:

$$\nabla_\mu \left(n_{\rm b} u^\mu\right) = 0 \,, \tag{5.6}$$

$$\nabla_\mu \left(n_{\rm e} u^\mu\right) = \mathcal{R}/m_{\rm b} \,, \tag{5.7}$$

$$\nabla_\mu T^{\mu\nu} = \mathcal{Q} u^\nu \,, \tag{5.8}$$

which correspond to conservation laws for the baryon number $n_{\rm b}$, lepton number $n_{\rm e}$, and the GRHD stress-energy tensor

$$T^{\mu\nu} = \rho h u^\mu u^\nu + P g^{\mu\nu}, \tag{5.9}$$

respectively. Here, $\rho = m_{\rm b} n_{\rm b}$ is the fluid rest mass density, $m_{\rm b}$ is the baryon mass, $h = 1 + \epsilon + P/\rho$ is the enthalpy, ϵ is the specific internal energy, u^μ is the fluid four-velocity, and P is the pressure. In addition, $\mathcal{R}$ and $\mathcal{Q}$ represent the source term contributions associated with neutrino emission via a leakage scheme (see e.g., Refs. [27, 120]).

Following the detailed derivations of Refs. [36, 43], we recast the conservation laws in a form appropriate for numerical integration. To bring the equations closer to the formulation presented in Ref. [43] for the case of pure GRHD, we rewrite the above equations as

$$\nabla_\mu \left(\rho u^\mu\right) = 0 \,, \tag{5.10}$$

$$\nabla_\mu \left(\rho Y_{\rm e} u^\mu\right) = \mathcal{R} \,, \tag{5.11}$$

where we have introduced the electron fraction $Y_{\rm e} = n_{\rm e}/n_{\rm b}$. In the following subsections, we show how to rewrite these conservation laws using a reference metric formulation.

5.2.1 Conservation of Lepton and Baryon Number

Following Ref. [43] and using the identity

$$\nabla_\mu V^\mu = \frac{1}{\sqrt{|g|}}\partial_\mu \left(\sqrt{|g|}V^\mu\right) \ , \tag{5.12}$$

we rewrite Eq. (5.11) as

$$\frac{1}{\sqrt{-g}}\left[\partial_t\left(\sqrt{-g}\rho Y_{\rm e}u^0\right) + \partial_i\left(\sqrt{-g}\rho Y_{\rm e}u^i\right)\right] = \mathcal{R}. \tag{5.13}$$

Now, using $\sqrt{-g} = \alpha\sqrt{\gamma} = \alpha e^{6\phi}\sqrt{\bar{\gamma}}$, we rewrite the second term in the square brackets of Eq. (5.13) as

$$\partial_i\left(\alpha e^{6\phi}\sqrt{\bar{\gamma}}\rho Y_{\rm e}u^i\right) = \partial_i\left(\sqrt{\hat{\gamma}}\alpha\xi\rho Y_{\rm e}u^i\right). \tag{5.14}$$

Note that since Eq. (5.12) is valid for any metric, we rewrite this last equation in terms of the covariant derivative associated with the reference metric $\hat{\gamma}_{ij}$, i.e.,

$$\partial_i\left(\sqrt{\hat{\gamma}}\alpha\xi\rho Y_{\rm e}u^i\right) = \sqrt{\hat{\gamma}}\hat{\nabla}_i\left(\alpha\xi\rho Y_{\rm e}u^i\right). \tag{5.15}$$

Replacing the spatial terms in Eq. (5.13) with this result and dividing by $\sqrt{\hat{\gamma}}$ gives the equation

$$\partial_t\left(\alpha\xi\rho Y_{\rm e}u^0\right) + \hat{\nabla}_i\left(\alpha\xi\rho Y_{\rm e}u^i\right) = \alpha\xi\mathcal{R}\ . \tag{5.16}$$

We define the conserved baryonic density as $D \equiv W\rho$, where $W = \alpha u^0$ is the Lorentz factor between observers in the fluid and normal frames, the conserved electron fraction as $DY_{\rm e}$, and the flux as

$$\left(f_{\tilde{Y}_{\rm e}}\right)^i \equiv \alpha\xi\rho Y_{\rm e}u^i = \xi D Y_{\rm e}v^i = \tilde{Y}_{\rm e}v^i. \tag{5.17}$$

Here, $v^i = u^i/u^0$ is the fluid three-velocity and $\tilde{Y}_{\rm e} \equiv \xi D Y_{\rm e}$ is the "densitized" conserved electron fraction. We thus obtain the evolution equation

$$\partial_t \tilde{Y}_{\rm e} + \hat{\nabla}_i \left(f_{\tilde{Y}_{\rm e}} \right)^i = \alpha \xi \mathcal{R}. \tag{5.18}$$

Note that Eqs. (5.10) and (5.11) have similar mathematical forms, with the main difference being that Eq. (5.10) does not have a source term. Introducing the "densitized" conserved density $\tilde{D} \equiv \xi D$, it immediately follows that

$$\begin{aligned} \partial_t \tilde{D} + \hat{\nabla}_i \left(f_{\tilde{D}} \right)^i &= 0 & (5.19) \\ \implies \partial_t \tilde{D} + \partial_i \left(\tilde{D} v^i \right) &= -\hat{\Gamma}^i_{ij} \tilde{D} v^j, & (5.20) \end{aligned}$$

where $\hat{\Gamma}^i_{ij}$ is the connection associated with the reference metric, and

$$\left(f_{\tilde{D}} \right)^i \equiv \xi D v^i = \tilde{D} v^i. \tag{5.21}$$

In Cartesian coordinates, connections vanish and we recover the Valencia formalism, where mass is conserved to machine precision. In curvilinear coordinates, source terms appear and lead to mass being conserved to truncation error. The same conclusion holds for the electron fraction evolution equation (Eq. 5.18). These truncation errors converge away in the continuum limit of increased resolution. Thus, as long as our numerical scheme is convergent at the expected order, which we demonstrate in Section 5.4, this consequence is not expected to affect our numerical solutions at sufficient resolution.

5.2.2 Conservation of Momentum

To derive the Euler equation from Eq. (5.8), we begin with the identity

$$\nabla_\nu A^\nu{}_\mu = \frac{1}{\sqrt{-g}} \partial_\nu \left(\sqrt{-g} A^\nu{}_\mu \right) - {}^{(4)}\Gamma^\rho_{\nu\mu} A^\nu{}_\rho, \tag{5.22}$$

valid for some tensorial quantity $A^{\nu}{}_{\mu}$, where ${}^{(4)}\Gamma^{\rho}_{\nu\mu}$ is the metric connection associated with $g_{\mu\nu}$. We then take a spatial projection of Eq. (5.8), i.e.,

$$\begin{aligned} g_{i\nu}\mathcal{Q}u^{\nu} &= g_{i\nu}\nabla_{\mu}T^{\mu\nu} = \nabla_{\mu}\big(g_{i\nu}T^{\mu\nu}\big) && (5.23)\\ &= \frac{1}{\sqrt{-g}}\partial_{\mu}\big(\sqrt{-g}T^{\mu}{}_{i}\big) - {}^{(4)}\Gamma^{\mu}_{i\nu}T^{\nu}{}_{\mu} && (5.24)\\ &= \frac{1}{\sqrt{-g}}\left[\partial_{t}\big(\sqrt{-g}T^{0}{}_{i}\big) + \partial_{j}\big(\sqrt{-g}T^{j}{}_{i}\big)\right] - {}^{(4)}\Gamma^{\mu}_{i\nu}T^{\nu}{}_{\mu}. \end{aligned}$$

Similar to the approach in Section 5.2.1, we expand terms with spatial derivatives as

$$\partial_{j}\Big(\sqrt{-g}T^{j}{}_{i}\Big) = \partial_{j}\Big(\sqrt{\hat{\gamma}}\alpha\xi T^{j}{}_{i}\Big) = \sqrt{\hat{\gamma}}\hat{\nabla}_{j}\Big(\alpha\xi T^{j}{}_{i}\Big) + \alpha e^{6\phi}\sqrt{\bar{\gamma}}\hat{\Gamma}^{k}_{ij}T^{j}{}_{k}. \tag{5.25}$$

Now, using this expansion in the previous equation and replacing $\sqrt{-g}$, then dividing by $\sqrt{\hat{\gamma}}$, we obtain

$$\partial_{t}\Big(\alpha\xi T^{0}{}_{i}\Big) + \hat{\nabla}_{j}\Big(\alpha\xi T^{j}{}_{i}\Big) = -\alpha\xi\Big(\hat{\Gamma}^{k}_{ij}T^{j}{}_{k} - {}^{(4)}\Gamma^{\mu}_{i\nu}T^{\nu}{}_{\mu}\Big) + \alpha\xi\mathcal{Q}u_{i}.$$

Since the expansion of the source terms has been done in Ref. [43], we refer the reader there for more details. Defining the densitized momentum as $\tilde{S}_{i} \equiv \alpha\xi T^{0}{}_{i}$, the flux term as $(f_{\tilde{S}})^{j}_{i} \equiv \alpha\xi T^{j}{}_{i}$, and the source term as

$$L_{i} \equiv -T^{00}\alpha\partial_{i}\alpha + T^{0}{}_{k}\hat{\nabla}_{i}\beta^{k} + \frac{1}{2}\hat{\nabla}_{i}\gamma_{jk}\left(T^{00}\beta^{j}\beta^{k} + 2T^{0j}\beta^{k} + T^{jk}\right), \tag{5.26}$$

we write the equations in compact form

$$\partial_{t}\tilde{S}_{i} + \hat{\nabla}_{j}\left(f_{\tilde{S}}\right)^{j}_{i} = \alpha\xi\left(L_{i} + \mathcal{Q}u_{i}\right). \tag{5.27}$$

5.2.3 Conservation of Energy

To derive the energy equation, we begin by using Eq. (5.12) again, applying it to the projection of Eq. (5.8)

$$n_{\mu}\nabla_{\nu}T^{\mu\nu} - \nabla_{\mu}\left(\rho u^{\mu}\right) = n_{\mu}\mathcal{Q}u^{\mu} \tag{5.28}$$

where $n_\mu = (-\alpha, 0, 0, 0)$ is the vector normal to each hypersurface. We rewrite this as

$$\nabla_\nu \left(n_\mu T^{\mu\nu} - \rho u^\nu \right) = T^{\mu\nu} \nabla_\nu n_\mu - \alpha \mathcal{Q} u^0. \tag{5.29}$$

For convenience, we define the variable $\eta^\nu \equiv n_\mu T^{\mu\nu} - \rho u^\nu$, giving

$$\nabla_\nu \eta^\nu = T^{\mu\nu} \nabla_\nu n_\mu - \alpha \mathcal{Q} u^0. \tag{5.30}$$

Continuing in the same fashion as Section 5.2.1, we arrive at

$$\partial_t \left(\alpha \xi \eta^0 \right) + \hat{\nabla}_i \left(\alpha \xi \eta^i \right) = \alpha \xi \left(T^{\mu\nu} \nabla_\nu n_\mu - \alpha \mathcal{Q} u^0 \right). \tag{5.31}$$

After further calculations, we obtain

$$\partial_t \tilde{\tau} + \hat{\nabla}_i \left(f_{\tilde{\tau}} \right)^i = \alpha \xi \left(\alpha \mathcal{Q} u^0 - T^{\mu\nu} \nabla_\nu n_\mu \right), \tag{5.32}$$

with

$$\tilde{\tau} = \alpha^2 \xi T^{00} - \tilde{D}, \tag{5.33}$$

$$\left(f_{\tilde{\tau}} \right)^i \equiv \xi \left(\alpha^2 T^{0i} \right) - \tilde{D} v^i, \tag{5.34}$$

where $\tilde{\tau}$ is the conserved energy. We then expand the source terms, directing the reader again to Ref. [43] for a detailed derivation. After additional work, we arrive at the definition of the source term

$$\begin{aligned} s_{\tilde{\tau}} \equiv{}& T^{00} \left(\beta_i \beta_j K^{ij} - \beta^i \partial_i \alpha \right) \\ &+ T^{0i} \left(2 \beta_j \gamma_{ik} K^{jk} - \partial_i \alpha \right) + T^{ij} K_{ij}, \end{aligned} \tag{5.35}$$

where K^{ij} is the extrinsic curvature. Collecting all terms, we finally write

$$\partial_t \tilde{\tau} + \hat{\nabla}_i \left(f_{\tilde{\tau}} \right)^i = \alpha \xi \left(s_{\tilde{\tau}} + \alpha \mathcal{Q} u^0 \right). \tag{5.36}$$

5.2.4 Final Equations

To conclude this Section we rewrite our evolution equations in a modified version of the standard flux-conservative form, and set $\bar{\gamma} = \hat{\gamma}$. Our equations then take the form

$$\partial_t \boldsymbol{C} + \hat{\nabla}_i \boldsymbol{F}^i = \boldsymbol{S}, \tag{5.37}$$

and can be explicitly written as

$$\partial_t \begin{bmatrix} \tilde{D} \\ \tilde{Y}_{\rm e} \\ \tilde{\tau} \\ \tilde{S}_j \end{bmatrix} + \partial_i \begin{bmatrix} \left(f_{\tilde{D}}\right)^i \\ \left(f_{\tilde{Y}_{\rm e}}\right)^i \\ \left(f_{\tilde{\tau}}\right)^i \\ \left(f_{\tilde{S}}\right)^i_j \end{bmatrix} = \begin{bmatrix} -\hat{\Gamma}^i_{ij}\left(f_{\tilde{D}}\right)^j \\ \alpha e^{6\phi}\mathcal{R} - \hat{\Gamma}^i_{ij}\left(f_{\tilde{Y}_{\rm e}}\right)^j \\ \alpha e^{6\phi}\left(s_{\tilde{\tau}} + W\mathcal{Q}\right) - \hat{\Gamma}^i_{ij}\left(f_{\tilde{\tau}}\right)^j \\ \alpha e^{6\phi}\left(L_j + \mathcal{Q}u_j\right) - \hat{\Gamma}^i_{ik}\left(f_{\tilde{S}}\right)^k_j + \hat{\Gamma}^k_{ij}\left(f_{\tilde{S}}\right)^i_k \end{bmatrix}, \tag{5.38}$$

where the conservative variables $\boldsymbol{C}$ in Eq. (5.37) are related to the primitive variables via

$$\boldsymbol{C} = \begin{bmatrix} \tilde{D} \\ \tilde{Y}_{\rm e} \\ \tilde{\tau} \\ \tilde{S}_j \end{bmatrix} \equiv e^{6\phi} \begin{bmatrix} W\rho \\ W\rho Y_{\rm e} \\ \alpha^2 T^{00} - W\rho \\ \alpha T^0_{\ j} \end{bmatrix}. \tag{5.39}$$

The conserved variables, fluxes $\boldsymbol{F}$ and source terms $\boldsymbol{S}$ are defined in terms of the primitive quantities $\boldsymbol{P} \equiv [\rho, P, v^i, Y_{\rm e}, T]$, where T is the temperature. Note that our equations differ from those presented in Ref. [43] due to the inclusion of lepton number conservation and neutrino emission. To summarize, transitioning from the formulation presented in Ref. [27] to our approach using a reference metric for pure hydrodynamics involves:

1. Replacing partial derivatives with covariant derivatives with respect to a reference metric, and

2. Replacing factors of $\sqrt{\gamma}$ with $e^{6\phi}$.

5.3 Numerical Implementation

Our code employs a second-order accurate finite-volume scheme that approximates cell-averaged quantities using their cell-centered values. The BlackHoles@Home infrastructure [29] provides a uniform grid discretization for coordinates $x^i = (x^1, x^2, x^3)$ of the form

$$x^i\,(j) = \Delta x^i \left(j - N_G + \frac{1}{2} \right), \tag{5.40}$$

where j is the grid index, Δx^i is the grid spacing, and N_G is the number of ghost cells, used for finite differences, interpolations, and reconstruction. In our implementation both the BSSN and GRHD quantities are defined on this same grid.

To second-order accuracy, cell-averages may be approximated by the *volume centroid* in any given grid cell. While *geometric cell-centers* and *volume centroids* coincide in Cartesian coordinates, they differ in general curvilinear coordinates, thus cell-center values are generally not equivalent to their cell averages without corrections [31, 32]. However, Ref. [36] demonstrated that this care need not be taken when working within a reference-metric formulation of the GRHD equations, since their integral form is effectively "Cartesian". In short, because we reformulate the equations with respect to the background metric $\hat{\gamma}_{ij}$ and express them in an *orthonormal basis*, as shown in the following subsection, information about the underlying coordinate system is transferred to the source terms. In fact, if we take volume integrals over Eq. (5.38), in the *orthonormal basis* of our chosen coordinate system, $\hat{\gamma}_{ij}$ simplifies to the flat metric and the integrands do not acquire additional geometric scale factors. This is unlike Refs. [31, 32], where all scale factors appear in the integrands. We refer the reader to Ref. [36] for further discussion. In the following subsections we outline key details of our numerical scheme.

5.3.1 Rescaling Tensorial Quantities

Numerical integration of Eq. (5.38) in singular coordinate systems often leads to instability as tensor components can diverge or vanish at coordinate singularities. To mitigate this, we follow the approach of Ref. [36, 43], employing cell-centered grids and integrating

the equations in an *orthonormal basis*, which removes singular terms from tensors subject to evolution or finite differencing.

To implement this strategy, we define the rescaling quantities:

$$\mathcal{R}_i = \sqrt{\hat{\gamma}_{ii}}, \quad \mathcal{R}^i = \frac{1}{\mathcal{R}_i}, \quad \mathcal{R}^{ij} = \mathcal{R}^i \odot \mathcal{R}^j, \tag{5.41}$$

where $\odot$ denotes the Hadamard product, indicating element-wise multiplication (without summation over repeated indices).

As an example, when computing the finite difference of the flux term for the densitized momentum, we first rescale the flux tensor $F^j_{\ i} \equiv \alpha\xi T^j_{\ i}$ as

$$f^j_{\ i} = F^j_{\ i} \odot \mathcal{R}_j \odot \mathcal{R}^i. \tag{5.42}$$

This rescaling ensures that if the fluxes are smooth in Cartesian coordinates, the rescaled flux tensor $f^j_{\ i}$ remains smooth across coordinate singularities. The flux divergence $\partial_j F^j_{\ i}$ is then expressed in terms of $f^j_{\ i}$ as

$$\partial_j F^j_{\ i} = \partial_j \left(f^j_{\ i} \odot \mathcal{R}_i \odot \mathcal{R}^j \right). \tag{5.43}$$

In practice, we evaluate Eq. (5.43) using the product rule, taking finite differences of $f^j_{\ i}$ and computing derivatives of $\mathcal{R}^j$ analytically. Since the conserved momentum is a contravariant vector containing singular terms (first appearing from the transformation of initial data from Cartesian to curvilinear coordinates), we scale out these terms to avoid numerical issues. Thus, we evolve $\tilde{s}_i = \tilde{S}_i \odot \mathcal{R}^i$ in the Euler equation, and all GRHD equations are expressed in terms of the *rescaled* three-velocity. A similar rescaling procedure is applied to integrated, but not finite-differenced, quantities in the reference metric formulation of the BSSN equations, as discussed in Refs. [29, 119]. This ensures consistency across both finite-differenced and integrated terms, mitigating issues arising from coordinate singularities.

5.3.2 High-Resolution Shock Capturing Scheme

GRoovy is built on the NRPy/BlackHoles@Home framework [29, 94], which provides the core infrastructure for evolving the coupled matter and spacetime fields. The evolution equations are expressed symbolically, enabling NRPy to automatically generate optimized C code for all tensorial and finite-differenced equations. Written in covariant form with respect to a reference metric, these equations can be generated in any coordinate system at NRPy code-generation time with a single parameter choice.

While future applications will leverage BlackHoles@Home's multi-patch infrastructure, the current results solve the GRHD and Einstein's equations on a single patch. BlackHoles@Home provides various time-stepping algorithms and both extrapolation and radiation boundary conditions. For all tests in Section 5.4, we use RK4 time-stepping and apply Sommerfeld boundary conditions to the spacetime variables, while primitive variables are copied from the grid interior to the outer boundaries.

To handle complex boundary conditions in curvilinear coordinates—particularly near angular boundaries and the origin—we use BlackHoles@Home's native curvilinear boundary condition driver. This driver automatically addresses parity changes caused by basis vector sign flips across coordinate boundaries. Since BlackHoles@Home employs cell-centered grids, ghost cells at inner boundaries (e.g., $r < 0$ or $\theta > \pi$) directly overlap interior grid cells, requiring only mapping and accounting for basis vector direction changes. For further details, see Refs. [29, 36, 94, 119].

To close the evolution equations in Eq. (5.38), we incorporate an equation of state (EoS) via direct linkage with GRHayL, which performs point-wise EoS computations. GRHayL supports various EoS models, including tabulated and hybrid piecewise polytropic models, as well as a simple ideal gas model that does not distinguish between cold and thermal components.

After each sub-step of the time evolution scheme, we recover the primitive variables from the conservative variables, typically using a non-linear solver. GRHayL provides several widely used conservative-to-primitive solvers, including the two-dimensional Noble2D solver from Refs. [38] and the one-dimensional solvers from Refs. [40, 121, 122]. Backup options

are employed to maintain physical realism in cases where inversions fail, particularly in the low-density atmosphere, where truncation errors in the conservative variable evolution can push values outside physical bounds. Since these routines operate in a Cartesian basis, we transform the inputs to Cartesian coordinates before invoking the solver and convert the outputs back to the original coordinate system afterward.

Time evolution of the conservative variables is governed by Eq. (5.38), with flux terms computed using the Harten-Lax-van Leer (HLL) approximate Riemann solver [33]. As in most fluid-based codes with high-resolution shock-capturing schemes, we interpolate metric quantities and primitive variables to cell interfaces before calculating the HLL fluxes. For primitive variable reconstruction, we use GRHayL's routines, employing either the piece-wise linear method [123] or the piece-wise parabolic method (PPM) [124]. We apply reconstruction to the *rescaled* three-velocity to prevent the PPM shock detection algorithm from misinterpreting coordinate singularities as physical shocks. While modifications to PPM for curvilinear coordinates have been explored in past work [32, 125], their connection to our rescaling technique remains unclear. A close examination of the PPM algorithm reveals that the scheme considers gradients of the pressure, which may be the cause of the undesirable results noted by Ref. [36], as PPM reconstruction without modification yields incorrect values for the pressure gradient without additional geometric terms. We observe similar issues in our implementation, as discussed in Section 5.4.3.3.

Finally, we incorporate the `NRPyLeakage` [27] neutrino physics module from GRHayL, enabling our code to model neutrino emission via a leakage scheme. The specific reactions included in this scheme are detailed in Ref. [27]. At each time step, the module processes the primitive variables and outputs the numerical values of $\mathcal{R}$ and $\mathcal{Q}$, as defined in Eq. (5.7) and Eq. (5.8).

5.3.3 Algorithmic Differences between GRoovy and IllinoisGRMHD

IllinoisGRMHD is a GRMHD code within the Einstein Toolkit framework, designed for modeling magnetized fluids in dynamical spacetimes with neutrino leakage. Building on IllinoisGRMHD's core algorithms, GRoovy uses GRHayL to implement GRHD capabilities within the BlackHoles@Home infrastructure. While preserving most of IllinoisGRMHD's original im-

plementation choices, GRoovy introduces two key enhancements aimed at improving accuracy and physical fidelity.

First, unlike the original IllinoisGRMHD implementation, which interpolates BSSN quantities to *cell-interfaces* at third-order accuracy before applying second-order finite differencing to approximate source terms at *cell-centers*, GRoovy directly operates on cell-centered quantities. This approach eliminates interpolation errors in spacetime quantities used to compute the GRHD source terms, and enables the use of any finite-differencing order supported by NRPy, currently up to 12th order.

The advantages of this approach are detailed in Section 5.6.1, where we demonstrate that increasing the finite-differencing order for metric source terms from second to fourth order significantly reduces central density drift in equilibrium neutron star simulations. Beyond fourth order (e.g., sixth order), the improvements are negligible; therefore, GRoovy adopts fourth-order finite differencing for metric derivatives in GRHD source terms. Given these results, we plan to incorporate higher-order finite-difference schemes for metric derivative source terms into future versions of IllinoisGRMHD.

Second, in evolving initially cold hydrodynamic flows modeled by polytropic or piecewise polytropic EoSs, fluid pressures can deviate from their cold values only through shock heating; there is no mechanism for cooling below the initial cold pressure $P_{\rm cold}$. However, numerical errors can drive $P < P_{\rm cold}$, especially near neutron star surfaces. In Section 5.6.2, we show that applying a pressure floor set to the physical minimum, $P_{\rm cold}$, leads to larger central density drifts. Conversely, allowing un-physical pressures by decreasing the floor minimizes these drifts. The original IllinoisGRMHD mitigated this issue by setting the floor to $0.9P_{\rm cold}$, balancing diffusion rates against the allowance of un-physical pressures. In contrast, GRoovy and GRHayL adopt a stricter approach, setting the pressure floor to the minimum value allowed by the physics, $P_{\rm cold}$.

5.4 Code Tests

In the following subsections, we demonstrate our code's ability to solve standard numerical tests from the literature in both fixed (5.4.1, 5.4.2) and dynamical (5.4.3) spacetimes.

5.4.1 Flat, Static Spacetime Tests

5.4.1.1 Balsara 0 Shock Test

Standard MHD tests include the Balsara tests [126], which evolve shocks and contact discontinuities in a magnetized plasma in 1D, typically using a Γ-law EoS. To showcase the high-resolution shock-capturing capabilities of our implementation in curvilinear coordinates, we evolve the Balsara 0 test (the Balsara 1 test without magnetic fields) in spherical coordinates. The initial conditions for the stationary neutral plasma are

$$\rho(z) = \left\{ \begin{array}{lcl} 1.0 & ; & z \leq z_0 \\ 0.125 & ; & z > z_0 \end{array} \right\}, \tag{5.44}$$

$$P(z) = \left\{ \begin{array}{lcl} 1.0 & ; & z \leq z_0 \\ 0.1 & ; & z > z_0 \end{array} \right\}, \tag{5.45}$$

where z_0 is the initial shock location. Unlike previous studies [126, 127], which place the shock at the origin, we set $z_0 = 1.0$, propagating the shock along the z-axis, transforming the 1D Cartesian test into a significantly more challenging 2D spherical-coordinate test.

We set the outer boundary at $r_{\rm max} = 2.0$ and evolve the data to $t = 1.0$ using total variation diminishing (TVD) reconstruction with a minmod limiter [123] and the Noble2D hybrid EoS conservative-to-primitive solver from GRHayL. The resolution is $(N_r, N_\theta, N_\phi) = (520, 260, 2)$, and we adopt a Γ-law EoS of the form $P = (\Gamma - 1)\rho\epsilon$, where ϵ is the specific internal energy and $\Gamma = 2$. This setup uses the simple EoS option from GRHayL's EoS module.

In Fig. 5.1, we compare our results with the exact solution obtained using the Riemann solver from Ref. [128]. While our results show good agreement, small oscillations are present, likely caused by the uneven resolution along the shock front as the shock and contact discontinuity propagate in opposite directions along the z-axis in spherical coordinates. This demonstrates that our code can successfully handle standard numerical tests in more challenging curvilinear coordinate settings.

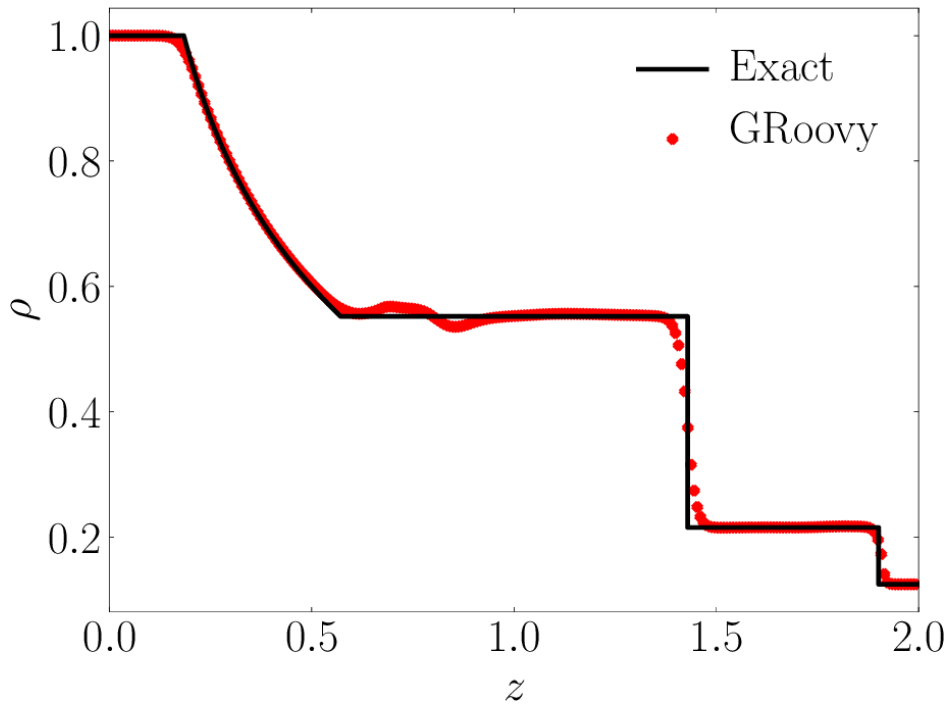

FIG. 5.1. Density profile at time $t = 1.0$ from the evolution of the Balsara 0 initial data.

5.4.1.2 Spherical Explosion

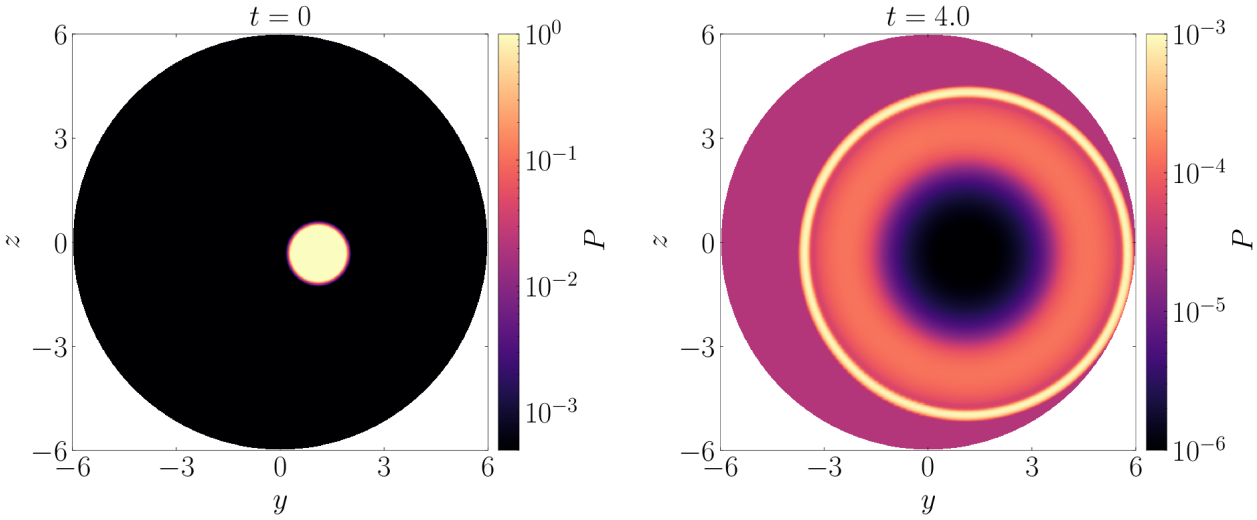

FIG. 5.2. Pressure profiles from the spherical explosion test, at $t = 0$ and $t = 4.0$.

To demonstrate our code's ability to handle hydrodynamic flows across coordinate singularities, we perform a standard hydrodynamic test: the spherical explosion [129]. This test models an over-dense ball of gas expanding into an ambient medium, a scenario most

efficiently evolved in spherical coordinates with the explosion centered at the origin. It becomes a "torture" test when the explosion is offset from the z-axis (as in Ref. [36]), requiring the code to handle hydrodynamic propagation across coordinate singularities while preserving spherical symmetry. This makes it a valuable benchmark for GRMHD and GRHD codes in spherical coordinates.

Following Ref. [36], we evolve the initial data in spherical coordinates, offsetting the explosion profile by $y = 1.1$, and additionally by $z = 0.3 \times y$. The initial density profile, following Ref. [127], is given by

$$\rho(r) = \begin{cases} \rho_{\rm in} & ; \quad r \leq r_{\rm in} \\ \exp \frac{(r_{\rm out}-r)\ln\rho_{\rm in} + (r-r_{\rm in})\ln\rho_{\rm out}}{r_{\rm out}-r_{\rm in}} & ; \quad r_{\rm in} < r < r_{\rm out} \\ \rho_{\rm out} & ; \quad r \geq r_{\rm out}. \end{cases} \tag{5.46}$$

A similar pressure profile is used, where $r_{\rm in}$ and $r_{\rm out}$ denote the inner and outer radii, respectively, with $r_{\rm in} = 0.8$ and $r_{\rm out} = 1.0$. The inner region has $\rho_{\rm in} = 10^{-2}$ and $p_{\rm in} = 1.0$, while the outer region has $\rho_{\rm out} = 10^{-4}$ and $p_{\rm out} = 3 \times 10^{-5}$. We adopt a Γ-law EoS with $\Gamma = 4/3$, use monotonized central (MC) reconstruction [123], and set the grid resolution to $(N_r, N_\theta, N_\phi) = (160, 80, 160)$.

We observe that the shock front successfully propagates through the z-axis coordinate singularity without significant artifacts, as shown in Fig. 5.2. While the shock front slows down artificially as the radial velocity component crosses zero at the singularity, this slowdown has no lasting effects at later times.

5.4.1.3 Neutrino Tests

We conclude our flat spacetime tests with a set of neutrino tests, demonstrating the successful integration of `NRPyLeakage` [27] into our code via GRHayL's neutrinos module.

The first test evaluates the code's ability to initialize optical depth in a constant-density sphere, representing an optically thick gas. The sphere has a radius of $r_{\rm Sph} = 2.5$, density $\rho^{\rm Sph} = 1.58 \times 10^{-4}$, electron fraction $Y_{\rm e}^{\rm Sph} = 0.1$, and temperature $T^{\rm Sph} = 8.0$, embedded in an ambient medium with $\rho^{\rm Ext} = 9.71 \times 10^{-11}$, $Y_{\rm e}^{\rm Ext} = 0.5$, and $T^{\rm Ext} = 0.01$. For this test, we use the SLy4 EoS [130] and a spherical coordinate grid with resolution $(N_r, N_\theta, N_\phi) =$

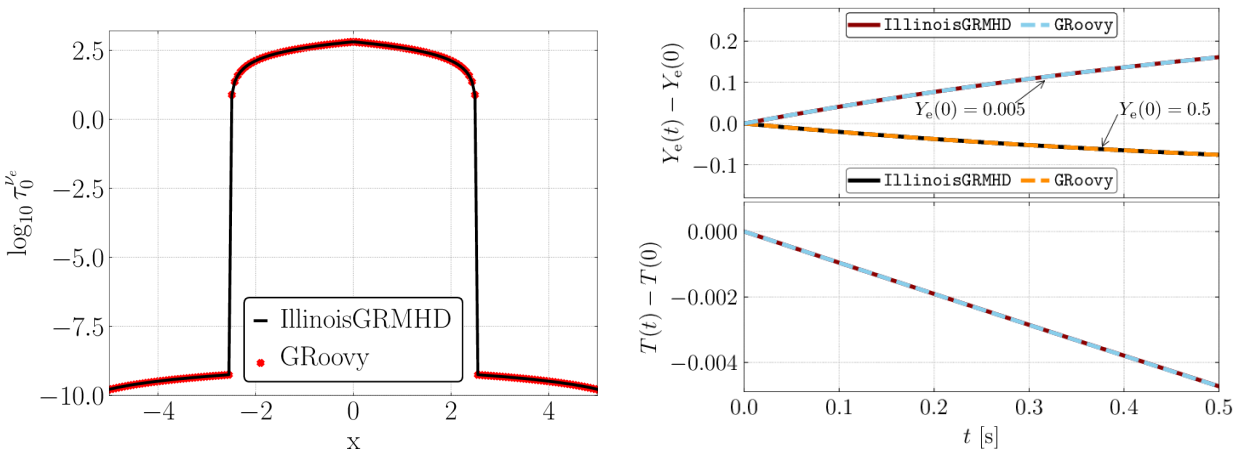

FIG. 5.3. **Left**: Radial profile of optical depth from the initialization procedure for an optically thick gas. **Right**: Time evolution of the electron fraction Y_e and temperature T for an optically thin gas.

$(60, 2, 2)$. After applying this algorithm, we find excellent agreement between the optical depth obtained from our initialization procedure and the result from IllinoisGRMHD [27], as shown in the left panel of Fig. 5.3.

The second test of our implementation of `NRPyLeakage` models an optically thin, neutrino-laden isotropic gas in spherical coordinates. We evolve two versions of the initial data, with $Y_e\,(0) = 0.5$ and $Y_e\,(0) = 0.005$. For both tests, we set $\rho = 10^{-12}$ and $T = 1$. In the right panel of Fig. 5.3, we compare our results against IllinoisGRMHD, confirming agreement between the solutions.

5.4.2 Curved, Static Spacetime Tests

5.4.2.1 Non-Rotating NS with a Hybrid EoS

An essential test for GRHD codes is the stable evolution of an isolated neutron star, where hydrodynamic fields are coupled to a curved spacetime. This test, often performed in Cartesian coordinates with adaptive mesh refinement (AMR), involves first setting up Tolman–Oppenheimer–Volkoff (TOV) initial data. The TOV equations provide a solution to Einstein's equations for a spherically symmetric star in hydrostatic equilibrium, but do

not assume an EoS. To this end, we choose a cold polytropic EoS of the form

$$P = K\rho^{\Gamma}, \tag{5.47}$$

where K is the polytropic constant and Γ is the adiabatic index. For all polytropic EoS tests in this chapter, we set a central density $\rho_c = 1.28 \times 10^{-3}$, $K = 100$, and $\Gamma = 2$. These parameters correspond to a star with a gravitational mass of $M = 1.4$ and a radius of $R = 8.1$ in code units, representing cold, degenerate nuclear matter. We generate these initial data using the TOV solver within NRPy [94].

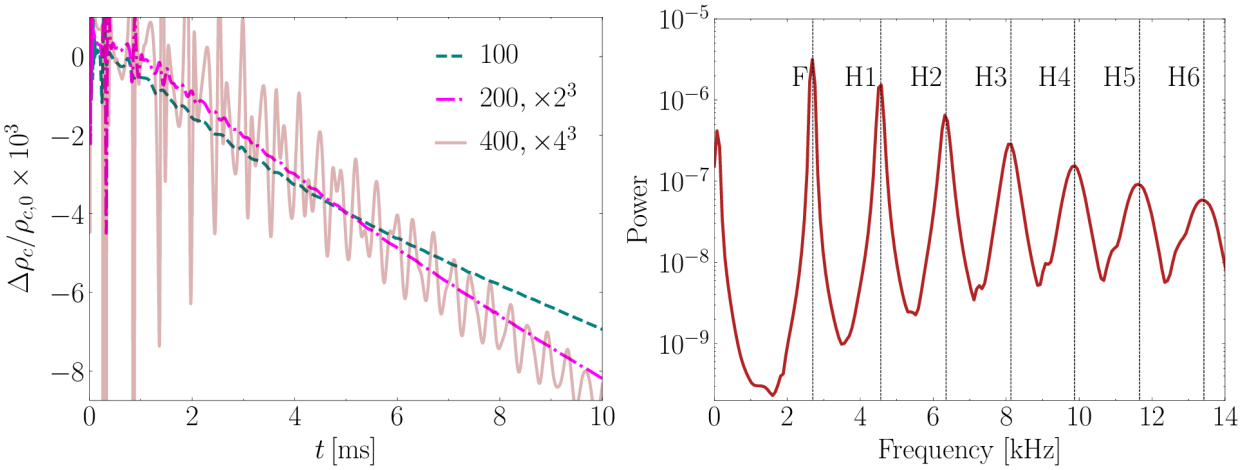

FIG. 5.4. Static-spacetime TOV evolution in spherical coordinates: central density drift. **Left**: Convergence study showing approximately third-order convergence, with radial resolutions $N_r = (100, 200, 400)$ and angular resolutions fixed at $N_\theta = N_\phi = 2$. The high-resolution run captures more power in the high-frequency overtones, as illustrated in the right panel. **Right**: Power spectrum of the central density for $N_r = 400$. Dashed vertical lines mark the fundamental mode (F) and overtones (H1–H6) from Ref. [131]. A Hann window is applied to the time series before performing the Fourier transform.

We test our code's ability to evolve these hydrodynamic fields atop a static spacetime—the Cowling approximation—in spherical coordinates. For these tests, we evolve the star with a hybrid EoS (with zero thermal part initially), set the outer boundary at $r_{\rm max} = 20$, and maintain a constant-density atmosphere with $\rho_{\rm atm} = \rho_c \times 10^{-13}$.

For this test, we use the Noble2D conservative-to-primitive solver with the 1D solver from Ref. [121] as a backup, and employ PPM reconstruction. Although the Noble2D solver occasionally fails to converge in low-density regions during the initial iterations, the 1D solver

reliably resolves these cases, allowing the simulation to proceed smoothly. Finally, we place the star at the origin, effectively making this a 1D test.

Since we employ a finite volume scheme to solve the GRHD equations, truncation errors induce an initial perturbation in the star, leading to oscillations. These perturbations arise both at the surface, where the density drops by orders of magnitude, and at the center, where PPM reduces to lower order at extrema [97]. These errors also cause the star to gradually diffuse over time, resulting in a downward drift of the central density. The left panel of Fig. 5.4 demonstrates that doubling the resolution from $N_r = 100$ to 200, and then from 100 to 400, reduces the diffusion by approximately a factor of 8 and 64, respectively. This result implies nearly third-order convergence, consistent with the expected convergence order of IllinoisGRMHD, which has been shown to achieve accuracy between second and third order.

Meanwhile, the magnitude of oscillations induced at the start of the evolution increases significantly as the radial resolution is increased from 200 to 400–a much higher radial sampling than is typical in numerical relativity simulations, with over 320 points across the diameter of the star. This behavior, as shown in the right panel of Fig. 5.4, results from higher harmonics from the initial perturbation being better resolved and not diffused away. When run at high resolution, our code successfully maintains power up to the sixth overtone for this TOV model, consistent with the frequencies reported in Ref. [131].

5.4.2.2 Non-Rotating NS with a Tabulated EoS and Neutrino Leakage

Hybrid EoS prescriptions have been widely adopted in the literature with considerable success, but they lack the capability to accurately model finite temperature effects compared to more realistic EoSs. In this Section we model a TOV solution using the SHT [132] EoS. We reproduce the sTOV-SHT model from Ref. [133] using the solver in NRPy, constructing the initial data from a beta-equilibrium, constant-entropy slice of the EoS table, at specific entropy $s = 1\,k_B$. The resulting solution has a central density of $9.3 \times 10^{14}\mathrm{g\,cm^{-3}}$, a gravitational mass of $2.741\,M_\odot$, and a central temperature of $\sim 30\,\mathrm{MeV}$. We verify the correctness of our optical depth calculation for this case as well, finding qualitative agreement with Fig. 1 of Ref. [134].

For the test presented here, we use spherical coordinates, PPM reconstruction, enable neutrino leakage, and use the one-dimensional conservative-to-primitive routine of Ref. [122] as implemented in GRHayL. To compare our results with those found in Ref. [134] using the GR1D [135] code, which solves the GRHD equations in spherical symmetry, we use a radial resolution of 25 m, and place the outer boundary at $r_{\rm max} = 62\,$km. Finally, we use a constant density atmosphere set to the minimum value of density in the EoS table. We show our results in Fig. 5.5, plotting the time evolution of the luminosities of electron neutrinos ν_e, electron antineutrinos $\bar{\nu}_e$, and heavy lepton neutrinos $\nu_x = \{\nu_\mu, \nu_\tau\}$, in the left panel. We compute neutrino luminosities using

$$\mathrm{L}_{\nu_i} = \int \alpha^2 W Q^{\rm eff}_{\nu_i} \sqrt{\gamma} d^3x, \tag{5.48}$$

where $Q^{\rm eff}_{\nu_i}$ is the effective emission rate for each $\nu_i = \{\nu_e, \bar{\nu}_e, \nu_x\}$. In the right panel we show the dominant frequencies in the luminosity oscillations, finding good agreement with the frequencies quoted in Ref. [133]. We note that while our luminosities are three orders of magnitude lower than those found in Ref. [133] and about two orders lower than in Ref. [134], both studies determined that these luminosities are strongly dependent on the atmosphere prescription. Thus, we attribute this result to the low numerical viscosity in the code when using spherical coordinates. This point will be expanded upon in the following Section. In our testing we find that setting the atmospheric density to approximately $\rho_{\rm atm} = \rho_c \times 10^{-6}$ leads to an increase in the luminosities by about two orders of magnitude, and they increase further when using cylindrical coordinates. Our results also seem reasonable given that Ref. [133] noted that the luminosities extracted from evolving an initially cold model, reaching a maximum of $\sim 10^{48}\mathrm{erg\,s^{-1}}$, should be treated as the minimum luminosity their leakage scheme can model.

5.4.3 Curved, Dynamical Spacetime Tests

5.4.3.1 Non-Rotating NS with a Hybrid EoS

Having demonstrated that GRoovy performs as expected in both flat and curved-but-static spacetime tests, we now turn to dynamical spacetime tests, which involve coupling

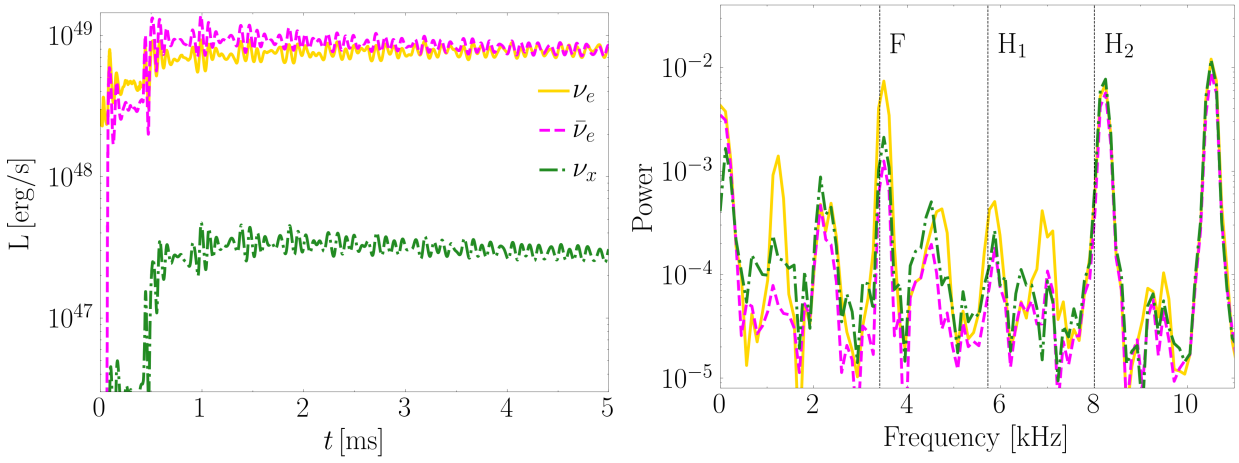

FIG. 5.5. Luminosities computed using the neutrino leakage module of GRHayL, from evolving a hot TOV model using the SHT [132] EoS in a static spacetime. **Left**: Time evolution of neutrino luminosities, using spherical coordinates with a radial resolution of 25 m. **Right**: Normalized power spectrum for all three neutrino species, showing the fundamental mode and overtones. Reference frequencies taken from Ref. [133].

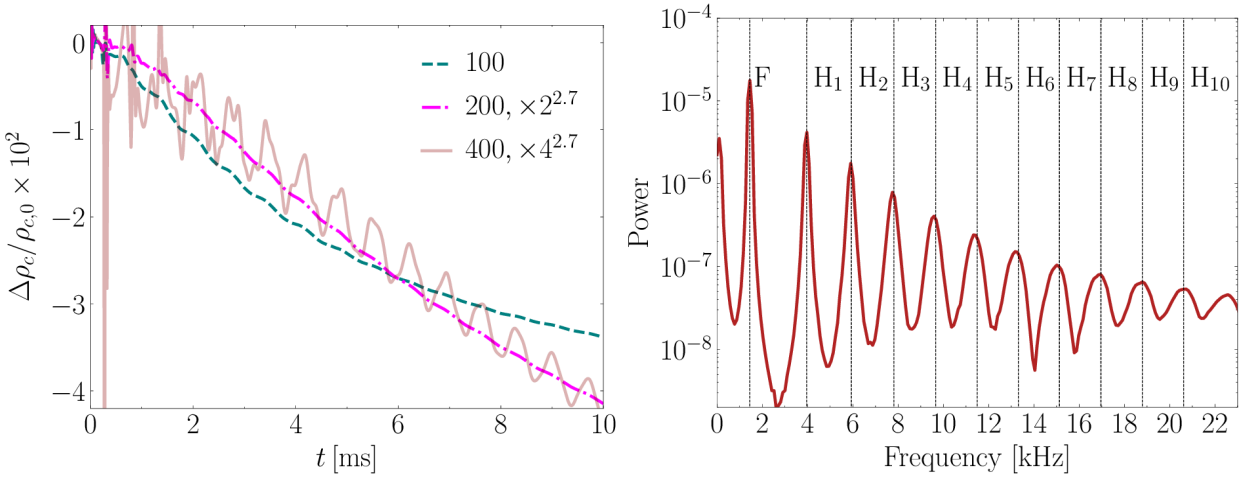

FIG. 5.6. Dynamical-spacetime TOV evolution in spherical coordinates: central density drift. **Left**: Similar to the left panel of Fig. 5.4, but with the vertical axis (density drift) rescaled by 10^2 instead of 10^3, and results at different radial resolutions rescaled assuming 2.7-order convergence. **Right**: Similar to the right panel of Fig. 5.4, but including overtones up to the tenth harmonic provided by [136].

to our spacetime solver. For spacetime evolution, we utilize the BSSN formulation with moving-puncture gauge conditions, incorporating the $1 + \log$ lapse evolution [14, 137] and a

second-order, non-covariant advecting shift evolution [138], governed by

$$\partial_t \alpha = \beta^i \partial_i \alpha - 2\alpha K, \tag{5.49}$$

$$\partial_t \beta^i = \beta^j \partial_j \beta^i + B^i, \tag{5.50}$$

$$\partial_t B^i = \beta^j \partial_j B^i + \frac{3}{4} \partial_0 \bar{\Lambda}^i - \eta B^i, \tag{5.51}$$

where K_{ij} is the extrinsic curvature, K is its trace, B^i and $\bar{\Lambda}^i$ are auxiliary evolution variables, and η is a damping parameter. We use fourth-order finite differencing to represent spatial derivatives in the BSSN equations, set $\eta = 0$, and apply Kreiss-Oliger dissipation [139] with strength $k = 0.2$, utilizing a fifth-order finite difference stencil.

In the left panel of Fig. 5.6, we present the results of our convergence study for this test, using the same resolutions and initial data as Sec. 5.4.2.1. The results are consistent with those from the static spacetime tests, showing that the drift in central density converges within the expected range (between second- and third-order), at approximately 2.7th order. The Fourier transform of the $N_r = 400$ central density (right panel of Fig. 5.6) shows frequencies in excellent agreement with predictions from a code deriving these frequencies from linear perturbations [136], extending up to the tenth overtone.

We next compare the evolution of identical TOV initial data simulated using our code in spherical and cylindrical coordinates, alongside evolutions in Cartesian coordinates performed with GRHayLHD [95, 96], a pure HD version of IllinoisGRMHD based on GRHayL in the Einstein Toolkit. The results are presented in Fig. 5.7. All simulations are performed with the same evolution parameters; only the coordinates and grid resolutions are changed.

To facilitate comparison, we use a spherical coordinates simulation with $\Delta r_m = 0.2$ as our reference case. Simulations in other coordinate systems use either the same resolution (e.g., $\Delta x = 0.2$ for Cartesian coordinates) or the equivalent "effective" resolution. For Cartesian coordinates, the effective resolution corresponds to Δx such that $\Delta x^2 + \Delta y^2 + \Delta z^2 = 3\Delta x^2 = \Delta r_m^2$. For the reference $\Delta r_m = 0.2$, this yields an effective $\Delta x \approx 0.11$.

The top left panel of Fig. 5.7 shows that the Cartesian simulation at $\Delta x = 0.2$ (solid black) exhibits significantly less diffusion than the spherical simulation at $\Delta r_m = 0.2$ (solid blue). The cylindrical simulation (dashed orange), which combines elements of Cartesian

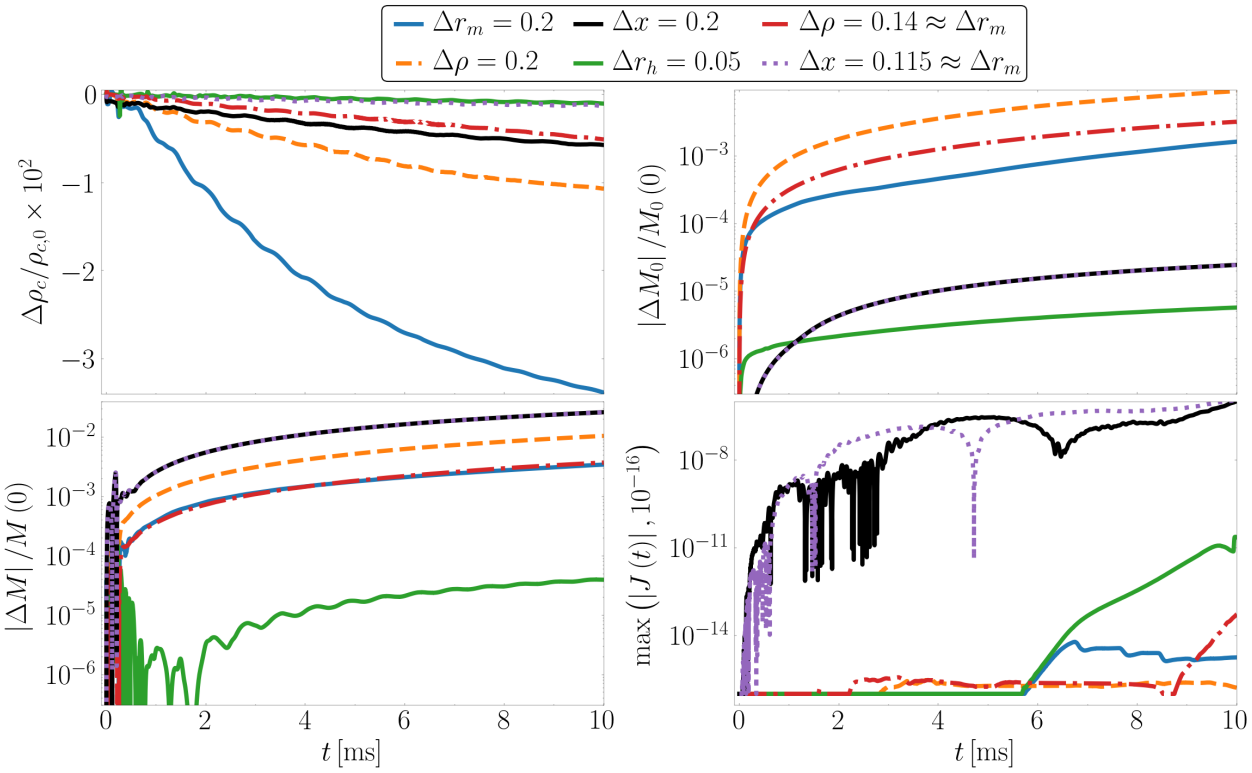

FIG. 5.7. Evolution of a TOV solution simulated in Cartesian, spherical, and cylindrical coordinates. We compare simulations using identical *coordinate* resolutions and identical *effective* resolutions, relative to the reference spherical case ($\Delta r_m = 0.2$). The panels show the time evolution of fractional changes in: (**top left**) central rest-mass density; (**top right**) rest mass; (**bottom left**) gravitational mass; and (**bottom right**) spin angular momentum. The rest mass, gravitational mass, and spin are computed using Eq. (5.52), Eq. (5.53), and Eq. (5.54), respectively.

and spherical coordinate systems, falls between the two. When using equivalent effective resolutions, the Cartesian (dotted purple) and cylindrical (dot-dashed red) simulations are even less diffuse, closely approaching the high-resolution spherical results at $\Delta r_h = 0.05$ (solid green).

These results may initially appear counterintuitive, as spherical grids naturally align with the symmetry of the TOV solution. However, the behavior of high-resolution shock-capturing (HRSC) schemes explains this trend. In HRSC schemes, hydrodynamic fluxes are computed across cell faces, and diffusion occurs fastest along coordinate lines. On Cartesian grids, the stellar surface diffuses most rapidly along the x, y, and z axes, as shown in Fig. 3 of Ref. [140]. Diffusion along diagonal lines (e.g., the $x = y = z$ line) is slower, as radial diffusion requires fluxes to traverse multiple orthogonal cell faces in a zigzag pattern. In

contrast, on spherical polar grids, radial fluxes align directly with cell faces, enabling faster radial diffusion. This alignment smooths the stellar surface more rapidly, resulting in greater overall diffusion compared to Cartesian grids.

As illustrated in the top left panel of Fig. 5.7, spherical coordinates require a resolution of approximately $\Delta r_h = 0.05$ (solid green) to achieve results comparable to $\Delta x = 0.115$ Cartesian simulations. However, the high-resolution simulation with spherical coordinates requires less than *1/25000* the computational cost, using $400\times2\times2$ grid points compared to the 346^3 required for the high-resolution Cartesian simulation.

The top right and bottom panels of Fig. 5.7 show the time evolution of the fractional change in rest mass M_0, gravitational mass M, and spin angular momentum J, which we compute using

$$M_0 = \int d^3x\sqrt{\gamma}\alpha u^0\rho, \tag{5.52}$$

$$M = \frac{1}{16\pi}\int d^3x\sqrt{\bar{\gamma}}\bigg(16\pi\psi^5\rho_\epsilon + \psi^{-7}\bar{A}_{ij}\bar{A}^{ij} - \psi\bar{R} - \frac{2}{3}\psi^5K^2\bigg), \tag{5.53}$$

$$J = \int d^3xT^0_\phi\alpha\sqrt{\gamma}, \tag{5.54}$$

where $\rho_\epsilon = (1+\epsilon)$ is the mass—energy density, $\bar{A}^{ij}$ is the conformal trace-free part of the extrinsic curvature, and $\bar{R} = \gamma^{ij}\bar{R}_{ij}$, with $\bar{R}_{ij}$ being the conformal Ricci tensor. Note that Eq. (5.54) is only valid for systems with axisymmetry.

The top right panel of Fig. 5.7 shows that the runs in Cartesian coordinates conserve rest-mass to at least the level of truncation error associated with our midpoint-rule approximation for the integral in Eq. (5.52). Although the simulations in cylindrical coordinates and the reference simulation in spherical coordinates do not conserve rest-mass as effectively, the high-resolution simulation in spherical coordinates achieve a comparable level of conservation. As expected, the cylindrical coordinate results are the least accurate due to the system's geometry not conforming well to the star. These inaccuracies are compounded by truncation errors from the source terms in Eq. (5.38); however, the errors demonstrably converge away with increasing resolution.

In the bottom left panel, the Cartesian runs exhibit the poorest conservation of gravi-

tational mass, as conservation is not guaranteed in this case, while the high-resolution simulation in spherical coordinates again does the best. Finally, the bottom right panel shows that while all simulations exhibit reasonable behavior in the evolution of the spin angular momentum (which should ideally remain at the round-off level), the curvilinear coordinate simulations are significantly closer to this ideal level before outer boundary effects become apparent.

5.4.3.2 Non-Rotating NS with a Tabulated EoS

Here we model a TOV solution with a tabulated EoS in spherical coordinates within the framework of full general relativity—an approach, to the best of our knowledge, undertaken for the first time. We construct initial data using a beta-equilibrium, $T = 0.01\,\mathrm{MeV}$ constant-temperature slice of the SLy4 tabulated EoS [130]. Atmospheric values for the primitives are set to $\rho_{\rm atm} = 1.4 \times 10^{-12}$, $T = 0.01\,\mathrm{MeV}$, and $Y_{\rm e} = 0.5$. For the specific TOV model considered, we select a gravitational mass of $M = 1.4$, corresponding to a central density of $\rho_c = 1.42 \times 10^{-3}$ and a radius of $R = 6.5$.

The star is modeled in spherical coordinates with radial resolutions $N_r = (100, 150, 200)$ and angular resolutions set to the minimum allowed in NRPy: $N_\theta = N_\phi = 2$. For primitive recovery, we again use the one-dimensional routine of Ref. [122]. Fig. 5.8 presents the results of this test at varying resolutions. Initially, the high-resolution run exhibits problematic high-frequency oscillations in the central density, attributed to insufficient numerical viscosity and the low resolution or inherent non-smoothness of table quantities. However, these oscillations are mitigated when a higher resolution EoS table is used (blue curve), in clear contrast to the oscillatory behavior observed with the lower resolution table (orange curve).

We generated this new table using the SRO code [1], modifying only the sampling options. Specifically, we increased the steps per decade in $(Y_{\rm e}, n, T)$, where n is the number density, from $(67, 30, 30)$ to $(88, 80, 80)$ in the final table. In future work, we aim to implement a spectral representation for piecewise polytropic and tabulated EoSs [141–144], which will help address the challenges associated with using raw tables.

[1] Available at http://stellarcollapse.org.

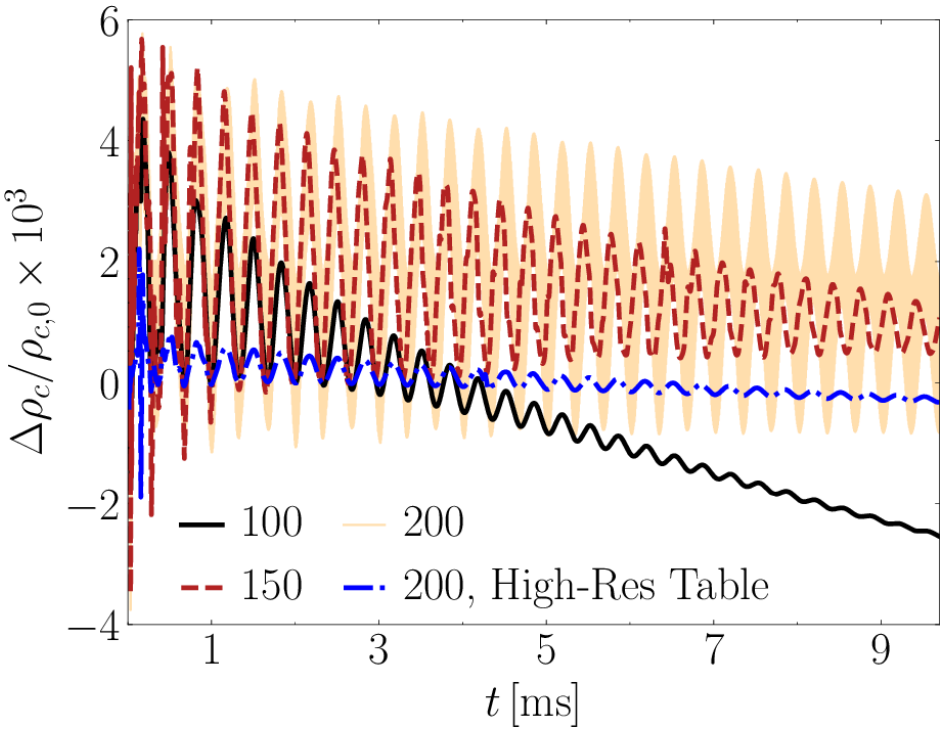

FIG. 5.8. Evolution of the normalized change in central density over time for a TOV model using the SLy4 tabulated EoS in spherical coordinates. We employ radial resolutions of $N_r = (100, 150, 200)$ while fixing the angular resolution at $N_\theta = N_\phi = 2$. The EoS table resolution is varied to illustrate the emergence and resolution of un-physical high-frequency oscillations.

5.4.3.3 Uniformly Rotating NS with a Hybrid EoS

In our final test, we examine GRoovy's capability to model a uniformly rotating neutron star (NS). This is particularly relevant for potential "handoff" studies, where the code could model binary neutron star (BNS) or black hole-neutron star (BHNS) post-merger remnants over seconds-long timescales. For this test, we evolve the BU2 model from Ref. [145] with $K = 100$ and $\Gamma = 2$. This model describes a rotating NS with the following properties: $M = 1.47$, equatorial coordinate radius $R_e = 8.54$, and dimensionless spin angular momentum $J/M^2 = 3.19 \times 10^{-1}$. Initial data for this model is generated using the open-source FUKA initial data solver [146].

FUKA is based on an extended version of the KADATH spectral solver library, which is specifically designed for solving numerical relativity problems [147]. KADATH employs a novel non-overlapping multi-grid spectral domain decomposition such that regions of high physical gradients such as the stellar surface can be defined along a domain boundary using appropriate boundary conditions. This method facilitates robust convergence and reduces numerical artifacts such as the *Gibbs phenomenon* produced by modeling shocks and steep gradients with spectral methods. FUKA computes initial data solutions using the eXtended Conformal Thin Sandwich (XCTS) decomposition of Einstein's constraint equations [148, 149]. Finally, FUKA adopts the conformal flatness approximation ($\gamma_{ij} = \psi^4 \delta_{ij}$ in Cartesian coordinates) and maximal slicing ($K = 0$) conditions.

For isolated neutron stars, FUKA models matter as an isentropic fluid in equilibrium and co-moving with the fluid undergoing uniform rotation which is parameterized by a rotational velocity ω (see, e.g.,Refs. [146, 150]). To close the system of equations, an EoS must be specified. For this test, we use a simple polytropic EoS. However, FUKA also supports piecewise polytropic and 1D cold tabulated EoSs for modeling isentropic fluids, which may be adopted in future studies. In this work, we leverage FUKA's native Python interface that enables full access to the initial data solution from Python, which we incorporate into our NRPy workflow.

PPM's lower-order accuracy at extrema is undesirable when evolving neutron stars in either Cartesian [97] or spherical coordinates [36]. Therefore, for this test we use MC

reconstruction, which exhibits more consistent—though second-order—convergence with increasing resolution. The spacetime evolution settings remain consistent with those used in the previous subsections. A uniform spherical grid is adopted for this test, with the outer boundary set at $r_{\rm max} = 25$. Spatial resolutions of $(N_r, N_\theta, N_\phi) = (300, 12, 2)$, $(400, 16, 2)$, and $(500, 20, 2)$ are employed, while maintaining a constant density atmosphere of $\rho_{\rm atm} = 1.29 \times 10^{-11}$.

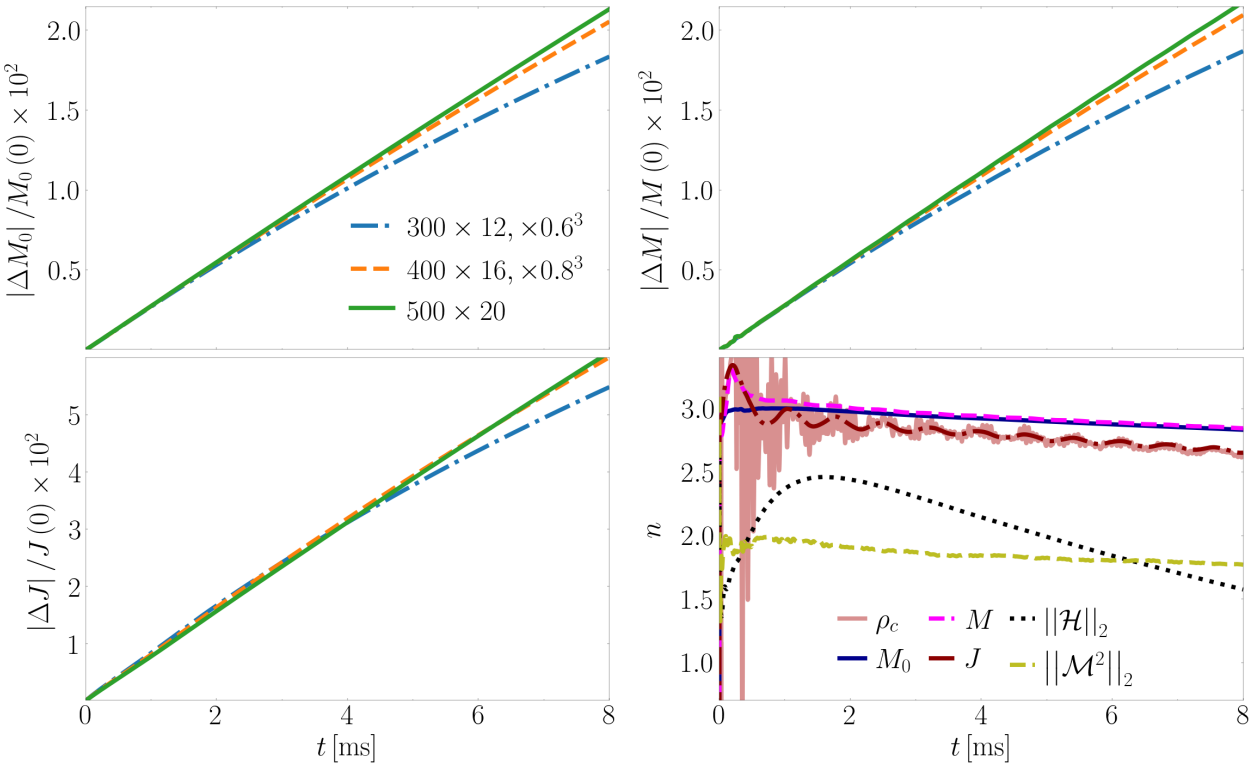

FIG. 5.9. Convergence study for the evolution of a uniformly rotating neutron star using three spatial resolutions: $(N_r, N_\theta, N_\phi) = (300, 12, 2)$, $(400, 16, 2)$, and $(500, 20, 2)$. **Top left, top right, and bottom left:** Evolution of the normalized change in rest mass, gravitational mass, and spin angular momentum, respectively. The results from the lower-resolution simulations are rescaled assuming third-order convergence to demonstrate error reduction relative to the high-resolution run. **Bottom right:** The convergence order n, calculated using Eq. (5.55) with data from the medium and high-resolution simulations. n is plotted for central density, rest and gravitational mass, spin angular momentum, and the L^2 norms of the Hamiltonian and momentum constraint violations. The observed convergence orders are consistent with the numerical scheme.

For this test we also monitor the L^2 norms of the Hamiltonian $\mathcal{H}$ and momentum $\mathcal{M}^2 = \mathcal{M}^i \mathcal{M}_i$ constraint violations, computed using Eqs. (46) and (47) from Ref. [29]. The top left, top right, and bottom left panels of Fig. 5.9 show the normalized change in rest

mass, gravitational mass, and spin angular momentum, respectively. In the high-resolution simulation, the rest and gravitational masses change by no more than 2.5% over 8 ms, while the angular momentum changes by less than 6%. Further, we rescale the results from the lower resolution simulations assuming a third-order convergence order. Given that the spacetime evolution employs a fourth-order finite difference scheme, the hydrodynamic finite-volume scheme is approximately second order, while reconstruction schemes cannot be better than first-order at shocks like the NS's surface, and both are coupled to a fourth-order timestepping scheme, the expected convergence order of our numerical scheme should fall between first and fourth order. This is seen in the bottom right panel of Fig. 5.9, where we use data from our medium and high resolution simulations to plot the computed convergence order n of the normalized change in central density, rest- and gravitational mass, spin angular momentum, and the Hamiltonian and momentum constraints. Assuming our scheme is truncation-error dominated, we compute n using

$$n = \frac{1}{\log_{10}(f)} \log_{10}\left(\frac{|\varepsilon|^{\text{med}}}{|\varepsilon|^{\text{high}}}\right), \tag{5.55}$$

where f is the ratio of the grid spacing of the high and medium resolution simulations, and ε is the computed error at the given resolution. In the continuum limit towards infinite resolution, ε should converge to zero at some order between one and four. We observe that after initial transient behavior these quantities all converge at expected order. We note that the downward drift in the convergence order of the Hamiltonian constraint is most likely due to our use of a fairly close outer boundary, which is expected considering that the simulations last for over 30 light-crossing times.

5.5 Conclusions and Future Work

In this chapter, we introduced GRoovy, a numerical relativity code for evolving GRHD fluids with advanced EoSs in diverse coordinate systems, including singular curvilinear (e.g., spherical and cylindrical) and non-singular (Cartesian) coordinates. This development broadens our ability to simulate a wide range of astrophysical systems. Building on

the open-source, IllinoisGRMHD-based [26, 27] GRHayL [95, 96], GRoovy dynamically integrates the core GRHD equations and extends them to include key physical processes such as lepton number conservation and neutrino leakage. Its flux-conservative formulation mitigates numerical instabilities near coordinate singularities by solving the equations within an orthonormal basis. The versatility and robustness of GRoovy were validated through a comprehensive set of tests in flat, static curved, and dynamical curved spacetimes.

In flat and static spacetime scenarios, we demonstrated GRoovy's ability to accurately capture shock dynamics in hydrodynamical flows, and model both optically thin and thick gases using a neutrino leakage scheme. For curved, static and dynamical spacetimes, we successfully modeled non-rotating neutron stars with both polytropic and tabulated equations of state, as well as uniformly rotating neutron stars in spherical coordinates. In dynamical spacetime tests, GRoovy integrated seamlessly with our BSSN solver in the moving-puncture gauge, showcasing its capability to evolve spacetime metrics coupled with hydrodynamic fields. These results establish this preliminary version of GRoovy as a first step towards developing a powerful tool for simulating binary neutron star (BNS) and black hole-neutron star (BHNS) post-merger remnants.

In the future, we plan to implement a mean-field prescription for magnetic field evolution [151–155]. This addition will allow us to study the large-scale effects of magnetic field dynamos on the overall dynamics in a primarily hydrodynamic context, as demonstrated in Ref. [153]. We also intend to leverage the in-development multi-patch, GPU, and `Charm++` parallelization infrastructures currently being integrated into BlackHoles@Home to conduct efficient full 3D simulations of important astrophysical phenomena.

Ultimately, our goal is to fully incorporate MHD, equipping GRoovy with another essential component to accurately capture the critical physics of compact object mergers. By enabling long-term simulations in coordinate systems conforming to the near-symmetries of compact binary remnants, GRoovy serves as a powerful and computationally efficient tool for investigating phenomena such as gamma-ray bursts, nucleosynthesis, and the evolution of post-merger remnants. This capability has the potential to drive new discoveries and deepen our understanding of compact astrophysical systems.

5.6 Detailed Analysis of Algorithmic Enhancements in GRoovy over IllinoisGRMHD

5.6.1 Finite-Differencing Accuracy Study

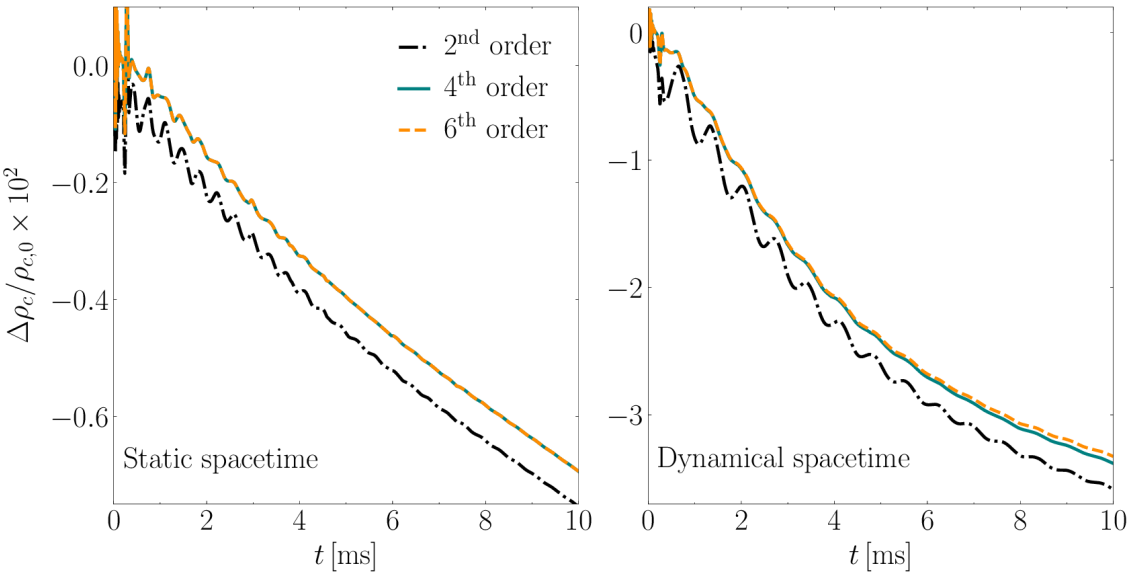

FIG. 5.10. Impact of GRHD metric source term finite differencing order on diffusion of TOV stars. **Left**: Drift of central density holding the spacetime evolution fixed (Cowling approximation). **Right**: Same as left panel, but evolving the spacetime.

We evolve the same Tolman-Oppenheimer-Volkoff (TOV) initial data described in Sec. 5.4.2 in both static and dynamical spacetimes to assess the impact of second-, fourth-, and sixth-order finite-difference stencils on stellar diffusion in source term calculations. The results, presented in Fig. 5.10, show that increasing the finite-differencing order from second to fourth significantly reduces central density drift in both static and dynamical spacetime evolutions. However, further increasing the order to sixth provides negligible improvements, indicating diminishing returns beyond fourth-order accuracy.

These findings suggest that fourth-order stencils strike a practical balance between computational cost and accuracy for equilibrium neutron star simulations. Our results confirm that the higher-order stencils supported by NRPy significantly outperform the second-order stencils used in the original IllinoisGRMHD implementation, reducing the rate at which the star diffuses.

5.6.2 Pressure Floor Study

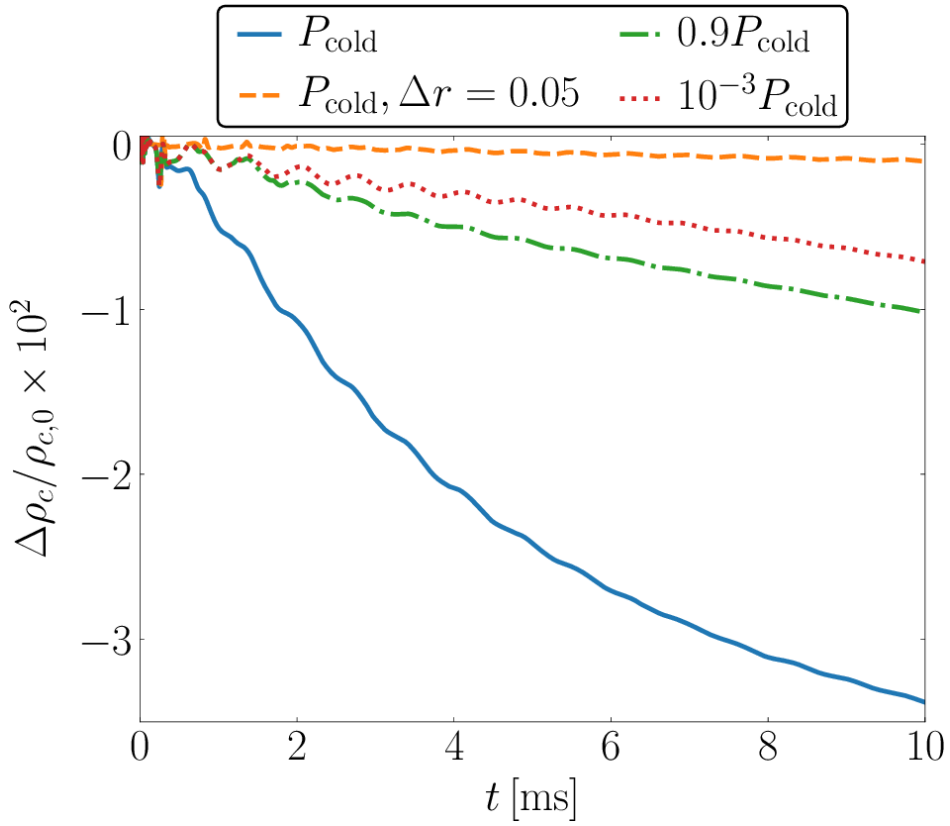

FIG. 5.11. Comparison of the effects of different pressure floors on TOV stellar models using a hybrid EoS, with evolutions performed in spherical coordinates. The blue, green, and red curves represent simulations at a medium coordinate resolution of $\Delta r_m = 0.2$, while the orange curve corresponds to a high-resolution simulation with $\Delta r_h = 0.05$.

Fig. 5.11 illustrates the impact of varying pressure floors on the evolution of polytropic TOV initial data in spherical coordinates, using the same numerical setup as in Sec. 5.4.3.1. The blue, green, and red curves correspond to simulations at a medium resolution of $\Delta r = 0.2$, while the orange curve represents a high-resolution run at $\Delta r = 0.05$. For the hybrid EoS used here, the correct pressure floor is the cold pressure $P_{\rm cold}$ defined by the $\Gamma = 2, K = 100$ polytrope.

While setting the floor below $P_{\rm cold}$ introduces artificial cooling, which is not physically consistent with our GRHD equations or EoS, it reduces stellar diffusion, as shown in the figure. Despite these numerical benefits, lowering the floor sacrifices physical realism. Therefore, we choose $P_{\rm cold}$ as the pressure floor in our simulations to maintain physical fidelity.

Chapter 6

Modeling the Bar-mode Instability using Realistic EoSs

6.1 Introduction

The stability of rotating NSs has been a subject of active research for several decades, with analyses focusing primarily on non-axisymmetric instabilities. These instabilities can redistribute angular momentum in the NS, generating gravitational waves in the process. Magnetic fields may also be reconfigured in the process, leading to potential electromagnetic signatures. Proto-NSs, born from core-collapse supernovae, may experience these instabilities after formation, as they cool, contract and gradually spin up. NSs in low or high mass binaries may also be spun up by accretion. NS instabilities generate high frequency GWs, which may be detectable by future ground based observatories, such as Cosmic Explorer or the Einstein telescope. These signals could then be used to place more stringent constraints on the NS EoS, as well as provide more insights into the stability of rapidly rotating NSs.

In this Chapter we focus primarily on the bar-mode instability, which arises in rapidly rotating, quasi-toroidal NSs. These NSs undergo a transition to a bar-like configuration before redistributing matter and angular momentum to a more stable configuration. Such a configuration can be classified using the parameter $\beta = T/|W|$, where T is the rotational kinetic energy and W the gravitational potential energy. Susceptibility to the bar-mode instability occurs after β surpasses some critical threshold value β_c.

Several studies have attempted to find the value of β_c, in both Newtonian gravity and full GR (e.g.,[156, 157]). However, it has become apparent that the parameter is dependent on both mass and EoS. To somewhat control for this, Ref. [158] extrapolated β_c to the zero-mass limit, and presented an explicit dependence solely on EoS stiffness. However, these

previous studies were limited by the use of a simplified model for the NS EoS of the form

$$P = K\rho^{\Gamma}. \tag{6.1}$$

As discussed previously in Chapter 3, this analytic model does not accurately account for finite-temperature effects, or effects due to neutrino emission. Thus, while the study done in Ref. [158] placed some bounds on the value for β_c, it is unclear how well this constraint fares when using a more realistic EoS. There is therefore a need to conduct a similar study, but in which finite-temperature effects are also included. Therefore, this Chapter presents preliminary results for such a study, comparing two different EoSs in their value for β_c. We present some ongoing work to constrain β_c for two different EoSs. Throughout this Chapter we adopt geometrized units with $c = G = M_{\odot} = 1$.

6.2 Numerical Methods

Here we describe the numerical methods we use to conduct this study of the bar-mode instability in quasi-stable rapidly rotating NSs. Specifically, we outline the numerical methods used for this study, and the changes to the evolution methods used in Chapter 5.

6.2.1 Charm++ Parallelization

The bar-mode instability results in a non-axisymmetric transient state which occurs for rapidly rotating, quasi-toroidal NS models. As such, we need to model them in three dimensions, with moderate to high sampling throughout the bulk of the star. While we can use GRoovy to model these systems in curvilinear coordinates, we can no longer assume axisymmetry, as was done for the bulk of the tests presented in Chapter 5, which greatly increases computational costs.

To circumvent this issue, we opt to build upon the work presented in Ref. [159], where we modified the BlackHoles@Home infrastructure to be parallelized across HPC nodes. We do so using the Charm++ [160, 161] task-based parallelization infrastructure, in which units of computation are managed by a runtime manager, and can be executed asynchronously. Most notably, Charm++ allows for codes to be highly scalable across hundreds of CPUs. Ref. [159]

presented the SuperB framework, which is built on top of BlackHoles@Home, allowing for all NRPy-generated codes, including GRoovy, to be parallelized with minimal modifications. Ref. [159] presented excellent strong scaling up to $\sim$ 7200 CPUs. We therefore modify GRoovy to work with SuperB, and the code can now scale efficiently across HPC nodes. This is a dramatic improvement compared to the version of GRoovy presented in 5, which is restricted to only single node performance. In terms of computational costs, the previous version is restricted to spherically symmetric or axisymmetric simulations, while this new version is capable of conducting full, three-dimensional simulations.

6.2.2 Improved Finite-Temperature Evolution

As demonstrated in Chapter 5, numerical grids have varying degrees of effects on the dynamics within a simulation. In this study we choose to model these quasi-stable configurations using cylindrical coordinates, which is a convenient compromise between minimizing angular momentum loss to due grid effects, e.g.,from using Cartesian coordinates, and a reasonable time step to ensure stability and efficiency, as compared to the extremely small time steps required when using spherical coordinates (a consequence of ensuring numerical stability and the extreme grid focusing near the polar axis). However, in tests we have still encountered numerical instabilities when using tabulated EoSs. We therefore add additional features to GRoovy to mitigate these instabilities.

The first modification concerns our reconstruction step. In Chapter 5 we found that only second-order reconstruction algorithms gave satisfactory results for rotating systems and did not introduce un-physical oscillations. These oscillations occur for higher order schemes because such algorithms, such as PPM or MP5, require gradients of the fields being reconstructed. In singular curvilinear coordinates these gradients require additional geometric factors. Thus, instead of putting in work to make our current PPM algorithm more generic, or implement a generic version of the MP5 algorithm, we opt to instead implement the WENO5Z reconstruction algorithm. WENO5Z does not consider gradients of the underlying fields, and instead uses a smoothness indicator based on trial polynomial interpolations. As a result, we are able implement fifth-order reconstruction without having to consider geometric factors.

The second modification is in the choice of reconstructed primitive variables. Historically, the original IllinoisGRMHD code reconstructed the fluid density, pressure, and three-velocity, in the realm of pure hydrodynamics. When using a tabulated EoS, the original code would reconstruct these variables along with the electron fraction, and would make the approximation that temperature is constant within a grid-cell. However, in the Frankfurt version of IllinoisGRMHD [162], the code instead reconstructs density, temperature and electron fraction, and uses these to compute the pressure at the cell-interfaces. The authors of Ref. [162] cited improved behavior in the atmospheres of NSs, so we also opt for this approach. Further, we also include entropy S as a "passive scalar" primitive variable into our evolution scheme. Starting with the conservation of entropy,

$$\nabla_\nu \left(S u^\mu\right) = 0. \tag{6.2}$$

We follow the derivation for conserved baryon number, as we saw in Chapter 5. The evolution for conserved entropy $\tilde{S} \equiv e^{6\phi} S$ then takes the form

$$\partial_t \tilde{S} + \partial_i \left(\tilde{S} v^i\right) = -\hat{\Gamma}^i_{ij} \tilde{S} v^j. \tag{6.3}$$

While the system is fully described by our original choice of primitive variables, there can be some regions of EoS tables that have degeneracy between the temperature and specific internal energy, which in some instances can lead to un-physical heat up. To mitigate these errors, we therefore passively evolve the entropy, such that its evolution is only used within the conservative-to-primitive routines. Once given the entropy, we use the EoS tables to recover temperature, instead of relying on the internal energy. We note, however, that conservation of entropy breaks down at shocks, so this recovery method is not always reliable.

Third, in an effort to minimize errors associated with the artificial atmosphere, we also implement a more robust conservative-to-primitive routine. Instead of relying on a backup strategy of routines, which is commonly done in other codes, we separately use the energy and entropy versions of the solvers described in Ref. [122] and Ref. [121], for a total of four simultaneous routines. We then select the primitive variables associated with the smallest change in the conserved energy and electron fraction. In this way, we ensure that we acquire

the best results from the solver which may be more robust for a given region of an EoS table. We find that the combination of all these strategies lead to the most robust simulations of our models, and minimize un-physical heating in the atmosphere.

Finally, we implement the five-stage, fourth order accurate time stepping scheme presented in Ref. [163]. This new algorithm, SSPRK54, has the desired feature of being strong stability-preserving, allowing us to relax the CFL condition, such that we may take larger time steps in our simulations. SSPRK54 allows us to more than quadruple the CFL condition used in Ref. [44]. The cost of this algorithm is that it requires five stages instead of the four required for fourth order Runge-Kutta. Nevertheless, we are usually not memory-bound in simulations using curvilinear coordinates, so we expect at least a factor of two in overall speed up of our simulations, given that our time step is increased.

6.3 Initial Data

To model bar-model unstable NSs in full GR, we first need initial data for the matter and spacetime fields, such that the matter is in hydrostatic equilibrium and the spacetime satisfies the Einstein's equations. Previous studies modeling the instability have used the RNS code [164] available in the Einstein Toolkit. However, while RNS does include tabulated EoS support, we find that our integration with stellar collapse EoS tables and interpolation to our evolution grids in BlackHoles@Home is much simpler with the FUKA initial data code. FUKA solves for initial data by solving the extended conformal thin sandwich formulation of the 3+1 constraint equations [165], in which the data are assumed to be in time-symmetry. Different to Chapter 5, however, is how initial data are imported to GRoovy.

In Chapter 5, we followed a workflow in which we first generated the initial data using FUKA, then defined the data on our pre-defined numerical grid to be used by GRoovy. The downside to this approach was that the larger the grid, the longer this process required, especially given that this importer utilized Python tools. Another, more subtle issue was that with the new Charm++-parallelized version of GRoovy, it is not possible to know a priori which computational unit will read in any given chunk of the initial data on the numerical grid, which inevitably leads to race-conditions and undefined behavior. The remedy to both

these issues is to directly link GRoovy to the FUKA code. This way we can explicitly call the data import functions at the C++ level. GRoovy can now import any FUKA initial data set at runtime, including binary NS initial data.

6.3.1 Validation Tests of Initial Data

Table 6.1: Comparison of the S6 and U13 models, computed using the RNS code [164] and FUKA. Model S6 has a gravitational mass of $M = 1.449$, while U13 has $M = 1.462$. Listed are β, the ratio between the rotational kinetic energy and the gravitational potential energy, the central density ρ_c, and the dimensionless spin measure J/M^2, all in units of $c = G = M_\odot = 1$.

S6	β	$\rho_c\,[10^{-4}]$	J/M^2	**U13**	β	$\rho_c\,[10^{-5}]$	J/M^2
Ref. [157]	0.240	2.261	1.411	Ref. [157]	0.281	5.990	1.753
XCTS.R17	0.236	2.261	1.410	XCTS.R17	0.279	5.992	1.760

To confirm the accuracy of FUKA in solving for the extreme NS configurations that are bar-mode unstable, we first reproduce known stable and unstable models, taken from Ref. [157]. Namely, we consider the S6 and U13 models from Ref. [157], and list their properties in Table 6.1. We also show the properties of the models obtained using FUKA, where we use 17 spectral collocation points for the initial data. For the EoS, as described by Eq. (6.1), we use $K = 100$ and $\Gamma = 2$. We observe good agreement between these sets of models.

We note that the models simulated in this Chapter have rotation rates well beyond the break-up limit, where centrifugal forces surpass the gravitational forces at the surface of the star. However, these NSs are differentially rotating, meaning that the angular velocity is a function of radius. The particular rotation profile we adopt in this study is the Komatsu-Eriguchi-Hachisu (KEH) differential law [166], also known as a constant specific angular momentum law, of the form [167]

$$A^2\left(\Omega_c - \Omega\right) = \frac{\left(\Omega - \omega\right) r^2 \sin^2\theta e^{2(\beta-\nu)}}{1 - \left(\Omega - \omega\right)^2 r^2 \sin^2\theta e^{2(\beta-\nu)}}, \tag{6.4}$$

where A is a parameter representing the degree of differential rotation, with $A \to \infty$ representing uniform rotation. Ω is the angular rotation rate, with Ω_c being the rotation rate at

the center of the star, r is the coordinate radius, and ω, ν are metric functions.

Next, we evolve these models to confirm that they are indeed stable (S6) or unstable (U13). We use the IllinoisGRMHD code, housed within the Einstein Toolkit, to evolve the models for about 30ms. We use the version of the code presented in Ref. [26], for models constructed with a polytropic EoS. We use the same EoS parameters as the initial data models. However, when evolving matter IllinoisGRMHD uses a hybrid EoS of the form

$$P = K\rho^{\Gamma} + (\Gamma_{\rm th} - 1)\,\rho\epsilon_{\rm th}, \tag{6.5}$$

where $\epsilon_{\rm th}$ is the thermal contribution to the internal energy originating from shocks. In the tests presented here we set $\Gamma_{\rm th} = \Gamma$.

For each simulation we use a constant density atmosphere $\rho_{\rm atm} = 1.28 \times 10^{-12}$ in geometrized units. We also do not include magnetic field evolution, similar to the studies presented in [158, 168]. We employ two resolutions in order to confirm convergence of the initial data models in an evolution scheme. For our medium resolution, we use only two refinement levels centered on the NSs, with the coarse grid having a resolution $dx = 0.8$ with side-lengths 204.8 in all three directions. In our evolution of these models we do not impose any symmetries, and only allow truncation errors to initiate the bar-mode instability. Previous studies, such as those done in [158, 168], introduce a small amplitude perturbation to the initial data to induce the instability. However, these and previous studies have found that the amplitude of the perturbation only affects the time until the onset of the instability. Therefore, depending on the spin profile and rate of rotating NS, the instability may occur in nature, as proto-NSs formed from core-collapse supernovae may experience perturbations from in-falling matter. For our high resolution simulations we decrease the grid spacing by 25%. We use the Baikal thorn within the Einstein Toolkit to evolve the spacetime. Baikal employs the BSSN formulation [116–118], and we use fourth order finite differencing, with fourth-order Runge-Kutta time-stepping to evolve the coupled matter-spacetime system, as implemented in the Einstein Toolkit, using a CFL factor of 0.2. For the spacetime evolution we also employ fifth order Kreiss-Oliger dissipation [139], with strength 0.2, and use 1+log lapse evolution [14, 137] and a second-order, non-covariant advecting shift evolution [138]. For

outer boundary conditions we use outgoing radiation boundary conditions using the NewRad thorn, while IllinoisGRMHD enforces outflow boundary conditions on the fluid velocity.

At $t = 41M_{\odot}$ of evolution time, we plot the Hamiltonian constraint, computed using Eqs. (46-47) from Ref. [29], shown in Fig. 6.1 for both models at both resolutions. We rescale the constraint violations associated with the lower resolution simulations assuming second order convergence. We see that these curves overlap well with the curves representing the high-resolution runs, indicating that the resolutions used here are in the convergent regime. We expect our numerical scheme to be between first and fourth order accurate, given that our WENO5Z reconstruction algorithm reduces to first-order accuracy at shocks, while we also employ fourth-order accurate finite-differencing in the spacetime evolution, and fourth-order accurate time stepping.

During evolution we monitor the azimuthal Fourier modes using

$$P_m = \int_V dv \rho e^{im\phi}, \tag{6.6}$$

where dv is the infinitesimal volume element. The bar-mode instability is characterized by an exponential increase in the $m = 2$ mode, after which the mode decrease over time as the NS migrates to a more stable configuration. We measure P_m out to a radius of $r = 45$ in geometrized units, and plot the time evolution of each mode up to $m = 4$ for both models in Fig. 6.2, showing only the results from the high-resolution simulations. In the left panel we plot the results for S6, and observe that while each mode is excited during the evolution, none of them increase exponentially over any given time. This is in contrast to U13, shown in the right of Fig. 6.2, where we observe an exponential increase in the $m = 2$ and $m = 4$ modes.

To further differentiate between the evolution of both models we also show three representative snapshots of two-dimensional color-maps of the baryon density, presented in Fig. 6.3, with S6 shown in the left panels and U13 in the right panels. While S6 shows some dynamics, it maintains a similar density profile throughout its evolution. U13, however, shows that it indeed is deformed into a bar-like configuration (middle right panel) after approximately 14 ms of evolution, before settling down to a more stable configuration with an

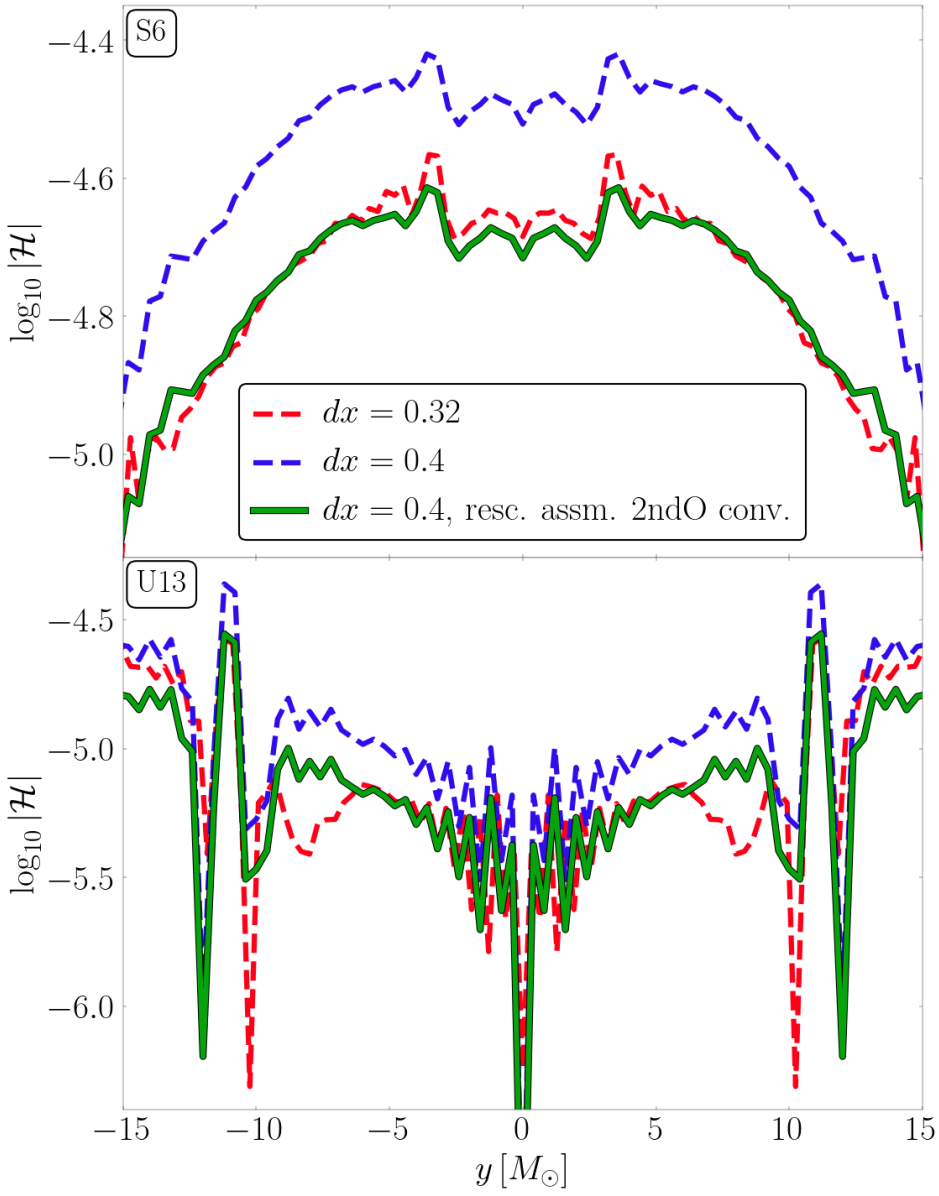

FIG. 6.1. Curves of the Hamiltonian constraint along the y-axis at $t = 41M_\odot$, at two different spatial resolutions. We also show the curve for the lower resolution simulation rescaled assuming second error convergence. We show these convergence plots for the S6 model, (**top**), and the U13 model (**bottom**).

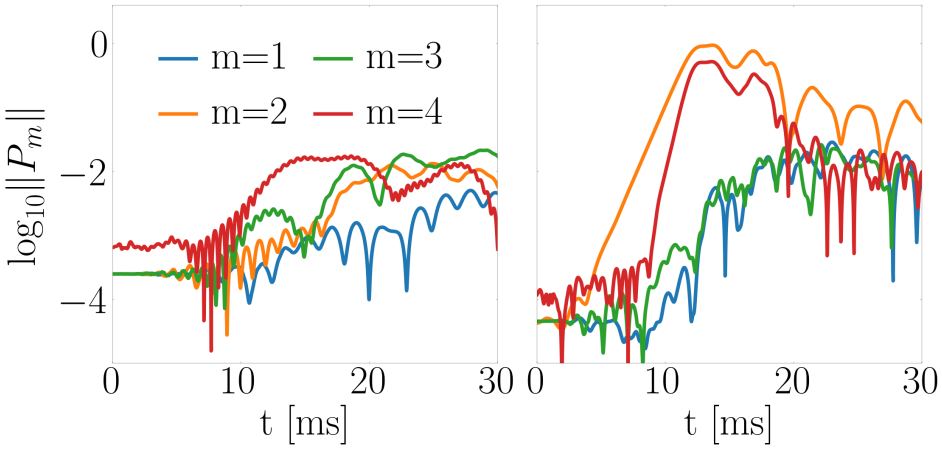

FIG. 6.2. Time evolution of the azimuthal Fourier modes, computed using Eq. (6.6), for models S6 (**left**) and U13 (**right**). We see that U13 is indeed bar-mode unstable, given the exponential increase in the $m = 2$ mode.

accretion disk (bottom right panel) at $t = 28$ ms. We therefore conclude that FUKA indeed faithfully solves for bar-mode unstable models, and we observe the transient bar feature during evolution.

6.4 Constraining the Instability Threshold

Now that we are confident FUKA can accurately solve for bar-mode unstable initial data, we move forward in our quest to constrain β_c for two different realistic, finite-temperature EoSs. Here we focus on cold initial data sets ($T = 0.01$ MeV $= 10^5$ K), comparing the SLy [130] and the DD2 [169] EoSs, with DD2 being the stiffer EoS. By stiffer, we mean that DD2 is able to support more mass for a non-rotating NS. As a result, since SLy is softer, initial data solutions with this EoS have a smaller radius compared to those constructed with DD2.

In this study we fix the gravitational mass of each NS to $M = 1.6M_\odot$, as well as the $A = 1$ parameter in Eq. (6.4), and vary $s = r_p/r_e$, the ratio between the polar and equatorial radii. Thus, as we decrease s below 1, the configurations become more extreme and quasi-toroidal, and β increases in turn. We first evolve models with s within the range $[0.2, 0.3]$. We do so using the latest version of GRoovy as described in previous sections in this Chapter, adopting cylindrical coordinates at resolution $(N_\rho, N_\phi, N_z) = (144, 96, 144)$,

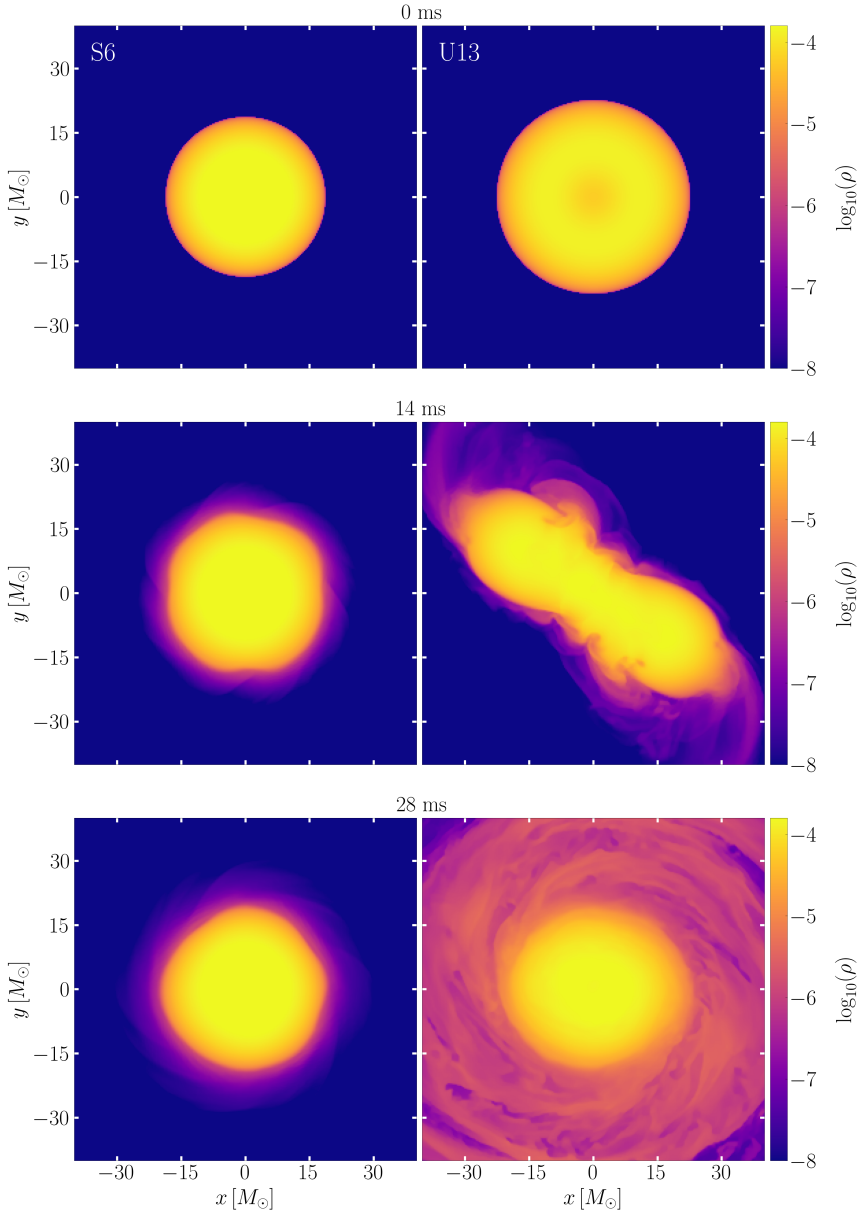

FIG. 6.3. Snapshots of the dynamical evolution of models S6 (**left** panels) and U13 (**right** panels). Shown are two-dimensional cuts along the $z = 0$ plane, plotting color-maps of the baryon density at $t = 0$ ms (**top** panels), $t = 14$ ms (**middle** panels), and $t = 28$ ms (**bottom** panels). U13 becomes bar-mode unstable at approximately 14 ms, then relaxes to a stable configuration afterwards.

and set the outer boundary at $\rho_{\max} = z_{\max} = 64M_{\odot}$. For both EoSs, we confirm that models with $s = 0.3$ are not bar-mode unstable, supported by the absence of an exponential increase in the $m = 2$ azimuthal Fourier mode, seen in the gray curves in Fig. 6.4, for the SLy (left panel) and DD2 (right panel) EoSs. This is in contrast with the $s = 0.2$ models represented by the pink curves, which show an obvious exponential increase in the $m = 2$ mode.

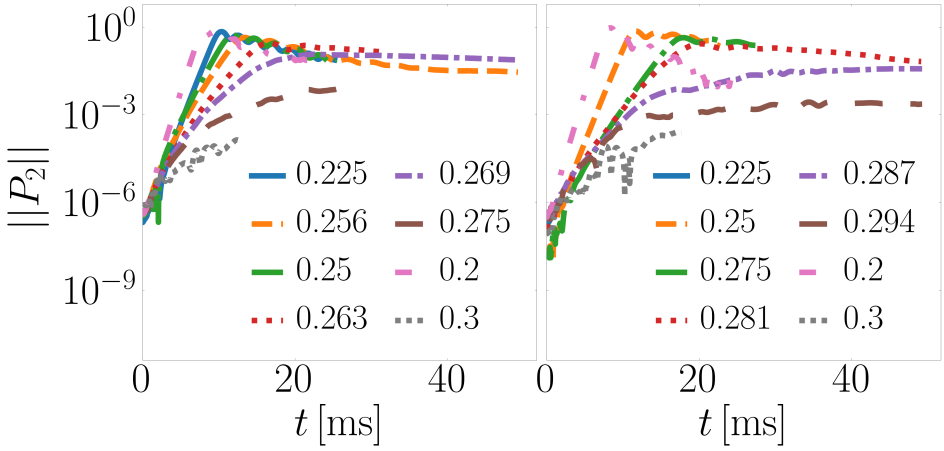

FIG. 6.4. Time evolution of the $m = 2$ azimuthal Fourier mode, for models with different values of the ratio between the polar and equatorial radii, $s = r_p/r_e$. **Left**: SLy EoS, **right**: DD2 EoS. Some simulations were terminated once it was clear the model was stable or unstable to the bar-mode instability.

Following our confirmation that β_c lies in this range, we then conduct a "tri-section" search, in which we successively bracket β_c using three intervals, and evolve the corresponding models. Once it is obvious that a model is stable or unstable, we stop the simulation. The result of the first two iterations of our tri-section is shown in remaining curves in both panels of Fig. 6.4. This is an ongoing study, in which we plan to do another iteration of tri-section. However, we can already observe from Fig. 6.4 that while $\beta_c < 0.263$ for the softer SLy EoS, for the stiffer DD2 EoS we have $\beta_c < 0.281$.

We also note that all models presented in this Chapter are cold models. In the future we plan to also include initially hot models, with central temperatures of about 30 MeV in our study, to gauge how realistic temperatures might affect β_c. Given that Ref. [133] demonstrated that different temperature profiles can affect the pressure profiles of NSs, as well as the maximum mass for non-rotating models, we expect β_c to have some non-negligible dependence on temperature profile. Further, given that newly formed hot proto-NSs, formed

from core-collapse supernovae, accretion-induced collapse of white dwarfs, or binary white dwarf mergers, will radiate neutrinos and cool down, their spin rates may increase as they contract, perhaps becoming susceptible to the bar-mode instability. It is therefore important to constrain β_c for hot NS models as well, so that we may determine the probability of observing bar-mode unstable NS in nature, whether in gravitational waves or electromagnetic signatures.

Chapter 7

Conclusion and Future Work

In this dissertation we have motivated the need for accurate and self-consistent modeling of relativistic fluids using numerical relativity. We began with an analysis of Maxwell's equations in vacuum, and drew parallels to the system of equations used to evolve Einstein's equations for dynamical spacetimes. We also described the mathematical foundation behind the Sommerfeld radiation boundary conditions we have implemented, which aim to minimize un-physical reflections from numerical boundaries when solving hyperbolic partial differential equations. We then described the different components that make up a general relativistic hydrodynamics code, motivating the formulation of the equations through a finite volume method, the methods used to approximate fluxes in such a method, and popular choices for equations of state. We then derived the full GRHD equations using a reference metric formulation, allowing us to evolve them in any singular, curvilinear coordinate system. We used these results with our understanding of GRHD codes to develop GRoovy, a GRHD code that can evolve fluids in dynamical spacetimes. We presented a wide battery of tests for GRoovy, modeling isolated non-rotating and rotating neutron stars in both fixed and dynamical spacetimes. We also modeled neutron stars using realistic, finite-temperature equations of state, as well as neutrino emission tests of optically thin and thick gases, and modeled the neutrino luminosity from a hot neutron star.

As we presented in the previous Chapter , current work is being done to model the bar-mode instability in rapidly rotating neutron stars. We validated the accuracy of the FUKA initial data code when generating bar-mode unstable neutron star initial data, showing excellent agreement with previous studies. We also presented ongoing work to constrain the bar-mode instability threshold for realistic equations of state, showing that there is a difference in the threshold between soft and stiff equations of state. We were able to obtain these preliminary results due to the various improvements to the GRoovy code, including

the development of a robust primitive recovery strategy, and inclusion of a higher-order reconstruction scheme that is appropriate for curvilinear coordinates.

Once our study on the instability threshold is concluded, we will then study its dependence on initial temperature profiles in the neutron star. Up until now, studies of the instability have only considered initially cold models, yet it is still unclear if proto-neutron stars born from core-collapse supernova are unstable to this instability. Given that these stars are fairly hot, with core temperatures of ≈ 30MeV [167], there needs to be more modeling of hot, rapidly rotating neutron stars. We also plan to study how neutrino emission may affect the onset and lifetime of the instability, and make predictions for the potentially observable neutrino luminosity.

Finally, we plan to show how the GW emission from these systems differ between soft and stiff equations of state, presenting another avenue in which the dense-matter equation of state may be constrained using GW observations from Galactic sources. Specifically, our we will extract GWs from our simulations, comparing them across EoSs and between initially cold and hot NS models. Previous studies have shown that GWs arising from the rapidly changing quadrupole moment of the deformed NS lay in the kHz regime. Our results will therefore be important for future ground based observatories, such as Cosmic Explorer and the Einstein Telescope. Conversely, our results may also show that any differences across EoS may ultimately be negligible. Nevertheless, such a study as yet to be done, thus warranting our current endeavor.

In this dissertation we have not only presented a rigorously-tested GRHD code that can be used to model matter configurations in strong gravity, but we also presented early results for the code's first application. Looking ahead, there are many improvements that we can make to GRoovy to improve its ability to more accurately model relativistic systems. One such improvement includes doing away with the second-order approximation in our finite-volume method, and implementing a fourth-order accurate scheme. This would entail faithfully computing cell-averages and keeping track of cell-averaged versus cell-centered quantities. Doing so would allow us to save considerably on computational costs, as higher accuracy corresponds to less required spatial resolution in our simulations. Another improvement would be including magnetic field evolution into the GRoovy code. In doing so,

we could then model magnetized fluids, and use the code to study magnetically driven winds and jets in systems such as binary neutron star merger remnants or black hole — accretion disk systems. Finally, we could also include more accurate treatments of neutrino emission and absorption, further increasing the realism of our simulations of some of the most energetic systems in the Universe.

Bibliography

[1] B. P. Abbott et al. "Multi-messenger Observations of a Binary Neutron Star Merger". ApJ 848.2, L12 (2017), p. L12. DOI: 10.3847/2041-8213/aa91c9. arXiv: 1710.05833 [astro-ph.HE].

[2] B. P. Abbott et al. "GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral". *Phys. Rev. Lett.* 119 (16 2017), p. 161101. DOI: 10.1103/PhysRevLett.119.161101. URL: https://link.aps.org/doi/10.1103/PhysRevLett.119.161101.

[3] V. Savchenko et al. "INTEGRAL Detection of the First Prompt Gamma-Ray Signal Coincident with the Gravitational-wave Event GW170817". ApJ 848.2, L15 (2017), p. L15. DOI: 10.3847/2041-8213/aa8f94. arXiv: 1710.05449 [astro-ph.HE].

[4] B. D. Metzger. "Kilonovae". *Living Reviews in Relativity* 23.1, 1 (2019), p. 1. DOI: 10.1007/s41114-019-0024-0. arXiv: 1910.01617 [astro-ph.HE].

[5] N. Sarin and P. D. Lasky. "The evolution of binary neutron star post-merger remnants: a review". *General Relativity and Gravitation* 53.6, 59 (2021), p. 59. DOI: 10.1007/s10714-021-02831-1. arXiv: 2012.08172 [astro-ph.HE].

[6] P. Mösta et al. "A Magnetar Engine for Short GRBs and Kilonovae". ApJ 901.2, L37 (2020), p. L37. DOI: 10.3847/2041-8213/abb6ef. arXiv: 2003.06043 [astro-ph.HE].

[7] D. Radice et al. "GW170817: Joint Constraint on the Neutron Star Equation of State from Multimessenger Observations". ApJ 852.2, L29 (2018), p. L29. DOI: 10.3847/2041-8213/aaa402. arXiv: 1711.03647 [astro-ph.HE].

[8] K. Takami, L. Rezzolla, and L. Baiotti. "Constraining the Equation of State of Neutron Stars from Binary Mergers". Phys. Rev. Lett. 113.9, 091104 (2014), p. 091104. DOI: 10.1103/PhysRevLett.113.091104. arXiv: 1403.5672 [gr-qc].

[9] W. H. Press, B. P. Flannery, and S. A. Teukolsky. *Numerical recipes. The art of scientific computing*. 1986.

[10] N. Afshordi et al. "Waveform Modelling for the Laser Interferometer Space Antenna". *arXiv e-prints*, arXiv:2311.01300 (2023), arXiv:2311.01300. DOI: 10.48550/arXiv.2311.01300. arXiv: 2311.01300 [gr-qc].

[11] T. W. Baumgarte and S. L. Shapiro. *Numerical Relativity: Solving Einstein's Equations on the Computer*. Cambridge University Press, 2010.

[12] A. M. Knapp, E. J. Walker, and T. W. Baumgarte. "Illustrating stability properties of numerical relativity in electrodynamics". Phys. Rev. D 65.6, 064031 (2002), p. 064031. DOI: 10.1103/PhysRevD.65.064031. arXiv: gr-qc/0201051 [gr-qc].

[13] L. Werneck et al. *The Einstein Toolkit*. Version The "Karl Schwarzschild" release, ET_2023_05. To find out more, visit http://einsteintoolkit.org. 2023. DOI: 10.5281/zenodo.7942541. URL: https://doi.org/10.5281/zenodo.7942541.

[14] J. G. Baker et al. "Gravitational-Wave Extraction from an Inspiraling Configuration of Merging Black Holes". *Phys. Rev. Lett.* 96 (11 2006), p. 111102. DOI: 10.1103/PhysRevLett.96.111102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.96.111102.

[15] J. G. Baker et al. "Binary black hole merger dynamics and waveforms". *Phys. Rev. D* 73 (10 2006), p. 104002. DOI: 10.1103/PhysRevD.73.104002. URL: https://link.aps.org/doi/10.1103/PhysRevD.73.104002.

[16] Y. Zlochower et al. "Accurate black hole evolutions by fourth-order numerical relativity". *Phys. Rev. D* 72 (2 2005), p. 024021. DOI: 10.1103/PhysRevD.72.024021. URL: https://link.aps.org/doi/10.1103/PhysRevD.72.024021.

[17] Z. B. Etienne et al. "General relativistic simulations of black-hole–neutron-star mergers: Effects of black-hole spin". *Phys. Rev. D* 79 (4 2009), p. 044024. DOI: 10.1103/PhysRevD.79.044024. URL: https://link.aps.org/doi/10.1103/PhysRevD.79.044024.

[18] Z. B. Etienne et al. "Improved moving puncture gauge conditions for compact binary evolutions". *Phys. Rev. D* 90 (6 2014), p. 064032. DOI: 10.1103/PhysRevD.90.064032. URL: https://link.aps.org/doi/10.1103/PhysRevD.90.064032.

[19] B. D. Farris et al. "Binary Black-Hole Mergers in Magnetized Disks: Simulations in Full General Relativity". *Phys. Rev. Lett.* 109 (22 2012), p. 221102. DOI: 10.1103/PhysRevLett.109.221102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.109.221102.

[20] T. W. Baumgarte, S. A. Hughes, and S. L. Shapiro. "Evolving Einstein's field equations with matter: The "hydro without hydro" test". *Phys. Rev. D* 60 (8 1999), p. 087501. DOI: 10.1103/PhysRevD.60.087501. URL: https://link.aps.org/doi/10.1103/PhysRevD.60.087501.

[21] C. F. Gammie, J. C. McKinney, and G. Tóth. "HARM: A Numerical Scheme for General Relativistic Magnetohydrodynamics". *The Astrophysical Journal* 589.1 (2003), p. 444. DOI: 10.1086/374594. URL: https://dx.doi.org/10.1086/374594.

[22] A. Murguia-Berthier et al. "HARM3D+NUC: A New Method for Simulating the Post-merger Phase of Binary Neutron Star Mergers with GRMHD, Tabulated EOS, and Neutrino Leakage". *The Astrophysical Journal* 919.2 (2021), p. 95. DOI: 10.3847/1538-4357/ac1119. URL: https://dx.doi.org/10.3847/1538-4357/ac1119.

[23] M. T. P. Liska et al. "H-AMR: A New GPU-accelerated GRMHD Code for Exascale Computing with 3D Adaptive Mesh Refinement and Local Adaptive Time Stepping". *Astrophys. J. Supp. Ser.* 263.2, 26 (2022), p. 26. DOI: 10.3847/1538-4365/ac9966. arXiv: 1912.10192 [astro-ph.HE].

[24] O. Gottlieb et al. "Large-scale Evolution of Seconds-long Relativistic Jets from Black Hole-Neutron Star Mergers". *Astrophys. J. Lett.* 954.1, L21 (2023), p. L21. DOI: 10.3847/2041-8213/aceeff. arXiv: 2306.14947 [astro-ph.HE].

[25] O. Gottlieb et al. "A Unified Picture of Short and Long Gamma-Ray Bursts from Compact Binary Mergers". *Astrophys. J. Lett.* 958.2, L33 (2023), p. L33. DOI: 10.3847/2041-8213/ad096e. arXiv: 2309.00038 [astro-ph.HE].

[26] Z. B. Etienne et al. "IllinoisGRMHD: an open-source, user-friendly GRMHD code for dynamical spacetimes". *Classical and Quantum Gravity* 32.17 (2015), p. 175009. DOI: 10.1088/0264-9381/32/17/175009. URL: https://dx.doi.org/10.1088/0264-9381/32/17/175009.

[27] L. R. Werneck et al. "Addition of tabulated equation of state and neutrino leakage support to illinoisgrmhd". *Phys. Rev. D* 107.4 (2023), p. 044037. DOI: 10.1103/PhysRevD.107.044037. arXiv: 2208.14487 [gr-qc].

[28] G. Tóth. "The ∇· B=0 Constraint in Shock-Capturing Magnetohydrodynamics Codes". *Journal of Computational Physics* 161.2 (2000), pp. 605–652. DOI: 10.1006/jcph.2000.6519.

[29] I. Ruchlin, Z. B. Etienne, and T. W. Baumgarte. "SENR /NRPy + : Numerical relativity in singular curvilinear coordinate systems". Phys. Rev. D 97.6, 064036 (2018), p. 064036. DOI: 10.1103/PhysRevD.97.064036. arXiv: 1712.07658 [gr-qc].

[30] A. Tchekhovskoy, J. C. McKinney, and R. Narayan. "WHAM: a WENO-based general relativistic numerical scheme - I. Hydrodynamics". MNRAS 379.2 (2007), pp. 469–497. DOI: 10.1111/j.1365-2966.2007.11876.x. arXiv: 0704.2608 [astro-ph].

[31] R. Monchmeyer and E. Muller. "A Conservative Second-Order Difference Scheme for Curvilinear Coordinates - Part One - Assignment of Variables on a Staggered Grid". *aap* 217 (1989), p. 351.

[32] A. Mignone. "High-order conservative reconstruction schemes for finite volume methods in cylindrical and spherical coordinates". *Journal of Computational Physics* 270 (2014), pp. 784–814. DOI: 10.1016/j.jcp.2014.04.001. arXiv: 1404.0537 [physics.comp-ph].

[33] A. Harten, P. D. Lax, and B. v. Leer. "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws". *SIAM Review* 25.1 (1983), pp. 35–61. DOI: 10.1137/1025002. eprint: https://doi.org/10.1137/1025002. URL: https://doi.org/10.1137/1025002.

[34] E. F. Toro, M. Spruce, and W. Speares. "Restoration of the contact surface in the HLL-Riemann solver". *Shock Waves* 4.1 (1994), pp. 25–34. DOI: 10.1007/BF01414629.

[35] D. S. Balsara and J. Kim. "A subluminal relativistic magnetohydrodynamics scheme with ADER-WENO predictor and multidimensional Riemann solver-based corrector". *Journal of Computational Physics* 312 (2016), pp. 357–384. DOI: 10.1016/j.jcp.2016.02.001.

[36] V. Mewes et al. "Numerical relativity in spherical coordinates: A new dynamical spacetime and general relativistic MHD evolution framework for the Einstein Toolkit". *Phys. Rev. D* 101 (10 2020), p. 104007. DOI: 10.1103/PhysRevD.101.104007. URL: https://link.aps.org/doi/10.1103/PhysRevD.101.104007.

[37] P. Mösta et al. "GRHydro: a new open-source general-relativistic magnetohydrodynamics code for the Einstein toolkit". *Classical and Quantum Gravity* 31.1 (2013), p. 015005. DOI: 10.1088/0264-9381/31/1/015005. URL: https://dx.doi.org/10.1088/0264-9381/31/1/015005.

[38] S. C. Noble et al. "Primitive Variable Solvers for Conservative General Relativistic Magnetohydrodynamics". *The Astrophysical Journal* 641.1 (2006), p. 626. DOI: 10.1086/500349. URL: https://dx.doi.org/10.1086/500349.

[39] S. C. Noble, J. H. Krolik, and J. F. Hawley. "Direct Calculation of the Radiative Efficiency of an Accretion Disk Around a Black Hole". *The Astrophysical Journal* 692.1 (2009), p. 411. DOI: 10.1088/0004-637X/692/1/411. URL: https://dx.doi.org/10.1088/0004-637X/692/1/411.

[40] J. A. Font et al. "Three-dimensional numerical general relativistic hydrodynamics: Formulations, methods, and code tests". *Phys. Rev. D* 61 (4 2000), p. 044011. DOI: 10.1103/PhysRevD.61.044011. URL: https://link.aps.org/doi/10.1103/PhysRevD.61.044011.

[41] Z. B. Etienne et al. "General relativistic simulations of black-hole–neutron-star mergers: Effects of magnetic fields". *Phys. Rev. D* 85 (6 2012), p. 064029. DOI: 10.1103/PhysRevD.85.064029. URL: https://link.aps.org/doi/10.1103/PhysRevD.85.064029.

[42] L. Ennoggi et al. “Relativistic gas accretion onto supermassive black hole binaries from inspiral through merger”. Phys. Rev. D 112.6, 063009 (2025), p. 063009. DOI: 10.1103/yc25-v1q4. arXiv: 2502.06389 [astro-ph.HE].

[43] P. J. Montero, T. W. Baumgarte, and E. Müller. “General relativistic hydrodynamics in curvilinear coordinates”. Phys. Rev. D 89.8, 084043 (2014), p. 084043. DOI: 10.1103/PhysRevD.89.084043. arXiv: 1309.7808 [gr-qc].

[44] T. Pierre Jacques et al. “General relativistic hydrodynamics code for dynamical spacetimes with curvilinear coordinates, tabulated equations of state, and neutrino physics”. *Phys. Rev. D* 112 (8 2025), p. 084044. DOI: 10.1103/hc9l-1thx. URL: https://link.aps.org/doi/10.1103/hc9l-1thx.

[45] J. A. Font. “Numerical Hydrodynamics and Magnetohydrodynamics in General Relativity”. *Living Rev. Rel.* 11 (2008), p. 7. DOI: 10.12942/lrr-2008-7.

[46] B. P. Abbott et al. “Multi-messenger Observations of a Binary Neutron Star Merger”. *Astrophys. J. Lett.* 848.2, L12 (2017), p. L12. DOI: 10.3847/2041-8213/aa91c9. arXiv: 1710.05833 [astro-ph.HE].

[47] B. P. Abbott et al. “Gravitational Waves and Gamma-Rays from a Binary Neutron Star Merger: GW170817 and GRB 170817A”. *Astrophys. J. Lett.* 848.2, L13 (2017), p. L13. DOI: 10.3847/2041-8213/aa920c. arXiv: 1710.05834 [astro-ph.HE].

[48] E. Troja et al. “The X-ray counterpart to the gravitational-wave event GW170817”. *Nature (London)* 551.7678 (2017), pp. 71–74. DOI: 10.1038/nature24290. arXiv: 1710.05433 [astro-ph.HE].

[49] E. Troja et al. “The outflow structure of GW170817 from late-time broad-band observations”. *Mon. Not. R. Astron. Soc.* 478.1 (2018), pp. L18–L23. DOI: 10.1093/mnrasl/sly061. arXiv: 1801.06516 [astro-ph.HE].

[50] J. J. Ruan et al. “Brightening X-Ray Emission from GW170817/GRB 170817A: Further Evidence for an Outflow”. *Astrophys. J. Lett.* 853.1, L4 (2018), p. L4. DOI: 10.3847/2041-8213/aaa4f3. arXiv: 1712.02809 [astro-ph.HE].

[51] G. P. Lamb, I. Mandel, and L. Resmi. "Late-time evolution of afterglows from off-axis neutron star mergers". *Mon. Not. R. Astron. Soc.* 481.2 (2018), pp. 2581–2589. DOI: 10.1093/mnras/sty2196. arXiv: 1806.03843 [astro-ph.HE].

[52] G. P. Lamb et al. "The Optical Afterglow of GW170817 at One Year Post-merger". *Astrophys. J. Lett.* 870.2, L15 (2019), p. L15. DOI: 10.3847/2041-8213/aaf96b. arXiv: 1811.11491 [astro-ph.HE].

[53] B. D. Metzger et al. "Electromagnetic counterparts of compact object mergers powered by the radioactive decay of r-process nuclei". *Mon. Not. R. Astron. Soc.* 406.4 (2010), pp. 2650–2662. DOI: 10.1111/j.1365-2966.2010.16864.x. arXiv: 1001.5029 [astro-ph.HE].

[54] B. D. Metzger. "Kilonovae". *Living Reviews in Relativity* 23.1, 1 (2019), p. 1. DOI: 10.1007/s41114-019-0024-0. arXiv: 1910.01617 [astro-ph.HE].

[55] M. Ruiz et al. "Binary Neutron Star Mergers: A Jet Engine for Short Gamma-Ray Bursts". *Astrophys. J. Lett.* 824.1, L6 (2016), p. L6. DOI: 10.3847/2041-8205/824/1/L6. arXiv: 1604.02455 [astro-ph.HE].

[56] P. Mösta et al. "A Magnetar Engine for Short GRBs and Kilonovae". *Astrophys. J. Lett.* 901.2, L37 (2020), p. L37. DOI: 10.3847/2041-8213/abb6ef. arXiv: 2003.06043 [astro-ph.HE].

[57] R. Ciolfi. "Collimated outflows from long-lived binary neutron star merger remnants". *Mon. Not. R. Astron. Soc.* 495.1 (2020), pp. L66–L70. DOI: 10.1093/mnrasl/slaa062. arXiv: 2001.10241 [astro-ph.HE].

[58] L. Sun et al. "Jet launching from binary neutron star mergers: Incorporating neutrino transport and magnetic fields". *Phys. Rev. D* 105 (10 2022), p. 104028. DOI: 10.1103/PhysRevD.105.104028. URL: https://link.aps.org/doi/10.1103/PhysRevD.105.104028.

[59] K. Kiuchi et al. "Self-Consistent Picture of the Mass Ejection from a One Second Long Binary Neutron Star Merger Leaving a Short-Lived Remnant in a General-Relativistic Neutrino-Radiation Magnetohydrodynamic Simulation". Phys. Rev. Lett.

131.1, 011401 (2023), p. 011401. DOI: 10.1103/PhysRevLett.131.011401. arXiv: 2211.07637 [astro-ph.HE].

[60] K. Takami, L. Rezzolla, and L. Baiotti. "Constraining the Equation of State of Neutron Stars from Binary Mergers". *Phys. Rev. Lett.* 113 (9 2014), p. 091104. DOI: 10.1103/PhysRevLett.113.091104. URL: https://link.aps.org/doi/10.1103/PhysRevLett.113.091104.

[61] D. Radice et al. "GW170817: Joint Constraint on the Neutron Star Equation of State from Multimessenger Observations". *Astrophys. J. Lett.* 852.2, L29 (2018), p. L29. DOI: 10.3847/2041-8213/aaa402. arXiv: 1711.03647 [astro-ph.HE].

[62] J. J. Cowan et al. "Origin of the heaviest elements: The rapid neutron-capture process". *Rev. Mod. Phys.* 93 (1 2021), p. 015002. DOI: 10.1103/RevModPhys.93.015002. URL: https://link.aps.org/doi/10.1103/RevModPhys.93.015002.

[63] D. M. Siegel and B. D. Metzger. "Three-Dimensional General-Relativistic Magnetohydrodynamic Simulations of Remnant Accretion Disks from Neutron Star Mergers: Outflows and r -Process Nucleosynthesis". Phys. Rev. Lett. 119.23, 231102 (2017), p. 231102. DOI: 10.1103/PhysRevLett.119.231102. arXiv: 1705.05473 [astro-ph.HE].

[64] L. Combi and D. M. Siegel. "GRMHD Simulations of Neutron-star Mergers with Weak Interactions: r-process Nucleosynthesis and Electromagnetic Signatures of Dynamical Ejecta". ApJ 944.1, 28 (2023), p. 28. DOI: 10.3847/1538-4357/acac29. arXiv: 2206.03618 [astro-ph.HE].

[65] B. P. Abbott et al. "GW190425: Observation of a Compact Binary Coalescence with Total Mass $\sim 3.4\ M_\odot$". *Astrophys. J. Lett.* 892.1, L3 (2020), p. L3. DOI: 10.3847/2041-8213/ab75f5. arXiv: 2001.01761 [astro-ph.HE].

[66] R. Abbott et al. "Observation of Gravitational Waves from Two Neutron Star-Black Hole Coalescences". *Astrophys. J. Lett.* 915.1, L5 (2021), p. L5. DOI: 10.3847/2041-8213/ac082e. arXiv: 2106.15163 [astro-ph.HE].

[67] R. Abbott et al. "GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo during the Second Part of the Third Observing Run". *Phys. Rev. X* 13 (4 2023), p. 041039. DOI: 10.1103/PhysRevX.13.041039. URL: https://link.aps.org/doi/10.1103/PhysRevX.13.041039.

[68] O. Porth et al. "The black hole accretion code". *Computational Astrophysics and Cosmology* 4.1, 1 (2017), p. 1. DOI: 10.1186/s40668-017-0020-2. arXiv: 1611.09720 [gr-qc].

[69] P. C.-K. Cheong et al. "Gmunu: paralleled, grid-adaptive, general-relativistic magnetohydrodynamics in curvilinear geometries in dynamical space-times". *Mon. Not. R. Astron. Soc.* 508.2 (2021), pp. 2279–2301. DOI: 10.1093/mnras/stab2606. arXiv: 2012.07322 [astro-ph.IM].

[70] V. Mewes et al. "Numerical relativity in spherical coordinates with the Einstein Toolkit". *Phys. Rev. D* 97 (8 2018), p. 084059. DOI: 10.1103/PhysRevD.97.084059. URL: https://link.aps.org/doi/10.1103/PhysRevD.97.084059.

[71] H. H.-Y. Ng et al. "Hybrid approach to long-term binary neutron-star simulations". *Phys. Rev. D* 109.6 (2024), p. 064061. DOI: 10.1103/PhysRevD.109.064061. arXiv: 2312.11358 [gr-qc].

[72] J. M. Stone et al. "Athena: A New Code for Astrophysical MHD". *apjs* 178.1 (2008), pp. 137–177. DOI: 10.1086/588755. arXiv: 0804.0402 [astro-ph].

[73] J. M. Stone et al. "The Athena++ Adaptive Mesh Refinement Framework: Design and Magnetohydrodynamic Solvers". *The Astrophysical Journal Supplement Series* 249.1 (2020), p. 4. DOI: 10.3847/1538-4365/ab929b. URL: https://dx.doi.org/10.3847/1538-4365/ab929b.

[74] L. Del Zanna et al. "ECHO: a Eulerian conservative high-order scheme for general relativistic magnetohydrodynamics and magnetodynamics". *Astron. Astrophys.* 473.1 (2007), pp. 11–30. DOI: 10.1051/0004-6361:20077093. arXiv: 0704.3206 [astro-ph].

[75] F. G. Lopez Armengol et al. "Handing off the outcome of binary neutron star mergers for accurate and long-term postmerger simulations". *Phys. Rev. D* 106 (8 2022), p. 083015. DOI: 10.1103/PhysRevD.106.083015. URL: https://link.aps.org/doi/10.1103/PhysRevD.106.083015.

[76] R. Ciolfi and J. V. Kalinani. "Magnetically Driven Baryon Winds from Binary Neutron Star Merger Remnants and the Blue Kilonova of 2017 August". *Astrophys. J. Lett.* 900.2, L35 (2020), p. L35. DOI: 10.3847/2041-8213/abb240. arXiv: 2004.11298 [astro-ph.HE].

[77] L. Combi and D. M. Siegel. "Jets from Neutron-Star Merger Remnants and Massive Blue Kilonovae". *Phys. Rev. Lett.* 131 (23 2023), p. 231402. DOI: 10.1103/PhysRevLett.131.231402. URL: https://link.aps.org/doi/10.1103/PhysRevLett.131.231402.

[78] S. Curtis et al. "Magnetized Outflows from Short-lived Neutron Star Merger Remnants Can Produce a Blue Kilonova". *Astrophys. J. Lett.* 961.1, L26 (2024), p. L26. DOI: 10.3847/2041-8213/ad0fe1. arXiv: 2305.07738 [astro-ph.HE].

[79] E. O'Connor. "An Open-source Neutrino Radiation Hydrodynamics Code for Core-collapse Supernovae". *Astrophys. J. Supp. Ser.* 219.2, 24 (2015), p. 24. DOI: 10.1088/0067-0049/219/2/24. arXiv: 1411.7058 [astro-ph.HE].

[80] E. P. O'Connor and S. M. Couch. "Exploring Fundamentally Three-dimensional Phenomena in High-fidelity Simulations of Core-collapse Supernovae". *Astrophys. J.* 865.2, 81 (2018), p. 81. DOI: 10.3847/1538-4357/aadcf7. arXiv: 1807.07579 [astro-ph.HE].

[81] S. M. Couch and E. P. O'Connor. "High-resolution Three-dimensional Simulations of Core-collapse Supernovae in Multiple Progenitors". *Astrophys. J.* 785.2, 123 (2014), p. 123. DOI: 10.1088/0004-637X/785/2/123. arXiv: 1310.5728 [astro-ph.HE].

[82] A. S. Schneider et al. "Equation of state effects in the core collapse of a 20 -$M_\odot$ star". Phys. Rev. C 100.5, 055802 (2019), p. 055802. DOI: 10.1103/PhysRevC.100.055802. arXiv: 1906.02009 [astro-ph.HE].

[83] P. Mösta et al. "r-process Nucleosynthesis from Three-dimensional Magnetorotational Core-collapse Supernovae". *Astrophys. J.* 864.2, 171 (2018), p. 171. DOI: 10.3847/1538-4357/aad6ec. arXiv: 1712.09370 [astro-ph.HE].

[84] V. Morozova et al. "The Gravitational Wave Signal from Core-collapse Supernovae". *Astrophys. J.* 861.1, 10 (2018), p. 10. DOI: 10.3847/1538-4357/aac5f1. arXiv: 1801.01914 [astro-ph.HE].

[85] A. Burrows, D. Radice, and D. Vartanyan. "Three-dimensional supernova explosion simulations of 9-, 10-, 11-, 12-, and 13-$M_\odot$ stars". *Mon. Not. R. Astron. Soc.* 485.3 (2019), pp. 3153–3168. DOI: 10.1093/mnras/stz543. arXiv: 1902.00547 [astro-ph.SR].

[86] S. Chon, T. Hosokawa, and N. Yoshida. "Radiation hydrodynamics simulations of the formation of direct-collapse supermassive stellar systems". *Mon. Not. R. Astron. Soc.* 475.3 (2018), pp. 4104–4121. DOI: 10.1093/mnras/sty086. arXiv: 1711.05262 [astro-ph.GA].

[87] M. A. Latif et al. "Radiation hydrodynamical simulations of the birth of intermediate-mass black holes in the first galaxies". *Mon. Not. R. Astron. Soc.* 508.2 (2021), pp. 1756–1767. DOI: 10.1093/mnras/stab2708. arXiv: 2012.09177 [astro-ph.GA].

[88] Y. Luo et al. "Direct Collapse to Supermassive Black Hole Seeds with Radiative Transfer: Isolated Halos". *Mon. Not. R. Astron. Soc.* 476.3 (2018), pp. 3523–3539. DOI: 10.1093/mnras/sty362. arXiv: 1801.08545 [astro-ph.GA].

[89] J. Smidt et al. "Radiation Hydrodynamical Simulations of the First Quasars". *Astrophys. J.* 865.2, 126 (2018), p. 126. DOI: 10.3847/1538-4357/aad7b8. arXiv: 1703.00449 [astro-ph.GA].

[90] M. Mościbrodzka et al. "Radiative Models of SGR A* from GRMHD Simulations". *Astrophys. J.* 706.1 (2009), pp. 497–507. DOI: 10.1088/0004-637X/706/1/497. arXiv: 0909.5431 [astro-ph.HE].

[91] M. Mościbrodzka et al. "Observational appearance of inefficient accretion flows and jets in 3D GRMHD simulations: Application to Sagittarius A*". *Astron. Astrophys.* 570, A7 (2014), A7. DOI: 10.1051/0004-6361/201424358. arXiv: 1408.4743 [astro-ph.HE].

[92] R. V. Shcherbakov, R. F. Penna, and J. C. McKinney. "Sagittarius A* Accretion Flow and Black Hole Parameters from General Relativistic Dynamical and Polarized Radiative Modeling". *Astrophys. J.* 755.2, 133 (2012), p. 133. DOI: 10.1088/0004-637X/755/2/133. arXiv: 1007.4832 [astro-ph.HE].

[93] R. Narayan et al. "GRMHD simulations of magnetized advection-dominated accretion on a non-spinning black hole: role of outflows". *Mon. Not. R. Astron. Soc.* 426.4 (2012), pp. 3241–3259. DOI: 10.1111/j.1365-2966.2012.22002.x. arXiv: 1206.1213 [astro-ph.HE].

[94] Z. B. Etienne. *nrpytutorial: The NRPy+ GitHub Repository*. https://github.com/zachetienne/nrpytutorial. 2022.

[95] S. Cupp et al. *GRHayL: General Relativistic Hydrodynamics Library*. https://github.com/GRHayL. 2023. URL: https://github.com/GRHayL.

[96] S. Cupp et al. "GRHayL: a modern, infrastructure-agnostic, extensible library for GRMHD simulations". in preparation. 2024.

[97] M. D. Duez et al. "Relativistic magnetohydrodynamics in dynamical spacetimes: Numerical methods and tests". *Phys. Rev. D* 72 (2 2005), p. 024028. DOI: 10.1103/PhysRevD.72.024028. URL: https://link.aps.org/doi/10.1103/PhysRevD.72.024028.

[98] D. Martin et al. "Neutrino-driven Winds in the Aftermath of a Neutron Star Merger: Nucleosynthesis and Electromagnetic Transients". *Astrophys. J.* 813.1, 2 (2015), p. 2. DOI: 10.1088/0004-637X/813/1/2. arXiv: 1506.05048 [astro-ph.SR].

[99] A. Perego et al. "Neutrino-driven winds from neutron star merger remnants". *Mon. Not. R. Astron. Soc.* 443.4 (2014), pp. 3134–3156. DOI: 10.1093/mnras/stu1352. arXiv: 1405.6730 [astro-ph.HE].

[100] C. Musolino, R. Duqué, and L. Rezzolla. ""Extended Emission" from Fallback Accretion onto Merger Remnants". *Astrophys. J. Lett.* 966.2 (2024), p. L31. DOI: 10.3847/2041-8213/ad3bb3. arXiv: 2402.11009 [astro-ph.HE].

[101] F. Foucart et al. "Estimating outflow masses and velocities in merger simulations: Impact of r-process heating and neutrino cooling". *Phys. Rev. D* 104 (12 2021), p. 123010. DOI: 10.1103/PhysRevD.104.123010. URL: https://link.aps.org/doi/10.1103/PhysRevD.104.123010.

[102] D. Radice and S. Bernuzzi. "Secular Outflows from Long-Lived Neutron Star Merger Remnants". *arXiv e-prints*, arXiv:2310.09934 (2023), arXiv:2310.09934. DOI: 10.48550/arXiv.2310.09934. arXiv: 2310.09934 [astro-ph.HE].

[103] M. Chabanov and L. Rezzolla. "Impact of bulk viscosity on the post-merger gravitational-wave signal from merging neutron stars". *arXiv e-prints*, arXiv:2307.10464 (2023), arXiv:2307.10464. DOI: 10.48550/arXiv.2307.10464. arXiv: 2307.10464 [gr-qc].

[104] M. Chabanov and L. Rezzolla. "Numerical modelling of bulk viscosity in neutron stars". *arXiv e-prints*, arXiv:2311.13027 (2023), arXiv:2311.13027. DOI: 10.48550/arXiv.2311.13027. arXiv: 2311.13027 [gr-qc].

[105] V. Paschalidis, Z. B. Etienne, and S. L. Shapiro. "Importance of cooling in triggering the collapse of hypermassive neutron stars". *Phys. Rev. D* 86 (6 2012), p. 064032. DOI: 10.1103/PhysRevD.86.064032. URL: https://link.aps.org/doi/10.1103/PhysRevD.86.064032.

[106] C. A. Raithel, V. Paschalidis, and F. Özel. "Realistic finite-temperature effects in neutron star merger simulations". *Phys. Rev. D* 104 (6 2021), p. 063016. DOI: 10.1103/PhysRevD.104.063016. URL: https://link.aps.org/doi/10.1103/PhysRevD.104.063016.

[107] P. Hammond, I. Hawke, and N. Andersson. "Thermal aspects of neutron star mergers". *Phys. Rev. D* 104 (10 2021), p. 103006. DOI: 10.1103/PhysRevD.104.103006. URL: https://link.aps.org/doi/10.1103/PhysRevD.104.103006.

[108] J. Fields et al. "Thermal Effects in Binary Neutron Star Mergers". *Astrophys. J. Lett.* 952.2, L36 (2023), p. L36. DOI: 10.3847/2041-8213/ace5b2. arXiv: 2302.11359 [astro-ph.HE].

[109] R. Ciolfi et al. "General relativistic magnetohydrodynamic simulations of binary neutron star mergers forming a long-lived neutron star". Phys. Rev. D 95.6, 063016 (2017), p. 063016. DOI: 10.1103/PhysRevD.95.063016. arXiv: 1701.08738 [astro-ph.HE].

[110] M. Ruiz, S. L. Shapiro, and A. Tsokaros. "GW170817, general relativistic magnetohydrodynamic simulations, and the neutron star maximum mass". Phys. Rev. D 97.2, 021501 (2018), p. 021501. DOI: 10.1103/PhysRevD.97.021501. arXiv: 1711.00473 [astro-ph.HE].

[111] S. D. Tootle et al. "Quasi-universal Behavior of the Threshold Mass in Unequal-mass, Spinning Binary Neutron Star Mergers". *Astrophys. J. Lett.* 922.1, L19 (2021), p. L19. DOI: 10.3847/2041-8213/ac350d. arXiv: 2109.00940 [gr-qc].

[112] S. Tootle et al. "Quark formation and phenomenology in binary neutron-star mergers using V-QCD". *SciPost Phys.* 13 (2022), p. 109. DOI: 10.21468/SciPostPhys.13.5.109. arXiv: 2205.05691 [astro-ph.HE].

[113] L. J. Papenfort et al. "Impact of extreme spins and mass ratios on the post-merger observables of high-mass binary neutron stars". *Mon. Not. Roy. Astron. Soc.* 513.3 (2022), pp. 3646–3662. DOI: 10.1093/mnras/stac964. arXiv: 2201.03632 [astro-ph.HE].

[114] X. Xie and A. T.-L. Lam. "Bridging relativistic jets from black hole scales to long-term electromagnetic radiation distances: A moving-mesh general relativistic hydrodynamics code with the HLLC Riemann solver". *Phys. Rev. D* 109 (8 2024), p. 084070. DOI: 10.1103/PhysRevD.109.084070. URL: https://link.aps.org/doi/10.1103/PhysRevD.109.084070.

[115] R. Arnowitt, S. Deser, and C. W. Misner. "The dynamics of general relativity". *Gravitation: An Introduction to Current Research*. Ed. by L. Witten. New York: Wiley, 1962, pp. 227–265. eprint: {arXiv:gr-qc/0405109}.

[116] J. D. Brown. "Covariant formulations of Baumgarte, Shapiro, Shibata, and Nakamura and the standard gauge". Phys. Rev. D 79.10, 104029 (2009), p. 104029. DOI: 10.1103/PhysRevD.79.104029. arXiv: 0902.3652 [gr-qc].

[117] M. Shibata and T. Nakamura. "Evolution of three-dimensional gravitational waves: Harmonic slicing case". *Phys. Rev. D* 52 (10 1995), pp. 5428–5444. DOI: 10.1103/PhysRevD.52.5428. URL: https://link.aps.org/doi/10.1103/PhysRevD.52.5428.

[118] T. W. Baumgarte and S. L. Shapiro. "Numerical integration of Einstein's field equations". *Phys. Rev. D* 59 (2 1998), p. 024007. DOI: 10.1103/PhysRevD.59.024007. URL: https://link.aps.org/doi/10.1103/PhysRevD.59.024007.

[119] T. W. Baumgarte et al. "Numerical relativity in spherical polar coordinates: Evolution calculations with the BSSN formulation". *Phys. Rev. D* 87 (4 2013), p. 044026. DOI: 10.1103/PhysRevD.87.044026. URL: https://link.aps.org/doi/10.1103/PhysRevD.87.044026.

[120] D. M. Siegel and B. D. Metzger. "Three-dimensional GRMHD simulations of neutrino-cooled accretion disks from neutron star mergers". *Astrophys. J.* 858.1 (2018), p. 52. DOI: 10.3847/1538-4357/aabaec. arXiv: 1711.00868 [astro-ph.HE].

[121] C. Palenzuela et al. "Effects of the microphysical equation of state in the mergers of magnetized neutron stars with neutrino cooling". *Phys. Rev. D* 92 (4 2015), p. 044045. DOI: 10.1103/PhysRevD.92.044045. URL: https://link.aps.org/doi/10.1103/PhysRevD.92.044045.

[122] W. I. Newman and N. D. Hamlin. "Primitive Variable Determination in Conservative Relativistic Magnetohydrodynamic Simulations". *SIAM Journal on Scientific Computing* 36.4 (2014), B661–B683. DOI: 10.1137/140956749. eprint: https://doi.org/10.1137/140956749. URL: https://doi.org/10.1137/140956749.

[123] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge: Cambridge University Press, 2004.

[124] P. Colella and P. R. Woodward. "The Piecewise Parabolic Method (PPM) for gas-dynamical simulations". *Journal of Computational Physics* 54.1 (1984), pp. 174–201. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(84)90143-8. URL: https://www.sciencedirect.com/science/article/pii/0021999184901438.

[125] K. G. Felker and J. M. Stone. "A fourth-order accurate finite volume method for ideal MHD via upwind constrained transport". *Journal of Computational Physics* 375 (2018), pp. 1365–1400. DOI: 10.1016/j.jcp.2018.08.025. arXiv: 1711.07439 [astro-ph.IM].

[126] D. Balsara. "Total Variation Diminishing Scheme for Relativistic Magnetohydrodynamics". *The Astrophysical Journal Supplement Series* 132.1 (2001), p. 83. DOI: 10.1086/318941. URL: https://dx.doi.org/10.1086/318941.

[127] F. Cipolletta et al. "Spritz: a new fully general-relativistic magnetohydrodynamic code". *Classical and Quantum Gravity* 37.13 (2020), p. 135010. DOI: 10.1088/1361-6382/ab8be8. URL: https://doi.org/10.1088%2F1361-6382%2Fab8be8.

[128] B. Giacomazzo and L. Rezzolla. "The exact solution of the Riemann problem in relativistic magnetohydrodynamics". *Journal of Fluid Mechanics* 562 (2006), pp. 223–259. DOI: 10.1017/S0022112006001145. arXiv: gr-qc/0507102 [gr-qc].

[129] S. S. Komissarov. "A Godunov-type scheme for relativistic magnetohydrodynamics". *Mon. Not. R. Astron. Soc.* 303.2 (1999), pp. 343–366. DOI: 10.1046/j.1365-8711.1999.02244.x.

[130] A. S. Schneider, L. F. Roberts, and C. D. Ott. "Open-source nuclear equation of state framework based on the liquid-drop model with Skyrme interaction". *Phys. Rev. C* 96 (6 2017), p. 065802. DOI: 10.1103/PhysRevC.96.065802. URL: https://link.aps.org/doi/10.1103/PhysRevC.96.065802.

[131] J. A. Font et al. "Three-dimensional numerical general relativistic hydrodynamics. II. Long-term dynamics of single relativistic stars". *Phys. Rev. D* 65 (8 2002), p. 084024. DOI: 10.1103/PhysRevD.65.084024. URL: https://link.aps.org/doi/10.1103/PhysRevD.65.084024.

[132] G. Shen, C. J. Horowitz, and S. Teige. "New equation of state for astrophysical simulations". Phys. Rev. C 83.3, 035802 (2011), p. 035802. DOI: 10.1103/PhysRevC.83.035802. arXiv: 1101.3715 [astro-ph.SR].

[133] F. Galeazzi et al. "Implementation of a simplified approach to radiative transfer in general relativity". *Phys. Rev. D* 88 (2013), p. 064009. DOI: 10.1103/PhysRevD.88.064009. arXiv: 1306.4953 [gr-qc].

[134] D. Neilsen et al. "Magnetized Neutron Stars With Realistic Equations of State and Neutrino Cooling". *Phys. Rev. D* 89.10 (2014), p. 104029. DOI: 10.1103/PhysRevD.89.104029. arXiv: 1403.3680 [gr-qc].

[135] E. O'Connor and C. D. Ott. "A New Open-Source Code for Spherically-Symmetric Stellar Collapse to Neutron Stars and Black Holes". *Class. Quant. Grav.* 27 (2010). Ed. by C. D. Ott, C. J. Pethick, and L. Rezzolla, p. 114103. DOI: 10.1088/0264-9381/27/11/114103. arXiv: 0912.2393 [astro-ph.HE].

[136] K. D. Kokkotas. *Private communication*, correspondence via email. 2024.

[137] M. Campanelli et al. "Accurate Evolutions of Orbiting Black-Hole Binaries without Excision". Phys. Rev. Lett. 96.11, 111101 (2006), p. 111101. DOI: 10.1103/PhysRevLett.96.111101. arXiv: gr-qc/0511048 [gr-qc].

[138] J. R. van Meter et al. "How to move a black hole without excision: Gauge conditions for the numerical evolution of a moving puncture". Phys. Rev. D 73.12, 124011 (2006), p. 124011. DOI: 10.1103/PhysRevD.73.124011. arXiv: gr-qc/0605030 [gr-qc].

[139] H. Kreiss and J. Oliger. *Methods for the approximate solution of time dependent problems*. 10. International Council of Scientific Unions, World Meteorological Organization, 1973.

[140] D. Radice, L. Rezzolla, and F. Galeazzi. "High-order fully general-relativistic hydrodynamics: new approaches and tests". *Classical and Quantum Gravity* 31.7 (2014), p. 075012. DOI: 10.1088/0264-9381/31/7/075012. URL: https://dx.doi.org/10.1088/0264-9381/31/7/075012.

[141] L. Lindblom. "Spectral representations of neutron-star equations of state". Phys. Rev. D 82.10, 103011 (2010), p. 103011. DOI: 10.1103/PhysRevD.82.103011. arXiv: 1009.0738 [astro-ph.HE].

[142] L. Lindblom. "Improved spectral representations of neutron-star equations of state". Phys. Rev. D 105.6, 063031 (2022), p. 063031. DOI: 10.1103/PhysRevD.105.063031. arXiv: 2202.12285 [astro-ph.HE].

[143] F. Foucart et al. "Smooth equations of state for high-accuracy simulations of neutron star binaries". Phys. Rev. D 100.10, 104048 (2019), p. 104048. DOI: 10.1103/PhysRevD.100.104048. arXiv: 1908.05277 [gr-qc].

[144] A. Knight et al. "Gravitational Waves from Binary Neutron Star Mergers with a Spectral Equation of State". *arXiv e-prints*, arXiv:2307.03250 (2023), arXiv:2307.03250. DOI: 10.48550/arXiv.2307.03250. arXiv: 2307.03250 [astro-ph.HE].

[145] N. Stergioulas, T. A. Apostolatos, and J. A. Font. "Non-linear pulsations in differentially rotating neutron stars: mass-shedding-induced damping and splitting of the fundamental mode". *Mon. Not. R. Astron. Soc.* 352.4 (2004), pp. 1089–1101. DOI: 10.1111/j.1365-2966.2004.07973.x. arXiv: astro-ph/0312648 [astro-ph].

[146] L. J. Papenfort et al. "New public code for initial data of unequal-mass, spinning compact-object binaries". *Phys. Rev. D* 104.2 (2021), p. 024057. DOI: 10.1103/PhysRevD.104.024057. arXiv: 2103.09911 [gr-qc].

[147] P. Grandclement. "Kadath: A Spectral solver for theoretical physics". *J. Comput. Phys.* 229 (2010), pp. 3334–3357. DOI: 10.1016/j.jcp.2010.01.005. arXiv: 0909.1228 [gr-qc].

[148] J. W. York Jr. "Conformal 'thin sandwich' data for the initial-value problem". *Phys. Rev. Lett.* 82 (1999), pp. 1350–1353. DOI: 10.1103/PhysRevLett.82.1350. arXiv: gr-qc/9810051.

[149] H. P. Pfeiffer and J. W. York Jr. "Extrinsic curvature and the Einstein constraints". *Phys. Rev. D* 67 (2003), p. 044022. DOI: 10.1103/PhysRevD.67.044022. arXiv: gr-qc/0207095.

[150] W. Tichy. "Initial data for binary neutron stars with arbitrary spins". *Phys. Rev. D* 84 (2011), p. 024041. DOI: 10.1103/PhysRevD.84.024041. arXiv: 1107.1440 [gr-qc].

[151] E. R. Most. "Impact of a mean field dynamo on neutron star mergers leading to magnetar remnants". Phys. Rev. D 108.12, 123012 (2023), p. 123012. DOI: 10.1103/PhysRevD.108.123012. arXiv: 2311.03333 [astro-ph.HE].

[152] L. Del Zanna et al. "General Relativistic Magnetohydrodynamics Mean-Field Dynamos". *Fluids* 7.2 (2022), p. 87. DOI: 10.3390/fluids7020087.

[153] D. Radice. "General-relativistic Large-eddy Simulations of Binary Neutron Star Mergers". *Astrophys. J. Lett.* 838.1, L2 (2017), p. L2. DOI: 10.3847/2041-8213/aa6483. arXiv: 1703.02046 [astro-ph.HE].

[154] D. Radice. "Binary Neutron Star Merger Simulations with a Calibrated Turbulence Model". *Symmetry* 12.8 (2020), p. 1249. DOI: 10.3390/sym12081249. arXiv: 2005.09002 [astro-ph.HE].

[155] C. Palenzuela et al. "Turbulent magnetic field amplification in binary neutron star mergers". *Phys. Rev. D* 106 (2 2022), p. 023013. DOI: 10.1103/PhysRevD.106.023013. URL: https://link.aps.org/doi/10.1103/PhysRevD.106.023013.

[156] M. Shibata, T. W. Baumgarte, and S. L. Shapiro. "The Bar-Mode Instability in Differentially Rotating Neutron Stars: Simulations in Full General Relativity". *Astrophys. J.* 542.1 (2000), pp. 453–463. DOI: 10.1086/309525. arXiv: astro-ph/0005378 [astro-ph].

[157] L. Baiotti et al. "Accurate simulations of the dynamical barmode instability in full General Relativity". *Phys. Rev. D* 75 (2007), p. 044023. DOI: 10.1103/PhysRevD.75.044023. arXiv: astro-ph/0609473.

[158] F. Löffler et al. "Stiffness effects on the dynamics of the bar-mode instability of Neutron Stars in full General Relativity". *Phys. Rev. D* 91 (2015), p. 064057. DOI: 10.1103/PhysRevD.91.064057. arXiv: 1411.1963 [gr-qc].

[159] N. Jadoo, T. P. Jacques, and Z. B. Etienne. "superB/NRPy: scalable, task-based numerical relativity for 3G gravitational wave science". *Classical and Quantum Gravity* 42.15, 155006 (2025), p. 155006. DOI: 10.1088/1361-6382/adee71. arXiv: 2505.00097 [gr-qc].

[160] L. V. Kale and S. Krishnan. "Charm++: A Portable Concurrent Object Oriented System Based on C++". *Proceedings of the 8th Annual Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*. Washington, D.C., USA: Association for Computing Machinery, 1993, pp. 91–108. DOI: 10.1145/165854.165873. URL: https://doi.org/10.1145/165854.165873.

[161] The Charm++ Development Team. *Charm++: The Parallel Programming System*. https://github.com/UIUC-PPL/charm. Accessed: 2025-10-22. 2025.

[162] E. R. Most, L. J. Papenfort, and L. Rezzolla. "Beyond second-order convergence in simulations of magnetized binary neutron stars with realistic microphysics". *Monthly Notices of the Royal Astronomical Society* 490.3 (2019), pp. 3588–3600. ISSN: 0035-8711. DOI: 10.1093/mnras/stz2809. eprint: https://academic.oup.com/mnras/article-pdf/490/3/3588/30338124/stz2809.pdf. URL: https://doi.org/10.1093/mnras/stz2809.

[163] S. Gottlieb, C.-W. Shu, and E. Tadmor. "Strong Stability-Preserving High-Order Time Discretization Methods". *SIAM Review* 43.1 (2001), pp. 89–112. DOI: 10.1137/S003614450036757X.

[164] N. Stergioulas and J. L. Friedman. "Comparing Models of Rapidly Rotating Relativistic Stars Constructed by Two Numerical Methods". ApJ 444 (1995), p. 306. DOI: 10.1086/175605. arXiv: astro-ph/9411032 [astro-ph].

[165] H. P. Pfeiffer and J. W. York. "Uniqueness and Nonuniqueness in the Einstein Constraints". *Phys. Rev. Lett.* 95.9, 091101 (2005), p. 091101. DOI: 10.1103/PhysRevLett.95.091101.

[166] H. Komatsu, Y. Eriguchi, and I. Hachisu. "Rapidly rotating general relativistic stars. II – Differentially rotating polytropes". *Mon. Not. R. Astron. Soc.* 239 (1989), pp. 153–171.

[167] V. Paschalidis and N. Stergioulas. "Rotating stars in relativity". *Living Reviews in Relativity* 20.1, 7 (2017), p. 7. DOI: 10.1007/s41114-017-0008-x. arXiv: 1612.03050 [astro-ph.HE].

[168] G. M. Manca et al. "Dynamical non-axisymmetric instabilities in rotating relativistic stars". *Classical and Quantum Gravity* 24.12 (2007), S171–S186. DOI: 10.1088/0264-9381/24/12/S12. arXiv: 0705.1826 [astro-ph].

[169] M. Hempel et al. "New Equations of State in Simulations of Core-collapse Supernovae". ApJ 748.1, 70 (2012), p. 70. DOI: 10.1088/0004-637X/748/1/70. arXiv: 1108.0848 [astro-ph.HE].